What Are Textons?

Song-Chun Zhu[1], Cheng-en Guo[1], Yingnian Wu[2], and Yizhou Wang[1]

[1] Dept. of Comp. and Info. Sci., Ohio State Univ., Columbus, OH 43210, USA
{szhu, cguo, wangyiz}@cis.ohio-state.edu,
[2] Dept. of Statistics, Univ. of California, Los Angeles, CA, 90095, USA,
ywu@stat.ucla.edu

Abstract. Textons refer to fundamental micro-structures in generic natural images and thus constitute the basic elements in early (pre-attentive) visual perception. However, the word "texton" remains a vague concept in the literature of computer vision and visual perception, and a precise mathematical definition has yet to be found. In this article, we argue that the definition of texton should be governed by a sound mathematical model of images, and the set of textons must be learned from, or best tuned to, an image ensemble. We adopt a generative image model that an image is a superposition of bases from an over-complete dictionary, then a texton is defined as a mini-template that consists of a varying number of image bases with some geometric and photometric configurations. By analogy to physics, if image bases are like protons, neutrons and electrons, then textons are like atoms. Then a small number of textons can be learned from training images as repeating micro-structures. We report four experiments for comparison. The first experiment computes clusters in feature space of filter responses. The second use transformed component analysis in both feature space and image patches. The third adopts a two-layer generative model where an image is generated by image bases and image bases are generated by textons. The fourth experiment shows textons from motion image sequences, which we call movetons.

1 Introduction

Texton refers to fundamental micro-structures in generic natural images and the basic elements in early (pre-attentive) visual perception[8]. In practice, the study of textons has important implications in a series of problems. Firstly, decomposing an image into its constituent components reduces information redundancy and, thus, leads to better image coding algorithms. Secondly, the decomposed image representation often has much reduced dimensions and less dependence between variables (coefficients), therefore it facilitates image modeling which is necessary for image segmentation and recognition. Thirdly, in biological vision the micro-structures in natural images provide an ecological clue for understanding the functions of neurons in the early stage of biological vision systems[1,13]. However, in the literature of computer vision and visual perception, the word "texton" remains a vague concept and a precise mathematical definition has yet to be found.

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2353, pp. 793–807, 2002.

One related mathematical theory for studying image components is harmonic analysis[5] which is concerned with decomposing some classes of mathematical functions. This includes Fourier transforms, wavelet transforms[4], and recently wedgelets and ridgelet[5] and various image pyramids in image analysis[14]. In recent years, there is a widespread consensus that the optimal set of image components should be learned from the *ensemble* of natural images. The natural image ensemble is known to be very different from those classic mathematical functional *classes* from which the Fourier and wavelet transforms were originally derived. This consensus leads to a vast body of work in the study of natural images statistics and image micro-structures, among which two streams are most remarkable.

One stream studies the statistical regularities of natural images. This includes the scale invariance[15], the joint density (histograms) of small image patches (e.g. 3×3 pixels)[10,9], and the joint histogram or correlation of filter responses[3]. Then probabilistic models are derived to account for the spatial statistics[7].

The other stream learns over-complete basis from natural images under the general idea of sparse coding[13]. In contrast to the orthogonal bases or tight frame in the Fourier and wavelet transforms, the learned bases are highly correlated, and a given image is coded by a sparse population in the over-complete basis. Added to the sparse coding idea is independent component analysis (ICA) which decomposes images as a linear superposition of some image bases which minimizes some measure of dependence between the coefficients of these bases[2].

While the over-complete basis presents a major progress in the pursuit of fundamental image elements, one may wonder what are the image structures beyond bases. By an analogy to physics, if we compare the image bases in the sparse or ICA coding to protons, neutrons, and electrons, then what are the "atoms", "molecules", and "polymers" in natural images? How do we learn such structures from generic images? This paper presents one step towards this goal.

We first examine the generative model in the sparse coding scheme. One basic assumption under this scheme is that the bases are independent and identically distributed. To release this assumption, we study the spatial structures of the bases under a generative model and define a texton as a mini-template that consists of a varying number of image bases with some geometric and photometric configurations. Like an atom in physics, a couple of bases in the texton have relatively large coefficients (heavy weights) and thus form the "nucleus" which is augmented by some bases with small coefficients (light weight) like electrons. Then a small number of textons can be learned from training images as repeating micro-structures.

We implement four experiments for comparison. The first experiment computes clusters in feature space of filter responses, as it is done in [11]. This is a discriminative model. The learned cluster centers are transferred into an image icon by pseudo-inverse. Then a typical image structure may appear multiple times as different image icons with translation, rotation and scaling. To address this problem, we did a second experiment which integrates the clustering with

some affine transform, in an idea of transformed component analysis (TCA)[6]. The number of learned clusters (TCA) is largely reduced, but they lack variability. In the third experiment, we adopt a generative model and assume that an image is generated by a number of bases from an over-complete dictionary, and the bases are generated by a set of textons. Both the base and texton sets need to be inferred. Then for each texton, we draw a set of typical examples to show the variety. In the fourth experiment, we show the learning of texton structure from motion image sequence. Motion provides extra information for identifying the basic elements which we call "movetons"

The paper is organized as follows. In Section (2), we first briefly review some previous work on learning over-complete bases and k-mean clustering. Then we report two experiments on transformed components analysis in Section (3). Section (4) presents a generative model for learning textons as mini-templates of bases, and we also show the learning of textons from image sequences. Section (5) discusses some future work.

2 Previous Work

In this section, we briefly review two previous work for computing image components. One is based on a generative model[13] and the other on a discriminative model[11].

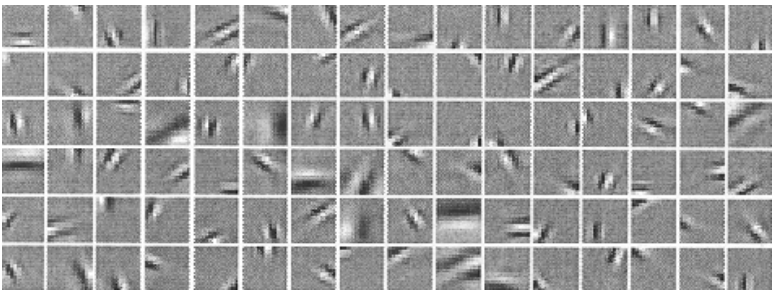

Fig. 1. Some image bases learned with sparse coding by (Olshausen and Field,1997).

2.1 Sparse Coding with Over-Complete Basis

Let $\boldsymbol{\psi} = \{\boldsymbol{\psi}_\ell(u, v), \ell = 1, ..., L\}$ be a set of 2D base functions (kernels or windows), then a dictionary (basis) of local image bases can be obtained by an orthogonal transform A (translating, rotating, and scaling) in a transform space $\Omega_A \ni A$,

$$\Delta = \{\boldsymbol{\psi}_\ell(u, v; A) : A = (x, y, \tau, \sigma) \in \Omega_A, \ell = 1, ..., L.\}$$

The sparse coding scheme[13] and other wavelet transforms[4] are based on a simple *generative image model* where an image $\mathbf{I}$ is a linear superposition of some n_B bases selected from Δ plus a Gaussian noise map $\mathbf{n}$,

$$\mathbf{I} = \sum_{i}^{n_B} \alpha_i \cdot \boldsymbol{\psi}_i + \mathbf{n}, \qquad \boldsymbol{\psi}_i \in \Delta, \ \forall i. \tag{1}$$

We denote the base representation by a *base map*, each base b_j is denoted by its type ℓ_j, coefficient α_j and transforms $x_j, y_j, \tau_j, \sigma_j$.

$$\mathbf{B}(\mathbf{I}) = \{b_j = (\ell_j, \alpha_j, x_j, y_j, \tau_j, \sigma_j) : j = 1, 2, ..., n_B.\}$$

When Δ is over-complete, $\mathbf{B}(\mathbf{I})$ have to be inferred from image $\mathbf{I}$ and a prior model is crucial for the selection of bases. In all current coding literature, the bases are assumed to be independently and identically distributed (iid), so

$$p(\mathbf{B}) = \prod_{j=1}^{n_B} p(b_j), \quad p(b_j) = p(\alpha_j) \cdot \text{unif}_\ell(\ell_j) \cdot \text{unif}(x_j, y_j) \cdot \text{unif}(\tau_j) \cdot \text{unif}(\sigma_j). \tag{2}$$

$\ell_j, x_j, y_j, \tau_j, \sigma_j$ are assumed to be independently and uniformly distributed. When $p(\alpha)$ is chosen to have high kurtosis (peaky at zero and heavy tails), it leads to the sparse coding idea by Olshausen and Field[13]. For example, $p(\alpha)$ can be a Laplacian distribution or a mixture of Gaussians,

$$p(\alpha) \sim \exp\{-|\alpha|/c\} \quad \text{or} \quad p(\alpha) = \sum_{j=1}^{2} \omega_j N(0, \sigma_j).$$

Using such priors, Olshausen and Field learned a set of bases Δ in a non-parametric form from a large ensemble of image patches. The learning is done by maximum likelihood estimation and some of the learned bases are shown in Fig. 1. Such bases capture some image structures and are believed to bear resemblance to the responses of simple cells in V1 of primates.

2.2 K-Mean Clustering in Feature Space

The other related work in computing image elements was shown by Leung and Malik who adopted a *discriminative model*.

For an image $\mathbf{I}$ on a lattice Λ, at each pixel (x, y), a pyramid of image filters $D = \{F_1, ..., F_N\}$ at various scales and orientations are convolved with the image. For illustration, we show the filter pyramid in a cone in Fig. 2. Thus a feature vector representation is extracted, and we denote it by set $\mathbf{F}(\mathbf{I})$

$$\mathbf{F}(\mathbf{I}) = \{F(x, y) = (F_1 * \mathbf{I}(x, y), \cdots, F_k * \mathbf{I}(x, y)) : \quad \forall (x, y) \in \Lambda\}.$$

If there are local image structures occurring repeatedly in image $\mathbf{I}$, then it is reasonable to believe that the vectors in set $\mathbf{F}(\mathbf{I})$ must form clusters. A K-mean clustering algorithm is applied in[11]. Because the feature vector over-constrains a local image patch, a pseudo-inverse can recover an image icon from

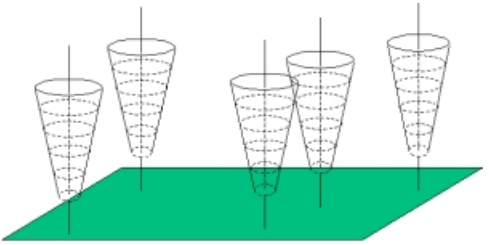

Fig. 2. At each pixel, a pyramid (cone) of filters at various scales and orientations are convolved with the image to extract a feature vector.

each cluster centers. More precisely, let $F_c = (f_{c1}, ..., f_{cN})$ be a cluster center in the N-dimensional feature space, then an image icon ϕ_c (say 15×15 pixels) is computed by

$$\phi_c = \arg\min \sum_{j=1}^{N} (F_j * \phi_c - f_{cj})^2, \quad c = 1, 2, ..., C. \tag{3}$$

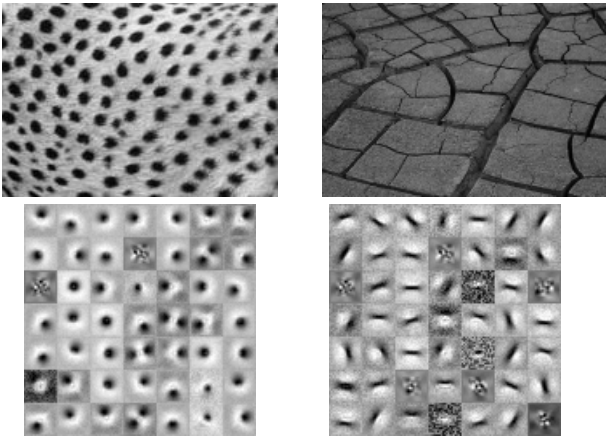

Fig. 3. Two texture images each with 49 image icons $\phi_i, i = 1, 2, ..., 49$ for the cluster centers.

We implement this work and some results are shown in Fig. 3 for $C = 49$ clusters on two typical texture images. Clearly, the cluster centers capture some essential image structures, such as blobs for the cheetah skin pattern, and bars for the crack pattern.

In comparison, though both the generative and discriminative approaches can compute image structures, they are fundamentally different. In a generative model, an image $\mathbf{I}$ is reconstructed by the addition of a number of n_B bases

where n_B is usually in the order of 10^2 times smaller than the number of pixels. This leads to tremendous dimension reduction for further image modeling. In contrast, in a discriminative model, $\mathbf{I}$ is constrained by a set of feature vectors. The number of features is $N \sim 10^2$ times larger than the number of pixels!

While both methods may use the same image pyramid, in the generative model, the base map $\mathbf{B}(\mathbf{I})$ are *random variables* subject to stochastic inference and therefore the computation of $\mathbf{B}(\mathbf{I})$ can be influenced by other variables in a bottom-up/top-down fashion if we introduce more sophisticated models on $p(\mathbf{B})$ as in later section. In contrast, in a discriminative method, the *responses* of *filters* in $\mathbf{F}(\mathbf{I})$ are *deterministic transforms* from the image in a bottom-up fashion which are fixed in the entire computational process.

The results in Figures 1 and 3 manifest one obvious problem that the potentially same image structure appears multiple times which are shifted, rotated, or scaled versions of each other. For the sparse coding scheme, this is caused by cutting natural images into small training patches centered at arbitrary locations. While in the K-mean clustering method, it is caused by extracting a feature vector at every pixel.

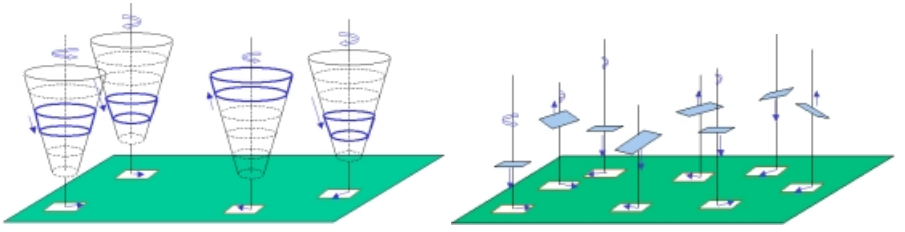

Fig. 4. The transform component analysis allows translation, rotation, and scaling of local image features or patches.

3 Learning Transformed Components

A rather straight-forward fix for the problem raised in the previous section is to introduce transformations as hidden (latent) variables. This is called transformed component analysis (TCA) in [6] and other neural computation literature.

Suppose we extract from image $\mathbf{I}$ a set of n features each with an unknown transformation $A \in \Omega_A$,

$$\Gamma(\mathbf{I}) = \{\gamma_j(A_j) : \quad A_j = (x_j, y_j, \tau_j, \sigma_j),\ j = 1, 2, ..., n\}.$$

We call $\Gamma(\mathbf{I})$ the transformed components of $\mathbf{I}$. In the following, we show two cases as Fig. 4 illustrates.

Transformed Components in Filter Space

In the first case, we compute a feature vector at each pixel by N filters as Section (2.2). Typically we use Laplacian of Gaussian (LoG), Gabor sine (Gsin)

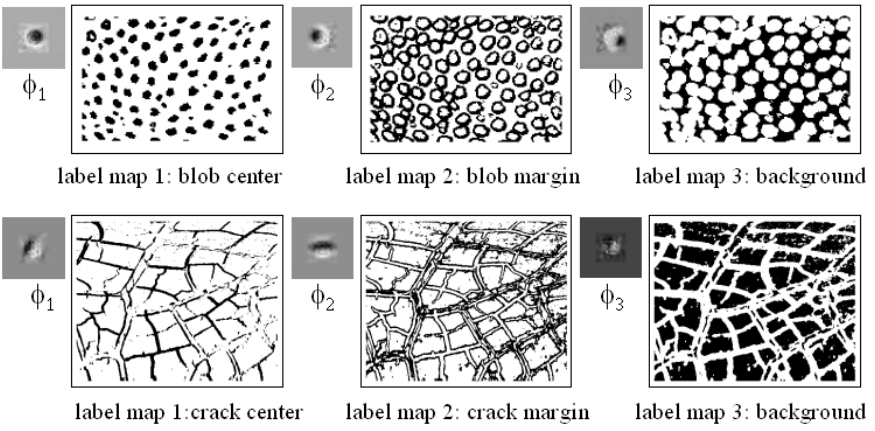

Fig. 5. The learned basic elements ϕ_1, ϕ_2, ϕ_3 for the two patterns are shown by the small image icons, to the right are label maps associated with these icons.

and Gabor cosine (Gcos) at 7 scales and 8 orientations. Thus $N = 7 + 7 \times 8 + 7 \times 8 = 119$. We subsample the image lattice Λ and select $n = |\Lambda|/\delta^2$ locations evenly with $\delta = 4 \sim 8$ pixels. We choose only 2 scales of the filter cone shown by the bold curves in Fig. 4.a. Thus a transformed component $\gamma(A)$ is an $M = 2+2\times 8+2\times 8 = 38$ dimensional feature vector. The translation (x_j, y_j, τ_j) corresponds to shift and rotation of the filter cone, and σ_j is the selection of scale from the cone, as the arrows in Fig. 4.a show. Therefore, the transform A indeed corresponds to the selection of filters in the filter cones.

$$\gamma(A) = (F_1(A) * \mathbf{I}, F_2(A) * \mathbf{I}, ..., F_M(A) * \mathbf{I}).$$

Γ poses constraints on the image $\mathbf{I}$ by a model $p(\mathbf{I}|\Gamma)$. We assume that $\gamma_j, j = 1, ..., n$ form a few tight clusters (mixture of Gaussians) after proper transforms $A_j, j = 1, ..., n$. These transforms are inferred as hidden variables so that the transformed components $\gamma_j, j = 1, 2, ..., n$ are aligned. Given a training image $\mathbf{I}^{\text{obs}}$, an EM-algorithm can be used to infer the hidden transforms and compute the cluster centers. The computation is governed by MLE.

Let $(f_{c1}, ..., f_{cM})$, $c = 1, ..., C$ be the cluster centers in the M-dimensional feature space, we can recover the C image icons $\phi_c, c = 1, ...C$ for the cluster centers by pseudo-inverse, as it is discussed in the previous section.

Fig. 5 shows $C = 3$ centers ϕ_1, ϕ_2, ϕ_3 for the cheetah and crack patterns. The image maps next to each center element $\phi_c, c = 1, 2, 3$ is a label map where the black pixels are classified to this cluster. Clearly, the three elements are respectively: ϕ_1 — the center the blobs (or cracks), ϕ_2 — the rim of the blobs (or cracks), and ϕ_3— the background. In the experiments, the translation of each filter cone is confined to within a local area (say 5×5 pixels), so that the image lattice are covered by the effective areas of the cone. Without such constraints, all cones may move to a background pixel to form a single cluster.

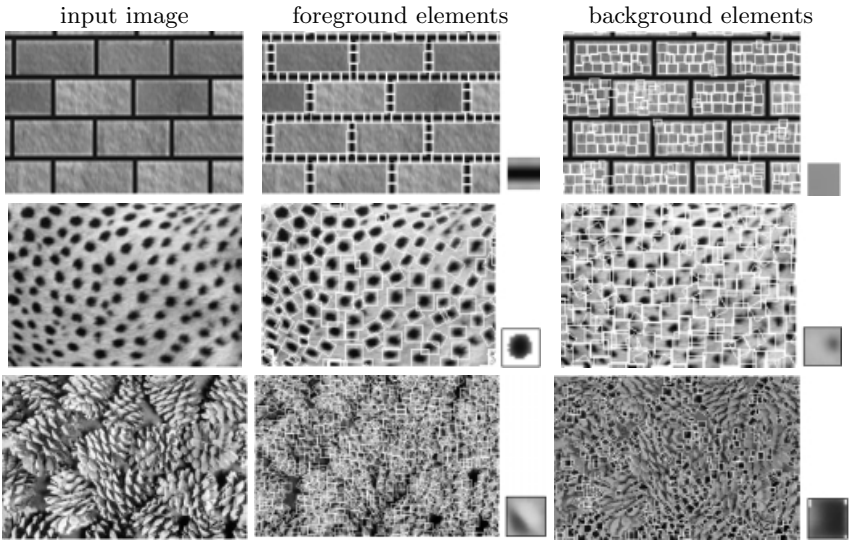

Fig. 6. The learned image icons (cluster centers) ϕ_1, ϕ_2 for the three patterns are shown by the small images, to the left are windows of transformed versions of the image patches associated with these icons.

Transformed Components in Image Patches

In this experiment, we replace the feature representation by image windows of $11 \times 11 = 121$ pixels. These windows can move within a local area and can be rotated and scaled as Fig. 4.b shows. Thus each transformed component $\gamma(A)$ is a local image patch. Like the TCA in feature space, these local patches are transformed to form tight clusters in the 121-space by an EM-algorithm. The cluster centers $\phi_c, c = 1, ..., C$ are the repeating micro image structures.

Fig. 6 shows the $C = 2$ centers for the cheetah, crack, and pine cone patterns. The image maps next to each center element $\phi_c, c = 1, 2$ is a set of windows which are transformed versions of the elements. ψ_1 corresponds to the blobs, bars, and contrasts for the three patterns respectively. ψ_2 are for the backgrounds.

In summary, the results in Figures 5 and 6 present a major improvement from those in Figures 1 and 3, due to the the inference of hidden variables for transforms. However, there are still two main problems which we should resolve in the next section.

1. The transformed components $\{\gamma_j, j = 1, ..., n\}$ only pose some constraints on image **I**, and they lack an explicit generative image model. As a result, the learned elements $\psi_\ell, \ell = 1, ..L$ are contaminated by each other, due to overlapping between adjacent image windows or filter cones, see Fig. 4.
2. There is a lack of variability in the learned image elements. Take the cheetah skin pattern as an example, the blobs in the input image display deformations, whereas the learned elements ϕ_1 are round-shaped. This is caused

by the assumption of Gaussian distribution for the clusters. In reality, the clusters have higher order structures which should be explored effectively.

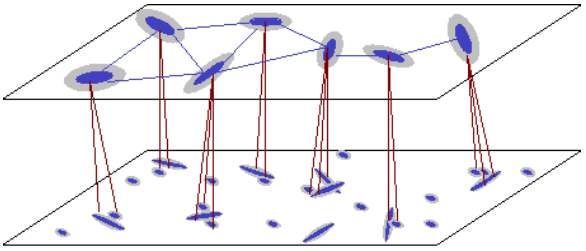

Fig. 7. A two-layer base map and textons are groups of bases.

4 Texton Learning: From Bases to Textons

To resolve the problems raised in the previous section, we propose to define "texton" as a mini-template that consists of a number of bases at some geometric and photometric configurations.

We adopt the explicit generative model in equation (1) where an image $\mathbf{I}$ is generated by n_B bases. $\mathbf{B}(\mathbf{I}) = \{b_j = (\ell_j, \alpha_j, x_j, y_j, \tau_j, \sigma_j) : j = 1, 2, ..., n_B.\}$ $\mathbf{B}(\mathbf{I})$ is selected from an over-complete basis $\mathit{\Delta}$ with three base functions $\boldsymbol{\psi} = \{\psi_1 = \text{LoG}, \psi_2 = \text{Gcos}, \psi_3 = \text{Gsin}\}$.

In the previous work, the bases are assumed to be independently distributed, see equation (2). To go beyond the sparse/image coding scheme, we need a more sophisticated probabilistic model for $p(\mathbf{B})$ which should account for the spatial relations between bases in $\mathbf{B}$.

Fig. 7 illustrates a model for the base map $\mathbf{B}$. It is generally true that the bases $\mathbf{B}(\mathbf{I})$ can be divided into two layers. Bases in the upper layer usually have relatively larger coefficients α_j (heavy) and capture some larger image structures. Bases in the lower layer have smaller α_j (light) and relatively higher frequencies. By an analogy to physics, we call the bases in the upper layer the "nucleus bases" as they have heavy weights like protons and neutrons, and the bases in the lower layer the "electron bases" which are light. It is generally observed that a nucleus base is surrounded by a few electron bases as the arrows in Fig. 7 show. Some bases in the lower level may not be associated with any nucleus bases, and we call them "free electrons". Furthermore the nucleus bases may also form some spatial groups in the upper layer, such as lines and curves, or other configurations.

Fig. 8 illustrates a real example of star pattern. For clarity, we choose a pattern where the texton elements are well separable, but this is not a necessary condition for the theory to work, as we show by examples later. The three bases and their corresponding symbolic sketches are shown on the left of the figure.

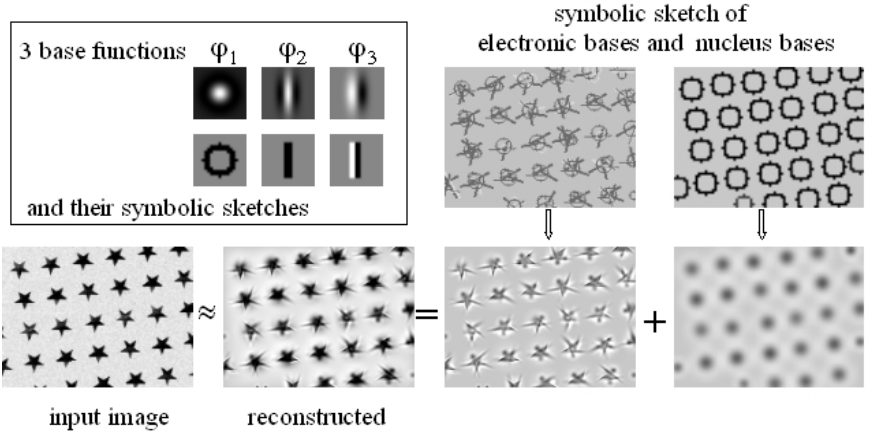

Fig. 8. Reconstructing a star pattern by two layers of bases. An individual star is decomposed into a LoG base in the upper layer for the body of the star plus a few other bases (mostly Gcos, Gsin) in the lower layer for the angles.

The symbolic sketches of the two base layers are shown on the right. In the bottom row, each layer of the bases generates an image (shown by arrow), and the reconstructed image is the linear sum of the two images. The residue between the input and the reconstructed image is assumed to be a Gaussian noise map **n**.

The regularity of the pattern is reflected by the organization of the bases. In this example, the nucleus base for a star is a LoG base, which has up to six electron bases (one LoG, five Gcos's) as Fig. 9 shows (leftmost). This forms a mini-template. The dash link means that this electron base may not appear in all instances (i.e. it appears with a probability). The right side of Fig. 9 displays some typical configurations of the texton template and their corresponding image appearances which are different variations of stars.

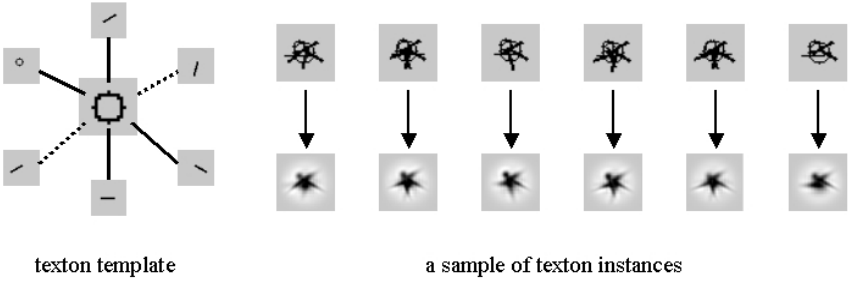

Fig. 9. Left: The texton template for the star pattern. Right: A sample of typical texton instances — the sketches (1st row) and image appearances (2nd row).

In the following, we explain our augmented generative model for discovering the textons from images.

Let $\boldsymbol{\pi} = \{\pi_1, ..., \pi_k\}$ be k deformable texton templates.[1] Usually $k \leq 3$ for an image and each template $\pi_\ell, \ell = 1, ..., k$ may include m_ℓ bases. For example, a template with $m_\ell = 3$ bases is

$$\pi = (\ (\ell_1, \alpha_1, \tau_1, \sigma_1),\ (\ell_2, \alpha_2, \delta x_2, \delta y_2, \delta \tau_2, \delta \sigma_2),\ \ (\ell_3, \alpha_3, \delta x_3, \delta y_3, \delta \tau_3, \delta \sigma_3)\ )$$

for the types of the three bases, and their relative positions, orientations and scales. Then a texton instance is a deformed version of one of the k texton templates. We denote a texton instance by

$$t_j = (\ell_j, \alpha_j, x_j, y_j, \tau_j, \sigma_j, (a_{jq}), q = 1, 2, ..., m_{\ell_j}),$$

where $\ell_j \in \{1, ..., k\}$ is the type of template, $x_j, y_j, \tau_j, \sigma_j$ are for the transform of the whole texton, and $a_{jq} \in \{0, 1\}$ indicates whether or not a base appears in the texton. This introduces a new level of variables called the "texton map" $\mathbf{T}$.

$$\mathbf{T} = \{t_j, j = 1, 2, ..., n_T\}$$

Therefore, an image $\mathbf{I}$ is generated by the base map $\mathbf{B}$ using some base functions $\boldsymbol{\psi}$, and the base map $\mathbf{B}$ is then generated by a texton map $\mathbf{T}$ using some texton templates $\boldsymbol{\pi}$

$$\mathbf{T} \xrightarrow{\boldsymbol{\pi}} \mathbf{B} \xrightarrow{\boldsymbol{\psi}} \mathbf{I}.$$

Without loss of generality, suppose we have one training image $\mathbf{I}^{\text{obs}}$, according to our generative model, the likelihood for $\mathbf{I}^{\text{obs}}$ is

$$p(\mathbf{I}^{\text{obs}}; \Theta) = \int p(\mathbf{I}^{\text{obs}}|\mathbf{B}; \boldsymbol{\psi}) p(\mathbf{B}|\mathbf{T}; \boldsymbol{\pi}) p(\mathbf{T}; \boldsymbol{\kappa})\ \ d\mathbf{B}\ d\mathbf{T}$$

where the latent variables $\mathbf{B}$ and $\mathbf{T}$ are summed out. $p(\mathbf{I}^{\text{obs}}|\mathbf{B}; \boldsymbol{\psi})$ is a Gaussian distribution for the noise $\mathbf{n}$ following equation (1). We divide the n_B bases in $\mathbf{B}$ into $n_T + 1$ classes

$$\mathbf{B} = \varpi_0 \cup \varpi_1 \cup \cdots \cup \varpi_{n_T}.$$

Bases in ϖ_0 are free electrons and are subject to the independence distribution $p(b_j)$ in equation (2). Bases in other classes form a deformable template.

$$p(\mathbf{B}|\mathbf{T}; \boldsymbol{\pi}) = p(|\varpi_0|) \prod_{b_j \in \varpi_0} p(b_j) \prod_{c=1}^{n_T} p(\varpi_c | t_c; \boldsymbol{\pi}_{\ell_c}).$$

$p(\mathbf{T}; \boldsymbol{\kappa})$ is another distribution which accounts for the number of textons n_T and the spatial relationship among them. It can be a Gibbs model as in [7] and for clarity, we assume the textons are independent at this moment.

[1] To clarify the notations, we use π for a texton template, ψ for a base function, and ϕ for an image icon by various clustering.

Then the goal is to learn the parameters $\Theta = (\boldsymbol{\psi}, \boldsymbol{\pi}, \boldsymbol{\kappa})$ by maximum likelihood, or equivalently minimize a Kullback-Leibler divergence between $p(\mathbf{I}; \Theta)$ and a underlying probability of images,

$$\Theta^* = (\boldsymbol{\psi}, \boldsymbol{\pi}, \boldsymbol{\kappa})^* = \arg\min KL(f(\mathbf{I})||p(\mathbf{I}; \Theta)) = \arg\max \log p(\mathbf{I}^{\text{obs}}; \Theta) + \epsilon,$$
$$= \int [\frac{\partial \log p(\mathbf{I}^{\text{obs}}|\mathbf{B}; \boldsymbol{\psi})}{\partial \boldsymbol{\psi}} + \frac{\partial \log p(\mathbf{B}|\mathbf{T}; \boldsymbol{\pi})}{\partial \boldsymbol{\pi}} + \frac{\partial \log p(\mathbf{T}; \boldsymbol{\kappa})}{\partial \boldsymbol{\kappa}}] p(\mathbf{B}, \mathbf{T}|\mathbf{I}^{\text{obs}}; \Theta) d\mathbf{B} d\mathbf{T}$$

ϵ is an approximation error which diminishes as sufficient data are available for training. In practice, ϵ may decide the complexity of the models, and thus the number of base functions L and template number k. The algorithm for solving MLE iterates two steps by stochastic gradient, like an EM algorithm, but it is guaranteed for global convergence.

1. Design a Markov chain Monte Carlo (MCMC) sampler to draw fair samples of the latent variables from posterior probability for a current Θ,

$$(\mathbf{B}, \mathbf{T}) \sim p(\mathbf{B}, \mathbf{T}|\mathbf{I}^{\text{obs}}; \Theta) \propto p(\mathbf{I}^{\text{obs}}|\mathbf{B}; \boldsymbol{\psi}) p(\mathbf{B}|\mathbf{T}; \boldsymbol{\pi}) p(\mathbf{T}; \boldsymbol{\kappa}).$$

   This includes Metropolis jump dynamics for the death/birth of bases and textons, the switching of base types and texton types, the assignment of bases to the classes ϖ_c etc., and diffusion dynamics which adjust the positions, scales, and orientations of bases and textons. The algorithm is initialized by a matching pursuit method[12] which often yields a very good initial base map $\mathbf{B}$.
2. Replace the integration by importance sampling, and then adjust the parameters Θ through MLE. In general, we can learn the base functions $\boldsymbol{\psi}$, the texton templates $\boldsymbol{\pi}$, and their spatial relation $\boldsymbol{\kappa}$. For simplification, we fix $\boldsymbol{\psi}$ to the Log, Gcos, Gsin bases which are proven to be good enough for many images.

Beside the running example of the star pattern, we show a few more examples for the cheetah skin and bird patterns in Fig. 10. For these patterns, the textons are well isolated, and we have one texton template for cheetah pattern and three texton templates for bird pattern. The nucleus of the texton is a LoG base, which is augmented by some electron bases to account for the blob deformations or bird wings. The texton instances display the variety of image appearances.

To go beyond separable textons, we further group the nucleus bases into polygons (k-gons). This is in spirit similar to Julesz's k-gon representation[8]. The difference is that Julesz's k-gons are based on points (pixels), whereas we define on bases. The texton templates discussed so far are 1-gon special cases. In general we can discover k-gon structures by clustering (i.e. improving the model $p(\mathbf{T}; \boldsymbol{\kappa})$. Pressed by space limitation, we choose not to discuss the details which is quite straightforward, and show two examples in Figures 11. The 2-gon for brick for elongated lines and "L"-shaped turns. The 3-gon for T-shaped junctions. The k-gons for the straw form aligned bars, or parallel and bifurcation structures.

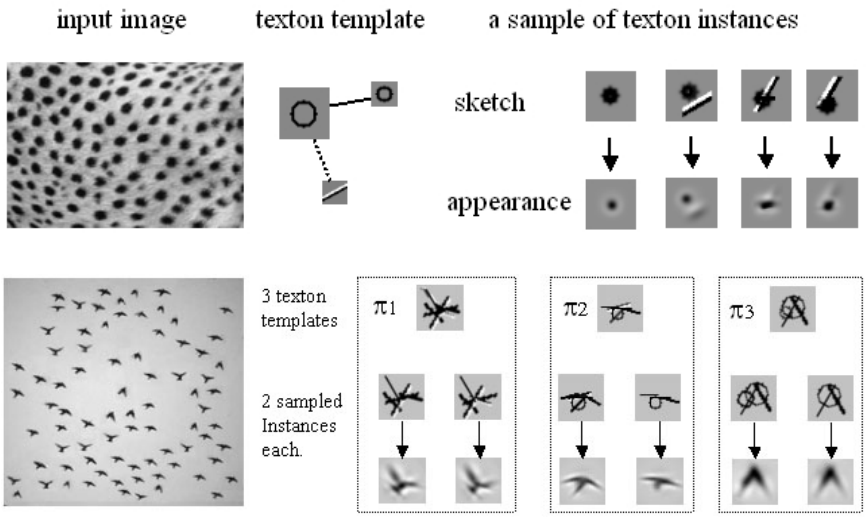

Fig. 10. The texton templates for the cheetah (flying bird) patterns, and a sample of typical texton instances — the sketches (1st row) and image appearances (2nd row)

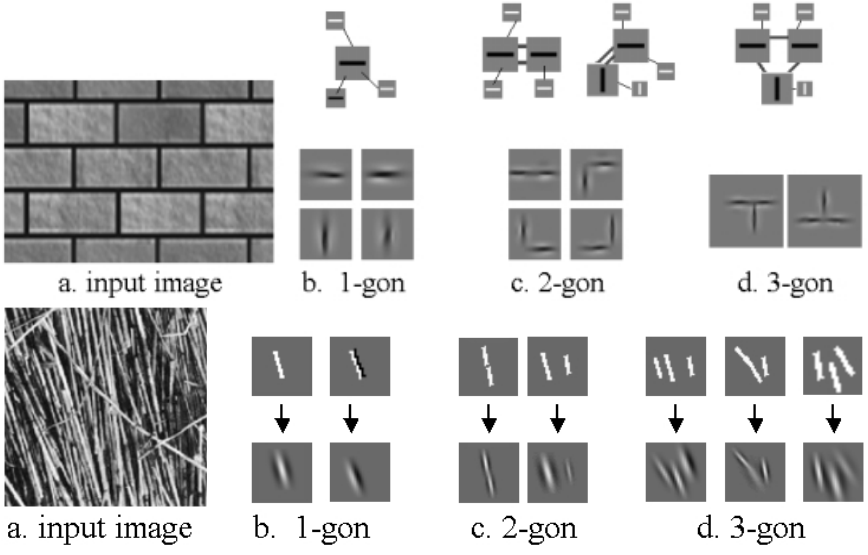

Fig. 11. The texton templates for the brick and straw patterns. The templates for k-gon $k = 1, 2, 3$ and corresponding image appearances.

Finally, we show one example that textons can be learned from motion sequence. Fig. 12 reports an experiment that various snow flakes can be tracked and their shape learned over the motion sequence. The blurred snow scenes make

the clustering of snow flakes difficult in single image, and the motion provides much strong cues.

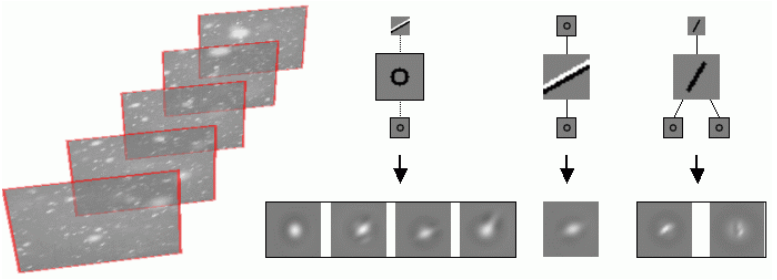

Fig. 12. The texton templates for the snow pattern learned from a movie sequence. The three types of templates for textons and corresponding image appearances.

5 Discussion

From a series of experiments, we learn that the generative model is a key for discovering fundamental structures. Texton should be explained as parameter functions in a generative model. We'd like to answer the following questions to conclude the paper.

1. Since the image appearance of a texton is a linear sum of some bases, would this be just equal to the one layer model? No, as shown in the experiments, the image appearance for a texton is NOT Gaussian distributed. The grouping of bases into textons accounts for high order statistics and enriches the variety of elements.

2. What if the image elements are not well separable? In many cases, bases are combined to form large structures, as atoms are grouped to molecules and polymers and share electrons. Some k-gon patterns are shown in Fig. 11, and more examples will be shown on our website http://www.cis.ohio-state.edu/oval.

3. What is next? The base function ψ which should be learned together with the textons, in particular, for patterns such as hair, water flow etc.

Acknowledgments. This work is supported partially by two NSF grants IIS 98-77-127 and IIS-00-92-664, and an ONR grant N000140-110-535.

References

1. Barlow, H.B. "Possible principles underlying the transformation of sensory messages". In *Sensory Communication*, ed. W.A. Rosenblith, pp217-234, MIT Press, Cambridge, MA, 1961.

2. Bell, A. J. and Sejnowski, T.J. "An information maximization approach to blind separation and blind deconvolution", *Neural Computation*, 7(6): 1129-1159, 1995.
3. Buccigrossi, R.W. and Simoncelli, E.P. "Image compression via joint statistical characterization in the wavelet domain", *IEEE trans on Image Processing*, 8(12):1688-701, 1999.
4. Coifman, R.R. and Wickerhauser, M.V. "Entropy based algorithms for best basis selection." *IEEE Trans. on Information Theory.*, Vol.38, pp713-718, 1992.
5. Donoho, D.L. Vetterli, M. DeVore, R.A. and Daubechie, I "Data compression and harmonic analysis", *IEEE Trans. Information Theory.* 6, 2435-2476, 1998.
6. Frey, B. and Jojic, N. "Transformed component analysis: joint estimation of spatial transforms and image components", *Proc. of Int'l Conf. on Comp. Vis.*, Corfu, Greece, 1999.
7. Guo, C. E. Zhu, S. C. and Wu, Y. N. "Visual learning by integrating descriptive and generative methods", *Proc. of Int'l Conf. on Computer Vision*, Vancouver, CA, July, 2001.
8. Julesz, B. "Textons, the elements of texture perception and their interactions", *Nature*, 290, 91-97, 1981.
9. Koloydenko, A. *Modeling natural microimage statistics*, Ph.D. Thesis, Dept. of Math and Stat., UMass, Amherst, 2000.
10. Lee, A.B. Huang, J.G. and Mumford, D.B. "Random collage model for natural images", *Int'l J. of Computer Vision*, oct. 2000.
11. Leung, T. and Malik, J. "Recognizing surface using three-dimensional textons", *Proc. of 7th ICCV*, Corfu, Greece, 1999.
12. Mallat, S. G. "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. on PAMI*, vol.11, no.7, 674-693, 1989.
13. Olshausen, B. A. and Field, D. J. "Sparse coding with an over-complete basis set: A strategy employed by V1?", *Vision Research*, 37:3311-3325, 1997.
14. Simoncelli, E.P. Freeman, W.T. Adelson, E.H. Heeger, D.J. "Shiftable multiscale transforms", *IEEE Trans. on Info. Theory*, 38(2): 587-607, 1992.
15. Zhu, S.C. and Mumford, D.B. "Prior learning and Gibbs reaction-diffusion", *IEEE Trans. PAMI*, vol.19, no.11, Nov. 1997.