To Appear in the Int'l Journal of Computer Vision: Special Issue on Texture.

What Are Textons?

Song-Chun Zhu, Cheng-en Guo, Yizhou Wang, and Zijian Xu

Departments of Statistics and Computer Science
University of California, Los Angeles
Los Angeles, CA 90095

{*sczhu*, *cguo*, *wangyz*, *zjxu@stat.ucla.edu*}

Abstract

Textons refer to fundamental micro-structures in natural images (and videos) and are considered as the atoms of pre-attentive human visual perception (Julesz, 1981). Unfortunately, the word "texton" remains a vague concept in the literature for lacking of a good mathematical model. In this article, we first present a three-level generative image model for learning textons from texture images. In this model, an image is a superposition of a number of image bases selected from an over-complete dictionary including various Gabor and Laplacian of Gaussian functions at various locations, scales, and orientations. These image bases are, in turn, generated by a smaller number of texton elements, selected from a dictionary of textons. By analogy to the *waveform-phoneme-word* hierarchy in speech, the *pixel-base-texton* hierarchy presents an increasingly abstract visual description and leads to dimension reduction and variable decoupling. By fitting the generative model to observed images, we can learn the texton dictionary as parameters of the generative model. Then the paper proceeds to study the *geometric*, *dynamic*, and *photometric* structures of the texton representation by further extending the generative model to account for motion and illumination variations. 1). For the geometric structures, a texton consists of a number of image bases with deformable spatial configurations. The geometric structures are learned from static texture images. 2). For the dynamic structures, the motion of a texton is characterized by a Markov chain model in time which sometimes can switch geometric configurations during the movement. We call the moving textons as "motons". The dynamic models are learned using the trajectories of the textons inferred from video sequence. 3). For photometric structures, a texton represents the set of images of a 3D surface element under varying illuminations and is called a "lighton" in this paper. We adopt an illumination-cone representation where a lighton is a texton triplet. For a given light source, a lighton image is generated as a linear sum of the three texton bases. We present a sequence of experiments for learning the geometric, dynamic, and photometric structures from images and videos, and we also present some comparison studies with K-mean clustering, sparse coding, independent component analysis, and transformed component analysis. We shall discuss how general textons can be learned from generic natural images.

1 Introduction

The purpose of vision, biologic and machine, is to compute a hierarchy of increasingly abstract interpretations of the observed images (or image sequences). Therefore it is of fundamental importance to know what are the descriptions used at each level of interpretation. By analogy to physics concepts, we wonder what are the visual "electrons", visual "atoms", and visual "molecules" for visual perception. The pursuit of basic images and perceptual elements is not just for intellectual curiosity but has important implications in a series of practical problems. For example,

1. *Dimension reduction.* Decomposing an image into its constituent components reduces information redundancy and leads to lower dimensional representations. As we will show in later examples, an image of 256×256 pixels can be represented by about 500 image bases, which are, in turn, reduced to 50-80 texton elements. The dimension of representation is thus reduced by about 100 folds. Further reductions are achieved in motion sequences and lighting models.

2. *Variable decoupling.* The decomposed image elements become more and more independent of each other and thus are spatially nearly decoupled. This facilitates image modeling which is necessary for visual tasks such as segmentation and recognition.

3. *Biologic modeling.* Micro-structures in natural images provide ecological clues for understanding the functions of neurons in early stages of biologic vision systems[3, 30].

In the literature, there are several threads of research investigating fundamental image structures from different perspectives, with many questions left unanswered.

Firstly, in *neurophysiology*, the cells in the early visual pathway (retina, LGN, and V1) of primates are found to compute some basic image structures at various scales and orientations [19]. This motivated some well-celebrated image pyramid representations including Laplacian of Gaussians (LoG), Gabor functions, and their variants [11, 32]. However, very little is known about how V1 cells are grouped into larger structures in higher levels (say, V2 and V4). Similarly, it is unclear what are the generic image representations beyond the image pyramids in image analysis.

Secondly, in *psychophysics*, Julesz [21] and colleagues discovered that pre-attentive vision is sensitive to some basic image features while ignoring other features. His experiments measured the response time of human subjects in detecting a target element among a number of distractors in the background. For example, Fig. 1 shows two pairs of elements in comparison. The response time for the left pair is instantaneous ($100 - 200\ ms$) and independent of the number of distractors. In contrast, for the right pair the response time increases linearly with the number of distractors. This discovery was very important in psychophysics and motivated Julesz to conjecture a pre-attentive stage that detects some atomic structures, such as elongated blobs, bars, crosses, and terminators [21], which he called "textons" for the first time.

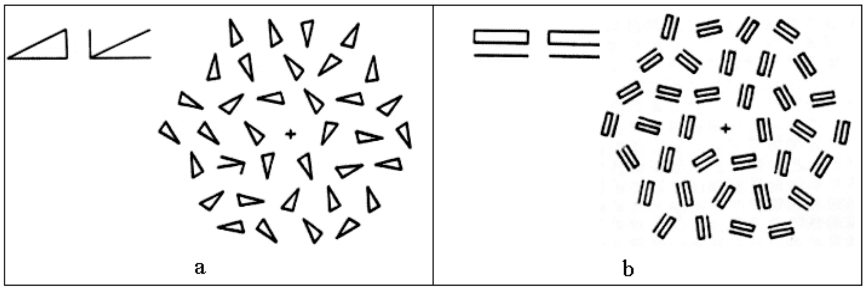


Figure 1: Two typical examples of searching a target element among a number of background distractors. The search time for the left pair is constant independent of the number of distractors, while it increases linearly with the number of distractors for the right pair. After (Julesz, 1981).

The early texton studies were limited by their exclusive focus on artificial texture patterns instead of natural images. It was shown that the perceptual textons could be adapted through training [22]. Thus the dictionary of textons must be associated with or learned from the ensemble of natural images. Despite the significance of Julesz's experiments, there have been no rigorous mathematical definitions for textons. Later in this paper, we argue that textons must be defined in the context of a generative model of images.

Thirdly, in *harmonic analysis*, one treats images as 2D functions, then it can be shown that some classes of functionals (such as Sobolev, Hölder, Besov spaces) can be decomposed into bases, for example, Fourier, wavelets [8], and more recently wedgelets and ridgelets [9]. It was proven that the Fourier, wavelets, and ridgelets bases are independent components for various functional spaces (see [9] and refs therein). But the natural image ensemble is known to be very different from those classic mathematical functional spaces.

The fourth perspective, and the most direct attack to the problem, is the study of *natural image statistics* and *image component analysis*. One important work is done by Olshausen and Field [30] who learned some over-complete image bases from natural image patches (12 × 12 pixels) with the idea of sparse coding. In contrast to the orthogonal and complete bases in Fourier analysis or tight frame in wavelet transforms, the learned bases are highly correlated, and a given image is coded by a sparse population in the over-complete dictionary. 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 [4]. Other interesting work includes micro-image (3 × 3 pixels) patches by Lee and Mumford who show the 3 × 3 pixel patches form very low dimensional and tight manifold in the 7-dimensional sphere [24].

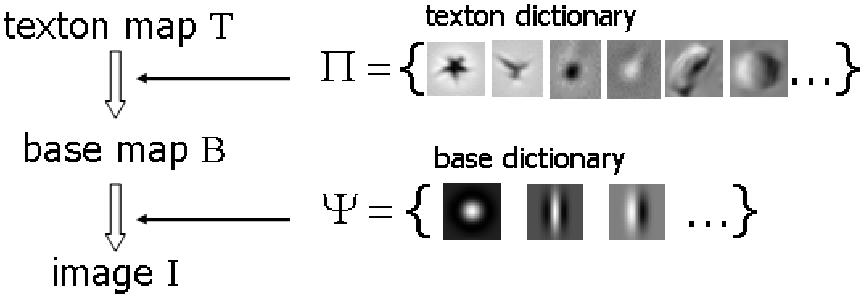


Figure 2: A three-level generative model: an image **I** (pixels) is a linear addition of some image bases selected from a base dictionary **Ψ**, such as Gabors and Laplacian of Gaussians. The base map is further generated by a smaller number of textons selected from a texton dictionary **Π**. Each texton consists of a number of bases in certain deformable configurations, for example, star, bird, cheetah blob, snowflake, bean, etc.

In this paper, we start with a three-level generative image model in Fig. 2. In this model, an image **I** is a superposition of a number of image bases selected from an over-complete dictionary **Ψ** including various Gabor and Laplacian of Gaussian bases at various locations, scales, and orientations. We represent the image bases as attributed points, and denote them by a base map **B**. The base map **B** is, in turn, generated by a smaller number of texton elements, denoted by a texton map **T**. The texton elements are selected from a dictionary of textons **Π**. In this generative model, the base map **B** and texton map **T** are *hidden (latent) variables* and the dictionaries **Ψ** and **Π** are *parameters* that should be learned through fitting the model to observed images. By analogy to the *waveform-phoneme-word* hierarchy in speech, the *pixel-base-texton* hierarchy presents an increasingly abstract visual description. This representation leads to enormous dimension reduction and variable decoupling. We conjecture that for natural images the size of the two dictionaries $\mathbf{\Psi}$ and $\mathbf{\Pi}$ should be in the order of $O(10)$ and $O(10^3)$ respectively which are the number of phonemes and words for most natural languages. Intuitively, textons are meaningful objects viewed at distance (i.e. small scale), such as stars, birds, cheetah blobs, snowflakes, beans, etc. (see Fig. 2).

This generative model extends existing work in the following aspects. Firstly, it is based on larger images instead of small image patches and thus accounts for the inter-relationship of the image components. Secondly, it no longer assumes independence of image bases and accounts for some spatial dependence of bases and larger image structures. Thirdly, it defines textons formally associated with a generative model, which is in contrast to some vague concepts in discriminative models.

Then the paper extends the generative model to motion sequence and lighting variations and studies the *geometric*, *dynamic*, and *photometric* structures of the texton representation.

1. For geometric structures, a texton consists of a small number of image bases with deformable spatial configurations. The geometric structures are learned from a static texture image with repeated elements.

2. For dynamic structures, the motion of a texton is charactered by a Markov chain model which may switch geometric configurations over time. We call the moving textons as "motons". The Markov chain models are learned using the trajectories of the textons and their constituent image bases are inferred from the video sequences.

3. For photometric structures, a texton represents a three-dimensional surface element under varying illuminations and is called a "lighton". A lighton is a triplet of 2D textons (i.e. it consists of three 2D textons). For a given light source, a lighton image is generated as a linear sum of the three textons.

To summarize, if we view a video sequence of 256×256 pixels with 256 frames as a point in 256^3-space, then the texton dictionary that we define above contains lower-dimensional manifolds living in such space, and corresponds to fundamental structures in images and videos. These manifolds are specified by parameters for geometric transforms and deformations, dynamics, and photometric variabilities.

Before we accept the three-level generative model, we tried other competitive ideas for defining textons, such as K-mean clustering [25] and transformed component analysis [14] on the feature and image patch space. We will present the results of our early studies for comparison.

The paper is organized as follows. In Section (2), we briefly review some previous work on learning over-complete image bases and K-mean clustering with experiments. In Section (3), we report two experiments on transformed components analysis on the feature space and image patch space respectively. Section (4) presents the generative model for learning textons from static images. Section (5) presents the motons which are learned from image sequences. Section (6) presents the lightons which are 3D surface elements under varying illuminations. Section (7) discusses some remaining issues and future work.

2 Background: over-complete basis and K-mean clustering

In this section, we review two previous studies for computing image components to provide some background and make comparisons. One is the sparse coding with over-complete dictionary – a work based on *generative modeling* [30] and the other is the K-mean clustering for textons – based on *discriminative modeling* [25]. The differences and relationship between generative and discriminative models are referred to [37].

2.1 Sparse coding with over-complete basis

In image coding, one starts with a dictionary of base functions

$$\mathbf{\Psi} = \{\boldsymbol{\psi}_\ell(u,v), \ell = 1, ..., L_\psi\}.$$

For example, some commonly used bases are Gabor, Laplacian-of-Gaussian (LoG), and other wavelet transforms. Let $A = (x, y, \tau, \sigma)$ denote the translation, rotation and scaling transform of a base function, and $G_A \ni A$ the orthogonal transform space (group), then we obtain a set of image bases Δ ,

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

A simple *generative image model*, adopted in almost all image coding schemes, assumes that an image $\mathbf{I}$ is a linear superposition of some image bases selected from Δ plus a Gaussian noise image $\mathbf{n}$.

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

where n_B is the number of bases and α_i is the *coefficient* of the i-th base $\boldsymbol{\psi}_i$.

As Δ is over-complete[1], the variables $(\ell_i, \alpha_i, x_i, y_i, \tau_i, \sigma_i)$ indexing a base ψ_i are treated as latent (hidden) variables and must be inferred probabilistically, in contrast to deterministic transforms such as the Fourier transform. All the hidden variables are summarized in a *base map*,

$$\mathbf{B} = (n_B, \{b_i = (\ell_i, \alpha_i, x_i, y_i, \tau_i, \sigma_i) : i = 1, 2, ..., n_B\}).$$

If we view each base $\boldsymbol{\psi}_i$ as an attributed point with attributes $b_i = (\ell_i, \alpha_i, x_i, y_i, \tau_i, \sigma_i)$, then $\mathbf{B}$ is an attributed spatial point process.

In the image coding literature, the bases are assumed to be independently and identically distributed (iid), and the locations, scales and orientations are assumed to be uniformly distributed, so

$$p(\mathbf{B}) = p(n_B) \prod_{i=1}^{n_B} p(b_i), \tag{2}$$

$$p(b_i) = p(\alpha_i) \cdot \text{unif}(\ell_i) \cdot \text{unif}(x_i, y_i) \cdot \text{unif}(\tau_i) \cdot \text{unif}(\sigma_i). \tag{3}$$

It was well-known that responses of image filters on natural images have high kurtosis histograms. This means that most of the time the filters have nearly zero response (i.e. they are silent) and they are activated with large response occasionally. This leads to the sparse coding idea by Olshausen and Field [30].[2] For example, $p(\alpha)$ is chosen to be a Laplacian distribution, or a mixture of two Gaussians with σ_1 close to zero. For all $i = 1, ..., n_B$,

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

In fact, as long as $p(\alpha)$ has high kurtosis, the exact form of $p(\alpha)$ is not so crucial. For example, one can choose a mixture of two uniform distributions on a range $[-\sigma_j, \sigma_j]$, $j = 1, 2$

[1] The number of bases in Δ is often 100 times larger than the number of pixels in an image.

[2] Note that the filter responses are convolutions of a filter with image in a deterministic way, and are different from the coefficients of the bases.

respectively, with σ_1 close to zero,

$$p(\alpha_i) = \sum_{j=1}^{2} \omega_j \text{unif}[-\sigma_j, \sigma_j].$$

A slight confusion in the literature is that the sparse coding scheme assumes $n_B = |\Delta|$, i.e. all bases in the set are "activated". The prior $p(\alpha)$ is supposed to suppress most of the activations to zero. It is simple to prove that this is equivalent to assume $p(\alpha)$ to be a uniform distribution and put a penalty to the model complexity, for example $p(n_B) \propto e^{-\lambda n_B}$. So the sparse coding prior is essentially a model complexity term.

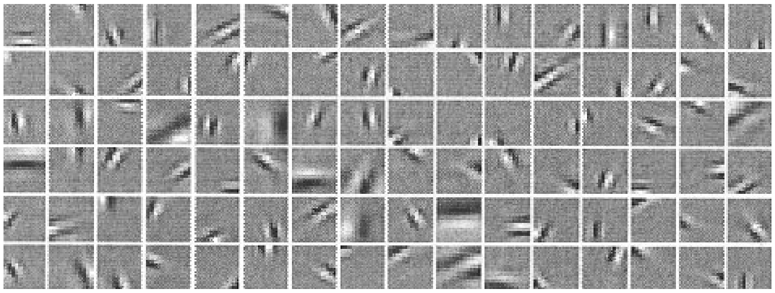


Figure 3: Some image bases learned with sparse coding by (Olshausen and Field, 1997) [30].

In the above image model, the base map $\mathbf{B}$ includes the hidden variables and the dictionary $\mathbf{\Psi}$ are parameters. For example, Olshausen and Field used $L_\psi = 144$ base functions, each being a 12×12 matrix. Following an EM-learning algorithm, they learned $\mathbf{\Psi}$ from a large number of natural image patches. Fig. 3 shows some of the 144 base functions. Such bases capture some image structures and are believed to bear resemblance to the responses of simple cells in V1 of primates.

In their experiments, the training images are chopped into 12×12 pixel patches, therefore they didn't really inferred the hidden variables for the transformation A_i. Thus the learned bases are not aligned at centers and are rather noisy.

2.2 K-mean clustering in feature space

There is also some effort of computing repeated image elements by Leung and Malik [25], who adopted a *discriminative method.* Most recently this method has been extended to textures with lighting variations and texture surface rendering[28, 13].

In discriminative method, the base functions are treated as "filters". By rotating and scaling these functions, one obtains a set of filters $\{F_1, F_2, ..., F_m\}$, which are convolved with an input image $\mathbf{I}$ at each location $(x, y) \in \Lambda$ on a lattice Λ. We denote all the filter responses as a set of $|\Lambda|$ points in a m-dimensional feature space,

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

In comparison to the hidden variable $\mathbf{B}$ in the previous generative model, $\mathbf{F}$ is deterministic transforms of the image $\mathbf{I}$.

If there are local structures occurring repeatedly in image $\mathbf{I}$, it is reasonable to believe that the vectors in set $\mathbf{F}$ must form clusters. A K-mean clustering algorithm is applied by Leung and Malik, and each cluster center was said to correspond to a "texton" [25]. The cluster center can be visualized by a pseudo-inverse which transfers a feature vector into an image icon. More precisely, let $F_c = (f_{c1}, ..., f_{cm})$ be a cluster center, then an image icon ϕ_c (say 15×15 pixels) is computed by a least square fit.

$$\phi_c = \arg\min \sum_{j=1}^{m} (F_j * \phi_c - f_{cj})^2, \quad c = 1, 2, ..., C. \tag{4}$$

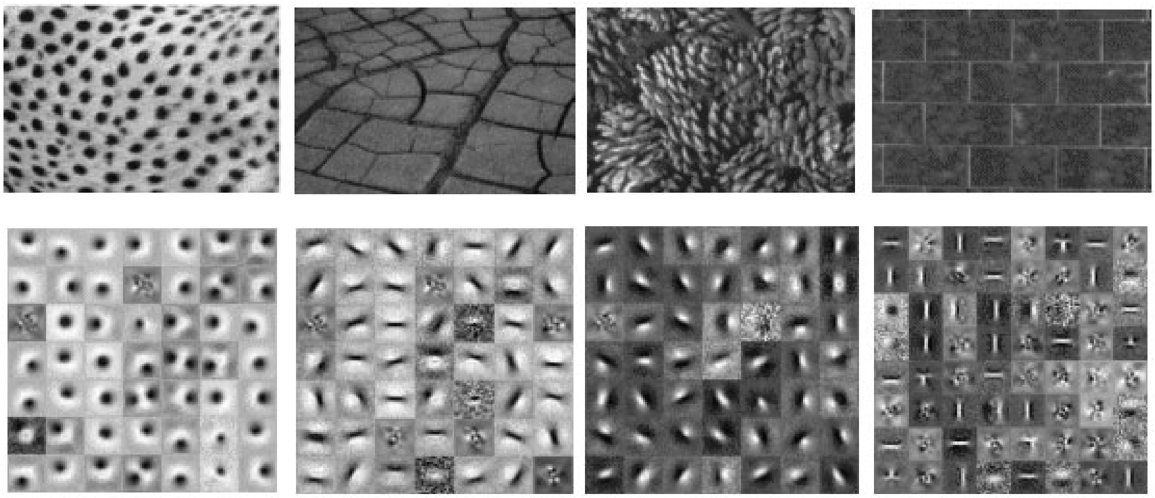


Figure 4: Upper row are four input texture images: cheetah skin, dry cracks, pine cone, and brick. The lower row displays $C = 49$ cluster centers arranged in a 7×7 mosaic for each image. The images are re-scaled and normalized for display.

We implement this work with some minor improvements and some results are shown in Fig. 4 for 49 clusters on four texture images. Clearly, the cluster centers capture some essential image structures, such as blobs for the cheetah skin pattern, bars for the crack and brick pattern, and edge contrasts for the pine cone.

We have two observations for the two methods presented above.

Firstly, the two methods have fundamental differences. In a generative model, an image $\mathbf{I}$ is "generated" in an explicit equation by the addition of a number of n_B bases where n_B is usually 100 times smaller than the number of pixels $|\Lambda|$. 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 feature description is often 100 times larger than the number of pixels! While both methods may use the same dictionary $\mathbf{\Psi}$, in the generative model, the base map $\mathbf{B}$ is a *random variable* subject to stochastic inference and therefore the computation of $\mathbf{B}$ can be influenced by other variables in a bottom-up/top-down fashion, i.e. lateral inhibitations in a neuroscience term. In the discriminative method, the *responses* of *filters* $\mathbf{F}$ are extracted from the image in a bottom-up fashion.

Secondly, the results in figures 3 and 4 manifest one obvious problem that the same image structure appears many 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 and there was no interaction or "explaining away" mechanism in this method.

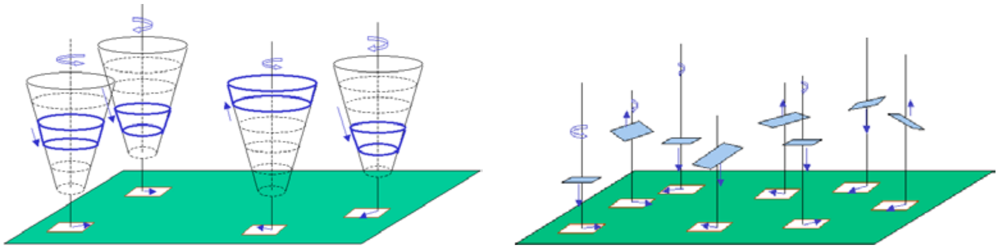
3 Learning transformed components

To remove redundancy among the learned image elements in previous section, one should explicitly infer the orthogonal transformation $A = (x, y, \tau, \sigma)$ as hidden (latent) variables and thus image components are merged if they are equivalent up to an orthogonal transform. This is a technique called transformed component analysis (TCA) by Frey and Jojic [14].

Suppose we extract from image $\mathbf{I}$ a set of N features or image patches, γ, each with an unknown transformation $A \in G_A$.

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

We call Γ the transformed components of $\mathbf{I}$. In the following, we present two experiments, filter TCA and patch TCA, as Fig. 5 illustrates.



a. Filter TCA b. Patch TCA

Figure 5: The transformed component analysis allows translation, rotation, and scaling of local image features or patches.

3.1 Transformed components in filter space

In the first case, we compute a feature vector at each pixel by a number of m filters as in the K-mean clustering method above. Typically we use Laplacian of Gaussian (LoG), Gabor sine (Gsin) and Gabor cosine (Gcos) at 7 scales and 8 orientations. Thus $m = 7 + 7 \times 8 + 7 \times 8 = 119$. These filters are arranged in a cone as shown in Fig. 5.a. We subsample the image lattice Λ by $4 - 8$ folds, and initialize the N cones at the subsampled lattice ($N = |\Lambda|/16$ or $|\Lambda|/64$).

Each feature γ_j chooses only 2 scales of the filter cone shown by the bold curves in Fig. 5.a. Thus a transformed component $\gamma(A)$ is an $2 + 2 \times 8 + 2 \times 8 = 34$ dimensional feature vector. The hidden variables (x_j, y_j, τ_j) correspond to shift and rotation of the filter cone, and σ_j corresponds to the selection of scales from the cone (jumping up and down the cone). The transforms are illustrated by the arrows in Fig. 5.

The movement of the filter cones are guided by the EM-algorithm to satisfy two constrains. 1). Each cone moves so that these filter vectors form tighter clusters by merging redundant clusters which are equivalent up to orthogonal transforms. 2). Collectively the cones should cover the entire image otherwise it yields a trivial solution. We form a likelihood probability $p(\mathbf{I}|\Gamma)$ by constraints from Γ.

The results of the computation include a set of transformed components Γ and C cluster centers. These cluster centers are again visualized by an image icon through pseudo-inverse.

Fig. 6 shows $C = 3$ center icons ϕ_1, ϕ_2, ϕ_3 for the cheetah and crack patterns. The image maps next to each icon are label maps where the black pixels are classified to this cluster. Clearly, the three elements are respectively: ϕ_1 — the center of the blobs (or cracks), ϕ_2

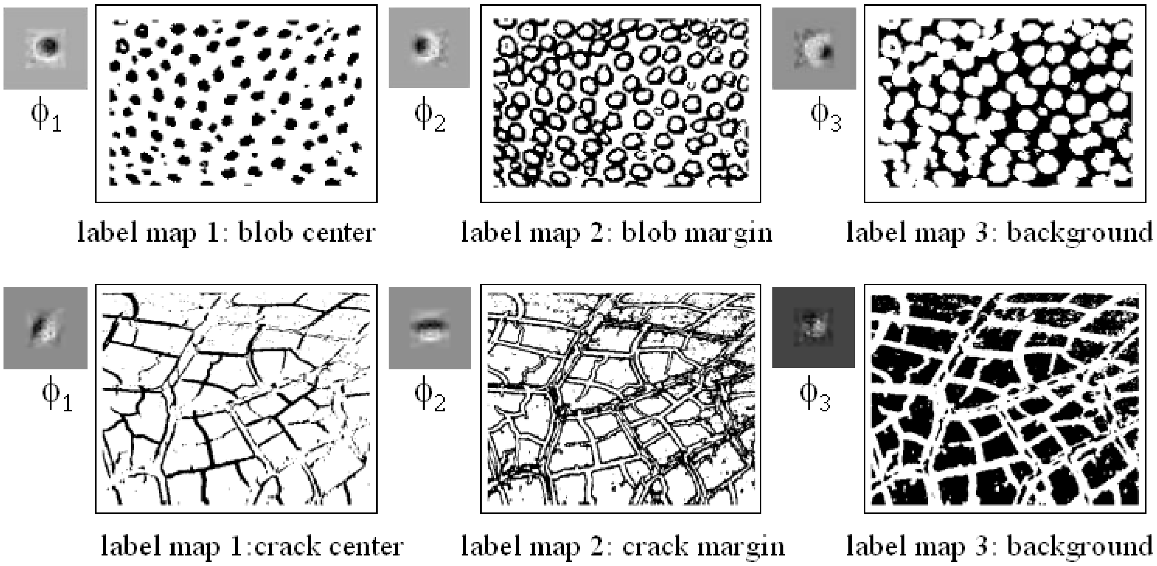


Figure 6: The learned basic elements ϕ_1, ϕ_2, ϕ_3 for the two patterns are shown by the small image icons. To the right are label maps associated with these icons.

— the rim of the blobs (or cracks), and ϕ_3 — the background. In the experiments, the translation of each filter cone is confined to be within a local area (say 5×5 pixels), so that the image lattice is covered by the effective areas of the cone.

3.2 Transformed components in the space of image patches

In a second experiment, we replace the feature representation by image windows of $11 \times 11 = 121$ pixels. These windows can be moved within a local area and can be rotated and scaled as Fig. 5.b illustrates. 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 and the patches collectively cover the observed image. The learned cluster centers $\phi_c, c = 1, ..., C$ are the repeating micro image structures.

Fig. 7 shows the $C = 2$ centers for the brick, cheetah, and pine cone patterns. The image maps next to each center element is a set of windows which are transformed versions of the elements. ϕ_1 corresponds to the blobs, bars, and contrasts for the three patterns respectively. ϕ_2 are for the backgrounds.

In summary, the results in figures 6 and 7 present a major improvement from those in figures 3 and 4, due to the inference of hidden variables for transforms. However, there are two main problems.

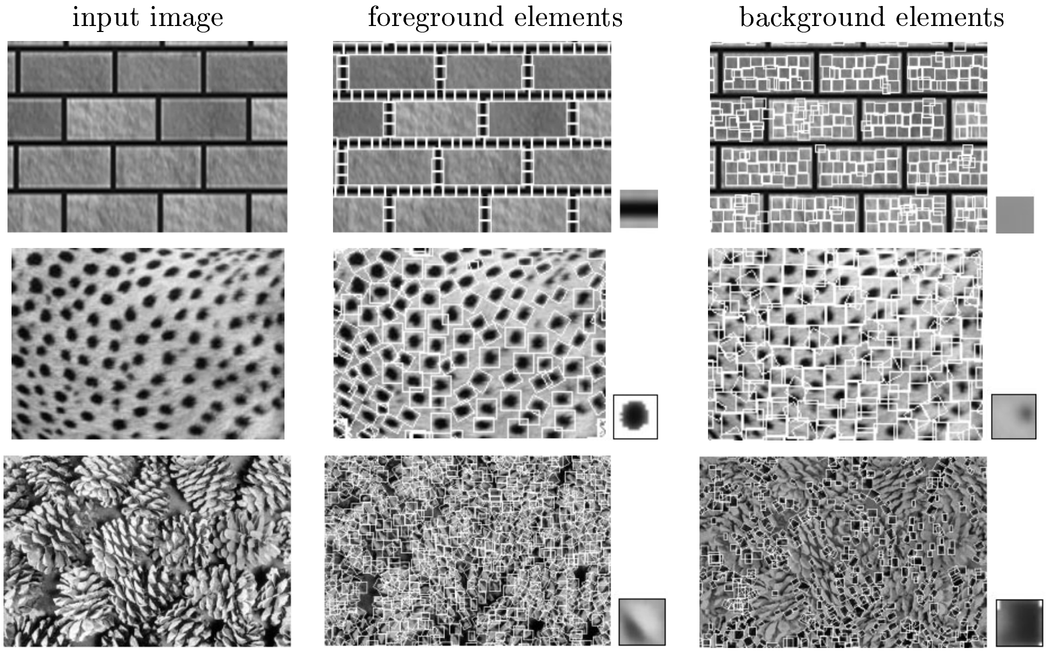


Figure 7: The learned image patches (cluster centers) ϕ_1, ϕ_2 for three texture patterns are shown by the small images. To the left are windows of transformed versions of the image patches associated with these icons.

1. The transformed components $\{\gamma_j, j = 1, ..., N\}$ only pose some constraints on image $\mathbf{I}$. An explicit generative image model is missing. As a result, the learned elements $\phi_\ell, \ell = 1, ..., C$ are contaminated by each other, due to overlapping between adjacent image windows or filter cones (see Fig. 5).

2. There is a lack of variability in the learned image elements. Taking 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 geometric structures which should be explored effectively.

To resolve these problems, we extend the TCA model to a three-level generative model. This extension allows us to explore the geometric, dynamic, and photometric structures of textons.

4 "Textons" – the basic geometric elements in images

4.1 A three-level generative model

Our comparison study leads us to a three-level generative model as shown in Fig. 2. In this model, an image **I** is generated by a base map **B** as in image coding, and the bases are selected from a dictionary $\boldsymbol{\Psi}$ with some orthogonal transforms. The base map **B** is, in turn, generated by a texton map **T**. The texton elements are selected from a texton dictionary $\boldsymbol{\Pi}$ with some orthogonal transforms. Each texton element in **T** consists of a few bases with a deformable geometric configuration. So we have,

$$\mathbf{T} \xrightarrow{\boldsymbol{\Pi}} \mathbf{B} \xrightarrow{\boldsymbol{\Psi}} \mathbf{I},$$

with

$$\boldsymbol{\Psi} = \{\boldsymbol{\psi}_\ell, \ell = 1, 2, ..., L_\psi\}, \quad \text{and} \quad \boldsymbol{\Pi} = \{\boldsymbol{\pi}_\ell; \ell = 1, 2, ..., \text{Ł}_\pi\}.$$

By analogy to the *waveform-phoneme-word* hierarchy in speech, the *pixel-base-texton* hierarchy presents an increasingly abstract visual description. This representation leads to dimension reduction and the texton elements account for spatial co-occurence of the image bases.

To clarify terminology, a base function $\psi \in \boldsymbol{\Psi}$ is like a mother wavelet and an image base b_i in the base map **B** is an instance under certain transforms of a base function. Similarly, a "texton" in a texton dictionary $\pi \in \boldsymbol{\Pi}$ is a deformable template, while a "texton element" is an instance in the texton map **T** which is a transformed and deformed version of a texton in $\boldsymbol{\Pi}$.

For natural images, it is reasonable to guess that the number of base functions is about $|\boldsymbol{\Psi}| = O(10)$, and the number of textons is in the order of $|\boldsymbol{\Pi}| = O(10^3)$ for various combinations. Intuitively, textons are meaningful objects viewed at distance (i.e. small scale), such as stars, birds, cheetah blobs, snowflakes, beans, etc. (see Fig. 2).

In this paper, we fix the base dictionary to three common base functions: Laplacian-of-Gaussian, Gabor cosine, Gabor sine, i.e.,

$$\boldsymbol{\Psi} = \{\ \psi_1, \psi_2, \psi_3\ \} = \{\ \mathrm{LoG}, \mathrm{Gcos}, \mathrm{Gsin}\ \}.$$

These base functions are not enough for patterns like hair or water, etc. But we fix them for simplicity and focus on the learning of texton dictionary $\boldsymbol{\Pi}$. This paper is also limited to learning textons for each individual texture pattern instead of generic natural images, therefore $|\mathbf{\Pi}|$ is a small number for each texture.

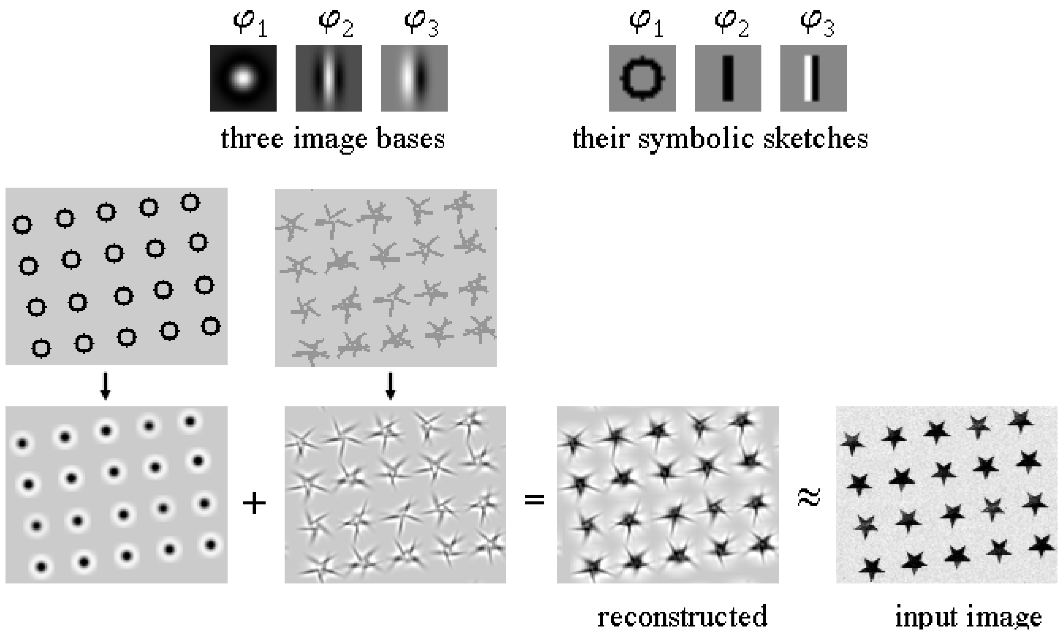

φ_1 φ_2 φ_3
three image bases
φ_1 φ_2 φ_3
their symbolic sketches
+
=
≈
reconstructed
input image

Figure 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.

Before we formulate the problem, we show an example of simple star pattern to illustrate the generative texton model. In Fig. 8, we first show the three base functions in $\mathbf{\Psi}$ (the first row) and their symbolic sketches. Then for an input image, a matching pursuit algorithm [29] is adopted to compute the base map $\mathbf{B}$ in a bottom-up fashion. This base map will be modified later by stochastic inference. It is generally observed that the base map $\mathbf{B}$ can be divided into two sub-layers. One sub-layer has relatively large ("heavy") coefficients α_i and captures some larger image structures. For the star pattern these are the LoG bases shown in the first column. We show both the symbolic sketch of these LoG bases (above) and the image generated by these bases (below). The heavy bases are usually surrounded by a number of "light" bases with relatively small coefficients α_i. We put these secondary bases in another sub-layer (see the second column of Fig. 8). When these image bases are superpositioned, they generate a reconstructed image (see the third column in Fig. 8). The residues of reconstruction are assumed to be Gaussian noise.

By an analogy to physics model, we call the heavy bases the "nucleus bases" as they have heavy weights like protons and neutrons, and the light bases the "electron bases". Fig. 9 displays an "atomic" model for the star texton. It is a LoG base surrounded by 5 electron bases.

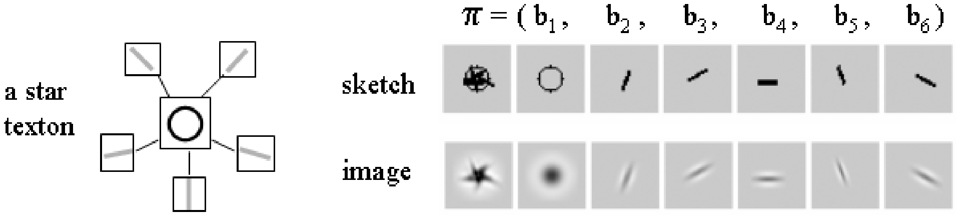


Figure 9: A texton for the star pattern π consists of a nucleus base (LoG) and five electron bases (Gsin). We show the sketches and the images for the texton and the six bases.

In the rest of this section, we show a statistical formulation and algorithm for inferring the base map $\mathbf{B}$ and the texton map $\mathbf{T}$ and learning the texton dictionary $\mathbf{\Pi}$.

4.2 Problem formulation

The three-level generative model is governed by a joint probability specified with parameters $\Theta = (\mathbf{\Psi}, \mathbf{\Pi}, \boldsymbol{\kappa})$.

$$p(\mathbf{I}, \mathbf{B}, \mathbf{T}; \Theta) = p(\mathbf{I}|\mathbf{B}; \mathbf{\Psi}) p(\mathbf{B}|\mathbf{T}; \mathbf{\Pi}) p(\mathbf{T}; \boldsymbol{\kappa}),$$

where $\mathbf{\Psi}$ and $\mathbf{\Pi}$ are dictionaries for two generating processes, and $p(\mathbf{T}; \boldsymbol{\kappa})$ is a descriptive (Gibbs) model for the spatial distribution of the textons as a stochastic attributed point process [16].

We rewrite the base map as

$$\mathbf{B} = (n_B, \{b_i = (\ell_i, \alpha_i, x_i, y_i, \tau_i, \sigma_i) : i = 1, 2, ..., n_B\}). \tag{5}$$

Because we assume Gaussian distribution $N(0, \sigma_o^2)$ for the reconstruction residues, we have

$$p(\mathbf{I}|\mathbf{B}; \mathbf{\Psi}) \propto \exp\{-\sum_{(u,v)\in\Lambda} (\mathbf{I}(u,v) - \sum_{i=1}^{n_B} \alpha_i \boldsymbol{\psi}_{\ell_i}(u, v; x_i, y_i, \tau_i, \sigma_i))^2 / 2\sigma_o^2\}. \tag{6}$$

The n_B bases in base map $\mathbf{B}$ are divided into $n_T + 1$ groups ($n_T < n_B$).

$$\{b_i = (\ell_i, \alpha_i, x_i, y_i, \tau_i, \sigma_i) : i = 1, 2, ..., n_B\} = \varpi_0 \cup \varpi_1 \cup \cdots \cup \varpi_{n_T}.$$

Bases in ϖ_0 are "free electrons" which do not belong to any texton, and are subject to the independent distribution $p(b_j)$ in equation (3). Bases in any other class form a texton element T_j, and the texton map is

$$\mathbf{T} = (n_T, \{T_j = (\ell_j, \alpha_j, x_j, y_j, \tau_j, \sigma_j, \delta_j) : j = 1, 2, ..., n_T\}).$$

Each texton element T_j is specified by its type ℓ_j, photometric contrast α_j, translation (x_j, y_j), rotation τ_j, scaling σ_j and deformation vector δ_j. A texton $\pi \in \mathbf{\Pi}$ consists of m image bases with a certain deformable configuration

$$\pi = (\ (\ell_1, \alpha_1, \tau_1, \sigma_1),\ (\ell_2, \alpha_2, \delta x_2, \delta y_2, \delta\tau_2, \delta\sigma_2), ...,\ \ (\ell_m, \alpha_m, \delta x_m, \delta y_m, \delta\tau_m, \delta\sigma_m)\ ).$$

The $(\delta x, \delta y, \delta\tau, \delta\sigma)$ are the relative positions, orientations and scales.

Therefore, we have

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

$p(\mathbf{T}; \boldsymbol{\kappa})$ is another distribution which accounts for the number of textons n_T and the spatial relationship among them. It can be a Gibbs model for attributed point process studied in [16]. For simplicity, we assume the textons are independent at this moment as a special Gibbs model.

By integrating out the hidden variables[3], we obtain a likelihood probability for any observable image $\mathbf{I}^{\text{obs}}$,

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

In $p(\mathbf{I}; \Theta)$ above, the parameters Θ (dictionaries, etc.) characterize the entire image ensemble, like the vocabulary for English or Chinese languages. In contrast, the hidden variables $\mathbf{B}, \mathbf{T}$ are associated with an individual image $\mathbf{I}$, and correspond to the parsing tree in language.

Our goal is to learn the parameters $\Theta = (\mathbf{\Psi}, \mathbf{\Pi}, \boldsymbol{\kappa})$ by maximum likelihood estimation, or equivalently minimizing a Kullback-Leibler divergence between a underlying probability of images $f(\mathbf{I})$ and $p(\mathbf{I}; \Theta)$.

$$\Theta^* = (\mathbf{\Psi}, \mathbf{\Pi}, \boldsymbol{\kappa})^* = \arg\min KL(f(\mathbf{I}) || p(\mathbf{I}; \Theta)) = \arg\max \sum_m \log p(\mathbf{I}_m^{\text{obs}}; \Theta) + \epsilon. \qquad (7)$$

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

[3] Some variables in $\mathbf{B}, \mathbf{T}$ are discrete, but we write the integration for notation clarity.

L_ψ and textons L_π. For clarity, we use only one large $\mathbf{I}^{\text{obs}}$ for training, because multiple images can be considered just patches of a larger image. For motion and lighting models in later sections, $\mathbf{I}^{\text{obs}}$ is extended to image sequence and image set with illumination variations.

4.3 Stochastic algorithm – Data-Driven Markov Chain Monte Carlo

Taking derivative of the log-likelihood with respect to Θ and set it to zero,

$$\frac{\partial \log p(\mathbf{I}^{\text{obs}}; \Theta)}{\partial \Theta} = 0,$$

we have by standard techniques,

$$\begin{aligned}
0 &= \frac{1}{p(\mathbf{I}^{\text{obs}}; \Theta)} \frac{\partial}{\partial \Theta} \int p(\mathbf{I}^{\text{obs}}, \mathbf{B}, \mathbf{T}; \Theta) d\mathbf{B} d\mathbf{T} \tag{8}\\
&= \frac{1}{p(\mathbf{I}^{\text{obs}}; \Theta)} \int \frac{\partial \log p(\mathbf{I}^{\text{obs}}, \mathbf{B}, \mathbf{T}; \Theta)}{\partial \Theta} p(\mathbf{I}^{\text{obs}}, \mathbf{B}, \mathbf{T}; \Theta) d\mathbf{B} d\mathbf{T} \tag{9}\\
&= \int \frac{\partial \log p(\mathbf{I}^{\text{obs}}, \mathbf{B}, \mathbf{T}; \Theta)}{\partial \Theta} \frac{p(\mathbf{I}^{\text{obs}}, \mathbf{B}, \mathbf{T}; \Theta)}{p(\mathbf{I}^{\text{obs}}; \Theta)} d\mathbf{B} d\mathbf{T} \tag{10}\\
&= \int [\frac{\partial \log p(\mathbf{I}^{\text{obs}} | \mathbf{B}; \boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}} + \frac{\partial \log p(\mathbf{B} | \mathbf{T}; \boldsymbol{\Pi})}{\partial \boldsymbol{\Pi}} + \frac{\partial \log p(\mathbf{T}; \boldsymbol{\kappa})}{\partial \boldsymbol{\kappa}}] p(\mathbf{B}, \mathbf{T} | \mathbf{I}^{\text{obs}}; \Theta)\ d\mathbf{B}\ d\mathbf{T} \tag{11}
\end{aligned}$$

Solving equation (11) needs stochastic algorithms which go a long way beyond the conventional EM-type algorithms [12]. Our objective is to find *globally optimal solutions* for $\boldsymbol{\Psi}^*, \boldsymbol{\Pi}^*$ while conventional EM is prone to local minima and furthermore the hidden variables $\mathbf{T}$ and $\mathbf{B}$ have changing dimensions. We propose to use the data driven Markov chain Monte Carlo (DDMCMC) algorithm [33].

The algorithm iterates two steps, like EM algorithm.

Step A: Design a Data-Driven Markov chain Monte Carlo (DDMCMC) sampler to draw samples of the latent variables from the posterior probability for a current Θ,

$$(\mathbf{B}_k^{\text{syn}}, \mathbf{T}_k^{\text{syn}}) \sim p(\mathbf{B}, \mathbf{T} | \mathbf{I}^{\text{obs}}; \Theta) \propto p(\mathbf{I}^{\text{obs}} | \mathbf{B}; \boldsymbol{\Psi}) p(\mathbf{B} | \mathbf{T}; \boldsymbol{\Pi}) p(\mathbf{T}; \boldsymbol{\kappa}), \quad k = 1, ..., K.$$

The DDMCMC sampling process includes designing two types of dynamics:

- Reversible jump dynamics for the death/birth of bases, the switching of base functions (i.e. types), the grouping and un-grouping of bases in texton elements, etc. The death and birth of bases realize a stochastic version of the matching pursuit method by Mallat et al. [29].

- Stochastic diffusions and Gibbs sampler for adjusting the positions, scales, and orientations of bases and texton elements.

The main idea of DDMCMC is to use data driven techniques, such as clustering, feature detection, matching pursuit, to compute heuristics expressed as *importance proposal probabilities* $q()$ in the reversible jumps. The convergence rate of MCMC critically depends on the importance proposal probabilities $q()$. Intuitively, the closer $q()$ approximates $p()$, the faster the convergence. The DDMCMC methods have been applied to image, range, and motion segmentation with very satisfactory speed in our experiments [33].

Step B: Calculate the integration in equation (11) (i.e. expectation with respect to $p(\mathbf{B}, \mathbf{T}|\mathbf{I}^{\rm obs}; \Theta)$) by importance sampling, therefore, the parameters in $\boldsymbol{\Psi}, \boldsymbol{\Pi}$ are learned by gradient descent. Let t be the time step and the $\lambda(t)$ is the step size at time t.

$$\begin{aligned}
\boldsymbol{\Psi}(t+1) &= (1-\lambda(t))\boldsymbol{\Psi}(t) + \lambda(t)\sum_{k=1}^{K} \frac{\partial \log p(\mathbf{I}^{\rm obs}|\mathbf{B}_k^{\rm syn}; \boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}} \\
\boldsymbol{\Pi}(t+1) &= (1-\lambda(t))\boldsymbol{\Pi}(t) + \lambda(t)\sum_{k=1}^{K} \frac{\partial \log p(\mathbf{B}_k^{\rm syn}|\mathbf{T}_k^{\rm syn}; \boldsymbol{\Pi})}{\partial \boldsymbol{\Pi}}
\end{aligned}$$

Thus we could select $K = 1$ if the step size $\lambda(t)$ is small enough. This algorithm converges to global maximum as the following theorem states [15].

Theorem 1. (Gu and Kong, 1998) *Under regularity conditions on the step size* $\lambda(t)$, *i.e.*

$$\sum_{t=1}^{\infty} \lambda(t) = \infty, \quad \sum_{t=1}^{\infty} \lambda(t)^2 < \infty,$$

and other mild conditions on the MCMC transition kernel and on a deterministic dynamics in the form of an ordinary differential equation derived from the algorithm, we have $(\Pi(t), \psi(t)) \to (\Pi^*, \psi^*)$ almost surely*, where* (Π^*, ψ^*) *is the globally optimal solution.*

Because $\boldsymbol{\Psi}$ and $\boldsymbol{\Pi}$ have both discrete and continuous variables, so the computational process consists of two types of dynamics.

- Reversible jump dynamics for creating/deleting base functions, adding and removing, splitting or merging bases in a texton $\pi \in \boldsymbol{\Pi}$.

- Stochastic diffusion dynamics and Gibbs sampler for adjusting the parameters in the base functions and the texton templates.

4.4 Experiments on learning textons from static texture images

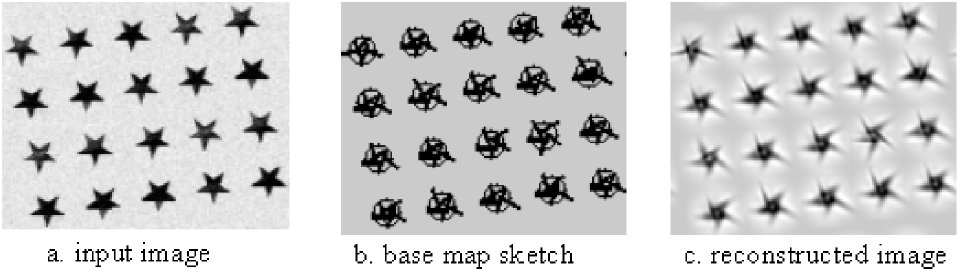


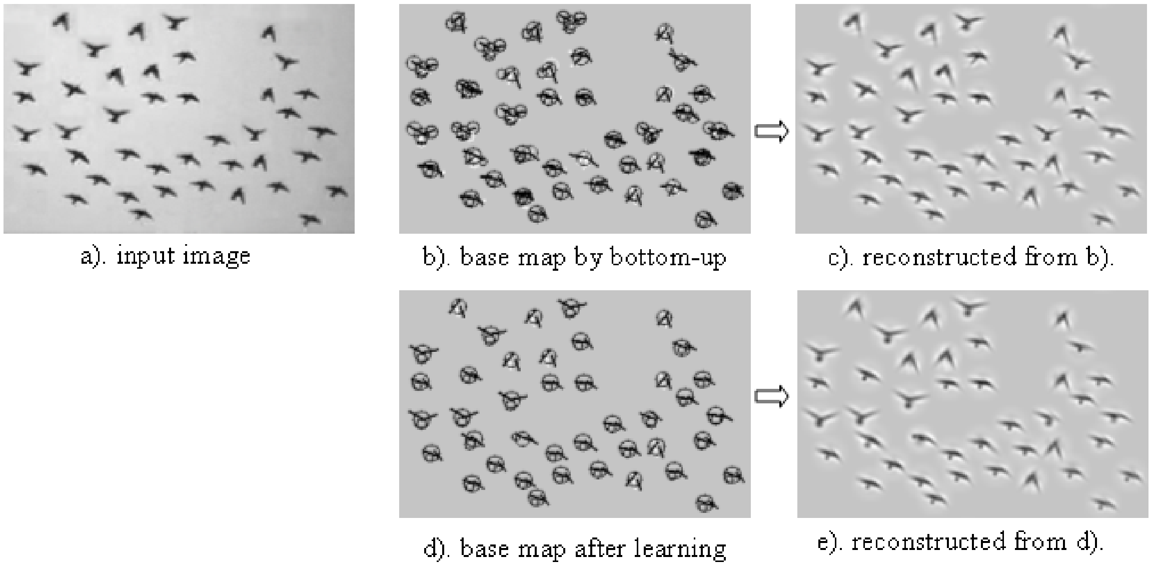
Figure 10: The base map b) of the star pattern becomes regular after the stochastic inference and learning, comparing to the base map in Fig. 9.

Now we present some experimental results. Given an input texture image, we fix $\mathbf{\Psi}$ to be the three bases, and run a matching pursuit algorithm for initialization. We first extract the bases with large coefficients and treat them as candidates for the nuclei of the texton elements. Then we add bases with small coefficients and group them to the nearest nucleus base. This step is a bottom-up. Then we run the stochastic learning algorithm which infers the base map and texton map, and find common spatial structures to form the texton dictionary $\mathbf{\Pi}$.

The first example is the star pattern. The base map from the bottom-up computation is shown in Fig. 8. Due to over-complete base representation and the noise in the image, a star can be reconstructed in various ways, and these are reflected in the variation of bases for each texton element. After learning, the inferred base map is shown in Fig. 10, and each star has nearly the same structure. The star texton is shown in Fig. 9. It consists of a LoG base as its "nucleus" and five Gsin bases as its "electrons". Our choice of base dictionary $\mathbf{\Psi}$ and additive model has obvious artifacts in reconstruction around sharp edges which are caused by occlusion.

In the second example, we show a bird image in Fig. 11. Again the bottom-up base map in Fig. 11.b is improved in Fig. 11.d after the learning process. Note that the reconstructed images in Fig. 11.c and Fig. 11.e are more or less the same. This bird image has three textons for various gestures of the birds. We show π_1, π_2, π_3 and their instances in Fig. 12.

Our third example is a cheetah skin pattern. Fig. 13.b displays the bottom-up base map, where the white circles are LoG bases that capture the strong lighting (brightness) as the image is not uniformly lighted. The inferred bases in Fig. 13.d are generated from



a). input image

b). base map by bottom-up

c). reconstructed from b).

d). base map after learning

e). reconstructed from d).

Figure 11: a). An input image of birds, b). the base map computed by matching pursuit in a bottom-up step, c). the reconstructed image from the base map in b), d). the base map after the learning process, e). the reconstructed image with the base map in d).

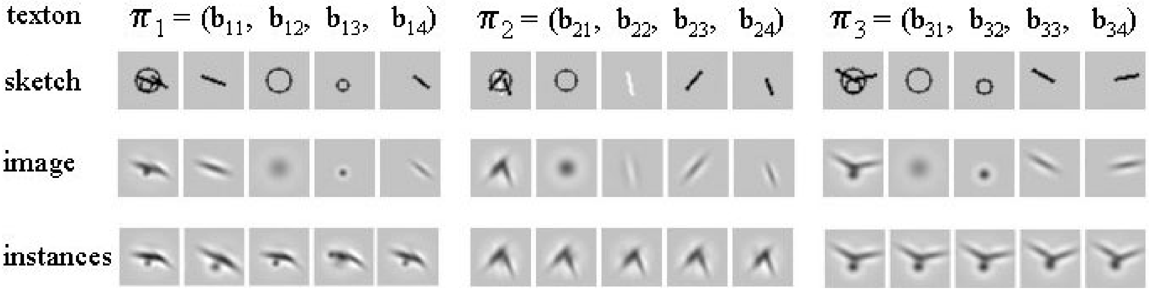
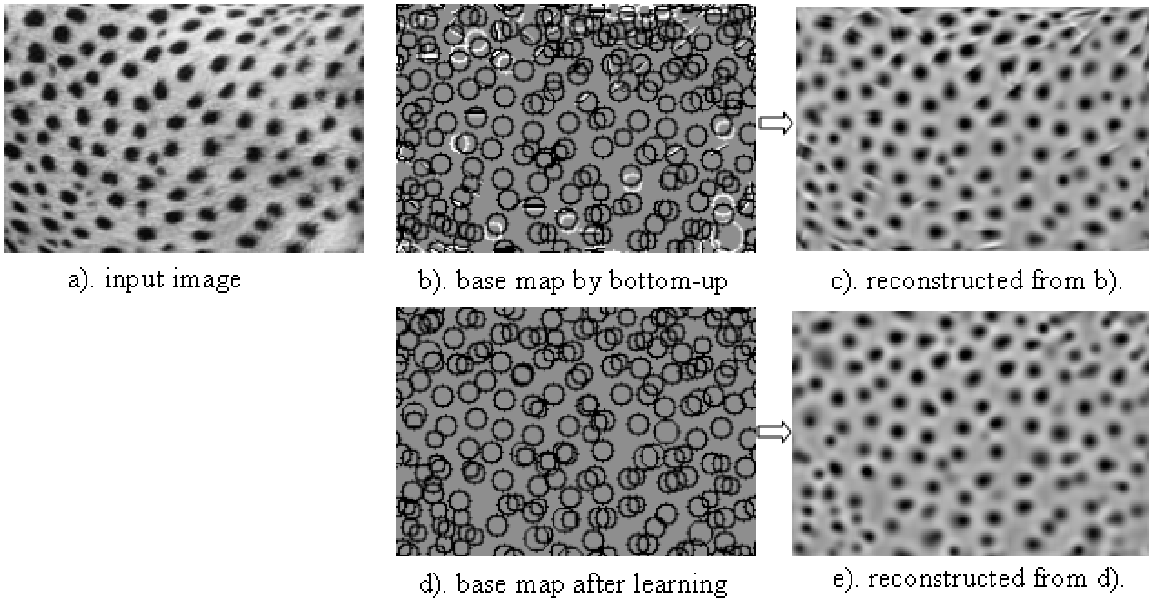

texton
$\pi_1 = (b_{11}, b_{12}, b_{13}, b_{14})$
$\pi_2 = (b_{21}, b_{22}, b_{23}, b_{24})$
$\pi_3 = (b_{31}, b_{32}, b_{33}, b_{34})$
sketch
image
instances

Figure 12: Three textons π_1, π_2, π_3 are learned for the bird image. Each texton has 4 bases. We show the sketches and images for these textons and bases. The last row shows five instances for each of the three textons.

the texton map with the white LoG base removed. Two textons are computed and shown in Fig. 14. π_1 has two LoG bases for the elongated blobs and π_2 has one LoG base for the round blobs. We show 9 instances for π_1 and 4 instances for π_2.

The fourth example is a heart pattern in Fig. 15. A heart texton consists of five bases: a LoG base for the "nucleus" which is added by two LoG plus two Gsin as "electrons". The fifth example is an image with four letters shown in Fig. 16. Only Gsin bases are selected as "strokes" for the letters. Then four textons are learned and shown in Fig. 16. These


a). input image

b). base map by bottom-up

c). reconstructed from b).

d). base map after learning

e). reconstructed from d).

Figure 13: a). An input image of cheetah skin, b). the base map computed by matching pursuit in a bottom-up step, c). the reconstructed image from the base map in b), d). the base map after the learning process, e). the reconstructed image with the base map in d).

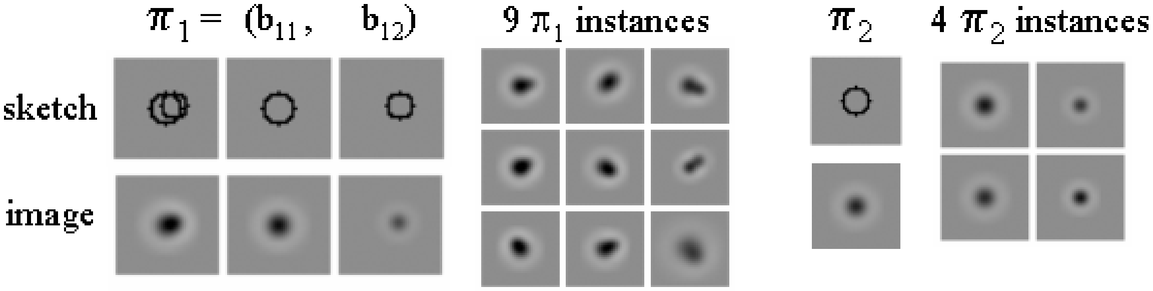


Figure 14: Two textons are learned for the cheetah skin pattern. π_1 has two LoG bases for the elongated blobs and π_2 has one LoG base for the round blobs. We show 9 instances for π_1 and 4 instances for π_2.

textons correspond to the four letters.

The last example is a leaf pattern as shown in Fig. 17. The learned texton has four bases. Ten texton element instances are shown in Fig. 17.d, which illustrates the variations of the leaves.

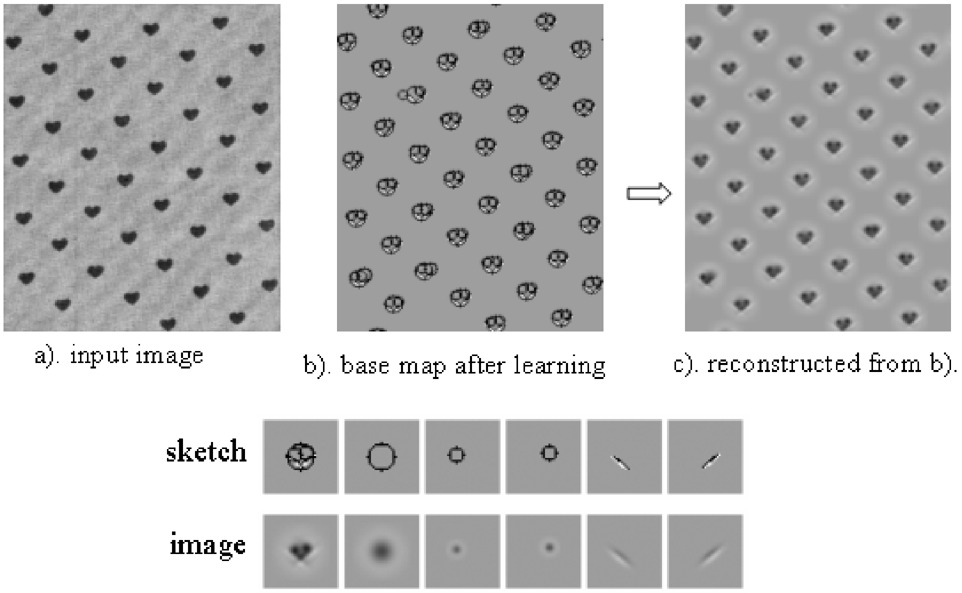

a). input image
b). base map after learning
c). reconstructed from b).

Figure 15: a). An input image of heart pattern, b). the base map after the learning process, c). the reconstructed image with the base map in b). A heart texton π consists of five bases.

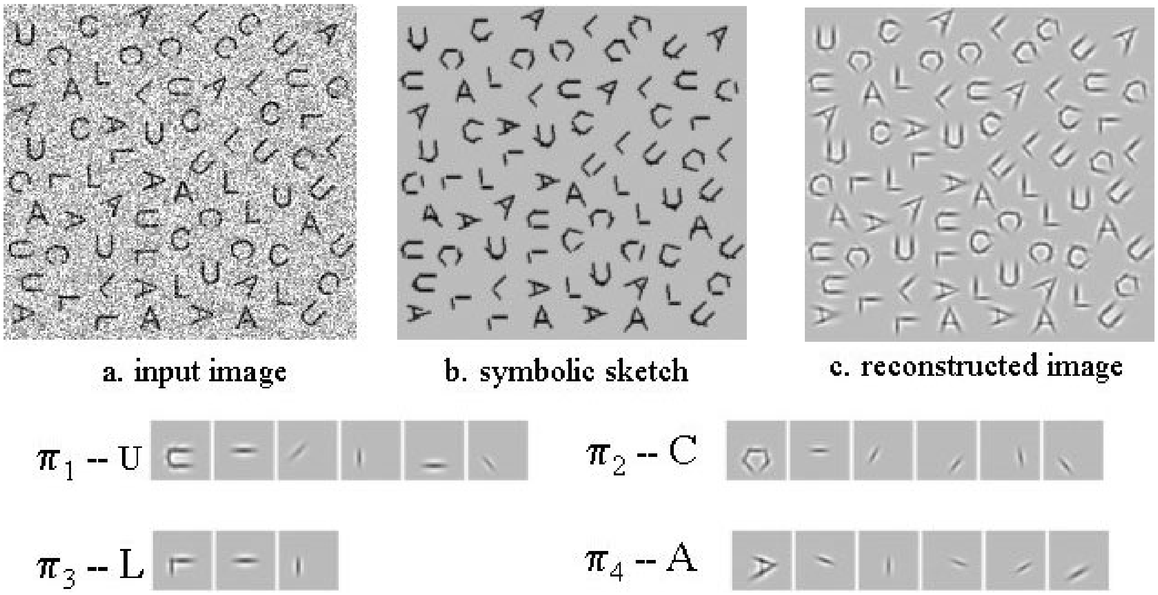
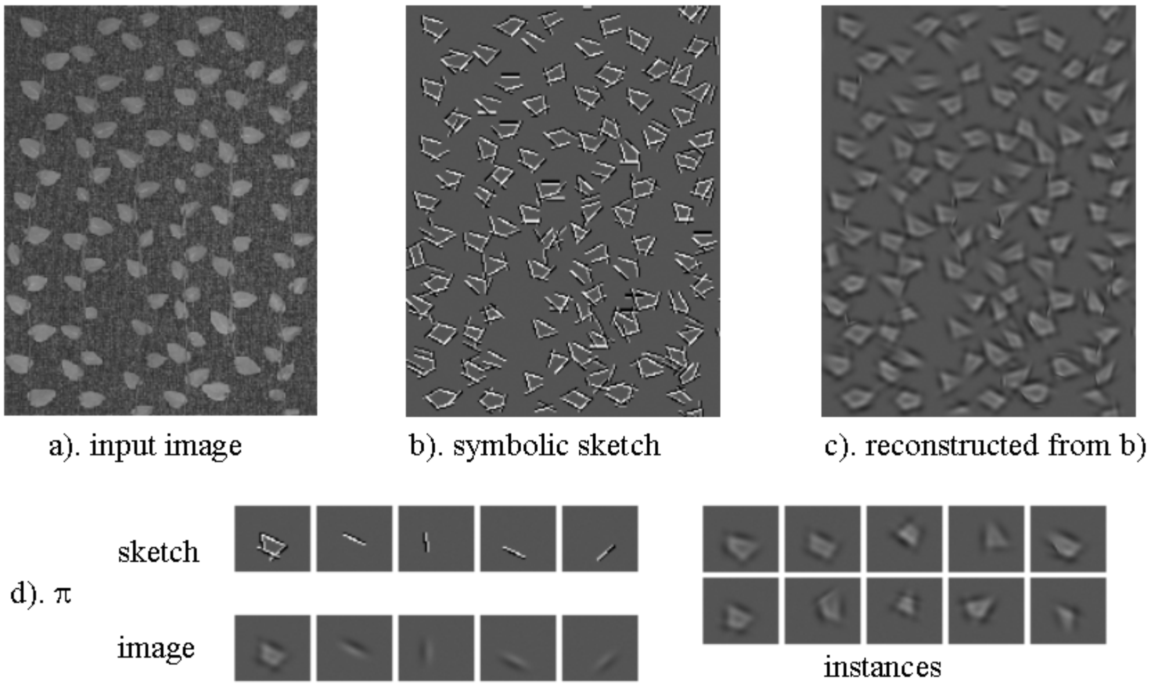

a. input image
b. symbolic sketch
c. reconstructed image

Figure 16: a). An input image with four letters, b). the base map after the learning process, c). the reconstructed image with the base map in b). Four textons $\pi_i, i = 1, 2, 3, 4$ are computed with Gcos as the strokes.

5 "Motons" – textons with motion dynamics

In this section, we augment the geometric structures of textons by studying their dynamic properties in video sequences. We call the moving textons as "motons". Examples include



a). input image b). symbolic sketch c). reconstructed from b)

d). π

Figure 17: a). An input image of leaves, b). the base map after the learning process, c). the reconstructed image with the base map in b), d). The sketch and image of the learned texton with four bases, and 10 texton element instances.

a falling snow flake, or a flying bird viewed in distance, etc. The modeling of motons originates from modeling textured motion patterns by Wang and Zhu[34, 35]. Generative texton models with motion dynamics is also used by Li, Wang and Shum in animating characters[26].

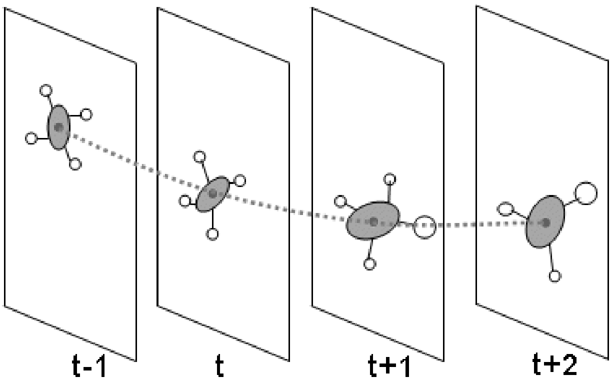


Figure 18: a). A linear "cable" Markov chain model for motons.

We start with a generative model for image sequence. Let $\mathbf{I}[0, \mathcal{L}]$ denote an image sequence with $\mathcal{L}+1$ frames, and $\mathbf{I}(t)$ a frame at time $t \in \{0,1,2,...,\mathcal{L}\}$. Each frame $\mathbf{I}(t)$ is generated by the three-level generative model in the previous section. Therefore a base map $\mathbf{B}(t)$ is computed from $\mathbf{I}(t)$, and the bases in $\mathbf{B}(t)$ are further grouped into a number of texton elements in a texton map $\mathbf{T}(t)$. Once these texton elements and bases are tracked over the image frames, we obtain a number of "moton elements". Fig. 18 shows a moton element. We call this a "cable" model where the trajectory of the nucleus base forms the core and the electronic base trajectories form coil of the cable through rotation. In practice, the core of a moton is relatively consistent through its life span, and the number of coil bases may change over time.

Each moton element has a life-span $[t^b, t^d] \subset [0, \mathcal{L}]$ and we call t^b, t^d the birth and death frames of a texton respectively. Thus a moton element ("cable") is denoted by

$$\mathcal{C} = \mathcal{C}[t^b, t^d] = (T(t^b), T(t^b+1), ..., T(t^d)), \tag{12}$$

where $T(t)$ is a 2D texton element at time t. Sometimes, the moton element may change its texton type over time, for example in the bird flying example that we will present shortly.

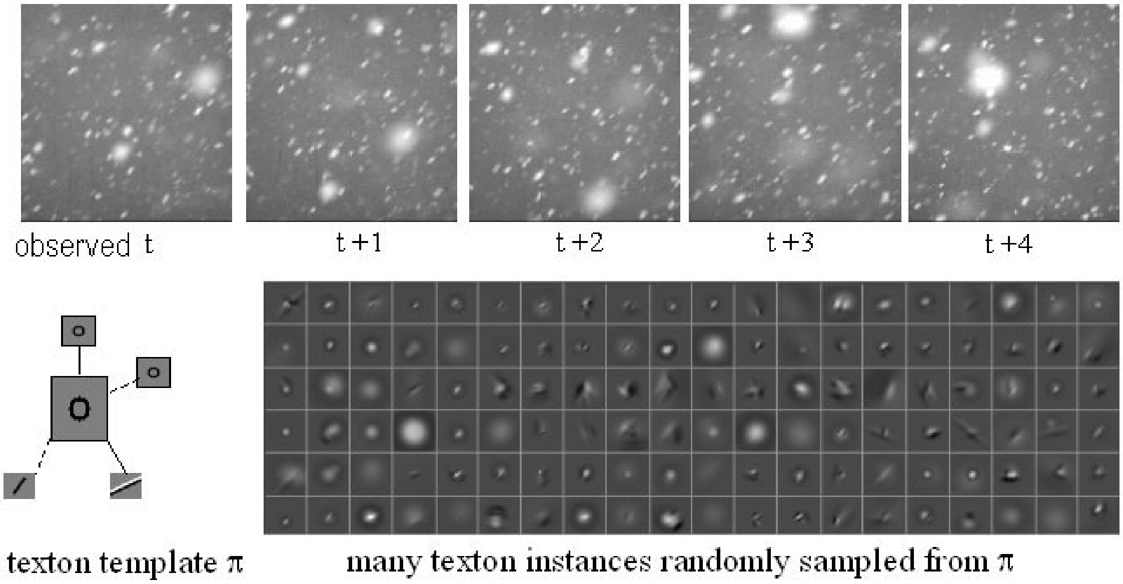


Figure 19: An example of snowing sequence. A texton π is learned from a snowing sequence and a variety of snowflake instances at various scales and orientations are randomly sampled from the template π.

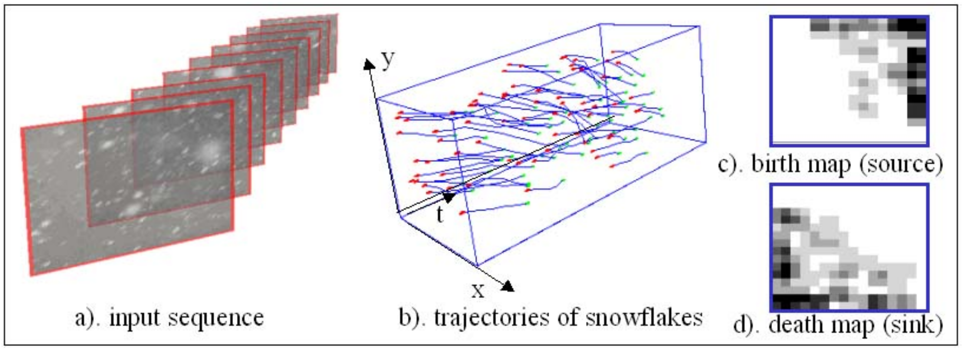


Figure 20: The computed trajectories of snow flakes and the source and sink maps.

The moton map $\mathbf{M}$ for $\mathbf{I}[0,\mathcal{L}]$ consists of a number of n_M moton elements (cables),

$$\mathbf{M} = (n_M, \{\mathcal{C}_i[t_i^b, t_i^d], i = 1, 2, ..., n_M\}).$$

Figures 20.b and 23.b show two examples of $\mathbf{M}$ for the snowing and bird flying sequences, where each trajectory is a moton element. These moton elements are instances of a moton dictionary,

$$\mathbf{\Xi} = \{\xi_\ell = (\eta_b^\ell, \eta_d^\ell, \eta_{mc}^\ell), \ell = 1, 2, ..., L_\Xi\}.$$

In the above notation, η_b^ℓ, η_d^ℓ are respectively the parameters specifying the birth, death probability of a certain moton ℓ and η_{mc}^ℓ is the parameter for the probability specifying the motion dynamics.

In summary, we have the following generative model for a video,

$$\mathbf{M} \xrightarrow{\mathbf{\Xi}} \mathbf{T}[0,\mathcal{L}] \xrightarrow{\mathbf{\Pi}} \mathbf{B}[0,\mathcal{L}] \xrightarrow{\boldsymbol{\psi}} \mathbf{I}[0,\mathcal{L}].$$

Thus we have a likelihood model for the video $\mathbf{I}[0,\mathcal{L}]$,

$$p(\mathbf{I}^{\text{obs}}[0,\mathcal{L}];\Theta) = \int [\prod_{t=0}^{\mathcal{L}} p(\mathbf{I}^{\text{obs}}(t)|\mathbf{B}(t);\ \mathbf{\Psi})p(\mathbf{B}(t)|\mathbf{T}(t);\mathbf{\Pi})]p(\mathbf{T}[0,\mathcal{L}]|\mathbf{M})p(\mathbf{M};\mathbf{\Xi})d\mathbf{M}d\mathbf{B}d\mathbf{T}$$

For simplicity, we assume $\mathbf{M}$ generates $\mathbf{T}[0,\mathcal{L}]$ as a deterministic function. In practice, it is more complicated due to base occlusions.

For clarity, we assume only one moton $L_\Xi = 1$ for a texture motion sequence, and the moton elements are independent of each other. Therefore the probability for the moton map is

$$p(\mathbf{M};\mathbf{\Xi}) \quad = \quad p(n_M)\prod_{i=1}^{n_M} p_B(T_i(t_i^b);\eta_b)p_D(T_i(t_i^d), t_i^d - t_i^b;\eta_d)p_{\text{mc}}(T_i(t_i^b+1)|T_i(t_i^b))$$

$$\prod_{t=t_i^b+2}^{t_i^d} p_{\mathrm{mc}}(T_i(t)|T_i(t-1), T_i(t-2); \eta_{mc}).$$

The initial and ending probabilities of the moton element are represented by the birth (source) and death (sink) probability maps $p_B(), p_D()$. Such probabilities account for where the moton elements are likely to come and leave in the image. For example, figures 20.c and 20.d are the birth and death maps for the snowing sequence. Dark intensity means high probability. Thus the algorithm automatically learns that the snowflakes enter the picture from the upper-right corner and leave at the bottom-left corner. Similarly, figures 23.c and 23.d show the sources and sinks for the birds. It is quite sparse due to the small number of birds in the observed sequence.

The conditional probabilities $p_{\mathrm{mc}}(T_{i,j}|T_{i,j-1}, T_{i,j-2}; \eta_{mc})$ determine the motion dynamics (Markov chain model) of each type of motons. We use the conventional second order autoregression (AR) model to fit the motion trajectories, which works fine for the snow and flying birds. For the flying bird cable, we have three textons as we discussed in the last section, and the birds switch among these textons over time. Thus we adopt a Markov chain model to switch its status using a 3×3 transition probability matrix. This simple model is shown in Fig. 22. The AR model parameters and the Markov chain transition probabilities are included in η_{mc}.

We demonstrate two examples and more examples are reported in [34]. The first example is a snowing sequence in Fig. 19. A texton π is learned and a variety of snowflake instances at various scales and orientations are randomly sampled from the template π. This demonstrates the variability of the snow texton. The moton elements, their birth/death maps are shown in Fig. 20.

Fig. 21 shows a portion of the bird flying sequence and its symbolic sketch. We further show the atomic model for each bird instance in Fig. 21.d. Fig. 22 shows the transition of bird texton states: wings up, gliding, and wings down. Fig. 23 displays the motons, the sources, and the sinks.

6 "Lightons" – textons under varying lighting conditions

The previous two sections discussed the geometric and dynamic properties of textons (motons). In the generative models, the photometric property of textons is represented by a

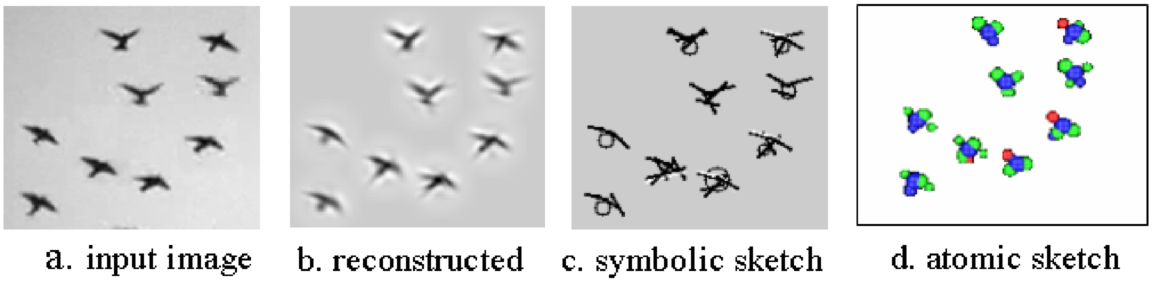


Figure 21: An example of bird sequence and the symbolic sketch and atomic models for the bird instances.

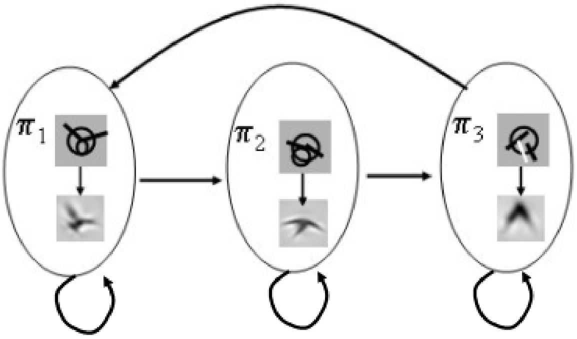


Figure 22: The Markov chain model for flying birds which switch among the three texton states.

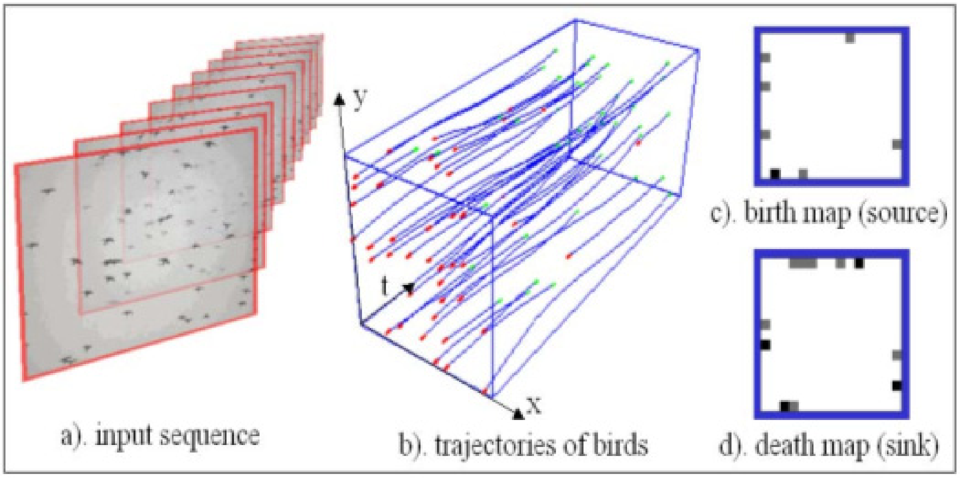


Figure 23: The computed trajectories of flying birds and the source and sink maps.

coefficient α for image bases which represents the intensity contrast. This is essentially a two-dimensional representation and is over-simplified for surfaces with 3D structures. For example, Koenderink et al. studied 3D pitted surfaces empirically. These surfaces have micro-image structures that change appearance with varying lighting conditions [23]. In this section, we slightly extend the generative model to a higher dimension representation and account for illumination variations.

Consider a rough 3D surface with unit surface normal $\vec{n}(x,y) = (n_1, n_2, n_3)(x,y)$ and surface albedo $\rho(x,y)$, and suppose the surface is illuminated by a point light source which is at relatively far distance and comes from direction $\vec{S} = (s_1, s_2, s_3)$, then under Lambertian reflectance model, one arrives at a classic image model,

$$\mathbf{I}(x,y) = \rho(x,y) < \vec{n}(x,y), \vec{S} >= s_1\mathbf{b}_1(x,y) + s_2\mathbf{b}_2(x,y) + s_3\mathbf{b}_3(x,y),$$

with $\mathbf{b}_i(x,y) = \rho(x,y)n_i(x,y), \ \ \forall i, x, y$. Rewrite images $\mathbf{I}$ and $\mathbf{b}_i, i = 1,2,3$ as long vectors in the $\Lambda|$-space, we have a simple additive model,

$$\mathbf{I} = s_1\mathbf{b}_1 + s_2\mathbf{b}_2 + s_3\mathbf{b}_3. \tag{13}$$

It was known that all images of the surface (from a fixed view point) under varying illuminations span a 3-dimensional space (or illumination cone) defined by the three images (axes of the cone) $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ [5]. Therefore the three axis images $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ characterize all the images of a 3D surface under the assumptions made above.

Given a set of images of a 3D surface, one can solve for the three axis images $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ by uncalibrated photometric stereo algorithms [20, 31] using singular value decomposition (SVD). The solution is up to a generalized Bas-relief (GBR) transform [31, 6].

We show two simple examples in this section to illustrate the ideas. In Fig. 24 we show six images of a 3D surface (many small spheres) under various illuminations. The computed three axis images $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are shown in Fig. 25 by photometric stereo. Similarly Fig. 26 shows six images of a 3D surface with beans, and the computed three axis images are shown in Fig. 27.

Following the additive image model (eq.(1)), we decompose the axis images as the sum of a number of image bases

$$\mathbf{b}_i = \sum_{j=1}^{n_{B_i}} \beta_{ij}\psi_{ij} + \mathbf{n}_i, \quad \psi_{ij} \in \Delta, i = 1,2,3.$$

So we have three base maps, one for each axis image,

$$\mathbf{B}_i = (n_{B_i}, \{\psi_{ij} : j = 1,2,...,n_{B_i}\}), \quad \text{for } i = 1,2,3.$$

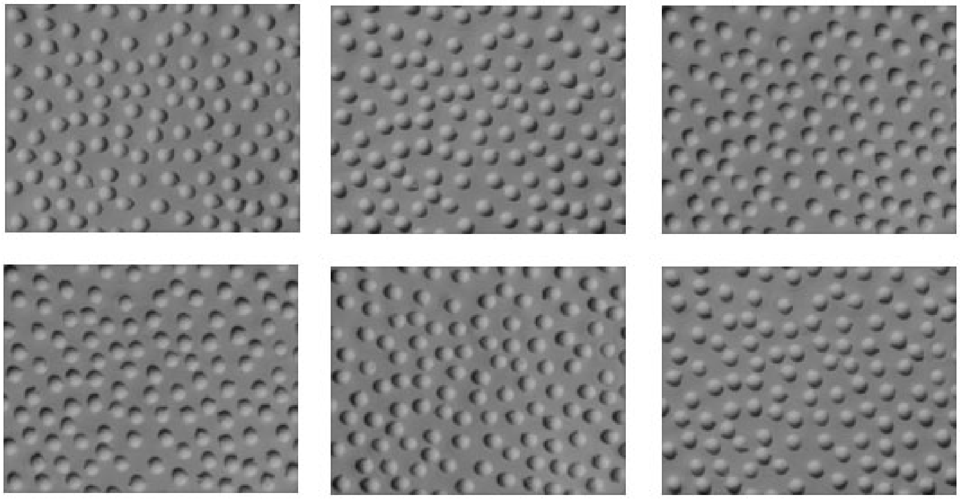


Figure 24: A set of input images for small spheres under varying illuminations. We use 20 images with lighting directions sampled evenly over the illumination hemisphere. Six of them are shown here.

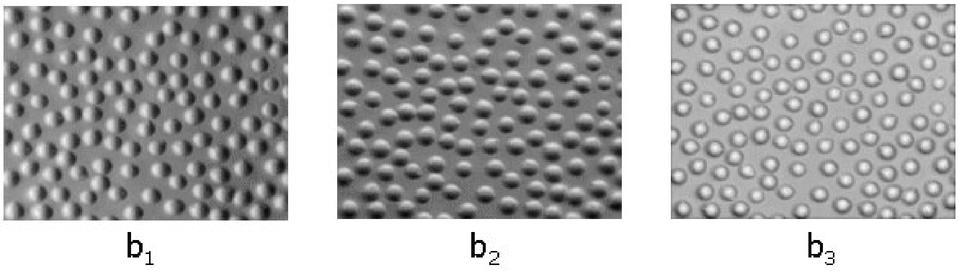


Figure 25: Three image bases computed by SVD from a set of input images under varying illuminations.

The bases in each base map are further grouped into a number of textons and form three texton maps $(\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3)$.

Because the bases $\psi_{ij}, i = 1, 2, 3; j = 1, ..., n_{B_i}$ are rendered by 3D surface structures, like the beans and spheres, the textons must be coupled across the three texton maps. Therefore we define a new element called "lighton". A lighton, denoted by ω, is a triplet of coupled 2D textons – one from each axis image, and $\mathbf{\Omega}$ is the set of lightons. Therefore

$$\mathbf{\Omega} = \{\omega_k : k = 1, 2, ..., L_\omega\}, \quad \text{with } \omega_k = (\pi_{k1}, \pi_{k2}, \pi_{k3}). \tag{14}$$

Suppose we denote the lighton map by $\mathbf{L}$, then we have the following generative model

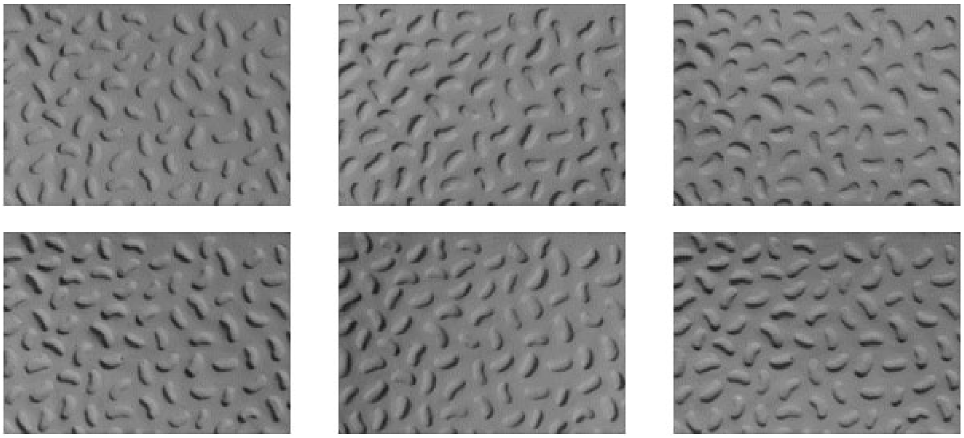


Figure 26: A set of input images for beans under varying illuminations. We use 20 images with lighting directions sampled evenly over the illumination hemisphere. Six of them are shown here.

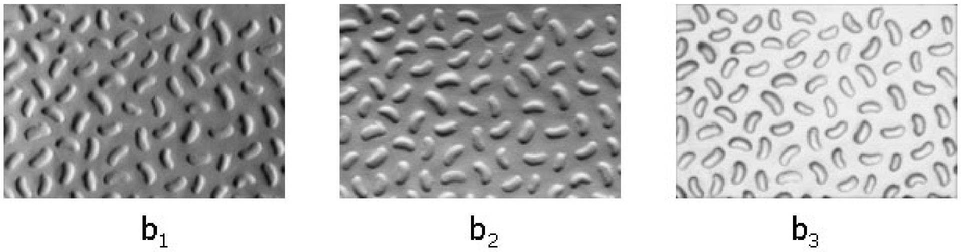


Figure 27: Three image bases computed by SVD from a set of input images under varying illuminations.

of images,

$$\mathbf{L} \xrightarrow{\boldsymbol{\Omega}} (\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3) \xrightarrow{\boldsymbol{\Pi}} (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3) \xrightarrow{\boldsymbol{\Psi}} (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3) \xrightarrow{(s_1, s_2, s_3)} \mathbf{I}. \tag{15}$$

In summary, the lighton map $\mathbf{L}$ generates three coupled texton maps $(\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3)$ using the lighton dictionary, which in turn generates three base maps $(\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)$ using the texton dictionary. The base maps generate the three axis images $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ using the base dictionary. Under a given lighting direction (s_1, s_2, s_3), the axis images create an image $\mathbf{I}$.

The learning of the lightons follows the same formulation and stochastic algorithm presented for learning textons and motons. The key difference from previous models is that the grouping of bases into textons and orthogonal transforms must be done by coupling the three axis images.

For clarity of presentation, we present some detailed analysis in an appendix and explain

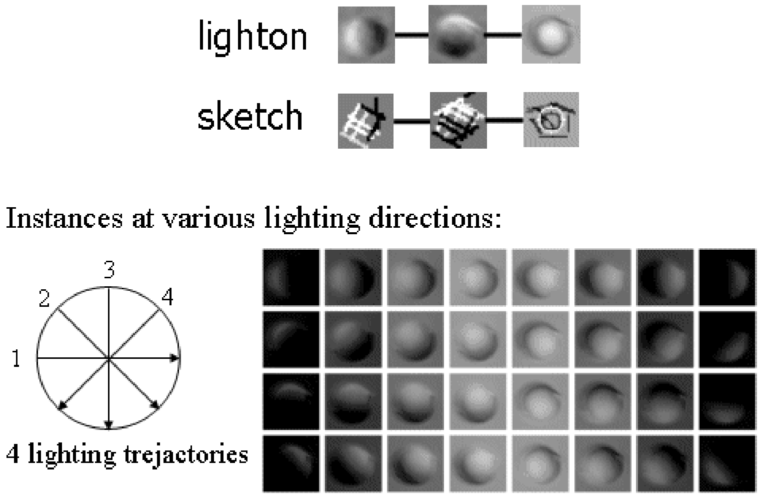


Figure 28: The lighton for the spheres is a texton triplet (i.e. three coupled textons). 32 instances are shown under various lighting directions.

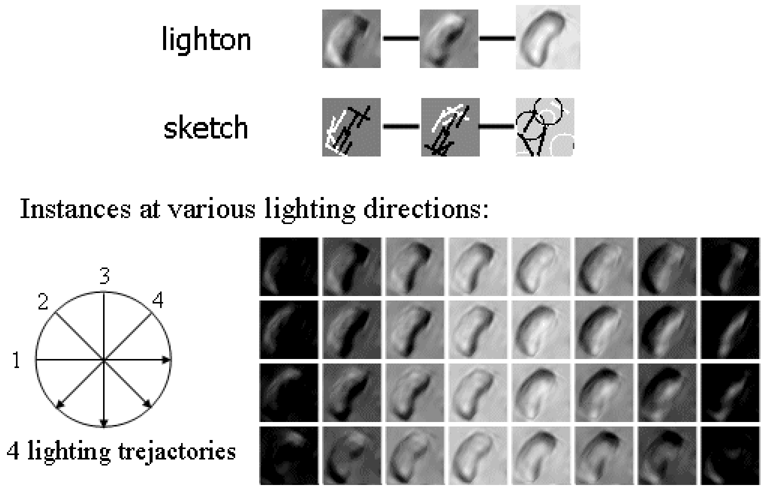


Figure 29: The lighton for the beans is a texton triplet (i.e. three coupled textons). 32 instances are shown under various lighting directions.

why the texton triplet (or lighton) corresponds to fundamental 3D surface structures at various locations, orientations, and scales. In the following, we show two examples of the lightons for the sphere and bean images in figures 28 and 29 respectively. For the simple case, the algorithm captures the sphere and bean as repeated elements in the 20 images. Each lighton (sphere or bean) is shown by the texton triplet (sketch) and three axis image patches in the top. Then we sample 32 lighting directions in the illumination sphere (8 angle for 4 directions), and we have 32 instances of the lightons.

7 Discussion

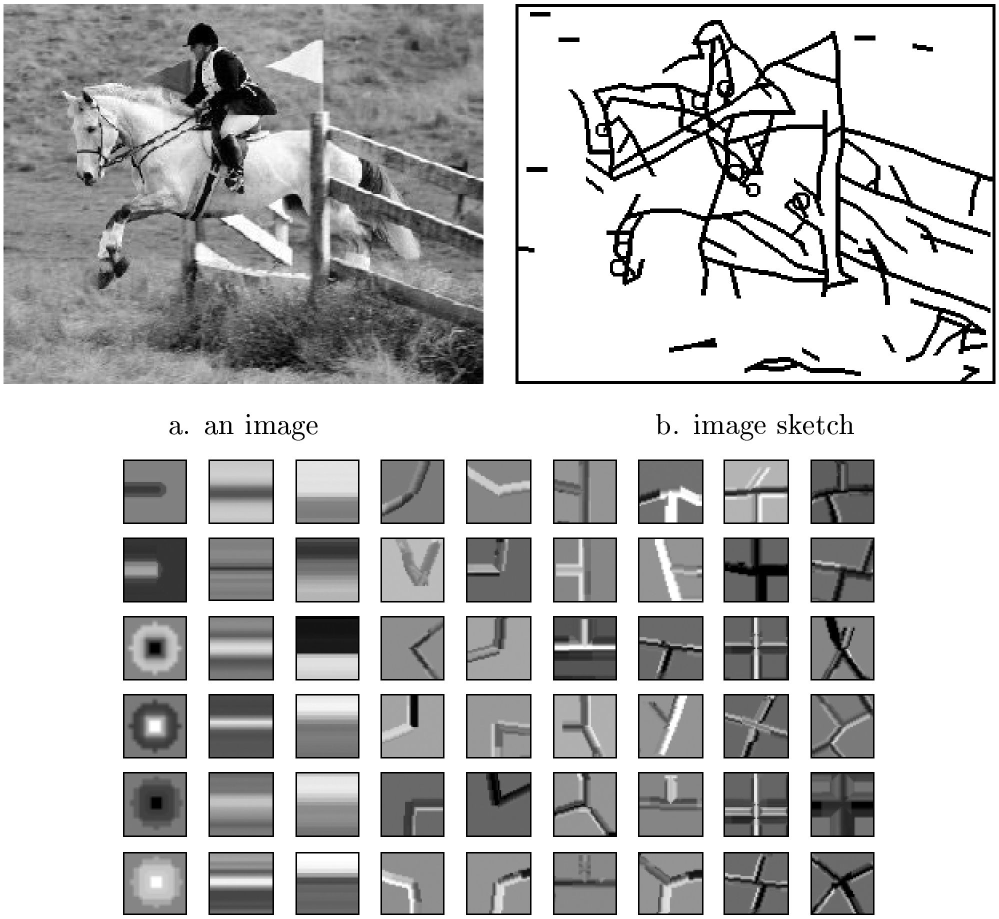
In this paper, we present a series of generative models and experiments for learning the fundamental image structures from textural images. The results are expressed in three dictionaries $\mathbf{\Pi}, \mathbf{\Xi}, \mathbf{\Omega}$ for the textons, motons, and lightons respectively which characterize the geometric, dynamic and photometric properties of the basic structures. These concepts are defined as parameters of the generative image models and thus can be learned by model fitting.

In future research, we plan to expand the work to learning the full texton/moton/lighton dictionaries from generic natural images and videos. The following problems are currently under investigation.

1). The base functions $\mathbf{\Psi}$ in this paper are limited to the LoG, Gcos, and Gsin functions, which must be extended to better account for images, such as, water, hair, shading[18]. Some bases must also be global, and form nearly periodic patterns, for example, the patterns discussed in Liu et al. [27].

2). The current model works on elements which are well separable. When severe occlusions present among elements, then it becomes more difficult to group the bases into textons. In such case, textons simply do not exist in free-form. By analogy to physics, the atoms (textons) exist in the form of large structure molecules and polymers by sharing some electrons with each other. This must involve spatial processes for the base map which is called the Gestalt fields[16].

We show one of most recent results for learning textons from natural images[17] where the connectivity of textons are included in the model. Figure 30.a is an input natural image. We divide the image lattice into two parts. One has sharp geometric contrast and is said to be "sketchable". The other part (rest of the lattice) is stochastic texture with no distinguishable structures and is said to be "non-sketchable". The graph structure is computed for the sketchable part and is shown in Figure 30.b. We call it the primal sketch of the image. Each vertex in this graph is associated with an image primitive in the image dictionary. We show a subset of the dictionary in Figure 30.c where the primitives are sorted according to their degrees of connectivities: blobs, endpoints, bars, t-junctions and cross junctions etc. With these primitives we can reconstruct the sketchable part of the image by generative models. Then the non-sketchable parts are modeled by texture models. More details are referred to [17]. This example shows that we can infer textons in a global



a. an image

b. image sketch

c. examples of textons learned from natural images.

Figure 30: Learning 2D textons from natural images by primal sketch. After (Guo, Zhu and Wu, 2003)

context from natural images.

3). In mathematical terms, the textons/motons/lightons are lower dimensional manifolds embedded in extremely high dimensional image (video) spaces. These manifolds are controlled by the parameters in the three dictionaries $\mathbf{\Pi}, \mathbf{\Xi}, \mathbf{\Omega}$. In our current study, the photometric structures are studied separately from the geometric and dynamic properties. But in the real world scene, the three aspects must be studied jointly, for example, the flashing light reflected from a swimming pool, etc.

We are studying these problems in ongoing projects.

Acknowledgments

This work is supported partially by NSF grants IIS 02-44763 and IIS 02-22967. We'd like to thank Yingnian Wu for intensive discussion during the development of the work.

Appendix: Deriving lightons from 3D surface elements

In this appendix, we briefly show physical meaning of the lightons and the relationship between the lighton representation (i.e. texton triplet) and the three dimensional surface elements.

Suppose a 3D textured surface consists of a number of n_L 3D elements, such as the spheres and beans in figures 24 and 26. Many other examples are shown in [23]. The visible surface of the 3D element at unit scale is represented by a height function $h_o(u, v)$ at a 2D domain D_0, and it has surface albedo $\rho_o(u, v)$. So we have a basic representation of the element by

$$(\rho_o(u, v), h_o(u, v)), \quad \forall (u, v) \in D_o.$$

It is well known that the height $h_o(u, v)$ can also be represented by the unit surface normal maps

$$\vec{n}_o(u, v) = (n_{o1}(u, v), n_{o2}(u, v), n_{o3}(u, v)) = \frac{(\partial h_o/\partial u, \partial h_o/\partial v, -1)}{\sqrt{1 + (\partial h_o/\partial u)^2 + (\partial h_o/\partial v)^2}}.$$

Furthermore, we can present the element by a triplet of images

$$(\mathbf{b}_{o1}, \mathbf{b}_{o2}, \mathbf{b}_{o3})(u, v) = \rho_o(u, v) \cdot (n_{o1}, n_{o2}, n_{o3})(u, v).$$

This is the physical model of the lightons Ω.

Now suppose the 3D texture surface is generated by embedding the 3D elements in a 2D flat plane β at various locations, scales (sizes), and orientations. It is straight-forward to show that the three axis images (that span the illumination cone) can be decomposed into the lightons with orthogonal transforms.

Each surface element $\mathbf{L}_j, j = 1, 2, ..., n_L$ is a translated, rotated and scaled version and has domain D_j at the texture plane. We denote the translation by (x_j, y_j), and the rotation (θ_j) and scaling (σ_j) by a 2×2 matrix

$$A_j = \frac{1}{\sigma_j} \left(\begin{array}{cc} cos(\theta_j), & sin(\theta_j) \\ -sin(\theta_j), & \cos(\theta_j) \end{array} \right).$$

We assume the plane β has constant height μ and constant albedo ν, thus the 3D texture surface has height

$$h(x,y) = \begin{cases} \sigma_j h_o(A_j((x-x_j),(y-y_j))'), & \text{if } (x,y) \in D_j, j = 1,2,...,n_L. \\ \mu, & \text{else.} \end{cases}$$

with albedo map

$$\rho(x,y) = \begin{cases} \rho_o(A_j((x-x_j),(y-y_j))'), & \text{if } (x,y) \in D_j, j = 1,2,...,n_L. \\ \nu, & \text{else.} \end{cases}$$

The three axis images are

$$(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)(x,y) = \rho(x,y) \frac{(\partial h(x,y)/\partial x, \partial h(x,y)/\partial y, -1)}{\sqrt{1 + (\partial h/\partial x)^2 + (\partial h/\partial y)^2}}.$$

Then it is straight-forward to show that at each domain

$$\begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{pmatrix}_{(x,y)} = \begin{pmatrix} cos(\theta_j), & -sin(\theta_j), & 0 \\ sin(\theta_j), & cos(\theta_j), & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{b}_{o1} \\ \mathbf{b}_{o2} \\ \mathbf{b}_{o3} \end{pmatrix}_{((x-x_j, y-y_j)A'_j)} \quad \text{for } (x,y) \in D_j.$$

This equation shows that the bases and textons in $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are coupled, and it is used in clustering the lightons.

This model will have problem when the 3D elements occlude each other.

References

[1] Adelson, E.H. and Pentland, A.P. 1996. "The perception of shading and reflectance", in *Perception as Bayesian Inference*, D. Knill and W. Richards (eds), pp409-423, New York, Cambridge Univ. Press.

[2] Atick, J.J. and Redlich, A.N. 1992. "What does the retina know about natural scenes?", *Neural Computation*, 4:196-210.

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

[4] Bell, A. J. and Sejnowski, T.J. 1995. "An information maximization approach to blind separation and blind deconvolution", *Neural Computation*, 7(6): 1129-1159.

[5] Belhumeur, P.N. and Kriegman, D. 1998. "What is the set of images of an object under all possible illumination conditions?", *Int'l J. Computer Vision*, 28(3).

[6] Belhumeur, P.N., Kriegman, D. and Yuille, A.L. 1999. "The Bas-Relief Ambiguity", IJCV, 35(1).

[7] Bergeaud, F. and Mallat, S. 1996. "Matching pursuit: adaptive representation of images and sounds." *Comp. Appl. Math.*, 15:97-109.

[8] Coifman, R.R. and Wickerhauser, M.V. 1992. "Entropy based algorithms for best basis selection." *IEEE Trans. on Information Theory.*, 38:713-718.

[9] Donoho, D.L., Vetterli, M., DeVore, R.A. and Daubechie, I. 1998. "Data compression and harmonic analysis", *IEEE Trans. Information Theory.* 6:2435-2476.

[10] Dana, K. and Nayar, S. 1999. "3D textured surface modelling", *Proc. of Workshop on Integration of Appearance and Geometric Methods in Object Recognition*, pp46-56.

[11] Daugman, J. 1985. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical Society of America*, 2(7):1160-1169.

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

[13] Dong, J. and Chantler, M.J. 2002, "Capture and synthesis of 3D surface texture", *Proceedings of 2nd Texture workshop.*

[14] Frey, B. and Jojic, N. 1999. "Transformed component analysis: joint estimation of spatial transforms and image components", *Proc. of Int'l Conf. on Comp. Vis.*, Corfu, Greece, 1999.

[15] Gu, M.G. and Kong, F.H. 1998. "A stochastic approximation algorithm with Markov chain Monte-Carlo method for incomplete data estimation problems", *Proceedings of the National Academy of Sciences*, U.S.A. 95, pp. 7270-7274.

[16] Guo, C.E., Zhu, S.C. and Wu, Y.N. 2001. "Visual learning by integrating descriptive and generative methods", *Proc. of Int'l Conf. on Computer Vision*, Vancouver, CA, July, 2001 (To Appear in IJCV).

[17] Guo, C.E., Zhu, S.C. and Wu, Y.N. 2001, "A mathematical theory for primal sketch and sketability", *Proc. of Int'l Conf. on Computer Vision*, Nice France, 2003.

[18] Haddon, J. and Forsyth, D.A. 1998, "Shading primitives: finding folds and shadow grooves", *Proc. 6th Int'l Conf. on Computer Vision*, Bambay, India.

[19] Hubel, D.H. and Wiesel, T.N. 1962. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex", *J. Physiology*, 160:106-154.

[20] Jacobs, D. 1997. "Linear fiting with missing data:applications to structure from motion and characterizing intensity images", *Proc. of CVPR*, 1997.

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

[22] Karni, A. and Sagi, D. 1991. "Where practice makes perfect in texture discrimination – evidence for primary visual cortex plasticity", *Proc. Nat. Acad. Sci. US*, 88:4966-4970.

[23] Koenderink, J.J., van Doorn, A.J., Dana, K.J. and Nayar, S. 1999. "Bidirectional Reflection Distribution Function of thoroughly pitted surfaces". IJCV, 31(2/3).

[24] Lee, A.B. Huang, J.G. and Mumford, D.B. 2000. "Random collage model for natural images", *Int'l J. of Computer Vision*, oct. 2000.

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

[26] Li, Y. Wang, T.S. and Shum, H.Y. 2002 "Motion textures: a two-level statistical model for character motion synthesis", *Proceedings of Siggraph.*

[27] Liu, Y.X. and Collins, R.T. 2000. "A computational model for repeated pattern perception using Frieze and wallpaper groups", *Proc. of Comp. Vis. and Patt. Recog.*, Hilton Head, SC. June, 2000.

[28] Liu, X.G., Yu, Y.Z. and Shum, H.Y. 2001. “Synthesizing bi-directional texture functions for real world surfaces”, *Proceeding of Siggraph.*

[29] Mallat, S.G. 1989. “A theory for multiresolution signal decomposition: the wavelet representation”, *IEEE Trans. on PAMI*, 11(7):674-693.

[30] Olshausen, B.A. and Field, D.J. 1997. “Sparse coding with an over-complete basis set: A strategy employed by V1?”, *Vision Research*, 37:3311-3325.

[31] Shashua, A. 1992. *Geometry and Photometry in 3D Visual Recognition*, Ph.D Thesis, MIT.

[32] Simoncelli, E.P., Freeman, W.T., Adelson, E.H. and Heeger, D.J. 1992. “Shiftable multiscale transforms”, *IEEE Trans. on Info. Theory*, 38(2):587-607.

[33] Tu, Z.W. and Zhu, S.C. 2002. “Image segmentation by Data-driven Markov chain Monte Carlo”, *IEEE Trans. on PAMI*, 24(5):657-673.

[34] Wang, Y.Z. and Zhu, S.C. 2002. “A generative model for textured motion: analysis and synthesis”, *Proc. of European Conf. on Computer Vision*, Copenhagen, Denmark, June 2002.

[35] Wang, Y.Z. and Zhu, S.C. 2002. “A generative model for textured motion: analysis and synthesis”, *Proc. of European Conf. on Computer Vision*, Copenhagen, Denmark, June 2002.

[36] Zhu, S.C., Guo, C.E., Wu, Y.N., and Wang, Y.Z. 2002. “What are textons”, *Proc. of European Conf. on Computer Vision*, Copenhagen, Denmark, June 2002.

[37] Zhu, S.C. 2002. “Statistical modeling and conceptualization of visual patterns”, *Preprint of UCLA Statistics Department.*