Submitted to the Int'l J. of Computer Vision, a short version appeared in ICCV01.

Modeling Visual Patterns by Integrating Descriptive and Generative Methods

Cheng-en Guo[1], Song-Chun Zhu[1,2] and Ying Nian Wu[2]
[1]Department of Computer Science,
[2]Department of Statistics,
University of California, Los Angeles
{cguo, sczhu, ywu}@stat.ucla.edu

Abstract

This paper presents a class of statistical models that integrate two statistical modeling paradigms in the literature: I). Descriptive methods, such as Markov random fields and minimax entropy learning [41], and II). Generative methods, such as principal component analysis, independent component analysis [2], transformed component analysis [11], wavelet coding [27, 5], and sparse coding [30, 24]. In this paper, we demonstrate the integrated framework by constructing a class of hierarchical models for texton patterns (the term "texton" was coined by psychologist Julesz in the early 80s). At the bottom level of the model, we assume that an observed texture image is generated by multiple hidden "texton maps", and textons on each map are translated, scaled, stretched, and oriented versions of a window function, like mini-templates or wavelet bases. The texton maps generate the observed image by occlusion or linear superposition. This bottom level of the model is generative in nature. At the top level of the model, the spatial arrangements of the textons in the texton maps are characterized by minimax entropy principle, which leads to embellished versions of Gibbs point process models [34]. The top level of the model is descriptive in nature. We demonstrate the integrated model by a set of experiments.

Keywords: Descriptive models, Generative models, Gibbs point processes, Markov chain Monte Carlo, Markov random fields, Minimax entropy learning, Perceptual organization, Texton models, Visual learning.

1 Introduction

What a vision algorithm can accomplish depends crucially upon how much it knows about the contents of the visual scenes, and the knowledge can be mathematically represented by general and parsimonious models that can realistically characterize visual patterns in the ensemble of images. Due to the variations of the patterns across scenes and the richness of details within each scene, the models are often statistical in nature. Existing methods for statistical modeling can be generally divided into two categories. In this paper, we call one category the *descriptive methods* and the other category the *generative methods*.[1]

Descriptive methods construct the model for a visual pattern by imposing statistical constraints on features extracted from signals. Descriptive methods include Markov random fields, minimax entropy learning [41], deformable models, etc. For example, recent methods on texture modeling all fall into this category [17, 41, 7, 32]. These models are built on pixel intensities or some deterministic transforms of the original signals, such as linear filtering. The shortcomings of descriptive methods are two-fold. First, they do not capture high level semantics in visual patterns, which are often very important in human perception. For example, a descriptive model of texture can realize a cheetah skin pattern with impressive synthesis results but it does not have explicit notion of individual blobs. Second, as descriptive models are built directly on the original signals, the resulting probability densities are often of very high dimensions and the sampling and inference are computationally expensive. It is desirable to have dimension reduction or sparse representation so that the models can be built in a low dimensional space that often better reflects the intrinsic complexity of the pattern.

In contrast to descriptive methods, generative methods postulate hidden variables as the causes for the complicated dependencies in raw signals, and thus the models are hierarchical. Generative methods are widely used in vision and image analysis. For example, principle component analysis (PCA), independent component analysis (ICA) [2], transformed compo-

[1] There is a third category of methods that can be called discriminative. The goal of discriminative methods is not for modeling visual patterns explicitly but for approximating the posterior probabilities directly, for example, pattern recognition, feed-forward neural networks and classification trees, etc. Thus we choose not to discuss it because our focus is on statistical modeling. See, however, Tu and Zhu (2002) [36] that incorporates the discriminative methods in Markov chain Monte Carlo posterior sampling.

nent analysis (TCA) [11], wavelet image representation [27, 5], sparse coding [30, 24], and the random collage model for generic natural images [21]. The hidden variables employed to represent or generate the observed image usually follow very simple models. However, existing generative models appear to suffer from an over-simplified assumption that the hidden variables are independent and identically distributed.[2] As a result, they are not sophisticated enough to model realistic visual patterns. For example, a wavelet image coding model can easily reconstruct an observed image, but it cannot synthesize a texture pattern through independent random sampling because the spatial relationships between the wavelet coefficients are not captured.

The two modeling paradigms were developed almost independently by somewhat disjoint communities working on different problems, and their relationship has yet to be explored. In this paper, we present a class of probabilistic models that integrate both descriptive and generative methods, as well as the algorithm for computational inference.

The proposed method can be viewed from the following four perspectives:

Figure 1 Two examples of natural patterns with layered structures. We not only perceive the texture impression in terms of pixel intensities, but also the repeated texture elements.

First, it combines the advantages of both descriptive and generative methods, and provides a general scheme for modeling sophisticated visual patterns. In computer vision, a fundamental observation, stated in Marr's primal sketch paradigm [28], is that natural visual

[2]Interested readers are referred to a recent paper [31] for discussion of the problem with existing generative models.

patterns consist of multiple layers of stochastic processes. For example, Figure 1 displays two natural images. When we look at the ivy-wall image, we perceive not only the texture "impression" in terms of pixel intensities, but we also see the repeated elements in the ivy and bricks. To capture the hierarchical notion, we propose a multi-layer generative model as shown in Figure 2. Inspired by the seminal work of Olshausen and Field [30], we assume that an image is generated by a few layers of stochastic processes and each layer consists of a finite number of distinct but similar elements, called "textons" (following the terminology of Julesz). In our experiments, each texton covers more than 100 pixels on average, so the layered representation achieves nearly 100-fold dimension reduction or sparsity. With sparse representation, the next step should be the modeling of the spatial arrangements based on geometric features. In particular, the textons at each layer are characterized by Markov random field (MRF) models through the minimax entropy learning [41], and previous MRF texture models can be considered special cases where the models have only one layer and each "texton" is just a pixel. See also a recent paper of ours [38] that is directly built on the work of Olshausen and Field [30], where the geometry of the elongate linear bases is characterized by a causal sketch model. We feel that the integrated model is a natural next step for the linear superposition models in wavelet and sparse coding.

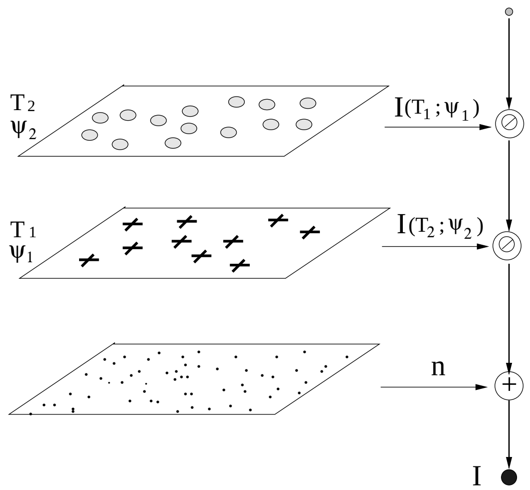


Figure 2 A generative model for an image $\mathbf{I}$ consists of multiple layers of texton maps $\mathbf{I}(\mathbf{T}_l; \Psi_l)$, $l = 1, ..., L$ superimposed with occlusion plus a background texture image $\mathbf{n}$.

It is our belief that *descriptive models can be precursors of generative models* and both are ingredients of the integrated learning process. In visual learning, the model can be initially built on image intensities via some features computed deterministically from the image intensities. Then we can replace the features by hidden causes, and such a process would incrementally discover more abstract elements or concepts such as textons, curves, flows, and so on, where elements at the more abstract levels become causes for the elements of lower abstractions. For instance, the flows generate curves, and the curves generate textons, which in turn generate pixel intensities. At each stage, the elements at the most abstract level have no further hidden causes and thus can be characterized by a descriptive model based on some deterministic features, and such models can be derived by the minimax entropy principle as demonstrated in [37]. When a new hidden level of elements is introduced, it replaces the current descriptive model by a simplified one. The learning process evolves until the descriptive model for the most abstract elements becomes simple enough for a certain vision purpose.

Second, the integrated scheme provides a representational definition of "textons". Texton has been an important notion in texture perception and early vision. Unfortunately, it was only expressed vaguely in psychology [19], and a precise definition of texton has yet to be found. In this paper, we argue that a definition of "texton" is possible only in the context of a generative model. In this paper, in contrast to the constraint-based clustering method by Malik, Leung, etc. [22, 23, 26], textons are naturally embedded in a generative model and are inferred as hidden variables of the generative model. This is consistent with the philosophy of ICA [11], TCA [11] and sparse coding [30, 24]. In this paper, the textons are defined in terms of image bases or window functions. In a related paper of ours [40], we explored other definitions of textons, such as combinations of linear bases, local elements of shape and shading, etc.

Third, we present a *Gestalt ensemble* to characterize the hidden texton maps as attributed point processes. The Gestalt ensemble corresponds to the grand canonical ensemble in statistical physics [4], and it differs from traditional Gibbs models by having an unknown number of textons whose neighborhood changes dynamically. The relationships between neighboring textons are captured by some Gestalt laws, such as proximity, continuity, etc.

Fourth, we adapt a stochastic gradient algorithm [16] for learning and inference. In the algorithm, we simplify the original likelihood function and solve the simplified maximum likelihood problem first. Starting from the initial solution, we then use the stochastic gradient algorithm to find refined solutions.

We demonstrate the proposed modeling method on texture images. For an input texture image, the learning algorithm can achieve the following four objectives:

1. Learning the appearance of textons for each stochastic process. Textons of the same stochastic process are translated, scaled, stretched, and oriented versions of a window function, like mini-templates or wavelet bases.

2. Inferring the hidden texton maps, each of which consists of an unknown number of similar textons that are related to each other by affine transformations.

3. Learning the minimax entropy models for the stochastic processes that generate the textons maps.

4. Verifying the learned window functions and generative models through stochastic sampling.

Recently, a variety of texture synthesis techniques have been proposed, notably the successful methods of Efros and Freeman [10] and Xu, Guo, and Shum [39], which are based on rearranging local image patches. Our work, however, is more concerned with learning parsimonious and sufficient models for texture patterns. Such models can be useful for image understanding in computer vision, and it may also lead to more graphics applications because the models may capture visually meaningful dimensions.

The paper is organized as follows. Section (2) introduces the background on both generative and descriptive methods. Section (3) discusses a hierarchical model for texture. Section (4) studies Gestalt ensembles for modeling texton processes. Then section (5) presents an integrated modeling scheme. Section (6) presents the algorithm for inferential computation. Some experiments are shown in Section (7). We conclude the paper with a discussion in section (8).

2 Background on Descriptive and Generative Models

Given a set of images $\mathcal{I} = \{\mathbf{I}_1^{\rm obs}, ..., \mathbf{I}_M^{\rm obs}\}$, where $\mathbf{I}_m^{\rm obs}, m = 1, ..., M$ are considered realizations of some underlying stochastic process governed by a frequency distribution $f(\mathbf{I})$. The objective of visual learning is to estimate a parsimonious probabilistic model $p(\mathbf{I})$ based on $\mathcal{I}$ so that $p(\mathbf{I})$ approaches $f(\mathbf{I})$ by minimizing a Kullback-Leibler divergence $KL(f||p)$ from f to p [6],

$$KL(f||p) = \int f(\mathbf{I}) \log \frac{f(\mathbf{I})}{p(\mathbf{I})} d\mathbf{I} = E_f[\log f(\mathbf{I})] - E_f[\log p(\mathbf{I})]. \tag{1}$$

In practice, the expectation $E_f[\log p(\mathbf{I})]$ is replaced by a sample average. Thus we have the standard maximum likelihood estimator (MLE),

$$p^* = \arg\min_{p \in \Omega_p} KL(f||p) \approx \arg\max_{p \in \Omega_p} \sum_{m=1}^{M} \log p(\mathbf{I}_m^{\rm obs}), \tag{2}$$

where Ω_p is the family of distributions where p^* is searched for. One general procedure is to search for p in a sequence of nested probability families of increasing complexities,

$$\Omega_0 \subset \Omega_1 \subset \cdots \subset \Omega_K \to \Omega_f \ni f,$$

where K indexes the dimensionality of the space. For example, K could be the number of free parameters in a model. As K increases, the probability family should be general enough to approach f to an arbitrary preset precision.

There are two choices of families for Ω_p in the literature.

The first choice is the exponential family, which can be derived by the descriptive method through maximum entropy, and has its root in statistical mechanics [4]. A descriptive method extracts a set of K feature statistics as *deterministic transforms* of an image $\mathbf{I}$, denoted by $\phi_k(\mathbf{I}), k = 1, ..., K$. Then it constructs a model p by imposing *descriptive constraints* so that p reproduces the observed statistics $\mathbf{h}_k^{\rm obs}$ extracted from $\mathcal{I}$,

$$E_p[\phi_k(\mathbf{I})] = \mathbf{h}_k^{\rm obs} = \frac{1}{M} \sum_{m=1}^{M} \phi_k(\mathbf{I}_m^{\rm obs}) \approx E_f[\phi_k(\mathbf{I})] = \mathbf{h}_k, \quad k = 1, ..., K. \tag{3}$$

One may consider $\mathbf{h}_k$ as a projected statistics of $f(\mathbf{I})$, thus when M is large enough, p and f will have the same projected (marginal) statistics on the K chosen dimensions. By the maximum entropy principle [18], this leads to the Gibbs model,

$$p(\mathbf{I}; \boldsymbol{\beta}) = \frac{1}{Z(\boldsymbol{\beta})} \exp\{-\sum_{k=1}^{K} \beta_k \phi_k(\mathbf{I})\}.$$

The parameters $\boldsymbol{\beta} = (\beta_1, ..., \beta_K)$ are Lagrange multipliers and they are computed by solving the constraint equations (3). The K features are chosen by a minimum entropy principle [41].

The descriptive learning method augments the dimension of the space Ω_p by increasing the number of feature statistics and generates a sequence of exponential families,

$$\Omega_1^d \subset \Omega_2^d \subset \cdots \Omega_K^d \to \Omega_f \ni f.$$

This family includes all the MRF and minimax entropy models for texture [41]. For example, a type of descriptive model for texture chooses $\phi_j(\mathbf{I})$ as the histograms of responses from some Gabor filters.

The second choice is the mixture family, which can be derived by integration or summation over some hidden variables $W = (w_1, ..., w_K)$,

$$p(\mathbf{I}; \Theta) = \int p(\mathbf{I}, W; \Theta) dW = \int p(\mathbf{I}|W; \Psi) p(W; \boldsymbol{\beta}) dW.$$

The parameters of a generative model include two parts $\Theta = (\Psi, \boldsymbol{\beta})$. It assumes a joint probability distribution $p(\mathbf{I}, W; \Theta)$, and that W generates $\mathbf{I}$ through a conditional model $p(\mathbf{I}|W; \Psi)$ with parameters Ψ. The hidden variables are characterized by a model $p(W; \boldsymbol{\beta})$. W should be *inferred* from $\mathbf{I}$ in a probabilistic manner, and this is in contrast to the deterministic features $\phi_k(\mathbf{I}), k = 1, ..., K$ in descriptive models. The generative method incrementally adds hidden variables to augment the space Ω_p and thus generates a sequence of mixture families,

$$\Omega_1^g \subset \Omega_2^g \subset \cdots \subset \Omega_K^g \to \Omega_f \ni f.$$

For example, principal component analysis, wavelet image coding [27, 5], and sparse coding [30, 24] all assume a linear additive model where an image $\mathbf{I}$ is the result of linear superposition of some window functions $\Psi_k, k = 1, ..., K$, plus a Gaussian noise process $\mathbf{n}$.

$$\mathbf{I} = \sum_{k=1}^{K} a_k \Psi_k + \mathbf{n},$$

where $a_k, k = 1, ..., K$ are the coefficients, Ψ_k are the eigen vectors in PCA, wavelet bases in image coding, or over-complete basis for sparse coding. The hidden variables are the K coefficients of bases plus the noise, so $W = (a_1, ..., a_K, \mathbf{n})$.[3] The coefficients are assumed to

[3]In PCA, since the bases are orthogonal, a_k can be computed as transform, but for over-complete basis, the a_k have to be inferred.

be independently and identically distributed,

$$a_k \sim p(a_k) = \frac{1}{Z} \exp\{-\lambda_o |a_k|^\rho\}, \quad k = 1, ..., K,$$

where Z is a normalizing factor. The norm $\rho = 1$ for sparse coding [30, 24] and basis pursuit [5], and $\rho = 2$ for principal component analysis. Thus we have a simple distribution for W,

$$p(W; \boldsymbol{\beta}) = \frac{1}{Z} \prod_{k=1}^{k} \exp\{-\lambda_o |a_k|^\rho\} \prod_{(x,y)} \exp\{-\frac{\mathbf{n}^2(x,y)}{2\sigma_o^2}\}.$$

In this example, the parameters are the K bases plus the parameters in $p(W; \boldsymbol{\beta})$, $\Theta = \{\Psi_1, ..., \Psi_K, \lambda_o, \sigma_o\}$. There are also occlusion models with randomly positioned discs called random collage or deadleaf models (see [21] and refs. therein).

In this model $p(W; \boldsymbol{\beta})$ is from the exponential family. However, in the literature, hidden variables $a_k, k = 1, ..., K$ are assumed to be iid Gaussian or Laplacian distributed. Thus the concept of descriptive models are trivialized.

3 A Multi-layered Generative Model for Texture

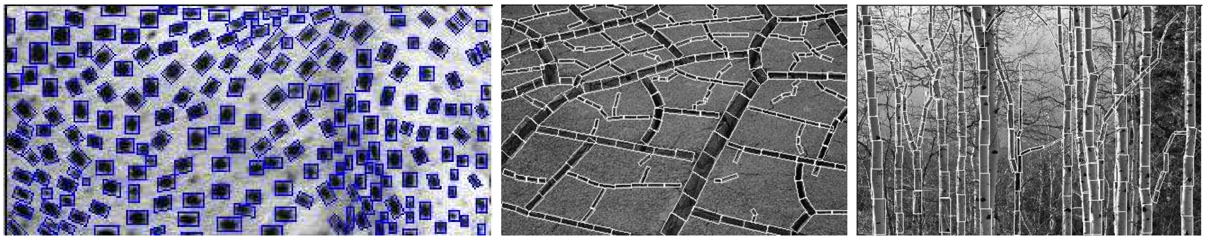


Figure 3 Texture images with texton processes. Each texton is represented by a rectangle window.

We focus on a multi-layer generative model for texture images and we believe that the same modeling method can be applied to other patterns such as object shapes. An image $\mathbf{I}$ is assumed to be generated by L layers of stochastic processes, and each layer consists of a finite number of distinct but similar elements, called "textons". Figure 3 shows three typical examples of texture images, and each texton is represented by a rectangular window. A layered model is shown in Figure 2.

Textons at layer l are image patches transformed from a square template Ψ_l. The j-th texton in layer l is identified by six transformation variables,

$$t_{lj} = (x_{lj}, y_{lj}, \sigma_{lj}, \tau_{lj}, \theta_{lj}, A_{lj}), \tag{4}$$

where (x_{lj}, y_{lj}) represents the texton center location, σ_{lj} the scale (or size), τ_{lj} the "stretch" (aspect ratio of height versus width), θ_{lj} the orientation, and A_{lj} for photometric transforms such as lighting variability. t_{lj} defines an affine transform denoted by $G[t_{lj}]$, and the pixels covered by a texton t_{lj} is denoted by D_{lj}. Thus the image patch $\mathbf{I}_{D_{lj}}$ of a texton t_{lj} is

$$\mathbf{I}_{D_{lj}} = G[t_{lj}] \odot \Psi_l, \quad \forall j, \ \forall l,$$

where $\odot$ denotes the transformation operator. Texton examples of a circular template at different scales, stretches, and orientations are shown in Figure 4.

Figure 4 A template Ψ and its three transformed copies. a). template Ψ; b). scaled copy; c). stretched copy; d). scaled/stretched/rotated copy.

We define the collection of all textons in layer l as a *texton map*,

$$\mathbf{T}_l = (n_l, \{t_{lj}, j = 1 \ldots n_l\}), l = 1 \ldots L,$$

where n_l is the number of textons in layer l.

In each layer, the texton map $\mathbf{T}_l$ and the template Ψ_l generate an image $\mathbf{I}_l = \mathbf{I}(\mathbf{T}_l; \Psi_l)$ deterministically. If several texton patches overlap at site (x, y) in $\mathbf{I}_l$, the pixel value is taken as average,

$$\mathbf{I}_l(x, y) = \frac{\sum_{j=1}^{n_l} \delta((x, y) \in D_{lj}) \mathbf{I}_{D_{lj}}(x, y)}{\sum_{j=1}^{n_l} \delta((x, y) \in D_{lj})},$$

where $\delta(\bullet) = 1$ if $\bullet$ is true, otherwise $\delta(\bullet) = 0$. In image $\mathbf{I}_l$, pixels not covered by any texton patches are transparent. The image $\mathbf{I}$ is generated in the following way,

$$\mathbf{I}(\mathbf{T}; \Psi) = \mathbf{I}(\mathbf{T}_1; \Psi_1) \ \oslash \ \mathbf{I}(\mathbf{T}_2; \Psi_2) \ \oslash \cdots \oslash \ \mathbf{I}(\mathbf{T}_L; \Psi_L), \qquad \text{and} \quad \mathbf{I}^{\text{obs}} = \mathbf{I}(\mathbf{T}; \Psi) \ + \ \mathbf{n}. \tag{5}$$

The symbol $\oslash$ denotes occlusion (or linear addition), i.e. $\mathbf{I}_1 \oslash \mathbf{I}_2$ means $\mathbf{I}_1$ occludes $\mathbf{I}_2$. $\mathbf{I}(\mathbf{T};\Psi)$ is called a *reconstructed image* and $\mathbf{n}$ is assumed to be Gaussian noise process $\mathbf{n}(x,y) \sim N(0,\sigma_0^2), \forall(x,y)$, although in general it should be a stochastic texture. Thus pixel value at site (x,y) in the image $\mathbf{I}$ is the same as the top layer image at that point, while uncovered pixels are only modeled by noises.

In this generative model, the hidden variables are

$$\mathbf{T} = (L, \{(\mathbf{T}_l, d_l) : l = 1, \ldots, L\},\ \mathbf{n}),$$

where d_l indexes the order (or relative depth) of the l-th layer.

To simplify computation, we assume that $L = 2$ and the two stochastic layers are called "background" and "foreground" respectively. The two texton process $\mathbf{T}_l, l = 1, 2$ are assumed to be independent of each other. This assumption seems okay for simple texture patterns studied in this paper, but for more sophisticated patterns, it is certainly necessary to have more levels and to consider the dependencies among these levels.

Thus the likelihood for an observable image $\mathbf{I}$ can be computed

$$p(\mathbf{I};\Theta) = \int p(\mathbf{I}|\mathbf{T};\Psi)p(\mathbf{T};\boldsymbol{\beta})d\mathbf{T}, \tag{6}$$

$$= \int p(\mathbf{I}|\mathbf{T}_1,\mathbf{T}_2;\Psi)\prod_{l=1}^{2} p(\mathbf{T}_l;\beta_{lo},\boldsymbol{\beta}_l)d\mathbf{T}_1 d\mathbf{T}_2, \tag{7}$$

where $\Psi = (\Psi_1, \Psi_2)$ be texton templates and $\boldsymbol{\beta} = (\beta_{1o}, \boldsymbol{\beta}_1, \beta_{2o}, \boldsymbol{\beta}_2)$ the parameters for the two texton processes which we shall discuss in the next section, and σ^2 the variance of the noise. The generative part of the model is a conditional probability $p(\mathbf{I}|\mathbf{T}_1,\mathbf{T}_2;\Psi)$,

$$p(\mathbf{I}^{\text{obs}}|\mathbf{T}_1,\mathbf{T}_2;\Psi) \propto \exp\{\frac{-\left\|\mathbf{I}^{\text{obs}} - \mathbf{I}(\mathbf{T}_1,\mathbf{T}_2;\Psi)\right\|^2}{2\sigma^2}\}, \tag{8}$$

where $I(\mathbf{T}_1,\mathbf{T}_2;\Psi)$ is the reconstructed image from the two hidden layers without noise (see eq. (5)). As the generative model is very simple, the texture pattern should be captured by the spatial arrangements of textons in models $p(\mathbf{T}_l;\beta_{lo},\boldsymbol{\beta}_l), l = 1, 2$, which are in much lower dimensional spaces and are more semantically meaningful than previous Gibbs models on pixels [41].

In the next section, we discuss the model $p(\mathbf{T}_l;\beta_{lo},\boldsymbol{\beta}_l), l = 1, 2$ for the texton processes.

4 A Descriptive Model of Texton Processes

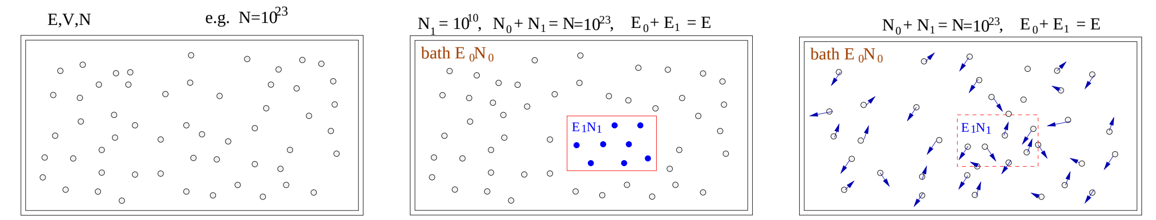
As the texton processes $\mathbf{T}_l$ are not generated by further hidden layers in the model[4], they can be characterized by descriptive models in exponential families. In this section, we first review some background on three physical ensembles, and then introduce a Gestalt ensemble for texton process. Finally we show some experiments for realizing the texton processes.

4.1 Background: The physics foundation for visual modeling

There are two main differences between a texton process $\mathbf{T}_l$ and a conventional texture defined on a lattice $\Lambda \subset Z^2$.

- A texton process has an unknown number of elements and each element has several attributes t_{lj}, while a texture image has a fixed number of pixels and each pixel has only one variable for intensity.
- The neighborhood of a texton can change depending on their relative positions, scales, and orientations, while pixels always have fixed neighborhoods.

Although a texton process is more complicated than a texture image, they share a common property that they all have large number of elements and global patterns arise from simple local interactions between elements. Thus a well-suited theory for studying these patterns is statistical physics – a subject studying macroscopic properties of a system involving a huge number of elements [4].



a). micro-canonical ensemble b). canonical ensemble c). grand-canonical ensemble

Figure 5 Three typical ensembles in statistical mechanics.

[4]We may introduce additional layers of hidden variables for curve processes that render the textons. But our model stops at the texton level in this paper.

To understand the intuitive ideas behind various texture and texton models, we find it revealing to discuss three physical ensembles which are shown in Figure 5.

1). *Micro-canonical ensemble.* Figure 5.a) is an insulated system of N elements. The elements could be atoms or molecules in systems such as solid ferro-magnetic material, fluid, or gas. N is nearly infinity, say $N = 10^{23}$. The system is decided by a configuration $S = (\mathbf{x}^N, \mathbf{m}^N)$, where $\mathbf{x}^N$ describes the coordinates of the N elements and $\mathbf{m}^N$ their momenta. The system is subject to some global constraints $\mathbf{h}_o = (N, E, V)$. That is, the number of elements N, the total system energy E, and total volume V are fixed. When it reaches equilibrium, this insulated system is characterized by a so-called *micro-canonical ensemble*,

$$\Omega_{mcn} = \{S : \mathbf{h}(S) = \mathbf{h}_o,\ f(S; \mathbf{h}_o) = 1/|\Omega_{mcn}|\}.$$

S is a microscopic state or *instance*, and $\mathbf{h}(S)$ is the macroscopic **summary** of the system. The state S is assumed to be uniformly distributed within Ω_{mcn}, thus it is associated with a uniform probability $f(S; \mathbf{h}_o)$. The system is identified by $\mathbf{h}_o$.

2). *Canonical ensemble.* Figure 5.b) illustrates a small subsystem embedded in a micro-canonical ensemble. The subsystem has $n << N$ elements, fixed volume $v << V$ and energy e. It can exchanges energy through the wall with the remaining elements which is called the "heat bath" or "reservoir". At thermodynamic equilibrium, the microscopic state $s = (\mathbf{x}^n, \mathbf{m}^n)$ for the small system is characterized by a canonical ensemble with a Gibbs model $p(s; \boldsymbol{\beta})$,

$$\Omega_{cn} = \{s;\ p(s; \boldsymbol{\beta}) = \frac{1}{Z}\exp\{-\boldsymbol{\beta} e(s)\}\}.$$

In our recent paper on texture modeling [37], the micro-canonical ensemble is mapped to a *Julesz ensemble* where $S = \mathbf{I}$ is an infinite image on 2D plane Z^2, and $\mathbf{h}_o$ is a collection of Gabor filtered histograms. The canonical ensemble is mapped to a FRAME model [41] with $s = \mathbf{I}_\Lambda$ being an image on a finite lattice Λ. Intuitively, s is a small patch of S viewed from a window Λ. The intrinsic relationship between the two ensembles is that the Gibbs model $p(s; \boldsymbol{\beta})$ in Ω_{cn} is derived as a conditional distribution of $f(S; \mathbf{h}_o)$ in Ω_{mcn}. There is a duality between $\mathbf{h}_o$ and $\boldsymbol{\beta}$ (see [37] and refs therein).

3). *Grand-Canonical ensemble.* Figure 5.c) illustrates a third system where the subsystem is open and can exchange not only energy but also elements with the bath. So v is fixed, but n and e may vary. This models liquid or gas materials. At equilibrium, the microscopic state s for this small system is governed by a distribution $p(s; \beta_o, \boldsymbol{\beta})$ with β_o controlling the density of elements in s. Thus a grand-canonical ensemble is

$$\Omega_{gd} = \{s = (n, \mathbf{x}^n, \mathbf{m}^n);\ p(s; \beta_o, \boldsymbol{\beta})\}$$

The grand-canonical ensemble is a mathematical model for visual patterns with varying numbers of elements, thus lays the foundation for modeling texton processes. In the next subsection, we map the grand-canonical ensemble to a *Gestalt ensemble* in visual modeling.

4.2 The Gestalt ensemble

Without loss of generality, we represent a spatial pattern by a set of attributed elements called textons as it was discussed in section (3). To simplify notation, we consider only one texton layer on a lattice Λ,

$$\mathbf{T} = (n, \{t_j = (x_j, y_j, \sigma_j, \tau_j, \theta_j, A_j), j = 1, ..., n\}).$$

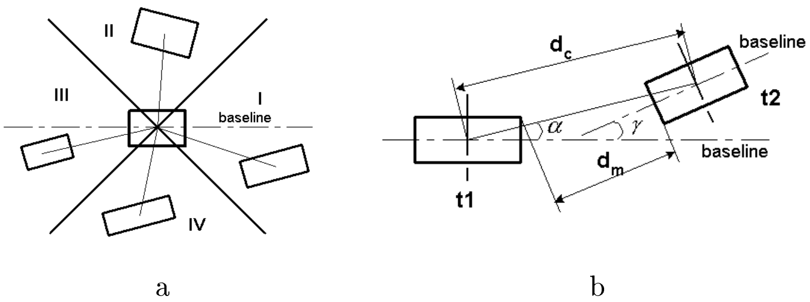


Figure 6 Texton neighborhood. a). a texton has four neighbors; b). Four measurements between texton t_1 and its neighbor t_2, d_c, d_m, α, and γ.

For a texton map $\mathbf{T}$, we define a neighborhood system $\partial(\mathbf{T})$.

$$\partial(\mathbf{T}) = \{\partial t : \ \ t \in \mathbf{T}, \partial t \subset \mathbf{T}\}$$

where ∂t is a set of neighboring textons for each texton t. In this paper, we use the nearest neighborhood. Because each texton covers a 15×15 patch on average, a pair of adjacent textons captures image features at the scale of often more than 30×30 pixels.

There are a few different ways of defining $\partial(\mathbf{T})$. One may treat each texton as a point, and compute a Voronoi diagram or Delaunay triangularization which provides graph structures for the neighborhood. For example, a Voronoi neighborhood was used in (Ahuja and Tuceryan 1989) [1] for grouping dot patterns. However, for textons, we need to consider other attributes such as orientation in defining neighborhood. Figure 6.a) shows a texton t. The plane is separated into four quadrants relative to the two axes of the rectangle. In each quadrant, the nearest texton is considered as the neighbor texton. Unlike the Markov random field on image lattice, the texton neighborhood is no longer translation invariant.

The above neighborhood is defined deterministically. In more general settings, $\partial(\mathbf{T})$ shall be represented by a set of hidden variables that can be inferred from $\mathbf{T}$. Thus a texton may have a varying number of neighbors referenced by some indexing or address variables. These address variables could be decided probabilistically depending on the relative positions, orientations, and scales or intensities. This leads to the so-called *mixed Markov random field* and is beyond the scope of this paper. Mumford and Fridman discussed such cases in other context (see [12]).

For a texton t_1 and its neighbor $t_2 \in \partial t$, we measure five features shown in Figure 6.b, which capture various Gestalt properties:

1. d_c: Distance between two centers, which measures *proximity.*

2. d_m: Gap between two textons, which measures *connectedness and continuation.*

3. α: Angle of a neighbor relative to the main axis of the reference texton. This is mostly useful in quadrants I and III. α/d_c measures the curvature of possible curves formed by the textons, or *co-linearity and co-circularity* in the Gestalt language.

4. γ: Relative orientations between the two textons. This is mostly useful for neighbors in quadrants II and IV and measures *parallelism.*

5. r: Size ratio which denotes the *similarity* of texton sizes. r is the width of t_2 divided by the width of t_1 for neighbors in quadrants I and III and r is length of t_2 divided by the length of t_1 for neighbors in quadrants II and IV.

Thus a total of $4 \times 5 = 20$ pairwise features are computed for each texton plus two features of each texton itself: The orientation θ_j and a two dimensional feature consisting of the scale and stretch (σ_j, τ_j). Following the notation of descriptive models in section (2), we denote these features by

$$\phi^{(k)}(t|\partial t), \quad \text{for } k = 1, ..., 22.$$

We compute 21 one dimensional marginal histograms and a two-dimensional histogram for (σ_j, τ_j), averaged over all textons.

$$H^{(k)}(z) = \sum_{j=1}^{n} \delta(z - \phi^{(k)}(t_j|\partial t_j)), \ \forall k.$$

We denote these histograms by

$$H(\mathbf{T}) = (H^{(1)}, ..., H^{(22)}), \quad \text{and} \quad \mathbf{h}(\mathbf{T}) = \frac{1}{n} H(\mathbf{T}).$$

The vector length of $\mathbf{h}(\mathbf{T})$ is the total number of bins in all histograms. One may choose other features and high order statistics as well. In the vision literature, (Steven, 1978) [33] was perhaps the earliest attempt to characterize spatial patterns using histogram of attributes (See [28] for some examples).

The distribution of $\mathbf{T}$ is characterized by a statistical ensemble in correspondence to the grand-canonical ensemble in Figure 5.c. We call it a *Gestalt ensemble* on a finite lattice Λ as it is the general representation for various Gestalt patterns,

$$\text{A Gestalt ensemble } = \Omega_{gst} = \{\mathbf{T} : p(\mathbf{T}; \beta_o, \boldsymbol{\beta})\}. \tag{9}$$

The Gestalt ensemble is governed by a Gibbs distribution,

$$p(\mathbf{T}; \beta_o, \boldsymbol{\beta}) = \frac{1}{Z} \exp\{-\beta_o n - < \boldsymbol{\beta}, H(\mathbf{T}) >\}, \tag{10}$$

where Z is the partition function, and β_o is a parameter controlling texton density. We can rewrite the vector valued potential functions $\boldsymbol{\beta}$ as energy functions $\beta^{(k)}()$, then we have

$$p(\mathbf{T}; \beta_o, \boldsymbol{\beta}) = \frac{1}{Z} \exp\{-\beta_o n - \sum_{j=1}^{n} \sum_{k=1}^{K=22} \boldsymbol{\beta}^{(k)}(\phi^{(k)}(t_j|t_{\partial j}))\}.$$

This model provides a rigorous way for integrating multiple feature statistics into one probability model, and generalizes existing point processes [34].

The probability $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$ is derived in the Appendix from the Julesz ensemble (or micro-canonical ensemble). We first define a close system with $N >> n$ elements on a lattice Λ, and we assume the density of textons is fixed

$$\lim_{N\to\infty} \frac{N}{|\Lambda|} = \rho, \quad \text{as } N \to \infty, \text{ and } \Lambda \to \mathrm{Z}^2.$$

Thus we obtain a Julesz ensemble on Z^2 [37],

$$\textit{A Julesz ensemble} = \Omega_{jlz} = \{\mathbf{T}_\infty : \mathbf{h}(\mathbf{T}_\infty) = \mathbf{h}_o, N \to \infty, f(\mathbf{T}_\infty; \mathbf{h}_o)\},$$

where $\mathbf{h}_o = (\rho, \mathbf{h})$ is the macroscopic summary of the system state $\mathbf{T}_\infty$. On any finite image, a texton process should be a conditional density of $f(\mathbf{T}_\infty; \mathbf{h}_o)$. There is a one-to-one correspondence between $\mathbf{h}_o = (\rho, \mathbf{h})$ and the parameters $(\beta_o, \boldsymbol{\beta})$ (See Appendix for details).

We can learn the parameters $(\beta_o, \boldsymbol{\beta})$ and select effective features $\phi^{(k)}$ by the descriptive method – the minimax entropy learning paradigm [41]. In the following subsection, we discuss some computational issues as well as experiments for learning $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$, and simulating the Gestalt ensembles.

4.3 Experiment I: Learning and sampling Gestalt ensembles

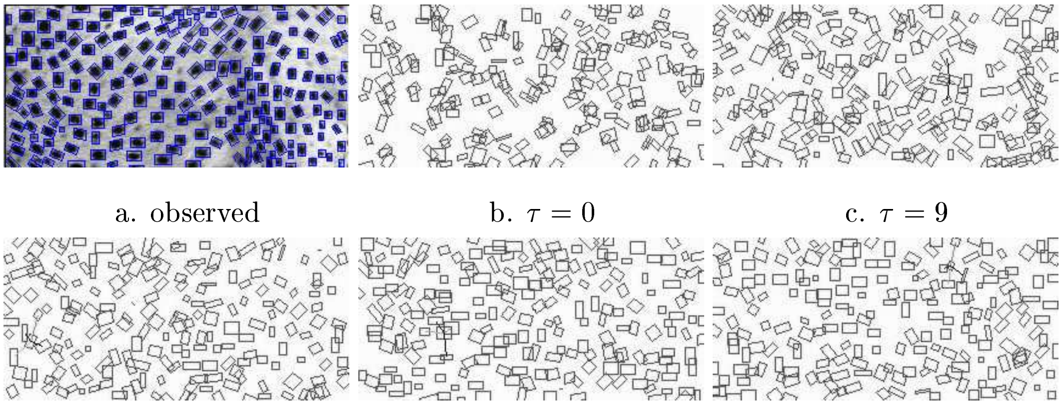
Suppose we have a set of texton maps, $\mathbf{T}_m$ on lattice $\Lambda_m, , m = 1, ..., M$, which are assumed to be independent realizations of the same texton processes. In this section, we assume these texton maps are known and they are manually drawn by a human observer. In the next section, the texton maps are estimated in a Bayesian inference step and thus the learning of the descriptive models for the texton maps shall be integrated with the estimation of the hidden texton maps. As long as the observation is large enough, i.e. $\sum_{m=1}^M |\Lambda_m|$ is large enough, we can estimate a texton model on a standard lattice Λ by the maximum likelihood estimator (MLE),

$$(\beta_o, \boldsymbol{\beta})^* = \arg\max \mathcal{L}(\beta_o, \boldsymbol{\beta}), \qquad \mathcal{L}(\beta_o, \boldsymbol{\beta}) = \sum_{m=1}^M \log p(\mathbf{T}_m; \beta_o, \boldsymbol{\beta}). \tag{11}$$

Thus by steepest ascent, let τ be time steps, we have,

$$\frac{d\beta_o}{d\tau} = \frac{\partial \mathcal{L}}{\partial \beta_o} = \frac{E_p[n]}{|\Lambda|} - \frac{\sum_{m=1}^M n_m}{\sum_{m=1}^M |\Lambda_m|},$$
$$\frac{d\boldsymbol{\beta}}{d\tau} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = E_p[\mathbf{h}(\mathbf{T})] - \frac{1}{M}\sum_{m=1}^M \mathbf{h}(\mathbf{T}_m).$$

Due to the concavity of the log-likelihood with respect to $(\beta_o, \boldsymbol{\beta})$, the solution is unique under mild regularity conditions. The expectation $E_p[n]$ and $E_p[\mathbf{h}(\mathbf{T})]$ often have to be estimated from Monte Carlo simulations as it is the case with texture learning [41].



a. observed b. $\tau = 0$ c. $\tau = 9$

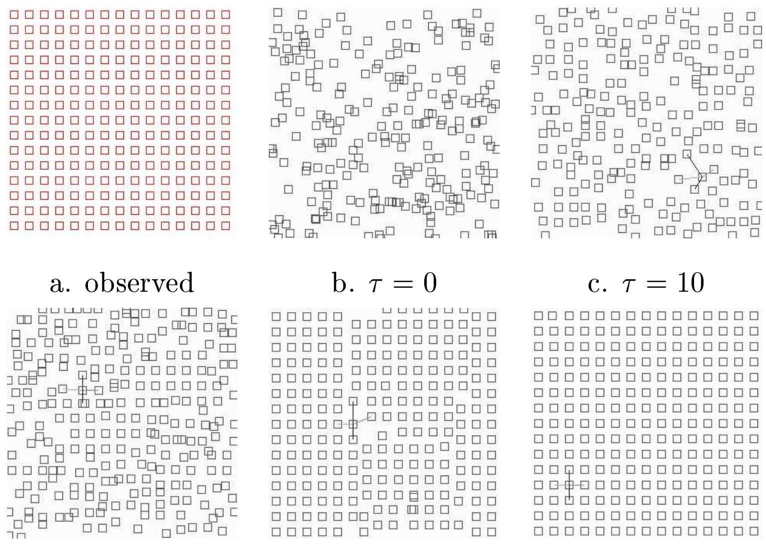
d. $\tau = 69$ e. $\tau = 141$ f. $\tau = 234$

Figure 7 a). The observed image with textons illustrated by the rectangular windows. b)-f) are typical texton maps sampled from a Gibbs model $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$ at various stages $\tau = 0, ..., 234$ of the learning procedure.

There are two different methods for simulating a Gestalt ensemble due to the fundamental link between the micro-canonical (Julesz) and grand-canonical (Gestalt) ensembles. In the first method, one can simulate a Julesz ensemble with a fixed number of textons on a large lattice. A Markov chain Monte Carlo (MCMC) algorithm for sampling a Julesz ensemble of texture images was presented by (Zhu, et al. 2000) [43]. Then different patches of the large synthesized texton map will be used as samples from $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$. The second method samples from $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$ directly and thus the Markov chain should have a death/birth dynamics to adjust the number of textons. We choose the second method because we can learn the parameter simultaneously as we draw samples from the model. Briefly stated, the Markov chain process includes two types of dynamics

1. A death/birth process: This is simulated by a reversible jump [15] that deletes or adds a texton.

2. A diffusion process: This updates the position, orientation, scale, and stretch of the



a. observed b. $\tau = 0$ c. $\tau = 10$

d. $\tau = 18$ e. $\tau = 51$ f. $\tau = 147$

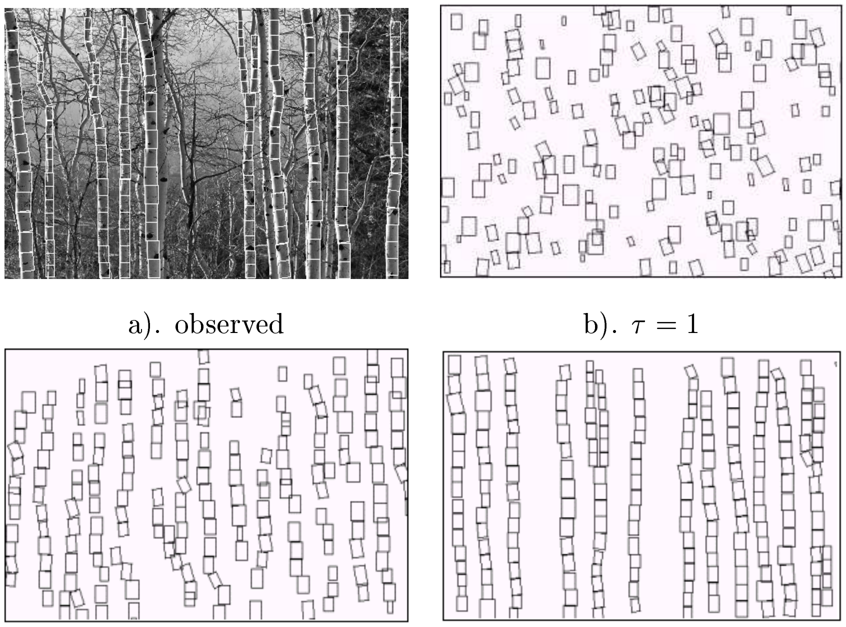
Figure 8 The simulation of a regular grid pattern at various stages $\tau = 0, ..., 147$ of the learning procedure.

textons by Gibbs sampler [13].

We show four typical examples for learning and sampling $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$ in Figures 7-10. The first example in Figure 7 is a cheetah skin pattern with textons (see the rectangles) being the blobs. Figure 7.a) is the observed image with textons illustrated by the rectangular windows. Figure 7.b)-f) are typical texton maps sampled from a Gibbs model $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$ at various stages of the learning procedure. At step $\tau = 234$, the synthesized texton map has statistics close to the observed with $< 5\%$ error in histograms. The spatial arrangements of the cheetah blobs are very random and this pattern is the easiest one among the four example.

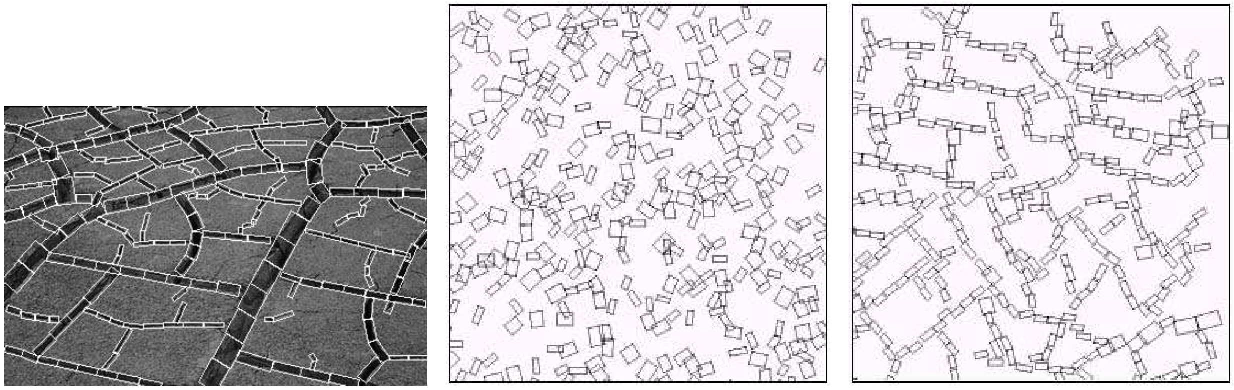
Figure 8 shows a very regular point pattern. It is much harder to simulate this pattern as it is extremely "cold". Thus a special annealing strategy is employed to sample this pattern. In each picture, we show the 4 neighbors for one texton.

Strictly speaking, the wood pattern in Figure 9 and the crack pattern in Figure 10 are not point processes. The textons form lines and curves for the trees and random graphs for the cracks. Thus it is desirable to introduce another layer of representation. In this experiment, we intend to demonstrate that such global curve and graph patterns can still be effectively



c). $\tau = 30$ d). $\tau = 332$

Figure 9 Markov chain Monte Carlo simulation of a woods pattern at various stages $\tau = 0, ..., 332$ of the learning procedure.



a). observed b). $\tau = 1$ c $\tau = 202$

Figure 10 Markov chain Monte Carlo simulation of a crack pattern at various stages $\tau = 0, ..., 202$ of the learning procedure.

characterized by the texton processes through Gestalt models.

The simulated patterns for woods and cracks in Figures 9 and 10 expose two drawbacks of the current texton models. First, the rectangular window representation is too rigid and often leaves some small gaps when two windows are supposed to be aligned seamlessly. To solve this problem, we should introduce more sophisticated texton representation as a linear superposition of wavelet bases. Second, the vertices and junctions in the crack pattern are missing, because we assume all textons play the same role. To solve this problem, we will have to label the textons as edge textons or vertex textons and then define neighborhood for each type of textons respectively. We shall address the two problems in future research.

5 An Integrated Model

After discussing the descriptive models for the hidden texton layers, we now return to the integrated framework presented in section (3).

The generative model for an observed image $\mathbf{I}^{\text{obs}}$ is rewritten from equation (7),

$$p(\mathbf{I}^{\text{obs}};\Theta) = \int p(\mathbf{I}^{\text{obs}}|\mathbf{T}_1,\mathbf{T}_2;\Psi)\prod_{l=1}^{2} p(\mathbf{T}_l;\beta_{lo},\boldsymbol{\beta}_l)d\mathbf{T}_1 d\mathbf{T}_2. \tag{12}$$

We follow the ML-estimate in equation (2),

$$\Theta^* = \arg\max_{\Theta\in\Omega_K^g} \log p(\mathbf{I}^{\text{obs}};\Theta).$$

The parameters Θ include the texton templates Ψ_l, the Lagrange multipliers $(\beta_{lo},\boldsymbol{\beta}_l)$, $l=1,2$ for two Gestalt ensembles, and the variance of the Gaussian noise, σ^2,

$$\Theta = (\Psi,\boldsymbol{\beta},\sigma), \quad \Psi = (\Psi_1,\Psi_2), \quad \text{and} \quad \boldsymbol{\beta} = (\beta_{1o},\boldsymbol{\beta}_1,\beta_{2o},\boldsymbol{\beta}_2).$$

To maximize the log-likelihood, we take the derivative with respect to Θ, and set it to zero. Let $\mathbf{T}=(\mathbf{T}_1,\mathbf{T}_2)$,

$$\begin{aligned}
&\frac{\partial \log p(\mathbf{I}^{\text{obs}};\Theta)}{\partial\Theta} \\
&= \int \frac{\partial \log p(\mathbf{I}^{\text{obs}},\mathbf{T};\Theta)}{\partial\Theta} p(\mathbf{T}|\mathbf{I}^{\text{obs}};\Theta)d\mathbf{T} \\
&= \int [\frac{\partial \log p(\mathbf{I}^{\text{obs}}|\mathbf{T};\Psi)}{\partial\Theta} + \sum_{l=1}^{2}\frac{\partial \log p(\mathbf{T}_l;\boldsymbol{\beta}_l)}{\partial\Theta}]\, p(\mathbf{T}|\mathbf{I}^{\text{obs}};\Theta)\, d\mathbf{T} \\
&= E_{p(\mathbf{T}|\mathbf{I}^{\text{obs}};\Theta)}[\frac{\partial \log p(\mathbf{I}^{\text{obs}}|\mathbf{T};\Psi)}{\partial\Theta} + \sum_{l=1}^{2}\frac{\partial \log p(\mathbf{T}_l;\boldsymbol{\beta}_l)}{\partial\Theta}] \;=\; 0.
\end{aligned}$$

In the literature, there are two well-known methods for solving the above equation. One is the EM algorithm [8], and the other is data augmentation [35] in the Bayesian context. We propose to use a stochastic gradient algorithm [16] which is more effective for our problem.

A Stochastic Gradient Algorithm

– **Step 0**. Initialize the hidden texton maps $\mathbf{T}$ and the templates Ψ using a simplified likelihood as discussed in the next section. Set $\boldsymbol{\beta} = 0$.

Repeat steps I and II below iteratively (like EM-algorithm).

– **Step I**. With the current $\Theta = (\Psi, \boldsymbol{\beta}, \sigma)$, obtain a sample of texton maps from the posterior probability

$$\mathbf{T}_m^{\text{syn}} \sim p(\mathbf{T}|\mathbf{I}^{\text{obs}};\Theta) \propto p(\mathbf{I}^{\text{obs}}|\mathbf{T}_1,\mathbf{T}_2;\Psi)p(\mathbf{T}_1;\beta_{1o},\boldsymbol{\beta}_1)p(\mathbf{T}_2;\beta_{2o},\boldsymbol{\beta}_2),\ m = 1,...,M. \quad (13)$$

This is *Bayesian inference*. The sampling process is realized by a Monte Carlo Markov chain which simulates a random walk with two types of dynamics.[5]

- I.a). A *diffusion dynamics* realized by a Gibbs sampler — sampling (relaxing) the transform group for each texton. For example, move textons, update their scales and rotate them, etc.

- I.b). A *jump dynamics* — adding or removing a texton (death/birth) by reversible jumps [15].

– **Step II**. We treat $\mathbf{T}_m^{\text{syn}}, m = 1,...,M$ as "observations", and estimate the integration in eq. (13).

We learn $\Theta = (\Psi, \boldsymbol{\beta}, \sigma)$ of the texton templates and Gibbs models respectively by gradient ascent:

- II.a). Update the texton templates Ψ by maximizing $\sum_{m=1}^{M} \log p(\mathbf{I}^{\text{obs}}|\mathbf{T}_m^{\text{syn}};\Psi)$; this is a model fitting process. In our experiment, the texton templates Ψ_1, Ψ_2 are represented by 15×15 windows and thus there are 2×225 unknowns.[6] The size of the windows

[5]This sampling process is almost identical to the simulation of the Gestalt ensemble in section (4.3), except that a likelihood $p(\mathbf{I}^{\text{obs}}|\mathbf{T}_1,\mathbf{T}_2;\Psi)$ is engaged in the posterior $p(\mathbf{T}|\mathbf{I}^{\text{obs}};\Theta)$.

[6]Each point in the window can be transparent, and thus the shape of the texton can change during the learning process.

seem adequate for our experiments, but for textures with larger local structures, we need to increase the window size. The transparency of the template is also learned. For each pixel in the foreground template, there is a boolean variable which indicates whether the pixel is transparent or not. Originally for all the pixels in the foreground template the transparency indicator is equal to 0. If we set the transparency equal to 1 then that pixel is not used in composing the foreground. A Gibbs sampler is used to decide the transparency indicators.

- II.b). Update $\beta_{lo}, \boldsymbol{\beta}_l, l = 1, 2$ by maximizing $\sum_{m=1}^{M} \log p(\mathbf{T}_m^{\mathrm{syn}}; \beta_{lo}, \boldsymbol{\beta}_l)$. This is exactly the maximum entropy learning process in the descriptive method (see eq. (11)) except that the texton processes are given by step I.

- II.c). Update σ for the noise process.

In step I, we choose to sample $M = 1$ example each time. There are two reasons for this choice. 1) The images are usually quite large and stationary, therefore, spatial averaging for one image already has large sample effect. 2) The iterative algorithm is cumulative. If the learning rate in steps II.a) and II.b) is slow enough, then the long run behavior also exhibits large sample effect. It has been proved in statistics [16] that such an algorithm converges to the optimal Θ if the step size in step II satisfies some mild conditions.

The following are some useful observations.

1. Descriptive model is part of the integrated learning framework, in terms of both representation and computing (Step II.b)).

2. Bayesian vision inference is a sub-task (step I) of the integrated learning process. A vision system, machine or biological, evolves by learning generative models $p(\mathbf{I}; \Theta)$ and makes inference about the world $\mathbf{T}$ using the current imperfect knowledge Θ – the Bayesian view of vision. What are missing in this learning paradigm are "discovery process" that introduces new hidden variables.

In this paper, we separate the learning of the templates Φ and the learning of $\boldsymbol{\beta}$ for computational efficiency. That is, we iterate Steps I and II while fixing $\beta_{lo} = 2.0$ and $\boldsymbol{\beta}_l = 0$, i.e., we only control the density of the textons. After that, we learn $\boldsymbol{\beta}_l$ based on the sampled texton maps, while keeping the learned Φ fixed.

6 Effective Inference by Simplified Likelihood

In this section, we address some computational issues in the integrated model, and propose a method for initializing the stochastic gradient algorithm (in step 0).

6.1 Initialization by likelihood simplification and clustering

The stochastic algorithm presented in the above section needs a long "burn-in" period if it starts from an arbitrary condition. To accelerate the computation, we use a simplified likelihood in step 0 of the stochastic gradient algorithm. Thus given an input image $\mathbf{I}^{\rm obs}$, our objective is to compute some good initial texton templates Ψ_1, Ψ_2 and hidden texton maps $\mathbf{T}_1, \mathbf{T}_2$, before the iterative process in steps I and II.

A close examination reveals that the computational complexity is largely due to the complex coupling between the textons in both the generative model $p(\mathbf{I}|\mathbf{T}_1, \mathbf{T}_2; \Psi)$ and the descriptive models $p(\mathbf{T}_1; \beta_{1o}, \boldsymbol{\beta}_1)$ and $p(\mathbf{T}_2; \beta_{2o}, \boldsymbol{\beta}_2)$. Thus we simplify both models by decoupling the textons.

Firstly, we decouple the textons in $p(\mathbf{T}_1; \beta_{1o}, \boldsymbol{\beta}_1)$ and $p(\mathbf{T}_2; \beta_{2o}, \boldsymbol{\beta}_2)$. We fix the total number of textons $n_1 + n_2$ to an excessive number, thus we do not need to simulate the death-birth process. We set $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ to 0, therefore $p(\mathbf{T}_l; \beta_{lo}, \beta_l)$ becomes a uniform distribution and the texton elements are decoupled from spatial interactions.

Secondly, we decouple the textons in $p(\mathbf{I}_{obs}|\mathbf{T}_1, \mathbf{T}_2; \Psi)$. Instead of using the image generating model in eq. (5) which implicitly imposes couplings between texton elements through eq. (8), we adopt a constraint-based model

$$p(\mathbf{I}^{\rm obs}|\mathbf{T}, \Psi) \propto \exp\{-\sum_{l=1}^{2}\sum_{j=1}^{n_l} ||\mathbf{I}^{\rm obs}_{D_{lj}} - G[T_{lj}] \odot \Psi_l||^2/2\sigma^2\}, \qquad (14)$$

where $\mathbf{I}^{\rm obs}_{D_{lj}}$ is the image patch of the domain D_{lj} in the observed image. For pixels in $\mathbf{I}^{\rm obs}$ not covered by any textons, a uniform distribution is assumed to introduce a penalty.

We run the stochastic gradient algorithm on the decoupled log-likelihood, which reduces to a conventional clustering problem. We start with two random texton maps and the algorithm iterates the following two steps.

I). Given Ψ_1 and Ψ_2, the algorithm runs a Gibbs sampler to change each texton t_{lj} respectively, by moving, rotating, scaling and stretching the rectangle, and changing the cluster into which each texton falls according to the simplified model of eq. (14). Thus the texton windows intend to cover the entire observed image, and at the same time try to form tight clusters around Ψ.

II). Given $\mathbf{T}_1$ and $\mathbf{T}_2$, the algorithm updates the texton Ψ_1 and Ψ_2 by averaging

$$\Psi_l = \frac{1}{n_l} \sum_{j=1}^{n_l} G^{-1}[T_{lj}] \odot \mathbf{I}^{\mathrm{obs}}_{D_{lj}}, \quad l = 1, 2,$$

where $G^{-1}[T_{lj}]$ is the inverse transformation. The layer order d_1 and d_2 are not needed for the simplified model.

This initialization algorithm for computing $(\mathbf{T}_1, \mathbf{T}_2, \Psi_1, \Psi_2)$ resembles the transformed component analysis [11]. It is also inspired by a clustering algorithm by (Leung and Malik, 1999) [23], which did not engage hidden variables, and thus compute a variety of textons Ψ at different scales and orientations. See also the work of Miller (2002) [29]. We also experimented with representing the texton template Ψ by a set of Gabor bases instead of a 15×15 window. However, the results were not as encouraging as in this generative model.

6.2 Experiment II: Texton clustering

In this subsection, we present one experiment for initialization and clustering using the method outlined in section (6.1).

Figure 11 shows an experiment on the initialization algorithm for a crack pattern. 1055 textons are used with the template size of 15×15. The number of textons is as twice as necessary to cover the whole image. In optimizing the likelihood in eq. (14), an annealing scheme is utilized with the temperature decreasing from 4 to 0.5. The sampling process converges to a result shown in Figure 11.

Figure 11.a) is the input image; Figure 11.b) and Figure 11.d) are the texton maps $\mathbf{T}_1$ and $\mathbf{T}_2$ respectively. Figure 11.c and Figure 11.e are the cluster centers Ψ_1 and Ψ_2, shown by rectangles respectively. Figure 11.f is the reconstructed image. The results demonstrate that the clustering method provides a rough but reasonable starting solution for generative modeling.

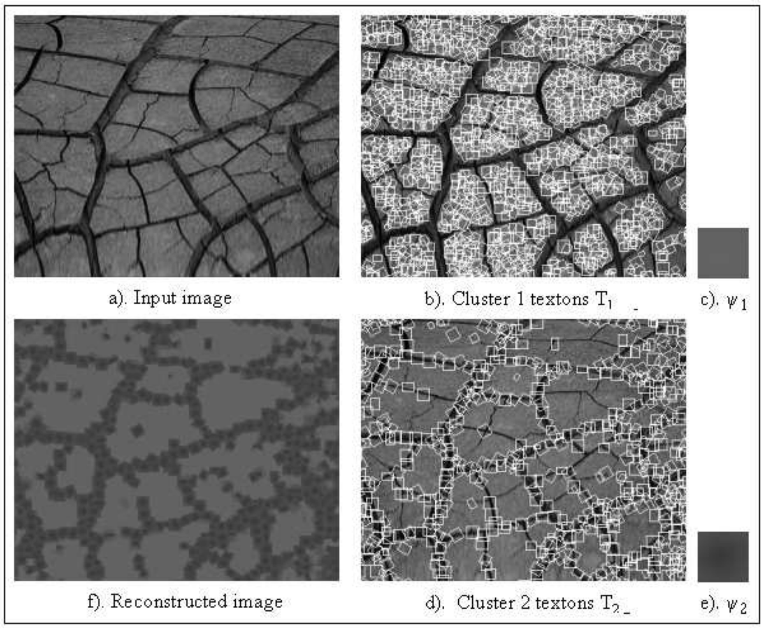


Figure 11 Result of the initial clustering algorithm, which provides a rough but reasonable starting solution for generative modeling. The initial clustering algorithm simplifies the models by decoupling the textons to accelerate the computation.

7 Experiment III: Integrated learning and synthesis

In this section, we show some experimental results obtained by the integrated model. For an input image, we first do a clustering step as section (6) showed. Then we run the stochastic gradient algorithm on the full models to refine the clustering results.

Figure 12 shows the result for the crack image obtained by the stochastic gradient algorithm, which took about 80 iterations of the two steps (Step I and Step II), following the initial solution (Step 0) shown in Figure 11. Figure 12.b) and Figure 12.d) are the background and foreground texton maps $\mathbf{T}_1$ and $\mathbf{T}_2$ respectively. Figure 12.c) and Figure 12.e) are the learned textons Ψ_1, Ψ_2 respectively. Figure 12.f) is the reconstructed image from learned texton maps and templates. Compared to the results in Figure 11, the results in Figure 12 have more precise texton maps and accurate texton templates due to an accurate generative model. The foreground texton Ψ_2 is a bar, and one pixel at corner of the left-top is transparent.

The integrated learning results for a cheetah skin image are shown in Figure 13. It can

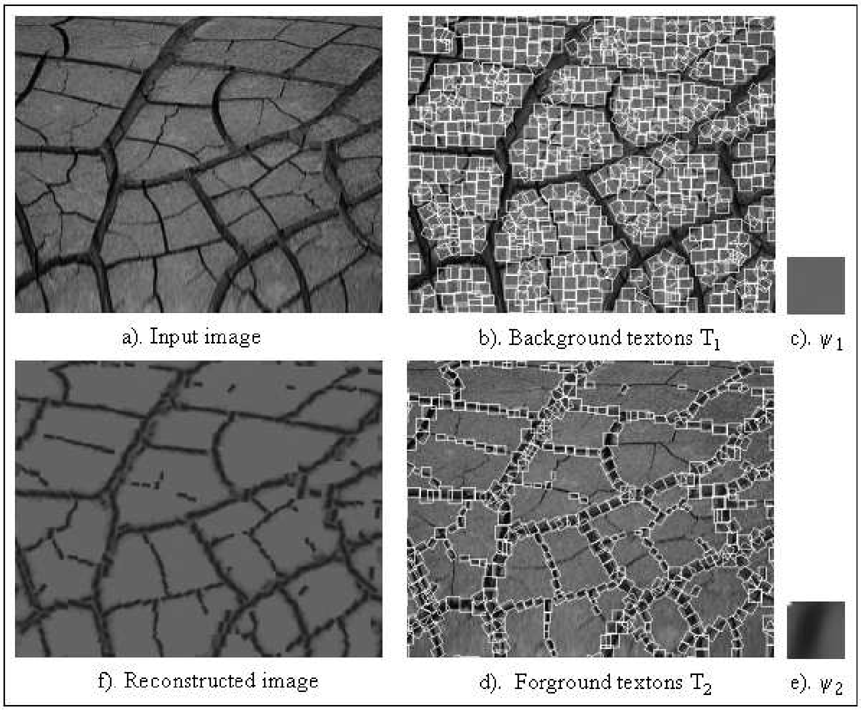


Figure 12 Generative model learning result for the crack image. a) input image, b) and d) are background and foreground textons discovered by the generative model, c) and e) are the templates for the generative model, f) is the reconstructed image from the generative model. Due to an accurate generative model, the results after learning have more precise texton maps and accurate texton templates compared to the initial results in Figure 11.

be seen that in the foreground template, the surround pixels are learned as being transparent and the blob is exactly computed as the texton. Figure 14 are the results for a brick image. No point in the template is transparent for the gap lines between bricks.

Figure 15 shows the learning of another short crack patterns. Figure 16 displays a pine corn pattern. The seeds and the black intervals are separated cleanly, and the reconstructed image keeps most of the pine structures. However the pine corn seeds are learnt as the background textons and the gaps between pine corns are treated as foreground textons.

We also do one experiment on a bark image (Figure 17). The result shows that the details of the bark are not modeled well. For such patterns, the linear superposition of the templates might do a better job. We shall investigate this issue in our future work.

We extend our model to three layers, i.e. $L = 3$ and do one experiment on a pattern of text (Figure 18), which has white background and two type of letters as foreground. Figure 19 shows the learning process. Three templates - white background, letter 'A' and letter 'B' were inferred gradually.

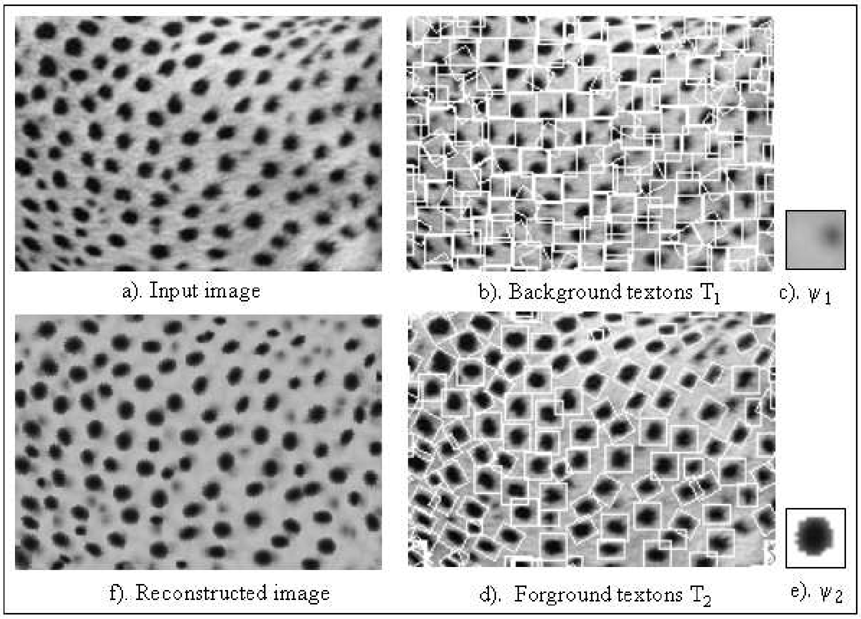


Figure 13 Generative model learning result for a cheetah skin image. See Figure 12 caption for explanations.

After the parameters Ψ and β of a generative model are estimated, new random samples could be drawn from the generative model. This proceeds in three steps: First, texton maps

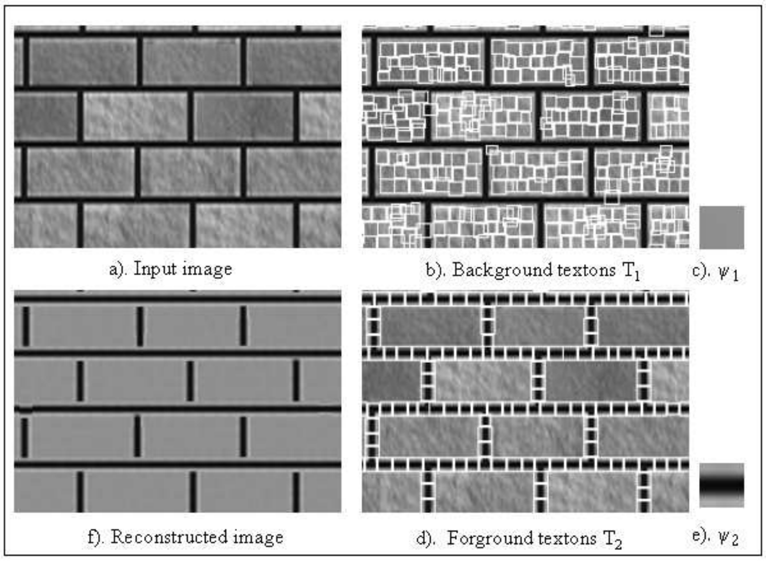


Figure 14 Generative model learning result for a brick image. See Figure 12 caption for explanations.

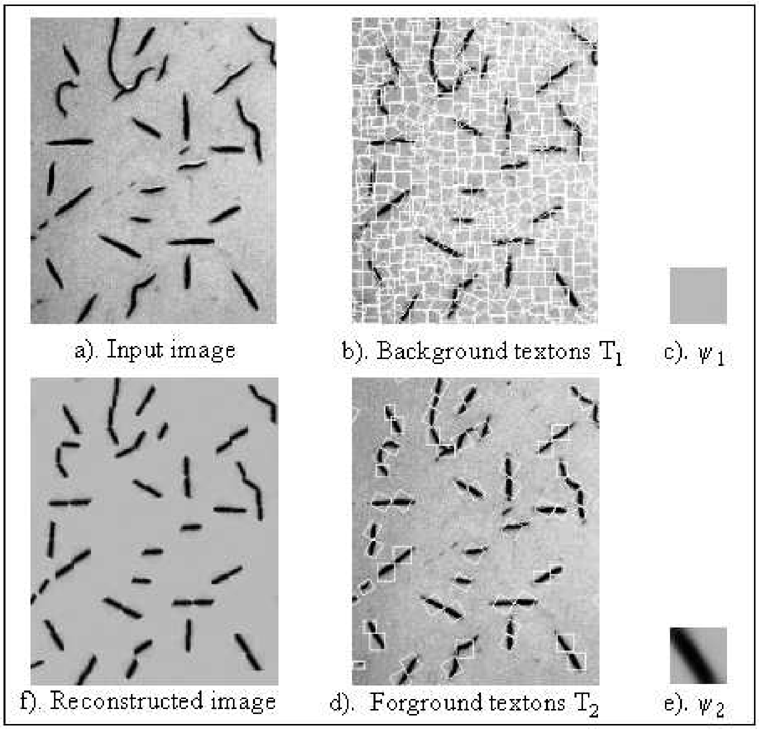


Figure 15 Generative model learning result for a crack image. See Figure 12 caption for explanations.

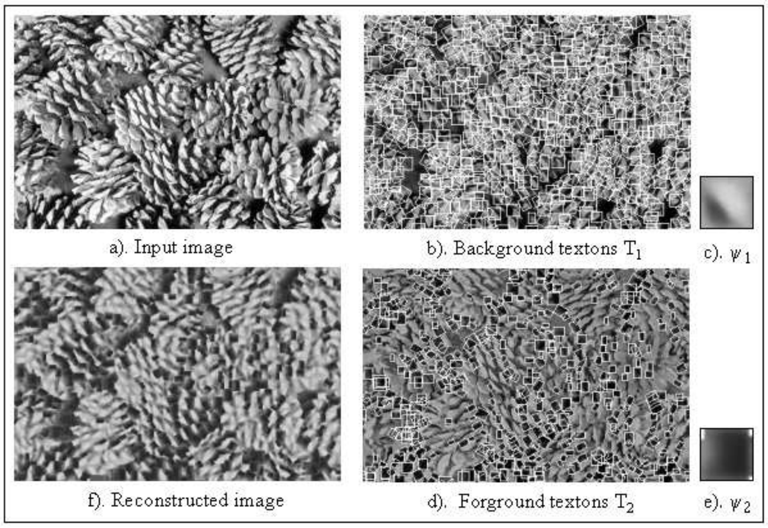


Figure 16 Generative model learning result for a pine corn image. See Figure 12 caption for explanations.

are sampled from the Gibbs models $p(\mathbf{T}_1; \boldsymbol{\beta}_1)$ and $p(\mathbf{T}_2; \boldsymbol{\beta}_2)$ respectively. Second, background and foreground images are synthesized from the texton maps and texton templates. Third, the final image is generated by combining these two images according to the occlusion model.

We show synthesis experiments on three patterns.

1. Figure 20 and Figure 21 are two synthesis examples of the two layered model for the cheetah skin pattern. The templates used here are the learned results in Figure 13.

2. Figure 22 shows texture synthesis for the crack pattern computed in Figure 15.

3. Figure 23 displays texture synthesis for the brick pattern in Figure 14. To capture the vertical and horizontal distances of the brick, we add four more neighbors in addition to four nearest neighbors to the feature space. The new four neighbors are those nearest neighbors which have the same orientation as the concerned texton. The T-junctions are not captured because we do not have such feature statistics.

Note that, in these texture synthesis experiments, the Markov chain operates with meaningful textons instead of pixels.

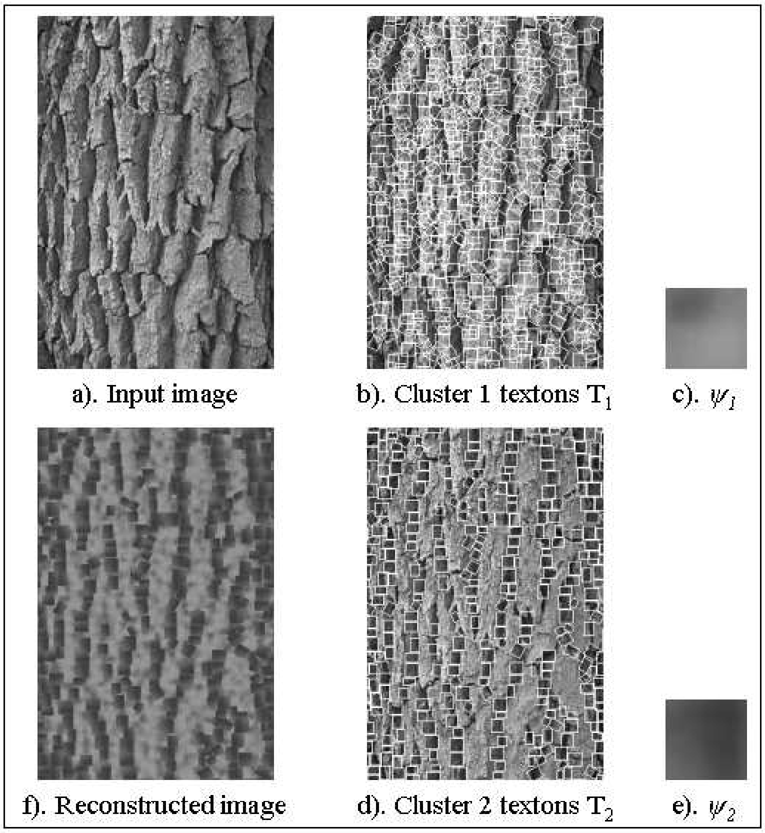


Figure 17 Generative model learning result for a bark image. The details of the bark are not modeled well by our current generative model.

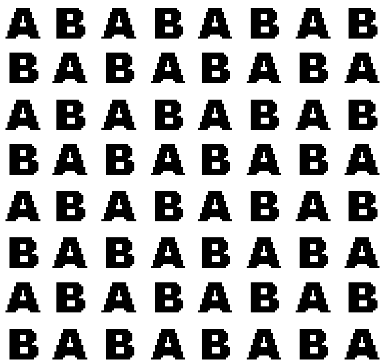


Figure 18 A text file with two foreground letters to test our model on three layers textons.

8 Discussion

In this paper, we present a class of statistical models for visual patterns. The models integrate and extend descriptive and generative methods, and provide a mathematical definition for

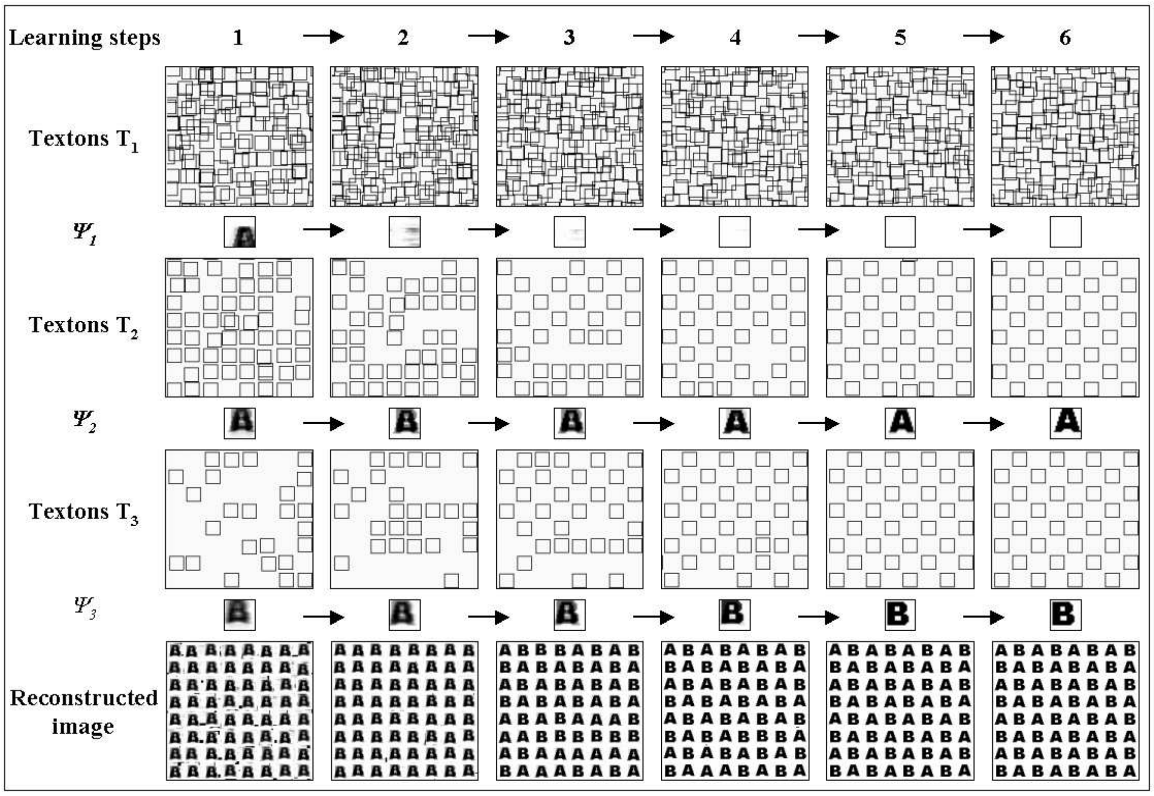


Figure 19 Generative model learning result for a text image. Six main steps are shown to illustrate the improving of textons and templates with learning.

textons and their perceptual organizations. The hierarchical model can be considered as a generalization of the hidden Markov model, and the hidden Markov structure is non-causal in our model.

The model has some advantages over previous pure descriptive method with Markov random fields on pixel intensities. First, from the representational perspective, the neighborhood in the texton map are much smaller than the pixel neighborhood in FRAME model [41]. The generative method captures more semantically meaningful elements on the texton maps. Second, from the computational perspective, the Markov chain operating the texton maps can move textons according to affine transforms and can add or delete a texton by death/birth dynamics, thus it is much more effective than the Markov chain used in traditional Markov random fields which flips one pixel intensity at a time.

We show that the integration of descriptive and generative methods is a natural path

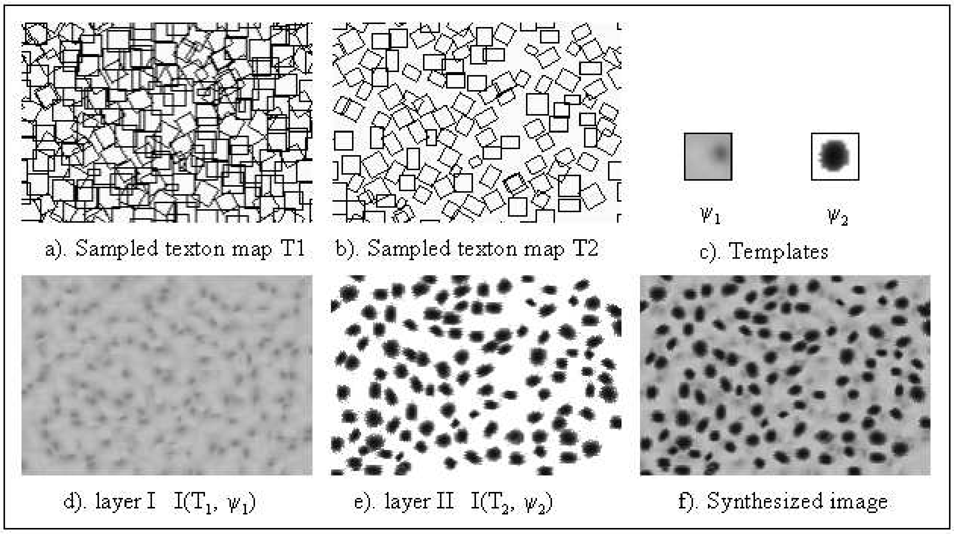


Figure 20 An example of a randomly synthesized cheetah skin image. a) and b) are the background and foreground texton maps respectively sampled from $p(\mathbf{T}_l; \beta_{lo}, \boldsymbol{\beta}_l)$; d) and e) are synthesized background and foreground images from the texton map and templates in c); f) is the final random synthesized image from the generative model.

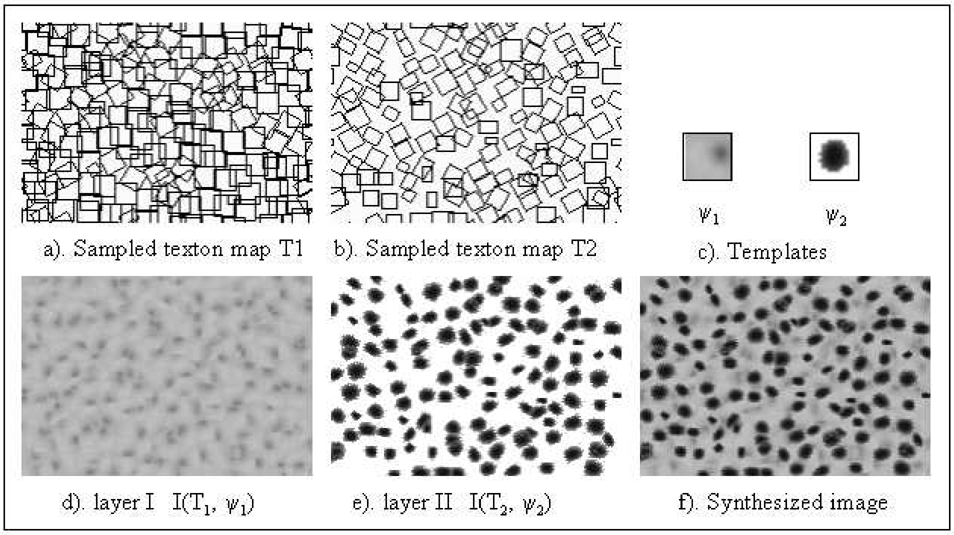


Figure 21 Second example of a randomly synthesized cheetah skin image.

for visual learning. We argue that a vision system could evolve by progressively replacing descriptive models with generative models, which realizes a transition from *empirical and statistical models* to *physical and semantical* models.

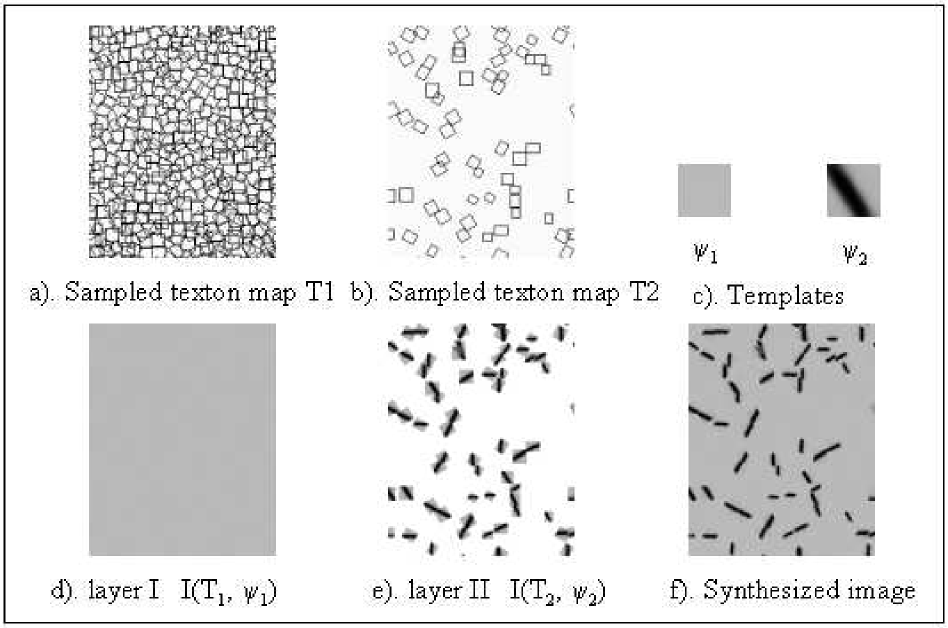


Figure 22 An example of a randomly synthesized crack image. See Figure 20 notation for explanations.

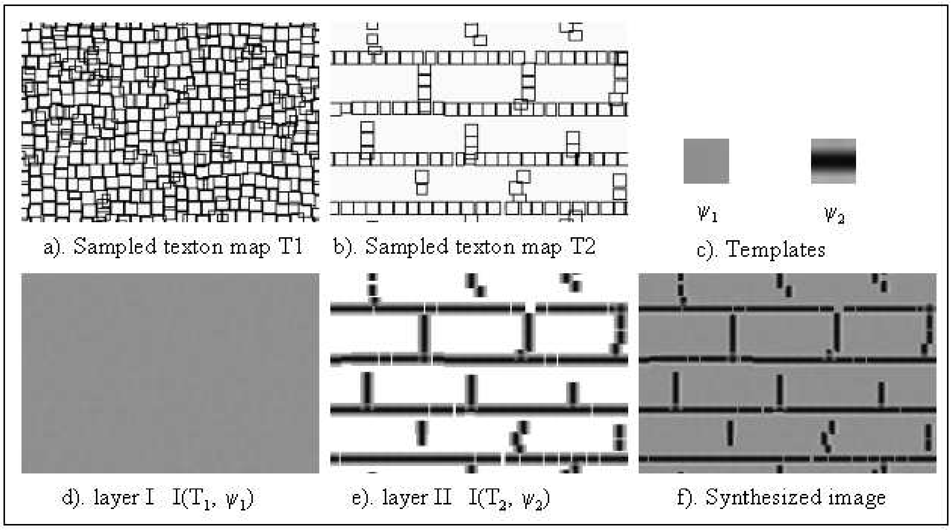


Figure 23 An example of a randomly synthesized brick image. See Figure 20 notation for explanations.

The following are important issues that should be addressed in future research.

First, the Gestalt model based on nearest neighbors is too simple for many spatial patterns. We need to introduce more descriptive feature statistics for descriptive modeling, or replace it with more abstract concepts such as curves and graphs as another hierarchy of generative model. We also need to explore more efficient inference and synthesis algorithms for Gestalt model.

Second, the model for local textons based on image windows is quite limited. In a recent paper [40], we explore combination of linear bases, and local shape and shading models. We also explore motion elements. But there is still much work to be done in order to find good local descriptors in term of generative models.

Third, some texture patterns (like foliage) are intrinsically complex (e.g., with a huge number of leaves), so that there may not exist low dimensional sparse representation in terms of textons. Such patterns may have to be modeled by the descriptive FRAME model [41]. On the other hand, some patterns may contain clear textons amid stochastic background (like twigs and straws), and in that case, the noise in the generative part of the model should be replaced by FRAME model [41].

Acknowledgments

The authors would like to thank the three reviewers for their helpful comments. This work is supported partially by NSF grants IIS 98-77-127, IIS-00-92-664, and DMS-0072538, and an ONR grant N000140-110-535.

References

[1] N. Ahuja and M. Tuceryan, "Extraction of early perceptual struct. in dot patterns", *CVGIP*, **48**, 1989.

[2] A. J. Bell and T. J. Sejnowski, "The independent components of natural images are edge filters", *Vision Research*, 37:3327-3338, 1997.

[3] J. Bergen and E. Adelson, "Early vision and texture perception", *Nature*, 333, 363-364, 1988.

[4] D. Chandler, *Introduction to Modern Statistical Mechanics*, Oxford University Press, 1987.

[5] S. Chen, D. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit", *SIAM Journal on Scientific Computing*, 20(1):33-61, 1999.

[6] T. Cover and Thomas, *Elements of Information Theory*, 1994.

[7] J. S. De Bonet and P. Viola, "A non-parametric multi-scale statistical model for natural images", *Advances in Neural Information Processing*, 10, 1997.

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

[9] R. Duda, P. Hart, and D. Stork, *Pattern Classification and Scene Analysis*, second edition, ohn Wiley & Sons, 2000.

[10] A. A. Efros and W. T. Freeman, "Image Quilting for Texture Synthesis and Transfer", SIGGRAPH 2001.

[11] B. Frey and N. Jojic, "Transformed component analysis: joint estimation of spatial transforms and image components", *ICCV*, 1999.

[12] A. Fridman, *Mixed Markov Models*, Doctoral dissertation, Division of Applied Math, Brown University. 2000.

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

[14] W. R. Gilks and R. O. Roberts, "Strategies for improving MCMC", chapter 6 in W. R. Gilks et al (eds) *Markov Chain Monte Carlo in Practice*, Chapman & Hall, 1997.

[15] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination", *Biometrika*, vol. 82, 711-732, 1995.

[16] M. G. Gu, "A stochastic approximation algorithm with MCMC method for incomplete data estimation problems", *Preprint*, Dept. of Math. and Stat., McGill Univ. 1998.

[17] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis". *SIGGRAPHS*, 1995.

[18] E. T. Jaynes, "Information theory and statistical mechanics", *Physical Review* 106, 620-630, 1957.

[19] B. Julesz, "Textons, the elements of texture perception and their interactions", *Nature*, 290, 91-97, 1981.

[20] K. Koffka, *Principles of Gestalt Psychology*, 1935.

[21] A B. Lee, D. B. Mumford, and J. G. Huang, "Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model", *Int'l J. of Computer Vision*, vol. 41, no. 1/2, pp 35-59, 2001.

[22] T. Leung and J. Malik, "Detecting, Localizing and Grouping Repeated Scene Elements from an Image", *Proc. 4th ECCV*, Cambridge, UK, 1996.

[23] T. Leung and J. Malik, "Recognizing surface using three-dimensional textons", *Proc. of 7th ICCV*, Corfu, Greece, 1999.

[24] M. S. Lewicki and B. A. Olshausen, "A probabilistic framework for the adaptation and comparison of image codes", *JOSA*, A. 16(7): 1587-1601, 1999.

[25] J. Malik, and P. Perona, "Preattentive texture discrimination with early vision mechanisms", *J. of Optical Society of America A*, vol 7. no.5, May, 1990.

[26] J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, Contours and Regions: Cue Integration in Image Segmentation", *ICCV*, 1999.

[27] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary", *IEEE trans. on Signal Processing*, vol.41, pp3397-3415, 1993.

[28] D. Marr, *Vision*, W.H. Freeman and Company, 1982.

[29] E. G. Miller, "Learning from One Example in Machine Vision by Sharing Probability Densities. " Ph.D. Thesis, Massachusetts Institute of Technology, 2002.

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

[31] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models", *Neural Computation*, vol. 11, no. 2, 1999.

[32] J. Portilla and E. P. Simoncelli, " A parametric texture model based on joint statistics of complex wavelet coefficients", IJCV, 40(1), 2000.

[33] K. A. Steven, "Computation of locally parallel structure", *Biol. Cybernetics*, 29, pp19-28, 1978.

[34] D. Stoyan, W. S. Kendall, J. Mecke, *Stochastic Geometry and its Applications*, 1985.

[35] M. Tanner, *Tools for Statistical Inference*, Springer, 1996.

[36] Z. Tu and S. C. Zhu, “Image Segmentation by Data-Driven Markov Chain Monte Carlo”, *IEEE Trans. PAMI*, vol. 24, no.5, May, 2002.

[37] Y. N. Wu, S. C. Zhu, and X. W. Liu, “Equivalence of Julesz and Gibbs Ensembles”, *ICCV*, 1999.

[38] Y. N. Wu, S. C. Zhu, and C. Guo (2002), ”Statistical Modeling of Texture Sketch”, ECCV, 2002.

[39] Y. Q. Xu, B. N. Guo, and H.Y. Shum, “Chaos Mosaic: Fast and Memory Efficient Texture Synthesis”, MSR TR-2000-32, April, 2000.

[40] S. C. Zhu, C. Guo, Y. N. Wu and Y. Wang, “What are Textons?” *ECCV*, 2002

[41] 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.

[42] S. C. Zhu, “Embedding Gestalt Laws in Markov Random Fields”, *IEEE Trans. PAMI.* vol. 21, no.11, 1999.

[43] S.C. Zhu, X.W. Liu, and Y.N. Wu, “Exploring Julesz Texture Ensemble by Effective Markov Chain Monte Carlo”, *IEEE Trans. PAMI*, vol. 22, no.6, June, 2000.

[44] S. C. Zhu and Cheng-en Guo, “Conceptualization and modeling of visual patterns”, *Proc. of 3rd Int’l Workshop on Perceptual Organization in Computer Vision*, Vancouver, Canada, July, 2001.

Appendix: Deriving the Gibbs model for texton process.

A texton pattern on a large lattice $\Lambda \to \mathrm{Z}$ is summarized by a Julesz ensemble (or micro-canonical ensemble),

$$\Omega_\Lambda(N, \mathbf{H}) = \{\mathbf{T}_\Lambda : N(\mathbf{T}_\Lambda) = N, \mathbf{H}(\mathbf{T}_\Lambda) = \mathbf{H}\}$$

where Λ is a large lattice (or more rigorously, $\Lambda \to \mathrm{Z}$), $\mathbf{T}_\Lambda$ is the texton map defined on lattice Λ, with $N(\mathbf{T}_\Lambda)$ being the number of textons on $\mathbf{T}_\Lambda$, and $\mathbf{H}(\mathbf{T}_\Lambda)$ the collection of the 22 histograms of Gestalt features. N and $\mathbf{H}$ are two parameters that defines the Julesz ensemble $\Omega_\Lambda(N, \mathbf{H})$.

Now, suppose we look at all the large texton maps $\mathbf{T}_\Lambda$ in the Julesz ensemble $\Omega_\Lambda(N, \mathbf{H})$ through a small window $\Lambda_0 \subset \Lambda$, and we are interested in the frequency distribution of all the small texton maps that we see from this window. This frequency distribution is called the Gestalt ensemble (or the grand-canonical ensemble). In probabilistic language, let $\mathbf{T}_\Lambda$ be a random texton map sampled from the uniform distribution over the Julesz ensemble $\Omega_\Lambda(N, \mathbf{H})$, and let $\mathbf{T}_{\Lambda_0}$ be the part of the large $\mathbf{T}_\Lambda$ on the small lattice Λ_0, then we are interested in the probability distribution of $\mathbf{T}_{\Lambda_0}$.

For a $\mathbf{T}_\Lambda \in \Omega_\Lambda(N, \mathbf{H})$, if $\mathbf{T}_{\Lambda_0} = \mathbf{T}_0$ for a specific $\mathbf{T}_0$, then $N(\mathbf{T}_{\Lambda \setminus \Lambda_0}) = N - N(\mathbf{T}_0)$ and $\mathbf{H}(\mathbf{T}_{\Lambda \setminus \Lambda_0}) = \mathbf{H} - \mathbf{H}(\mathbf{T}_0)$, where $\Lambda \setminus \Lambda_0$ is the rest of the lattice. Clearly, the number of large texton maps in $\Omega_\Lambda(N, \mathbf{H})$ with $\mathbf{T}_0$ on Λ_0 is the same as the number of textons maps $\mathbf{T}_{\Lambda \setminus \Lambda_0}$ in $\Omega_{\Lambda \setminus \Lambda_0}(N - N(\mathbf{T}_0), \mathbf{H} - \mathbf{H}(\mathbf{T}_0))$. Therefore, the frequency of $\mathbf{T}_0$

$$p(\mathbf{T}_0) \propto |\Omega_{\Lambda \setminus \Lambda_0}(N - N(\mathbf{T}_0), \mathbf{H} - \mathbf{H}(\mathbf{T}_0))|,$$

A Taylor expansion of $\log p(\mathbf{T}_0)$ at $(N, \mathbf{H})$ gives

$$\begin{aligned}
\log p(\mathbf{T}_0) &= C - \frac{\partial \log |\Omega_{\Lambda \setminus \Lambda_0}(N, \mathbf{H})|}{\partial N} N(\mathbf{T}_0) - \frac{\partial \log |\Omega_{\Lambda \setminus \Lambda_0}(N, \mathbf{H})|}{\partial \mathbf{H}} H(\mathbf{T}_0) \\
&= C - \beta_0 N(\mathbf{T}_0) - \boldsymbol{\beta} \mathbf{H}(\mathbf{T}_0),
\end{aligned}$$

where C is a constant, β_0 and $\boldsymbol{\beta}$ are identified with the derivatives of the log of the volumes of the Julesz ensemble $\Omega_\Lambda(N, \mathbf{H})$ with respect to N and $\mathbf{H}$. Therefore, the Gibbs form of the $p(\mathbf{T}_0)$ is derived.