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

Visual Learning By Integrating Descriptive and Generative Methods

Cheng-en Guo and Song-Chun Zhu
Department of Computer and Information Science
Ohio State University
{cguo, szhu}@cis.ohio-state.edu

Ying Nian Wu
Department of Statistics
University of California, Los Angeles
ywu@stat.ucla.edu

Abstract

This paper presents a mathematical framework for visual learning that integrates two popular statistical learning paradigms in the literature: I). Descriptive methods, such as Markov random fields and minimax entropy learning[35], and II). Generative methods, such as principal component analysis, independent component analysis[2], and transformed component analysis[10], wavelet coding[26, 5], and sparse coding[28, 23]. The integrated framework creates richer classes of probabilistic models for visual patterns. In this paper, we demonstrate the integrated framework by learning a class of hierarchical models for texton patterns (the term of textons 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, 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 or global organizations of the textons in the texton maps are characterized by minimax entropy principle, which leads to embellished versions of Gibbs point process models[32]. The top level of the model is descriptive in nature. The learning framework achieves four goals: i). Computing the window functions (or appearances) of different types of textons. ii). Inferring the hidden texton maps that generate the image. iii). Learning Gibbs point process models for the texton maps. iv). Verifying the learned window functions and Gibbs models through texture synthesis by stochastic sampling. We use a stochastic gradient algorithm for inferential computation. We demonstrate the learning framework 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 "understands" the contents of the observed images. Thus in computer vision and especially Bayesian image analysis, an important research theme is visual learning whose objective is to construct parsimonious and general models that can realistically characterize visual patterns in natural scenes. Due to the stochastic nature of visual patterns, visual learning is posed as a statistical modeling and inference problem, and existing methods in the literature 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 model a visual pattern by imposing statistical constraints on features extracted from signals. Descriptive methods include Markov random fields, minimax entropy learning[35], deformable models, etc. For example, recent methods on texture modeling all fall into this category[16, 35, 7, 30]. 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 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 in nature. Generative methods are widely used in vision and image analysis. For example, principle component analysis (PCA), independent component analysis (ICA)[2], transformed component analysis (TCA)[10], wavelet image representation[26, 5], sparse coding[28, 23],

[1] There is a third category of methods which are discriminative. The goal of discriminative methods is not for modeling visual patterns explicitly but for classification. For example, pattern recognition, feed-forward neural networks and classification trees etc. Thus we choose not to discuss it because our focus is statistical modeling.

and the random collage model for generic natural images[20]. Despite their simplicity, these generative models suffer from an over-simplified assumption that hidden variables are independent and identically distributed (iid).[2] As a result, they are not sophisticated enough to generate realistic visual patterns. For example, a wavelet image coding model can easily reconstruct an observed image, but it cannot synthesize a texture pattern through iid random sampling because the spatial relationships between the wavelet coefficients are not characterized.

The two learning paradigms were developed almost independently by somewhat disjoint communities working on different problems, and their relationship has yet to be studied. In this paper, we present a visual learning framework that integrates both descriptive and generative methods and extends them to a richer class of probabilistic models for computer vision.

The integrated learning framework makes contributions to visual learning in the following four aspects.

Figure 1 Two examples of natural patterns.

First, it combines the advantages of both descriptive and generative methods, and provides a general visual learning framework for modeling complex visual patterns. In computer vision, a fundamental observation, stated in Marr's primal sketch paradigm[27], is that natural visual patterns consist of multiple layers of stochastic processes. For example, Fig. 1 displays two

[2]Interested readers are referred to a recent paper[29] for discussion of the problem with existing generative models.

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 Fig 2. We assume that a texture 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 experiments, each texton covers more than 100 pixels on average, so the layered representation achieves a nearly 100-fold dimension reduction.[3] The spatial arrangements of the textons at each layer are characterized by Markov random field (MRF) models through the minimax entropy learning[35], and previous MRF texture models can be considered special cases where the models have only one layer and each "texton" is just a pixel.

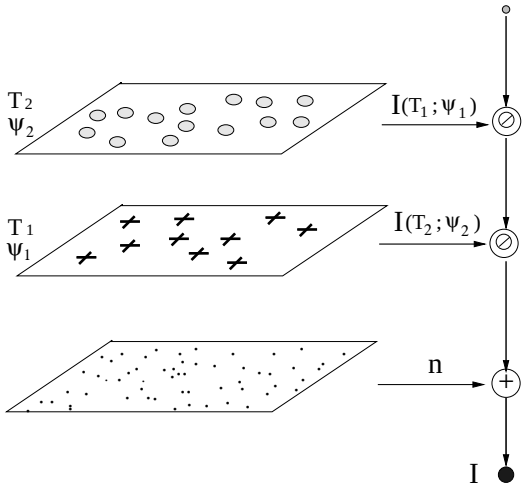


Figure 2 A generative model for an image $\mathbf{I}$ consists of multiple layers of texton maps $\mathbf{I}(\mathbf{T}_l; \Psi_l), l = 1, ..., L$ superimposed with occlusion plus an additive noise image $\mathbf{n}$.

It is our belief that *descriptive models are 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 discovers 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,

[3] A texton has to be described by a few variables for location, scale, orientation etc.

which in turn generate pixel intensities. At each stage, the elements at the most abstract level have no further hidden causes and thus they have to 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[34]. 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. By analogy, the learning process is very similar to the situation in physics, where experimental observations are explained by a hierarchy of elements (say from quarks, electrons, atoms, to molecules) and their interactions.

Second, the integrated learning framework 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[18], and a precise definition of texton has yet to be found. In this paper, we argue that the definition of "texton" is possible only in the context of a generative model. In contrast to the constraint based clustering method by Malik, Leung etc. [21, 22, 25], in this paper, 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[10], TCA[10] and sparse coding[28, 23].

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 and continuity etc.

Fourth, we adapt a stochastic gradient algorithm [15] for effective learning and inference, in contrast to the conventional EM algorithm [8]. In the adapted algorithm, we simplify the original likelihood function and solves 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 learning method on texture images. For an input texture image, the learning algorithm achieves the following four objectives.

1. Learning the appearance of textons for each stochastic process. Textons of the same

stochastic process are translated, scaled, 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 which 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.

It is worth noting that this is not another paper on texture synthesis, which is a computer graphics problem that has been successfully solved by a variety of texture sampling techniques[16, 35, 7, 30]. Instead this paper is about learning parsimonious and sufficient models for vision tasks, where texture synthesis only serves as a method for model checking.

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

2 Background on Descriptive and Generative Learning

Given a set of images $\mathcal{I} = \{\mathbf{I}_1^{\text{obs}}, ..., \mathbf{I}_M^{\text{obs}}\}$, where $\mathbf{I}_m^{\text{obs}}, m = 1, ..., M$ are considered realizations of some underlying stochastic process governed by a frequency $f(\mathbf{I})$. The objective of visual learning is to estimate a 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],

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

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),

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

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,

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

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 and both are general enough for approximating any distribution f.

The first choice is the exponential family. The exponential family 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 $\mathbf{h}_k^{\text{obs}}$ extracted from $\mathcal{I}$,

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

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[17], 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[35].

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,

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

This family includes all the MRF and minimax entropy models for texture[35]. 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[26, 5], and sparse coding[28, 23] 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_i 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 $W = (a_1, ..., a_K, \mathbf{n})$.[4] The coefficients are assumed to be independently and identically distributed,

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

The norm $\rho = 1$ for sparse coding[28, 23] and basis pursuit[5], and $\rho = 2$ for principal component analysis. Thus we have a simple distribution for W,

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

[4] 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.

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 deadleafs models (see [20] 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

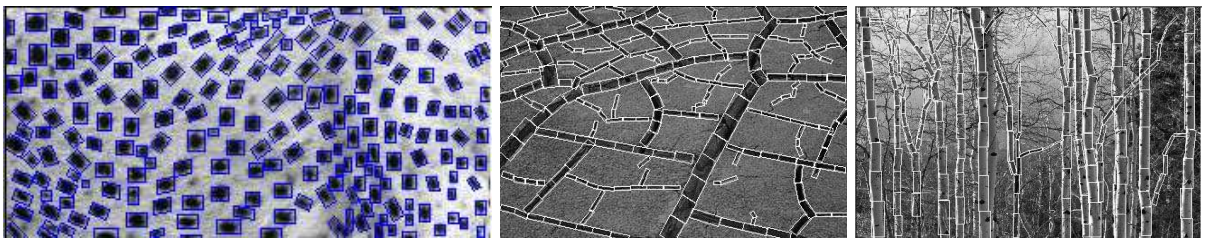


Figure 3 Texture images with texton processes.

We focus on a multi-layer generative model for texture images and we believe that the same learning framework 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". Fig 3 shows three typical examples of texture images, and each texton is represented by a rectangular window. A layered model is shown in Fig 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 "shear" (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, shears, and orientations are shown in Fig. 4.

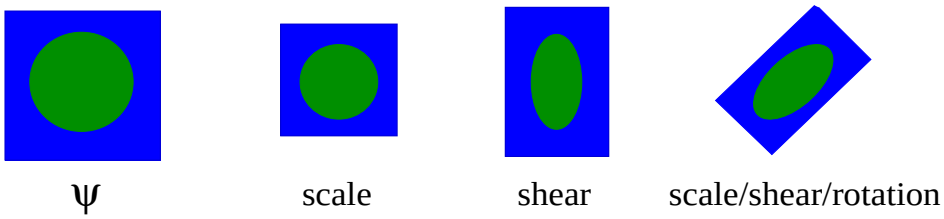


Figure 4 A template Ψ (leftmost) and its three transformed copies.

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

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

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,

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

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 a Gaussian noise process $\mathbf{n}(x, y) \sim N(0, \sigma_0^2), \forall (x, y)$. 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

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

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. We find that this assumption holds true for most of the texture patterns. Otherwise one has to make L as an unknown complexity parameter in the model.

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

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

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

Let $\Psi = (\Psi_1, \Psi_2)$ be texton templates and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ 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 $p(\mathbf{T}_l; \boldsymbol{\beta}_l), l = 1, 2$, which are in much lower dimensional spaces and are more semantically meaningful than previous Gibbs models on pixels[35].

In the next section, we discuss the model $p(\mathbf{T}_l; \boldsymbol{\beta}_l), l = 1, 2$ 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, [5] they must 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.

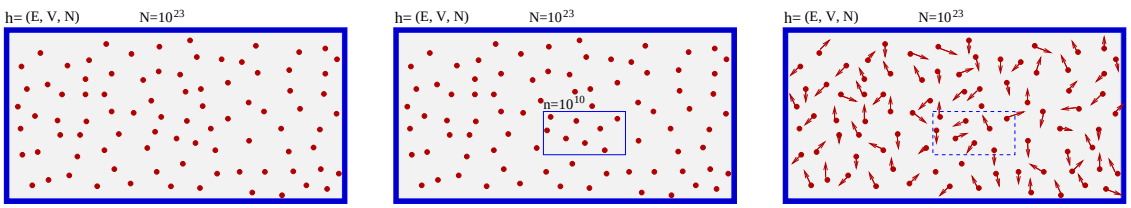
[5] We may introduce additional layers of hidden variables for curve processes that render the textons. But our model stops at the texton level in this paper.

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 $\Lambda \subset \mathrm{Z}^2$.

- A texton process has an unknown number of elements and each element has many 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.

To understand the intuitive ideas behind various texture and texton models, we find it revealing to discuss three physical ensembles which are shown in Fig. 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 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 $p(s; \boldsymbol{\beta})$,

$$\Omega_{cn} = \{s;\ p(s; \boldsymbol{\beta}) = c \cdot \exp\{-\boldsymbol{\beta} e(s)\}\}.$$

In our recent paper on texture modeling [34], 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[35] 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 [34] 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 change. 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,

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

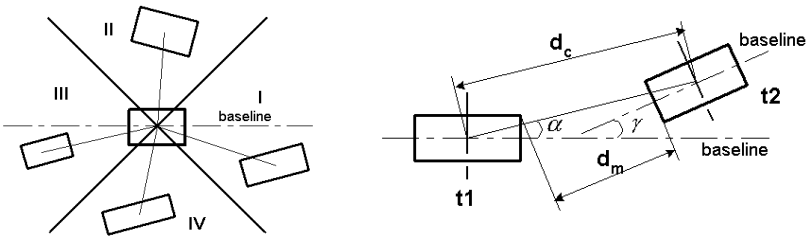


Figure 6 Texton neighborhood. a). a texton has four neighbors; b). Four measurements between texton t_1 and its neighbor t_2, d_c, d_m, α, and γ.

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

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

where ∂t is a set of neighboring textons for each texton t. We find the nearest neighbors are often enough. 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 [11]).

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 the scaling and shearing (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 one 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) was perhaps the earliest attempt for characterizing spatial patterns using histogram of attributes (See [27] for some examples).

The distribution of $\mathbf{T}$ is characterized by an statistical ensemble in correspondence to the grand-canonical ensemble in Fig. 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,

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

Z is a partition function. 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

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

This model provides a rigorous way for integrating multiple feature statistics into one probability model, and generalizes existing point processes[32].

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[34],

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

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[35]. In the following subsection, we discuss some computational issues and 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. 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,

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

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[35].

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)[37]. 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

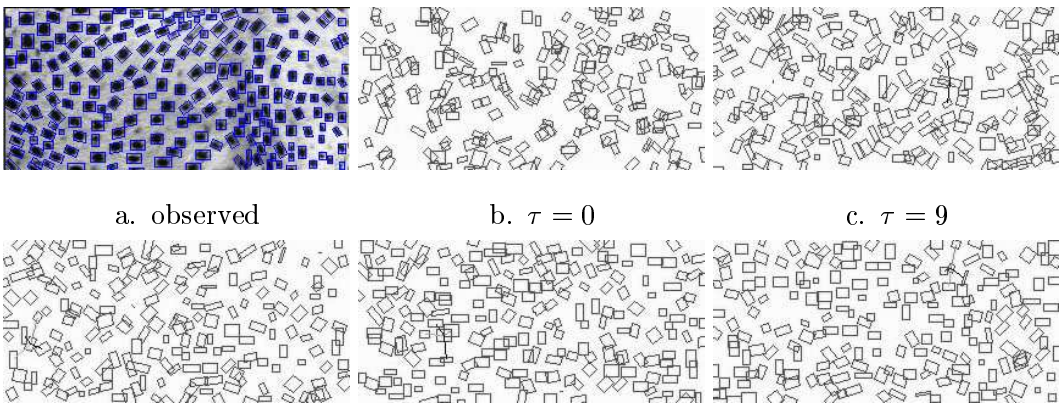


Figure 7 a). The observed image with textons illustrated by the rectangular windows. b)-f) are typical texton maps sampled from a Gibbs model $p(\mathbf{T}; \beta_o, \boldsymbol{\beta})$ at various stages $\tau = 0, ..., \tau = 234$ of the learning procedure.

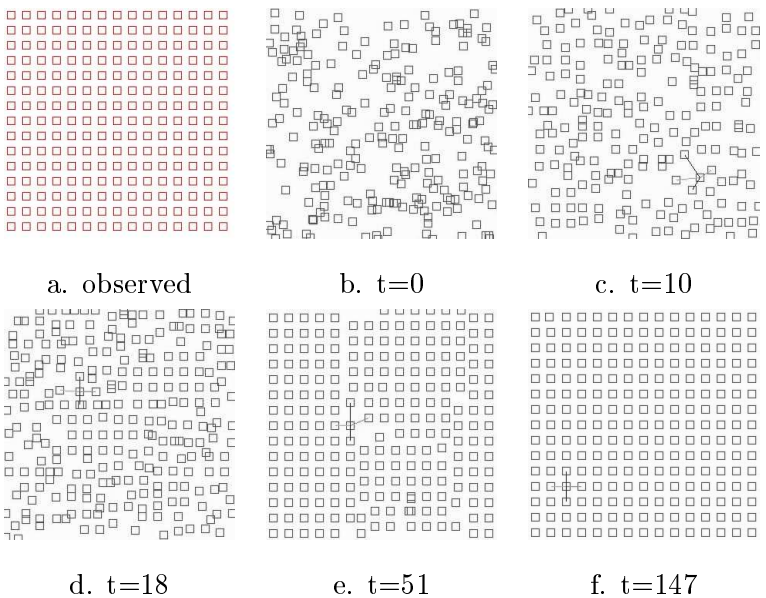
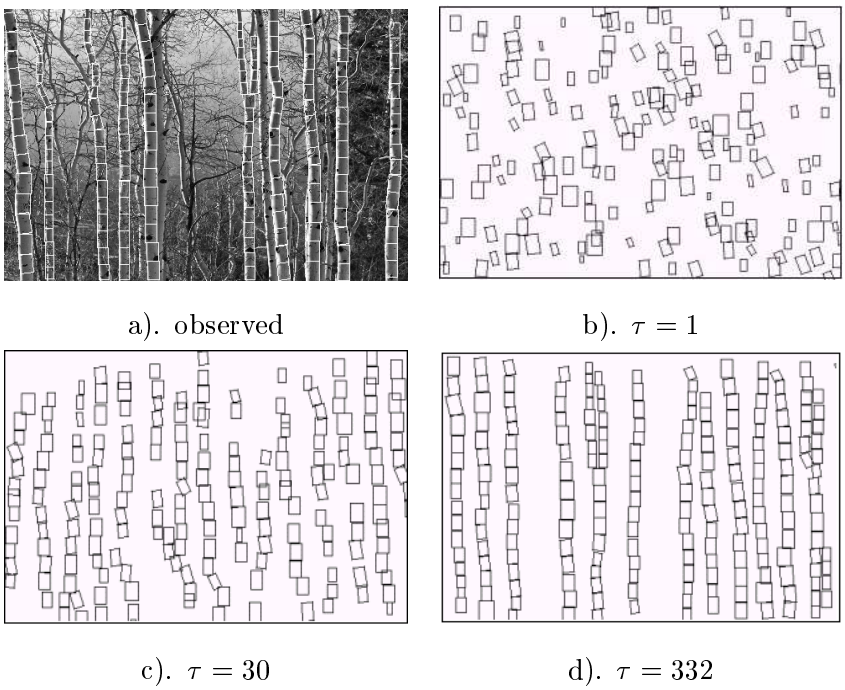


Figure 8 The simulation of a regular grid pattern.

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[14] that deletes or adds a texton.

2. A diffusion process: This updates the position, orientation, scale of the textons by a Gibbs sampler[12].

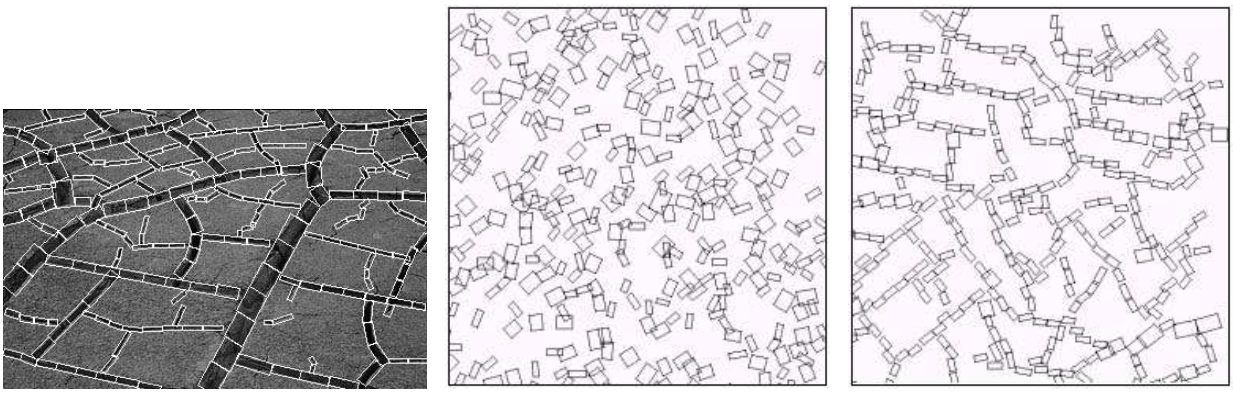


a). observed b). $\tau = 1$

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

Figure 9 The Markov chain Monte Carlo simulation of a woods pattern

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.



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

Figure 10 The Markov chain Monte Carlo simulation of a crack pattern.

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 used 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 characterized by the texton processes through Gibbs 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 leave some small gaps when they 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 Learning Framework

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

5.1 The Integrated Learning Paradigm

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

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

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 $\boldsymbol{\beta}_l$, $l = 1, 2$ for two Gestalt ensembles, and the variance of the Gaussian noise, σ^2

$$\Theta = (\Psi, \boldsymbol{\beta}, \sigma), \quad \Psi = (\Psi_1, \Psi_2), \quad \text{and} \quad \boldsymbol{\beta} = (\beta_{10}, \boldsymbol{\beta}_1, \beta_{20}, \boldsymbol{\beta}_2).$$

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

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

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[33] in the Bayesian context. We propose to use a stochastic gradient algorithm[15] 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.[6]

- 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[14].

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

$$\frac{\partial \log p(\mathbf{I}^{\text{obs}}|\mathbf{T}; \Psi)}{\partial \Psi} + \sum_{l=1}^{2} \frac{\partial \log p(\mathbf{T}_l; \boldsymbol{\beta}_l)}{\partial \boldsymbol{\beta}_l} = 0$$

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 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.[7]
- II.b). Update $\beta_{lo}, \boldsymbol{\beta}_l, l = 1, 2$ by maximizing $\sum_{m=1}^{M} \log p(\mathbf{T}_m^{\text{syn}}; \beta_{lo}, \boldsymbol{\beta}_l)$. 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.

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

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

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

In step I, we choose to sample $M = 1$ example each time. If the learning rate in steps II.a) and II.b) is slow enough, the expectation is estimated by importance sampling through samples $\mathbf{T}^{\text{syn}}$ over time. It has been proved in statistics[15] 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 models and learning 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 $\mathbf{T}$ (or W in more general generative models) 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.

5.2 Mathematical definitions of visual patterns in the integrated learning paradigm

Any visual learning paradigm must answer the question of *conceptualization*: How do we define, mathematically, a visual *concept* or a pattern? For example, a human face, a wood grain texture, and so on. In this section, we show how the descriptive and generative methods conceptualize a visual pattern.

The *concept* of a visual pattern is an *abstraction* (or *summary*) for an assembly of configurations (or instances) s that are not distinguished by human perception for a certain vision *purpose*. Because of the stochastic nature of the visual signal, instances in this assembly are governed by a frequency $f(s)$, and an instance can then be considered a random sample from the probability distribution $f(s)$. Thus a concept is said to be equal to an *ensemble*,

$$\text{a visual concept } c = \Omega_c = \{s \ : \ f(s)\}.$$

By a descriptive method, the ensemble is defined through statistical constraints (see equation.(3)),

$$\text{a descriptive concept } c = \Omega(\mathbf{h}_c) = \{s \ : \ f(s; \mathbf{h}_c), \ E_f[\mathbf{h}(s)] = \mathbf{h}_c\}.$$

For example, s could be a human face represented by a list of key points, or $s = \mathbf{I}$ could be a texture image. $\mathbf{h}(s)$ is the statistics extracted from s which are sufficient for a certain vision purpose. $\mathbf{h}_c$ is vector value that identifies this concept. $\mathbf{h}_c$ corresponds to a Gibbs model $p(s; \boldsymbol{\beta})$ with parameter $\boldsymbol{\beta}$. It is accepted that two concepts may have overlapping ensembles.

When the signal is homogeneous, such as texture or texton maps, the expectation $E_f[\mathbf{h}(s)]$ can be computed from a single instance through spatial average over a large enough lattice Λ, thus we can define a concept as an equivalence class – called the Julesz ensemble in (Zhu et al. 2000)[37].

$$\text{a descriptive concept } c = \Omega(\mathbf{h}_c) = \{s \ : \mathbf{h}(s) = \mathbf{h}_c \ \ \Lambda \to \mathrm{Z}^2, f(s; \mathbf{h}_c)\},$$

For example, for defining a texture, $\mathbf{h}_c$ is the sufficient and necessary statistics extracted in texture perception. The ensembles on large lattice $\Lambda \to \mathrm{Z}^2$ are disjoint and deterministic as $f(s; \mathbf{h}_c)$ is a uniform distribution.

Though the pure descriptive concept and Julesz ensemble are technically sound, they are only a coarse or first stage approximation to human perception in texture discrimination. Texture studies in psychology[3, 18] suggest that human vision is sensitive to the perception of some basic elements called textons, perhaps for some vision purpose. It was also argued and demonstrated by Malik et al. [24, 21, 25] that the detection of individual textons plays an important role in texture discrimination, segmentation and grouping. However, it was unclear what textons really are.

We argue that the mathematical definition of textons must be guided by a generative model. So the integrated learning paradigm extends the Julesz ensemble definition to a generative concept with W being the hidden variables,

$$\text{a generative concept } c = \{s \ : f(s; \Theta) = \int f(s|W; \Psi) f(W; \mathbf{h}_w) dW\},$$

where W are instances of a descriptive ensemble,

$$\{W \ : \mathbf{h}(W) = \mathbf{h}_w, \Lambda \to \mathrm{Z}^2, f(W; \mathbf{h}_w)\}.$$

Thus a concept is identified by the parameters $\Theta = (\Psi, \mathbf{h}_w)$ (or $\Theta = (\Psi, \boldsymbol{\beta})$, because of the duality between $\mathbf{h}_w$ and $\boldsymbol{\beta}$). In this definition, Ψ is a mathematically sound definition of texton.

6 Effective Inference by Simplified Likelihood

In this section, we address some computational issues in the integrated learning paradigm, 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 section (5.1)) 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 $\mathbf{I}^{\text{obs}}$, 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 analysis 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

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

where $\mathbf{I}^{\text{obs}}_{D_{lj}}$ is the image patch of the domain D_{lj} in the observed image. For pixels in $\mathbf{I}^{\text{obs}}$ 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 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

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

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[10]. It is also inspired by a clustering algorithm by (Leung and Malik, 1999)[22], which did not engage hidden variables, and thus compute a variety of textons Ψ at different scale and orientations. 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 some experiments for initialization and clustering as section (6.1) stated.

Fig. 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 Fig. 11.

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

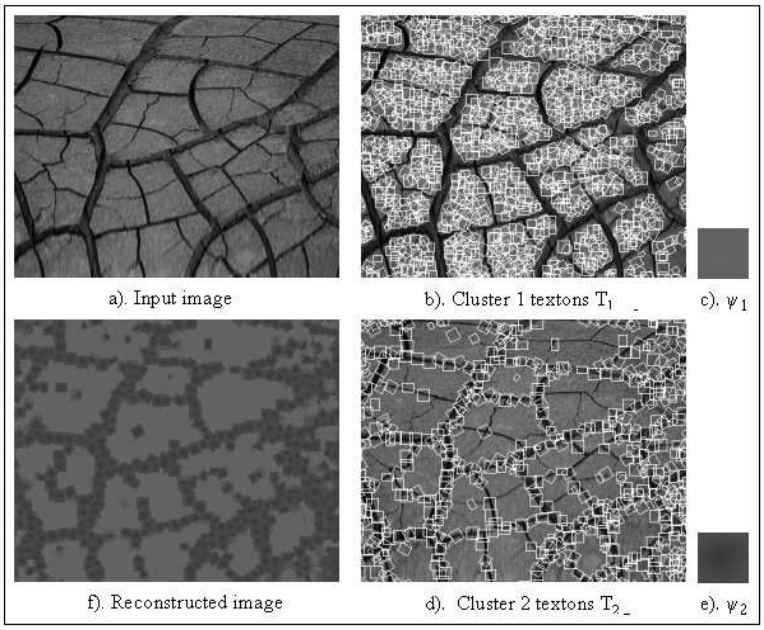


Figure 11 Result of the initial clustering algorithm.

7 Experiment III: Integrated learning and synthesis

In this section, we show some experimental results obtained by the integrated learning paradigm. 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.

Fig. 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 Fig. 11. Fig. 12.b) and Fig. 12.d) are the background and foreground texton maps $\mathbf{T}_1$ and $\mathbf{T}_2$ respectively. Fig. 12.c) and Fig. 12.e) are the learned textons Ψ_1, Ψ_2 respectively. Fig. 12.f) is the reconstructed image from learned Texton maps and templates. Compared to the results in Fig. 11, the results in Fig. 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 Fig. 13. It can be seen that in the foreground template, the surround pixels are learned as being transparent and the blob is exactly computed as the texton. Fig. 14 are the results for a brick image. No

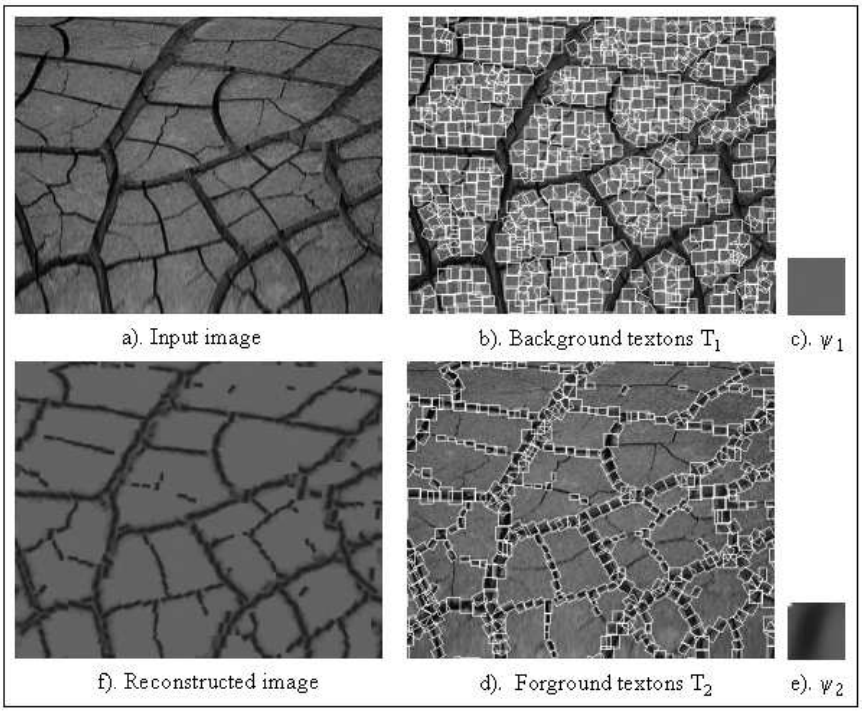


Figure 12 Generative model learning result for the crack image. a) input image, b) and d) are background and foreground textons discovered by the generative model, c) and e) are the templates for the generative model, f) is the reconstructed image from the generative model.

point in the template is transparent for the gap lines between bricks.

Figure 15 shows the learning of another short crack patterns. Fig. 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.

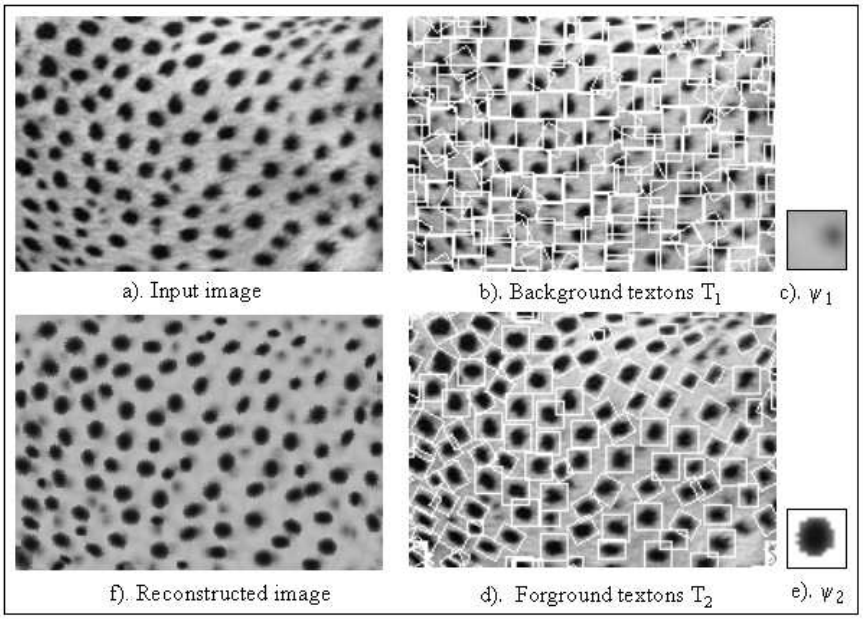


Figure 13 Generative model learning result for a cheetah skin image. The notations are the same as in Fig. 12.

After the parameters Ψ and $\boldsymbol{\beta}$ of a generative model are discovered for a type of texture images, new random samples could be drawn from the generative model. This proceeds in three steps: First, texton maps 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 the occlusion model.

We shown synthesis experiments on three patterns.

1. Fig 17 and Fig. 18 are two synthesis examples of the two layered model synthesis for the cheetah skin pattern. The templates used here are the learned results in Fig 13.
2. Figure 20 shows texture synthesis for the crack pattern computed in Figure 15.

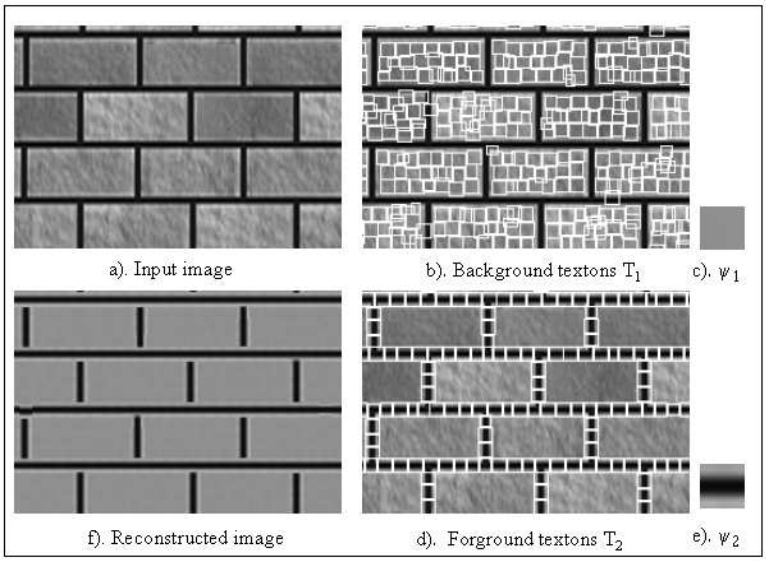


Figure 14 Generative model learning result for a brick image. The notations are the same as in Fig. 12.

3. Figure 20 displays texture synthesis for the brick pattern in Figure 14.

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

8 Discussion

In this paper, we present a visual learning paradigm which integrates and extends descriptive and generative models, and which also provides sound framework for visual conceptualization and for defining textons. The hierarchical model for textures has advantage over previous pure descriptive method with Markov random fields on pixel intensities.

Firstly, from the representational perspective, the neighborhood in the texton map are much smaller than the pixel neighborhood in a FRAME model [35]. The generative method captures more semantically meaningful elements on the texton maps.

Secondly, 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.

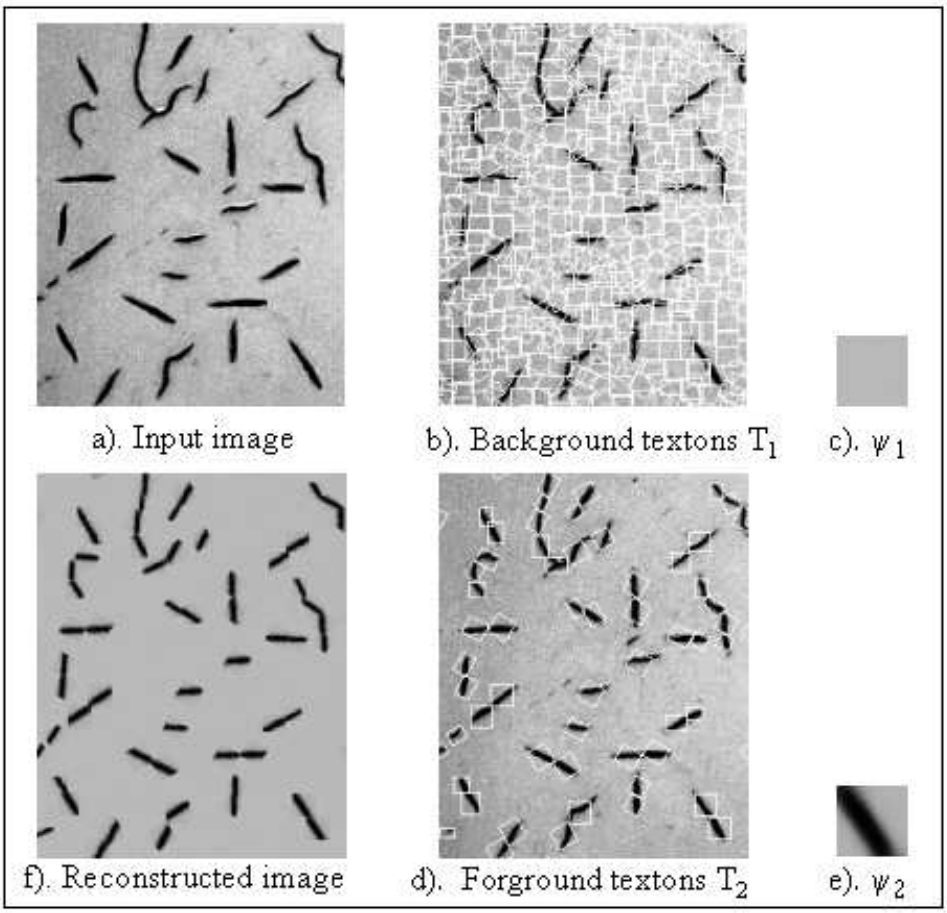


Figure 15 Generative model learning result for a crack image. The notations are the same as in Fig. 12.

We show that the integration of descriptive and generative methods is a natural path for visual learning. We argue that a vision system should evolve by progressively replacing descriptive models with generative models, which realizes a transition from *empirical and statistical models* to *physical and semantical* models. The work presented in this paper provides a step towards this goal.

In future research, we shall develop effective algorithms for sampling the Gestalt ensembles, and extend the generative model to curve and graph processes. These models will be employed for general purpose vision tasks, such as segmentation and grouping.

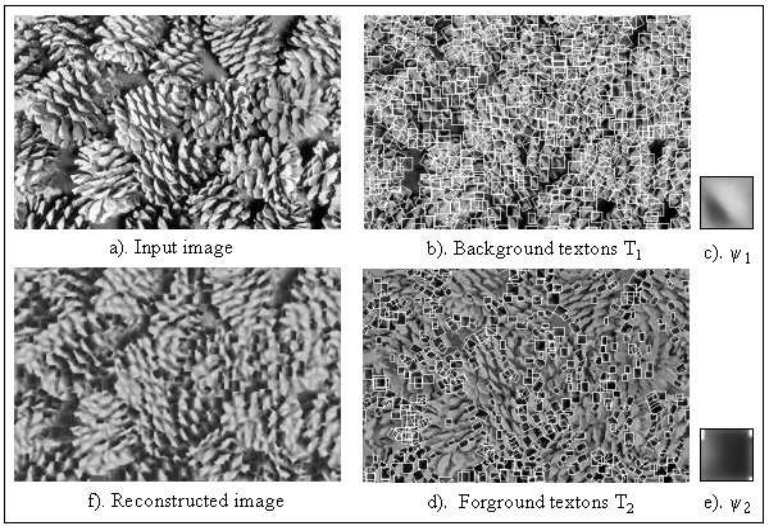


Figure 16 Generative model learning result for a pine corn image. The notations are the same as in Fig. 12.

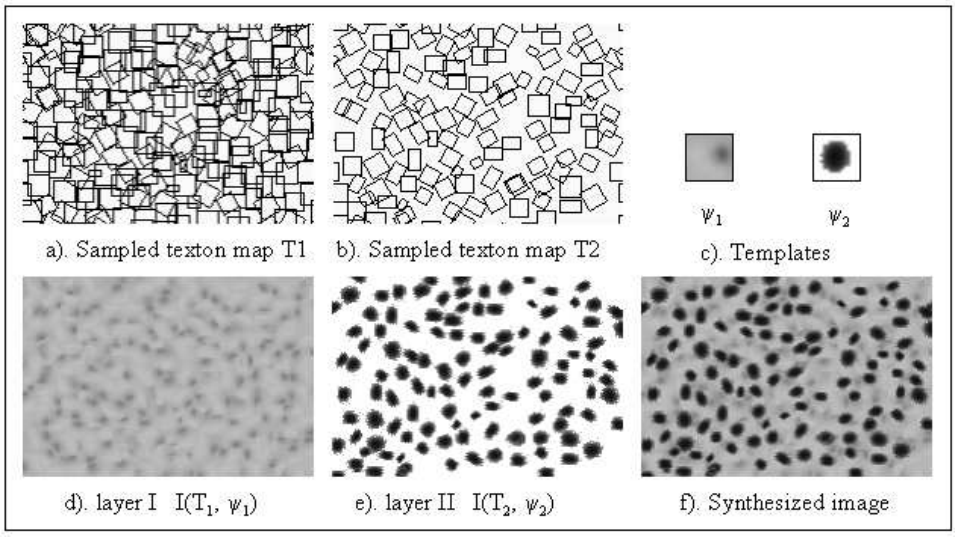


Figure 17 An example of a randomly synthesized cheetah skin image. a) and b) are the background and foreground texton maps sampled from $p(\mathbf{T}_l; \beta_l)$; d) and e) are synthesized background and foreground images from the texton map and templates in c); f) is the final random synthesized image from the generative model.

Acknowledgments

This work is supported partially by NSF grants IIS 98-77-127, IIS-00-92-664, and DMS-0072538, a NASA grant NAG-13-00039, and an ONR grant N000140-110-535.

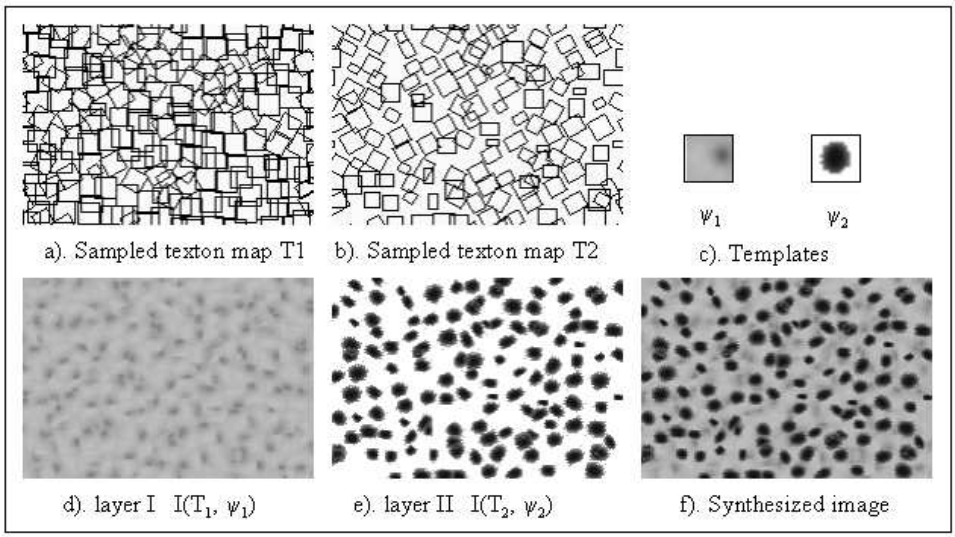


Figure 18 Second example of a randomly synthesized cheetah skin image. Notation is the same as in Fig. 17.

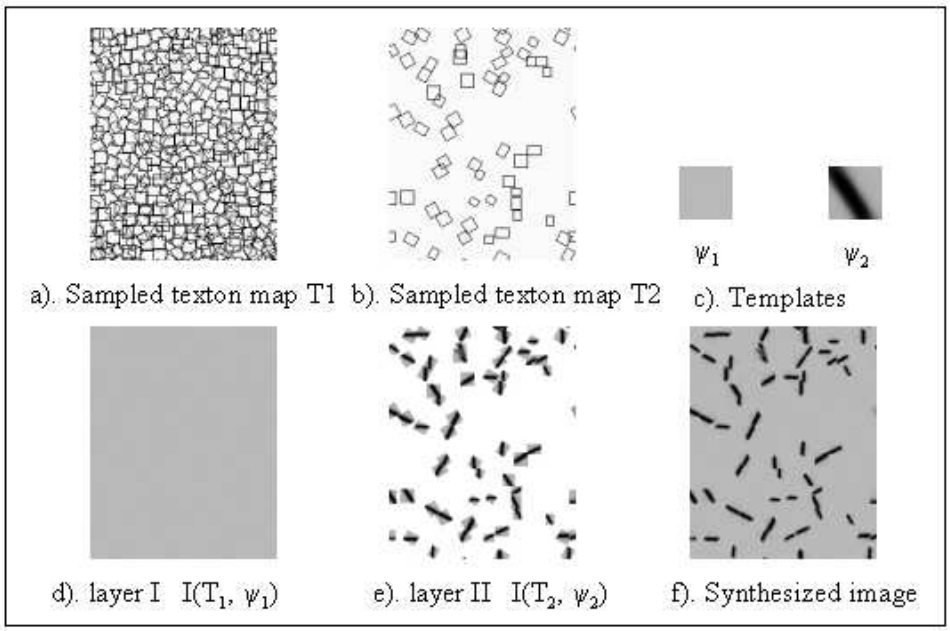


Figure 19 An example of a randomly synthesized crack image. Notations are the same as in Fig. 17.

References

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

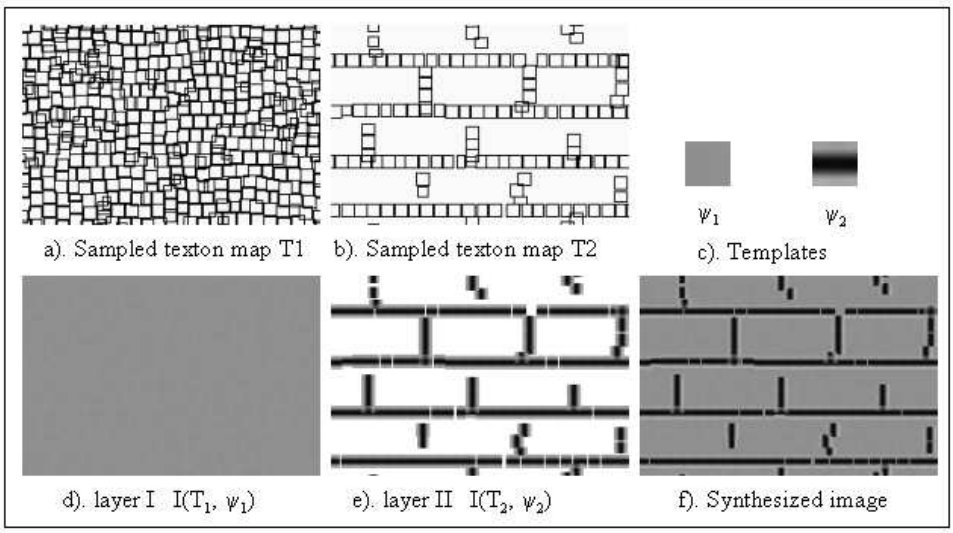


Figure 20 An example of a randomly synthesized brick image. Notation is the same as in Fig. 17.

[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 Nueral 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] B. Frey and N. Jojic, "Transformed component analysis: joint estimation of spatial transforms and image components", *Proc. of 7th ICCV*, Corfu, Greece, 1999.

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

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

[13] 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.

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

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

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

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

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

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

[20] 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.

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

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

[23] 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.

[24] 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.

[25] J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, Contours and Regions: Cue Integration in Image Segmentation", *Proc. of 7th ICCV*, Corfu, Greece, 1999.

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

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

[28] 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.

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

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

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

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

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

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

[35] 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.

[36] S. C. Zhu, "Embedding Gestalt Laws in Markov Random Fields", *PAMI.* vol. 21, no.11, 1999.

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

[38] 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*, Vancaouver, 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 Gelstalt 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 Gelstalt 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 $N(\mathbf{T}_{\Lambda/\Lambda_0}) = N - N(\mathbf{T}_0)$ and $\mathbf{H}(\mathbf{T}_{\Lambda/\Lambda_0}) = \mathbf{H} - \mathbf{H}(\mathbf{T}_0)$, where Λ/Λ_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 $\mathbf{T}_{\Lambda/\Lambda_0}$ in $\Omega_{\Lambda/\Lambda_0}(N - N(\mathbf{T}_0), \mathbf{H} - \mathbf{H}(\mathbf{T}_0))$. Therefore, the frequency of $\mathbf{T}_0$

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

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

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

where 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.