Proceedings of Int'l Conf. on Computer Vision, Vancouver, Canada, July, 2001.

Visual Learning By Integrating Descriptive and Generative Methods

Cheng-en Guo, Song-Chun Zhu
Ohio State University
$\{cguo, szhu\}$@cis.ohio-state.edu

Yingnian Wu
Univ. of California at Los Angeles
ywu@stat.ucla.edu

Abstract [1]

This paper presents a mathematical framework for visual learning that integrates two popular statistical learning paradigms in the literature: I). Descriptive learning, such as Markov random fields and minimax entropy learning, and II). Generative learning, such as PCA, ICA, TCA, image coding and HMM. We apply this integrated learning framework to texton modeling, and we assume that an observed texture image is generated by multiple layers of hidden stochastic "texton processes" with each texton being a window function, like a mini-template or a wavelet, under affine transformations. The spatial arrangements of the textons are characterized by minimax entropy models. The texton processes generate images by occlusion or linear addition. Thus given a raw input image, the learning framework achieves four goals: i). Computing the appearance of the textons. ii). Inferring the hidden stochastic texton processes. iii). Learning Gibbs models for each texton process. and iv). Verifying the learnt textons and Gibbs models through random sampling and texture synthesis. The integrated framework subsumes the minimax entropy learning paradigm and creates a richer class of probability models for visual patterns, which are suited for middle level vision representations. Furthermore we show that the integration of descriptive and generative methods yields a natural and general framework of visual learning. We demonstrate the proposed framework and algorithms on many real images.

1 Introduction

In Bayesian statistical image analysis, an important task is to learn probabilistic models that characterize visual patterns in real images. Existing methods for learning statistical models are generally divided into two categories. In this paper, we call one the *descriptive method* and the other *generative method*.

I). Descriptive method characterizes visual patterns by imposing statistical constraints and thus learns models at a "signal" level. This includes Markov random fields, minimax entropy learning[13], deformable models. For example, recent work on texture modeling fall in this category[13, 11]. These models are built on pixel intensities through complex interactions between image features, which are often reflected by complicated Gibbs potential functions. The shortcoming is that they do not capture high level semantics in the patterns. For example, a Gibbs model of texture can realize a cheetah skin pattern but it does not have explicit notion of individual blobs.

II). In contrast to descriptive method, generative method infers hidden causes (or semantics) from raw signals, and thus can learns hierarchical models. Examples of generative method are principle component analysis (PCA), independent component analysis (ICA), transformed component analysis (TCA)[2], image coding[9], and hidden Markov models (HMM). As a recent review paper[10] pointed out, existing generative models mentioned above suffer from the simplified assumption that hidden variables are independent and identically distributed. Therefore they are not powerful enough to model realistic visual patterns. For example, an image coding model cannot synthesize a texture patterns through random sampling.

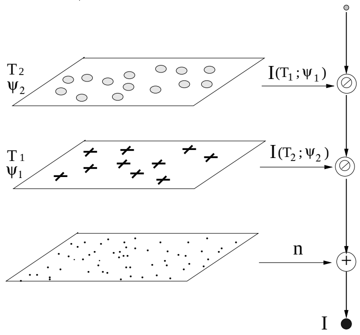
In this paper, we present a visual learning paradigm that integrates both descriptive and generative methods and we apply this learning paradigm to modeling texton patterns.

In early vision, a fundamental observation, dated back to Marr's primal sketch[8], is that natural visual patterns consist of multiple layers of stochastic processes. An example is shown in Fig. 1.a. When we look at this pattern, we perceive not only the texture "impression" and pixels but also the repeated elements for the ivy and bricks. In psychology, basic texture elements are called "texton" or "texel" vaguely[5], and a precise mathematical definition has yet to be found. In this paper, we propose to study a multiple layer generative model as Fig. 1 illustrates. It has three stochastic processes — two for the ivy and brick patterns respectively with two distinct "textons" and the third for noise process. The stochastic processes are hidden and the image is the only observable signal.

Given an input image, the integrated learning framework achieves the following four objectives.

[1] This work is supported by a NSF grant IIS 98-77-127, and a NASA grant NAG 13-00039. We refer to a web site www.cis.ohio-state.edu/oval for a detailed report and more results.

a). A visual pattern



b). A generative model with three layers

Figure 1: Ψ_i is a texton window function, T_i is a texton process with texton elements Ψ_i, and $\mathbf{I}(\mathbf{T}_i; \psi_i), i = 1, 2$ are texton images.

1. Learning the texton for each stochastic process. A texton is a represented as a window function, like a mini-template or wavelets.
2. Inferring the hidden stochastic processes each being a spatial pattern with a number of textons subject to affine transformations.
3. Learning minimax entropy models for the hidden processes.
4. Verifying the learnt textons and generative models through random sampling.

Furthermore we observed for hidden layers if there is no further hidden layer behind, the hidden variables must be characterized by the descriptive method, i.e. the minimax entropy models. Thus descriptive models are precursors of generative models, and Learning process evolves by discovering hidden causes. So the two learning paradigms must be integrated.

The integrated learning framework makes three interesting contributions to visual learning. 1). It subsumes the minimax entropy learning paradigm by extending from pixels to textons and creates a richer class of probability models for visual patterns. It is easy to show that existing texture models are special cases of this model where the textons are single pixels. 2). It introduces the minimax entropy learning paradigm for modeling hidden variables in generative models, and thus subsumes and extends existing generative models such as PCA, ICA, and TCA[2]. 3). It can automatically learn textons from images as the transformed components under the generative model. Our work is different from [6, 7] which used the clustering method in feature spaces of filter response. As a result, texton elements at various translations, rotations and scales are treated as distinct textons[7]. We demonstrate the proposed framework and algorithms on a number of real images.

2 Background on Visual Learning

Given a set of observable signals $S = \{\mathbf{I}_1^{\text{obs}}, \mathbf{I}_2^{\text{obs}}, ..., \mathbf{I}_M^{\text{obs}}\}$. Without loss of generality, we assume the observable signals are raw images. The goal of visual learning is to estimate a probabilistic model $p(\mathbf{I})$ from S so that $p(\mathbf{I})$ approaches the underlying frequency $f(\mathbf{I})$, which governs the ensemble of signals in an application, in terms of minimizing a Kullback-Leibler divergence $KL(f(\mathbf{I})||p(\mathbf{I}))$ between f and p. This leads to the standard maximum likelihood estimator (MLE).

$$p^* = \arg\min_{p \in \Omega_p} KL(f(\mathbf{I})||p(\mathbf{I})) \approx \arg\max_{p \in \Omega_p} \sum_{i=1}^{M} \log p(\mathbf{I}_i^{\text{obs}}). \quad (1)$$

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,

$$\Omega_0 \subset \Omega_1 \subset \cdots \subset \Omega_k \to \Omega_f \ni f.$$

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 contain the true distribution $f(\mathbf{I})$.

There are only two choices of families Ω_p in the literature and both are general enough for approximating any distributions $f(\mathbf{I})$.

The first choice is the exponential family of models, which is derived by descriptive method, and has deep root in statistical mechanics. A descriptive method extracts a set of K features as *deterministic transforms*, and computes the statistics for these features across images in S. The statistics are denoted by $\phi_j(\mathbf{I}), j = 1, 2, ..., K$. Then it constructs a model p through imposing *descriptive constraints* so that p reproduces the observed statistics while having maximum entropy. This leads to the following Gibbs form with $\boldsymbol{\beta} = (\beta_1, ..., \beta_K)$ are the parameters for the model,

$$p(\mathbf{I}; \boldsymbol{\beta}) = \frac{1}{Z(\boldsymbol{\beta})} \exp\{-\sum_{j=1}^{K} \beta_j \phi_j(\mathbf{I})\}.$$

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

$$\Omega^d_1 \subset \Omega^d_2 \subset \cdots \Omega^d_K \to \Omega_f.$$

This family includes all the MRF and minimax entropy models for texture[13].

The second choice is the mixture family of models, which is derived from integration or summation over some hidden variables $W = (w_1, w_2, ..., w_k)$.

$$p(\mathbf{I};\Theta) = \int \cdot \int p(\mathbf{I}, w_1, w_2, ..., w_k; \Theta) \prod_{i=1}^{k} dw_i.$$

In this way, we assume that there exists a joint probability distribution $f(\mathbf{I}, W)$, and that W generates $\mathbf{I}$ and W should be *inferred* from $\mathbf{I}$, instead of being computed as deterministic transforms. The generative method incrementally adds hidden variables to augment the space Ω_p and thus generates a sequence of mixture families,

$$\Omega^g_1 \subset \Omega^g_2 \subset \cdots \subset \Omega^g_K \to \Omega_f \ni f.$$

For example, in PCA and image coding[9], a simply generative model is an addition of some window functions $\Psi_i, i = 1, 2, ..., M$, such as over-complete wavelet bases, eigen vectors plus an iid Gaussian noise process $\mathbf{n}$.

$$\mathbf{I} = \sum_{i=1}^{K} a_i \Psi_i + \mathbf{n}; \quad \alpha_i \sim p(\alpha)\ \forall i.$$

In this example, the parameters are the K bases (or eigen vectors) $\Theta = \{\Psi_1, ..., \Psi_K\}$ and the hidden variables are the K coefficiencies of bases (or eigen vectors) plus the noise $W = (a_1, a_2, ..., a_K, \mathbf{n})$.

The forms of a mixture model $p(\mathbf{I};\Theta)$ are decided by the distribution of the hidden variables W. The latter must be from descriptive families. However, in the literature, hidden variables $a_i, i = 1, 2, ..., K$ are assumed to be iid Gaussian or Laplacian distributed. Thus the concept of descriptive models are trivialized.

In the following section, we study a learning paradigm that integrates both families and some interesting relationships are revealed.

3 An Integrated Learning Framework

3.1 A generative model of texture

In this section, we study a multi-layer generative model as Fig 1.b shows. We assume that a texture image $\mathbf{I}$ is generated by L layers of stochastic processes while each layer consists of a finite number of distinct elements, called "textons", which are image patches transformed from one square image template Ψ_i. The jth texton in layer i is represented by six transform variables on the template Ψ_i as

$$T_{ij} = (x_{ij}, y_{ij}, \sigma_{ij}, \tau_{ij}, \theta_{ij}, A_{ij}),$$

where (x_{ij}, y_{ij}) represents the texton center location. σ_{ij} is the scale of the size, τ_{ij} is called "shear" compressing the width of the texton, θ_{ij} is the orientation, and A_{ij} denotes photometric transforms such as lighting variability. The transformation operator on T_{ij} is denoted by $G[T_{ij}]$. The pixel domain in which the texton T_{ij} covers is denoted as $D_{ij} = D[T_{ij}]$. Thus the image patch $\mathbf{I}_{D_{ij}}$ of a texton T_{ij} is derived by

$$\mathbf{I}_{D_{ij}} = G[T_{ij}] \odot \Psi_i,$$

where $\odot$ denotes the transformation operation. Texton examples at different scales, shears, and orientations are shown in Fig. 2.

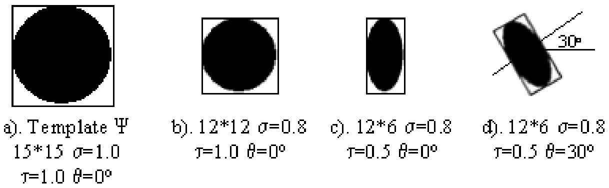


Figure 2: Texton examples at different scales, shears, and orientations.

We define all textons in layer i as a "texton map"

$$\mathbf{T}_i = (n_i, \{T_{ij}, j = 1 \ldots n_i\}), i = 1 \ldots L,$$

where n_i is the number of textons in layer i.

In each layer, the texton map $\mathbf{T}_i$ and the template Ψ_i generate an image $\mathbf{I}_i = \mathbf{I}(\mathbf{T}_i; \Psi_i)$ deterministically. If several textons overlap at site (x, y) in $\mathbf{I}_i$, the pixel value is averaged as

$$\mathbf{I}_i(x, y) = \frac{\sum_{j=1}^{n_i} \delta((x,y) \in D_{ij}) I_{D_{ij}}(x, y)}{\sum_{j=1}^{n_i} \delta((x,y) \in D_{ij})},$$

where $\delta(\bullet) = 1$ if $\bullet$ is true, otherwise $\delta(\bullet) = 0$. In image $\mathbf{I}_i$, pixels not covered by the textons are transparent. Then the final image $\mathbf{I}$ is generated by the following model as

$$\mathbf{I} = \mathbf{I}(\mathbf{T}_1; \Psi_1) \oslash \mathbf{I}(\mathbf{T}_2; \Psi_2) \oslash \cdots \oslash \mathbf{I}(\mathbf{T}_L; \Psi_L) + \mathbf{n}. \quad (2)$$

The symbol $\oslash$ denotes occlusion or linear addition, i.e. $\mathbf{I}_1 \oslash \mathbf{I}_2$ means $\mathbf{I}_1$ occludes $\mathbf{I}_2$. In this generative model, the hidden variables are

$$\mathbf{T} = (L, \{(\mathbf{T}_i, d_i) : i = 1, 2, \ldots, L\},\ \mathbf{n}),$$

where d_i indexes the order (or relative depth) of the i-th layer. $\mathbf{n}$ is the noise process. The 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.

To simplify computation, we assume that $L = 2$ and the two stochastic layers, called "background" and "foreground", are independent of each other. We find that this assumption holds true for most of the texture patterns. Otherwise one has to implement a jump process in Markov chain Monte Carlo to infer L. Thus we obtain a likelihood model for image $\mathbf{I}$.

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

$$= \int p(\mathbf{I}|\mathbf{T}_1, \mathbf{T}_2; \Psi) \prod_{i=1}^{2} p(\mathbf{T}_i; \boldsymbol{\beta}_i) d\mathbf{T}_1 d\mathbf{T}_2 d\mathbf{d_1} d\mathbf{d_2}, \quad (3)$$

where $\Theta = (\Psi, \boldsymbol{\beta})$ with $\Psi = (\Psi_1, \Psi_2)$ being texton templates and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ the parameters in the Gibbs models for texton processes, and d_1 and d_2 denote the layer order (background or foreground). The model $p(\mathbf{I}|\mathbf{T}_1, \mathbf{T}_2; \Psi)$ is simply Gaussian distributed as

$$p(\mathbf{I}^{\text{obs}}|\mathbf{T}_1, \mathbf{T}_2; \Psi) \propto \exp \frac{-\left\|\mathbf{I}^{\text{obs}} - \mathbf{I}(\mathbf{T}_1, \mathbf{T}_2; \Psi)\right\|^2}{2\sigma^2}, \quad (4)$$

where $I(\mathbf{T}_1, \mathbf{T}_2; \Psi)$ is the *reconstructed image* from the hidden layers without noise(see eq. (2)). $p(\mathbf{T}_i; \boldsymbol{\beta}_i), i = 1, 2$ are also exponential models which characterize the spatial relationships through a set of feature statistics. The details are discussed in the next subsection.

3.2 A descriptive model of texton map

The construction of descriptive model for texton maps follows the minimax entropy learning paradigm[13]. We only briefly discuss it and refer to a companion paper for detailed study[14].

For a given texton map $\mathbf{T}_i$ with n_i elements, we first define some neighborhood structures for each texton, and measure a set of features which characterize important spatial relationship between each elements in a local neighborhood. For example, the orientation and scale of a single texton, the distance and relative orientations and sizes of two neighboring textons. We then calculate the histograms of these features $H_j(\mathbf{T}_i), j = 1, 2, ..., K$.

A Gibbs (maximum entropy) model is then obtained by descriptive method[13],

$$p(\mathbf{T}_i; \boldsymbol{\beta}_i) = \frac{1}{Z(\boldsymbol{\beta}_i)} \exp\{-\beta_{i0} n_i - \sum_{j=1}^{K} < \beta_{ij}, H_j(\mathbf{T}_i) >\}.$$

In $p(\mathbf{T}_i; \boldsymbol{\beta}_i)$, β_{i0} controls the density of textons n_i on a given unit area, and $\beta_{ij}, j = 1, 2, ..., K$ are vector valued Lagrange multipliers.

$p(\mathbf{T}_i; \boldsymbol{\beta}_i)$ governs a texton ensemble which corresponds to a so-called grand-canonical ensemble in statistical mechanics. It can be simulated by a Markov chain Monte Carlo algorithm which utilizes Gibbs sampler for the position, scale, shear, and orientation of the textons and also reversible jumps[3] which simulate the death/birth of textons. The selection of important features is done by the minimum entropy principle[13].

Of course, the entire descriptive learning is an ML-estimator that maximizes the log-likelihood by steepest ascent,

$$\boldsymbol{\beta}^* = \arg\max \log p(\mathbf{T}_i; \boldsymbol{\beta}_i); \quad \frac{\log p(\mathbf{T}_i; \boldsymbol{\beta}_i)}{\partial \boldsymbol{\beta}_i} = 0. \quad (5)$$

3.3 The Integrated Learning Paradigm

To learn a generative model $p(\mathbf{I}; \Theta)$ in eq. (3), we follow the ML-estimate in eq. (1).

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

Note that Θ characterizes the visual pattern and the whole ensemble governed by $p(\mathbf{I}, \mathbf{T}; \Theta)$, while $\mathbf{T}$ is associated with only an image instance $\mathbf{I}$.

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

$$\begin{aligned}
&\frac{\partial \log p(\mathbf{I}^{\text{obs}}; \Theta)}{\partial \Theta} \\
= &\int \frac{\partial \log p(\mathbf{I}^{\text{obs}}, \mathbf{T}; \Theta)}{\partial \Theta} p(\mathbf{T}|\mathbf{I}^{\text{obs}}; \Theta) d\mathbf{T} \\
= &\int [\frac{\partial \log p(\mathbf{I}^{\text{obs}}|\mathbf{T}; \Psi)}{\partial \Psi} + \sum_{i=1}^{2} \frac{\partial \log p(\mathbf{T}_i; \boldsymbol{\beta}_i)}{\partial \boldsymbol{\beta}_i}] \\
&\quad p(\mathbf{T}|\mathbf{I}^{\text{obs}}; \Theta)\, d\mathbf{T} \\
= &\; E_{p(\mathbf{T}|\mathbf{I}^{\text{obs}}; \Theta)}[\frac{\partial \log p(\mathbf{I}^{\text{obs}}|\mathbf{T}; \Psi)}{\partial \Psi} + \sum_{i=1}^{2} \frac{\partial \log p(\mathbf{T}_i; \boldsymbol{\beta}_i)}{\partial \boldsymbol{\beta}_i}] \\
= &\; 0. \qquad (6)
\end{aligned}$$

In the literature, there are two well-known methods for solving the above equations. One is the EM algorithm[1], and the other is data augmentation[12]. We propose to use a stochastic gradient algorithm[4] which is more effective than the EM-algorithm and data augmentation.

A Stochastic Gradient Algorithm

Step 0. Initialize the hidden layers $\mathbf{T}$ and the templates Ψ from $\mathbf{I}^{\mathrm{obs}}$ using a data driven (clustering) method discussed in the next section. Set $\boldsymbol{\beta}=0$.

Step I. Given current $\Theta=(\Psi,\boldsymbol{\beta})$, it *samples typical* texton maps from the posterior probability $\mathbf{T}^{\mathrm{syn}}=(\mathbf{T}_1^{\mathrm{syn}},\mathbf{T}_2^{\mathrm{syn}},d_1,d_2)\sim p(\mathbf{T}|\mathbf{I}^{\mathrm{obs}};\Theta)$. This is the Bayes *perceptual inference.* The sampling process is realized by a Monte Carlo Markov chain which simulates a random walk with two types of dynamics.

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

- I.b). A *jump-dynamics* — adding or removing a texton (death/birth) by reversible jumps[3] using Metropolis-Hastings method. Also the layer order d_1 and d_2 are sampled between background and foreground.

Step II. We treat $\mathbf{T}^{\mathrm{syn}}$ as "observation", and estimate the integration in eq. (6) by importance sampling. Thus we have

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

We learn $\Theta=(\Psi,\boldsymbol{\beta})$ the texton and Gibbs model respectively by gradient ascent in two steps.

- II.a). Computing the texton templates Ψ by maximizing $\log p(\mathbf{I}^{\mathrm{obs}}|\mathbf{T}^{\mathrm{syn}};\Psi)$, and this is often done by regression. In our experiment, each texton is represented by a 15×15 window with 225 unknowns. Also each point in the window could be transparent, and thus the shape of the texton could change during the learning process.

- II.b). Computing $\boldsymbol{\beta}_i, i=1,2$ by maximizing $\log p(\mathbf{T}_i^{syn};\boldsymbol{\beta}_i)$. This is exactly the maximum entropy learning process in descriptive method (see eq. (5)).

The algorithm iterates steps I and II. If the learning rate in steps II.a and II.b is slow enough, the expectation is estimated by importance sampling through samples $\mathbf{T}^{\mathrm{syn}}$ over time. It has been proved in statistics[4] that such algorithm converges to the optimal Θ if the step size in step II satisfies some mild conditions.

In summary, we feel that the following observations are especially revealing.

1. Descriptive models and descriptive method are inherent part (Step II.b) in generative models and generative method. Existing generative models, such as image coding have weak (iid) descriptive models instead of the Gibbs model and this limits their expressive power.

2. Bayesian vision inference is a sub-task (step I) of generative learning.

3.4 Initialization by Data Clustering

Both the hidden texton maps $\mathbf{T}=(\mathbf{T}_1,\mathbf{T}_2)$ and the texton templates $\Psi=(\Psi_1,\Psi_2)$ need to be initialized in order to start the bootstrap procedure in the previous section. In this section, we present a stochastic algorithm to obtain the initial $\mathbf{T}^0$ and Ψ^0 by decoupling some variables with two simplifications from the model in eq. (3).

Firstly, we decouple the texton elements in the prior $p(\mathbf{T}_i;\beta_i)$. In the two texton maps $\mathbf{T}_1$ and $\mathbf{T}_2$, n_1+n_2 is fixed to an excessive number, thus we don't need to simulate the death-birth process. $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are set to be 0, therefore $p(\mathbf{T}_i;\beta_i)$ becomes a uniform distribution and all texton elements are decoupled from interactions.

Secondly, we further decouple the texton elements in the likelihood $p(\mathbf{I}^{obs}|\mathbf{T};\Psi)$. Instead of using the image generating model in eq. (2) which implicitly imposes couplings between texton elements through eq. (4), we adopt a constraint-based model

$$p(\mathbf{I}^{obs}|\mathbf{T},\Psi)\propto \exp\{-\sum_{i=1}^{2}\sum_{j=1}^{n_i}||\mathbf{I}^{\mathrm{obs}}_{D_{ij}}-G[T_{ij}]\odot\Psi_i||^2/2\sigma^2\}, \tag{7}$$

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

So far all the textons are decoupled of each other by simplifying the generative model of eq. (3) to eq. (7) without the integration of $\mathbf{T}$. Consequentially the searching problem of $\mathbf{T}^0$ and Ψ^0 turns into a conventional clustering issue.

We start with random texton maps and the algorithm iterates the following two steps. I). Given Ψ_1 and Ψ_2, it runs a Gibbs sampler to change each texton T_{ij} respectively, by moving, rotating, scaling the rectangle, and changing the cluster into which each texton falls according to the simplified model of eq. (7). 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 $\mathbf{T}_2$, it updates the texton Ψ_1 and Ψ_2 by averaging as

$$\Psi_i=\frac{1}{n_i}\sum_{j=1}^{n_i}G^{-1}[T_{ij}]\odot\mathbf{I}^{\mathrm{obs}}_{D_{ij}},\quad i=1,2,$$

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

This initialization algorithm for computing (T^0, Ψ^0) resembles transformed component analysis (TCA). It is also inspired by a clustering algorithm by (Leung and Malik, 1999)[7], 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 in some textures.

4 Experiments

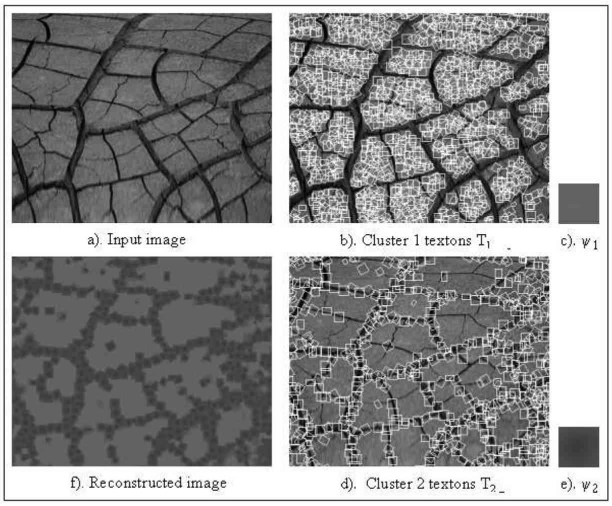


Figure 3: Result of the initial clustering algorithm.

Experiment I: Initialization by TCA. Fig. 3 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. (7), an annealing scheme is utilized with the temperature decreasing from 4 to 0.5. The sampling process converged to a result shown in Fig. 3.

Fig. 3.a is the input image; Figs 3.b and Figs 3.d are the texton maps $\mathbf{T}_1$ and $\mathbf{T}_2$ of two clusters respectively. Fig. 3.c and Fig. 3.e are the cluster centers Ψ_1 and Ψ_2, shown by rectangles respectively. Fig. 3.f is the reconstructed image. The results demonstrate that the clustering method provides a rough but reasonable starting solution for generative modeling.

Experiment II: Integrated Learning

Fig. 4 shows the result for the crack image obtained by the stochastic gradient algorithm, following the initial solution shown in Fig. 3. It took about 80 iterations of the two steps. Fig. 4.b and Fig. 4.d are the

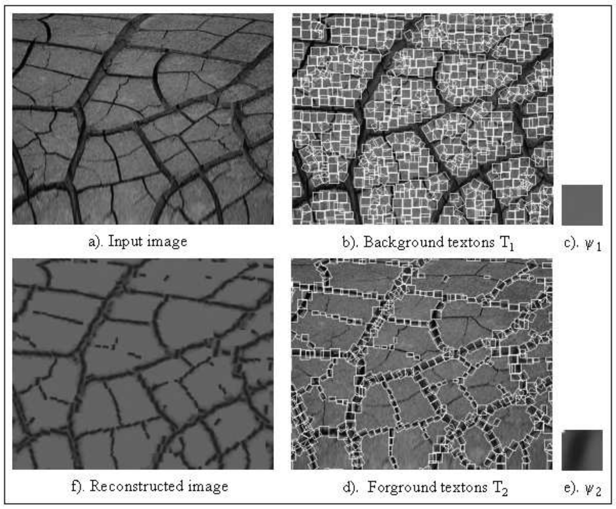


Figure 4: Generative model learning result for the crack image. a) input image, b) and d) are background and foreground textons discovered by the generative model, c) and e) are the templates for the generative model, f) is the reconstructed image from the generative model.

background and foreground texton maps $\mathbf{T}_1$ and $\mathbf{T}_2$ respectively. Fig. 4.c and Fig. 4.e are the learned textons Ψ_1, Ψ_2 respectively. Fig. 4.f is the reconstructed image from learned textons and templates. Compared to the results in Fig. 3, the results in Fig. 4 have more precise texton maps and 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. 5. 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. 7 are the results for a brick image. No point in the template is transparent for the gap lines between bricks. We refer to our web site for a long report and more results.

Experiment III: Random texture sampling and synthesis.

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: Firstly, texton maps are sampled from the Gibbs models $p(\mathbf{T}_1; \boldsymbol{\beta}_1)$ and $p(\mathbf{T}_2; \boldsymbol{\beta}_2)$ respectively. Secondly, background and foreground images are synthesized from the texton maps and texton templates. Thirdly, the final image is generated by combining these two images according the occlusion model. Fig 8 and Fig. 9 are two examples of the two layered model synthesis for the cheetah skin

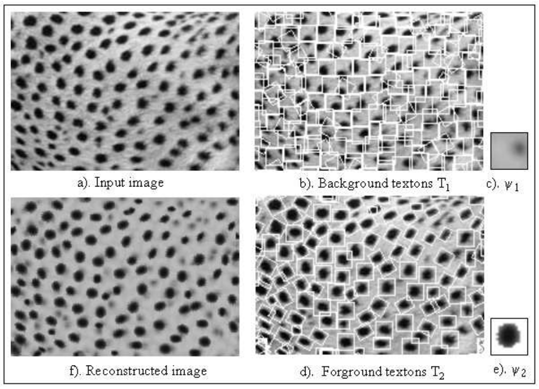


Figure 5: Generative model learning result for a cheetah skin image. The notations are the same as in Fig. 4.

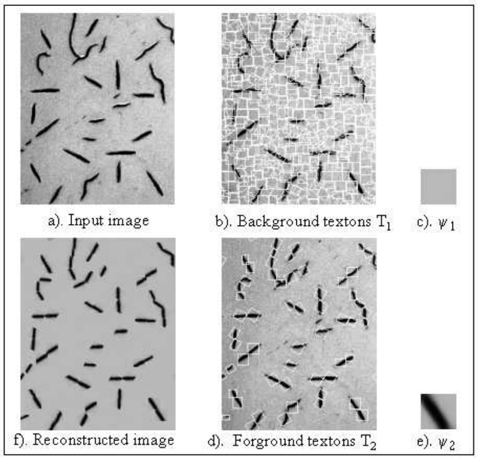


Figure 6: Generative model learning result for a crack image. The notations are the same as in Fig. 4.

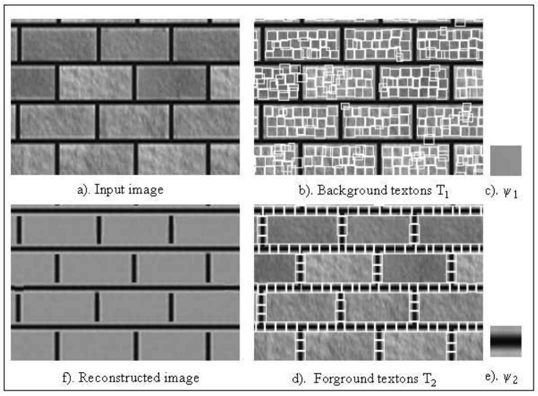


Figure 7: Generative model learning result for a brick image. The notations are the same as in Fig. 4.

pattern. The templates used here are the learned results in Fig 5.

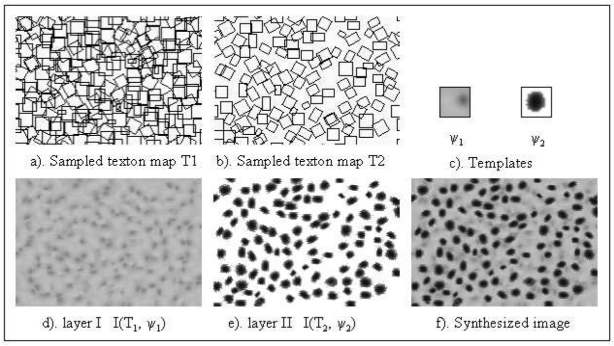


Figure 8: An example of a randomly synthesized cheetah skin image. a) and b) are the background and foreground texton maps sampled from $p(\mathbf{T}_i; \beta_i)$; d) and e) are synthesized background and foreground images from the texton map and templates in c); f) is the final random synthesized image from the generative model.

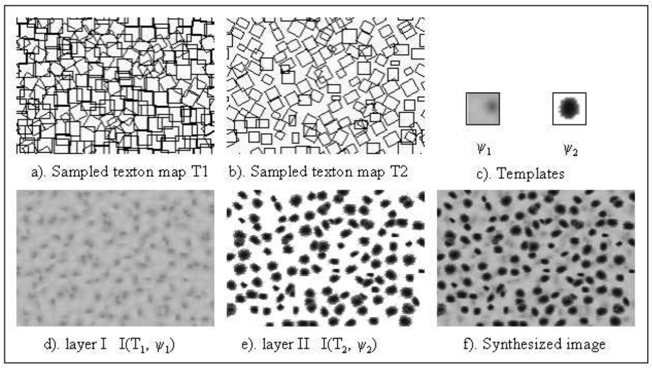


Figure 9: Second example of a randomly synthesized cheetah skin image. Notations are the same as in Fig. 8.

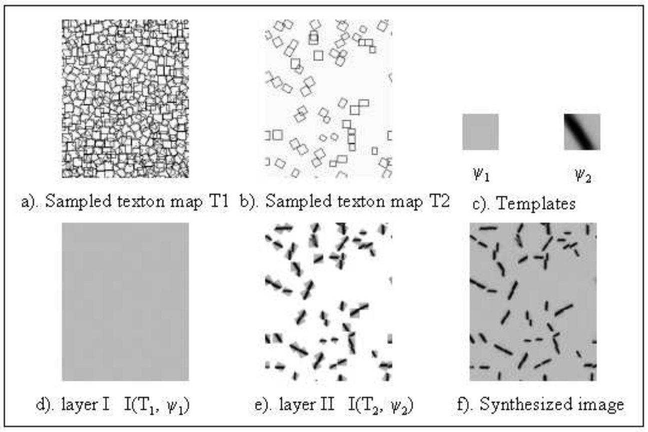


Figure 10: An example of a randomly synthesized crack image. Notations are the same as in Fig. 8.

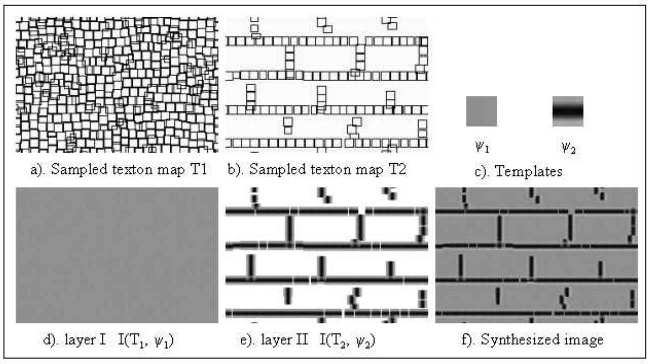


Figure 11: An example of a randomly synthesized brick image. Notations are the same as in Fig. 8.

Figure 11 shows texture synthesis for the crack pattern computed in Figure 6. Figure 11 displays texture synthesis for the brick pattern in Figure 7. Note that, in these texture synthesis experiments, the Markov chain operates with meaningful image elements instead of pixels. The lighting condition is not considered in current experiments. For some texture images, e.g. the cheetah skin image, the lighting globally changes. However, such information is lost in the generative model results. Future experiments will pay attention to this issue.

5 Discussion

The generative method has advantages over previous descriptive method with Markov random fields on pixel intensities.

I). In representation: The neighborhood in the texton map are much smaller than the pixel neighborhood in previous descriptive model [13]. The generative method captures more semantically meaningful features on the texton map.

II). In computation: The Markov chain operating in the texton map can move blobs according to affine transforms and can add or delete a blob through death/birth dynamics, and thus is much more effective than the Markov chain used in traditional Markov random fields which flips one pixel intensity at a time.

Furthermore we show that the integration of descriptive and generative methods is a natural and inevitable 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.

References

[1] 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.

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

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

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

[5] B. Julesz and J. R. Bergen, "Texton, the fundamental elements in preattentive vision and perception for textures", *Bell System Technical Journal*, 62(6), 1983.

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

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

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

[9] 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.

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

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

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

[13] 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.

[14] S. C. Zhu and Cheng-en Guo, "Conceptualization and modeling of visual patterns", Preprint, OSU Vision and Learning Laboratory, September, 2000.