Statistical Modeling of Texture Sketch

Ying Nian Wu[1], Song Chun Zhu[2], and Cheng-en Guo[2]

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

Abstract. Recent results on sparse coding and independent component analysis suggest that human vision first represents a visual image by a linear superposition of a relatively small number of localized, elongate, oriented image bases. With this representation, the sketch of an image consists of the locations, orientations, and elongations of the image bases, and the sketch can be visually illustrated by depicting each image base by a linelet of the same length and orientation. Built on the insight of sparse and independent component analysis, we propose a two-level generative model for textures. At the bottom-level, the texture image is represented by a linear superposition of image bases. At the top-level, a Markov model is assumed for the placement of the image bases or the sketch, and the model is characterized by a set of simple geometrical feature statistics.

Keywords: Independent component analysis; Matching pursuit; Minimax entropy learning; Sparse coding; Texture modeling.

1 Introduction

As argued by Mumford (1996) and many other researchers, the problem of vision can be posted in the framework of statistical modeling and inferential computing. That is, top-down generative models can be constructed to represent a visual system's knowledge in the form of probability distributions of the observed images as well as variables that describe the visual world, then visual learning and perception become a statistical inference (and model selection) problem that can be solved in principle by computing the likelihood or posterior distribution. To guide reliable inference, the generative models should be realistic, and this can be checked by visually examining random samples generated by the models.

Recently, there has been some progress on modeling textures. Most of the recent models involve linear filters for extracting local image features, and the texture patterns are characterized by statistics of local features. In particular, inspired by the work of Heeger and Bergen (1996), Zhu, Wu, and Mumford (1997) and Wu, Zhu, and Liu (2001) developed a self-consistent statistical theory for texture modeling; borrowing results from statistical mechanics, they showed that a class of Markov random field models is a natural choice under the assumption

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2352, pp. 240–254, 2002.

that texture impressions are decided by histograms of filter responses. Meanwhile, Portilla and Simoncelli (2000), using steerable pyramid, made a thorough investigation of feature statistics for representing texture patterns. These models appear to be very successful in capturing stochastic textures, but seem to be less effective for textures with salient local structures and regular spatial arrangements.

Meanwhile, there have been major developments in sparse coding (Olshausen and Field, 1996) and independent component analysis (Bell and Sejnowski, 1997). These methods represent natural images (actually image patches) by linear superposition of image bases. By asking for sparseness or independence of the coefficients in the linear decomposition, localized, elongate and oriented image bases can be learned from natural images. The learned bases constitute a general and efficient vocabulary for image representation. The idea of sparseness has also been studied in wavelet community (e.g., Mallat and Zhang, 1993; Candes and Donoho, 2000).

This inspired us to re-consider the role of linear filters in texture models. The filter responses are inner products between the image and localized image bases, and this is a bottom-up operation. However, generative models are for top-down representation, and therefore it appears to be more appropriate for the images bases to play a representational role as in sparse and independent component analysis. This consideration promoted us to construct texture models based on linear representation instead of linear operation. The resulting model is a two-level top-down model. At the bottom level, the texture image is represented by linear superposition of image bases, and at the top level, the spatial arrangements of image bases and their coefficients are to be modeled. The sparse and independent component analysis suggests that with well-designed image bases, the coefficients should become much sparser and less dependent than the raw image intensities, and thus more susceptible to modeling. Guided by the top-down model, local structures and their organizations can be more sharply captured, because variable selection or explaining-away effect can be easily achieved in model-based context.

To be more specific, the spatial model for the image bases and their coefficients incorporate sparseness in the sense that only a small number of image bases have non-zero coefficients or are active, so that the locations, orientations, and elongations of the active image bases capture the *geometrical* aspect of the texture image, or they tell us how to *sketch* the image, while the coefficients of these active bases capture the *photometrical* aspect of the image, or they tell us how to *paint* the image given the sketch. To visually illustrate the sketch, we can depict each active image base by a small line segment or a linelet of the same location, orientation, and length, as first proposed by Bergeaud and Mallat (1996). The sparse decomposition in general achieves about 100 folds of dimension reduction.

The two-level top-down model is a hidden Markov model, with the coefficients of image bases being latent variables or missing data. While rigorous model fitting typically involves EM-type algorithms (Dempster, Laird, and Rubin, 1977),

in this article, we shall isolate the problem of modeling the sketch of the texture images, while assuming that the sketch can be obtained from the image. Our sketch model is a causal Markov chain model whose conditional distributions are characterized by a set of simple geometrical feature statistics automatically selected from a pre-defined vocabulary.

Embellished versions of our model can be useful in the following regards. In computer vision, it can be used for image segmentation and perceptual grouping. In computer graphics, as the sketch captures the geometrical essence of images, it may be used for non-photo realistic rendering. For understanding human vision, it provides a model-based representational theory for Marr's primal sketch (Marr 1982).

2 Sketching the Image

2.1 Sparse Coding and Independent Component Analysis

The essential idea of sparse coding (Olshausen and Field, 1996), independent component analysis (Bell and Sejnowski, 1999), and their combination (Lewiki and Olshausen, 1999) is the assumption that an image $\mathbf{I}$ can be represented as the superposition of a set of image bases. The bases are selected from an over-complete basis (vocabulary) $\{\mathbf{b}_{(x,y,l,\theta,e)}\}$, where (x, y) is the central position of the base on the image domain, and (l, θ, e) is the type of the base. l is the scale or length of the base, θ the orientation, and e the indicator for even/odd bases. The DC components of the bases are 0, and the l_2 norm of the bases are 1. Therefore,

$$\mathbf{I} = \sum_{(x,y,l,\theta,e)} c_{(x,y,l,\theta,e)} \mathbf{b}_{(x,y,l,\theta,e)} + N(0, \sigma^2), \tag{1}$$

$$c_{(x,y,l,\theta,e)} \sim p(c) = \rho\delta_0 + (1-\rho)N(0, \tau^2), \text{independently}. \tag{2}$$

The base coefficients $c_{(x,y,l,\theta,e)}$ is assumed to be independently distributed according to a distribution $p(c)$, which is the mixture of a point mass at 0 (then the base is said to be inactive) and a Gaussian distribution with a large variance (for active state).

In general, the basis (vocabulary) can be learned from natural images by minimizing the number of active bases (i.e. sparse coding) or by maximizing a measure of independence between the coefficients.

In our experiments, we select a set of bases for vocabulary $\mathbf{b}_{(x,y,l,\theta,e)}$, as shown in Figure 1. We use 128 difference of offset Gaussian (DOOG) filters (Malik and Perona, 1989) at various scales and orientations, even and odd. There are 5 scales, and we adopt a curvelet design of Candes and Donoho (2000), that is, for smaller scales, the bases become more elongate and there are more orientations. We also use the center-surround Difference of Gaussian (DOG) filters at 8 scales to account for variations that cannot be efficiently captured by the DOOG bases.

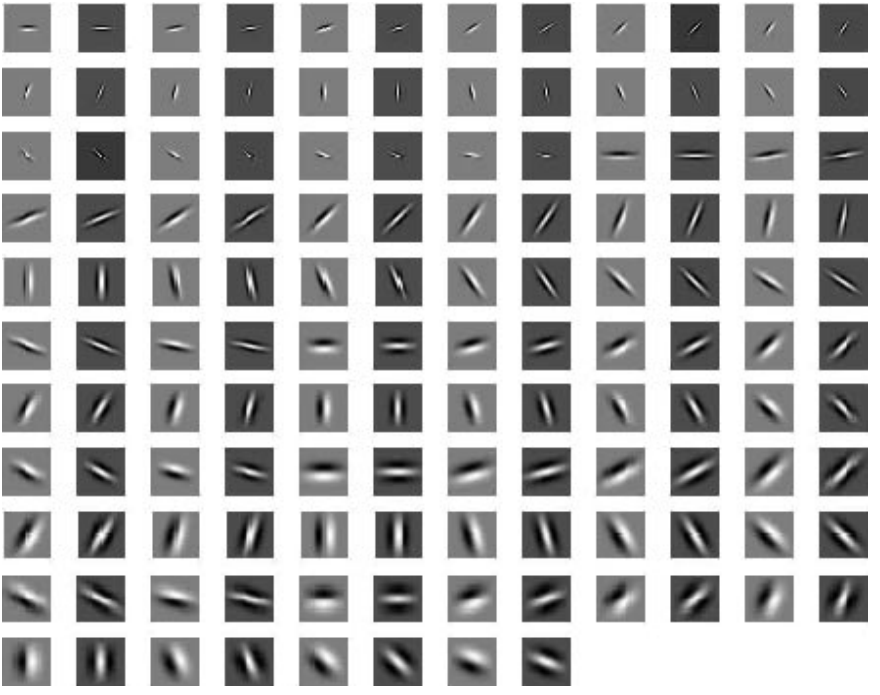

Fig. 1. Image bases used in representation and modeling.

2.2 Matching Pursuit and Its MCMC Version

Since the bases are over-completed, the coefficients have to be inferred by sampling a posterior probability. In this section, we extend the heuristic matching pursuit algorithm of Mallat and Zhang (1993) to a more principled Markov chain Monte Carlo (MCMC) algorithm.

We sample the posterior distribution of the coefficients $p(\{c_{(x,y,l,\theta,e)}\} \mid \mathbf{I})$ in Model (1) in order to find a symbolic representation or a sketch of image $\mathbf{I}$. We assume the parameters (ρ, τ^2, σ^2) are known for this model. First, let's consider the Gibbs sampler for posterior sampling. For simplicity, we use j or i to index (x, y, l, θ, e) and we define $z_j = 1$ if $\mathbf{b}_j$ is active, i.e., $c_j \neq 0$, and $z_j = 0$ otherwise. Then the algorithm is as follows:

1. Randomly select a base $\mathbf{b}_j$. Compute $\mathbf{R} = \mathbf{I} - \sum_{i \neq j} c_i \mathbf{b}_i$, i.e., the residual image. Let $r_j =< \mathbf{R}, \mathbf{b}_j > /(1 + \sigma^2/\tau^2)$, and $\sigma_\star^2 = 1/(1/\sigma^2 + 1/\tau^2)$.
2. Compute the Bayes factor by integrating out c_j,

$$\gamma_j = \frac{p(\mathbf{I} \mid z_j = 1; \{c_i, \forall i \neq j\}, \mathbf{I})}{p(\mathbf{I} \mid z_j = 0; \{c_i, \forall i \neq j\}, \mathbf{I})} = \exp\{\frac{r_j^2}{2\sigma_\star^2}\}\sqrt{\sigma_\star^2/\tau^2}.$$

Then the posterior probability

$$p(z_j = 1 \mid \{c_i, \forall i \neq j\}, \mathbf{I}) = \frac{(1-\rho)\gamma_j}{\rho + (1-\rho)\gamma_j}.$$

3. Sample z_j according to the above probability. If $z_j = 0$, then set $c_j = 0$, otherwise, sample $c_j \sim N(r_j, \sigma_\star^2)$. Go back to [1].

The problem with this Gibbs sampler is that if σ^2 is small, the algorithm is too willing to activate the base $\mathbf{b}_j$ even though the response r_j is not that large, or in other words, the algorithm is too willing to jump into a local energy minimum. The idea of matching pursuit of Mallat and Zhang (1993) can be used to remedy this problem. That is, instead of randomly selecting a base $\mathbf{b}_j$ to update, we randomly choose a window W on the image domain, and look at all the inactive bases within this window W. Then we sample one base by letting these bases compete for an entry. So we have the following windowed-Gibbs sampler or a Metropolized matching pursuit algorithm:

1. Randomly select a window W on the image domain, let A be the number of active bases within W, and let B be the number of inactive bases within W. With probability p_{birth} (a pre-designed number), go to [2]. With probability $p_{\text{death}} = 1 - p_{\text{birth}}$, go to [4].
2. For each inactive base $j = (x, y, l, \theta, e)$ with $(x, y) \in W$ and $z_j = 0$, compute γ_j as described in [1] and [2] of the Gibbs sampler. Then with probability

$$p_{\text{accept}} = \frac{p_{\text{death}} \sum_{j:z_j=0;\ (x,y)\in W} (1-\rho)\gamma_j/\rho}{p_{\text{birth}} \times (A+1)},$$

   go to [3], with probability $1 - p_{\text{accept}}$ go back to [1].
3. Among all the inactive bases j with $(x, y) \in W$ and $z_j = 0$, sample a base j with probability proportional to γ_j, then let $z_j = 1$ and sample c_j as described in [3] of the Gibbs sampler. Go back to [1].
4. If $A > 0$, then randomly select an active base $\mathbf{b}_k$ with $z_k = 1$ and $(x, y) \in W$. Then temporarily turn off $\mathbf{b}_k$, i.e., set $c_k = 0$, and $A \to A - 1$ temporarily. Then for all the inactive bases j with $z_j = 0$ and $(x, y) \in W$, including base k, do the same computation as [2] (as if $c_k = 0$), and compute p_{accept} as in [2].
5. With probability $1/p_{\text{accept}}$, accept the proposal of deleting the base k, i.e., set $c_k = 0$. Go back to [1]. With probability $1 - 1/p_{\text{accept}}$, reject the proposal of deleting base k, i.e., recover the original c_k. Go back to [1].

This is a Metropolis algorithm with two types of proposals, adding an inactive base and deleting an active base, and this is a birth and death move. In addition to that, we can easily incorporate updating schemes, including 1) perturbing the coefficient of an active base, i.e., updating c_j; 2) moving an active base to a different position, i.e., updating (x, y); 3) shortening or stretching an active base by changing its length l; 4) rotating an active base to a different orientation, i.e., updating θ.

For the simple model (1), if we let $\sigma^2 \to 0$, then the Metropolis algorithm described above goes to the windowed version of the matching pursuit algorithm. In this paper, we shall just use the latter algorithm for simplicity. We feel that the MCMC version of matching pursuit is useful in two aspects. Conceptually, it helps us to understand matching pursuit as a limiting MCMC for posterior sampling. Practically, we believe that the MCMC version, especially with the moves for updating coefficients, positions, lengths, orientations of the active bases, is useful for us to re-estimate the image sketch after we fit a better prior model for sketch.

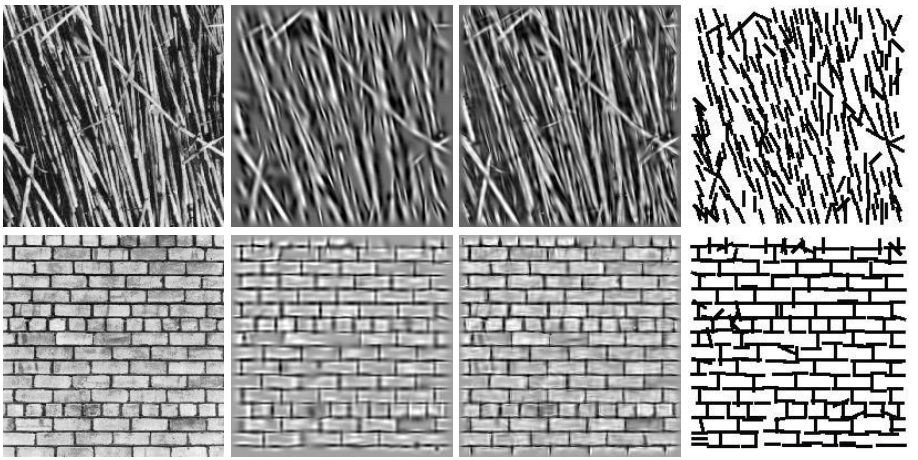

Fig. 2. Sparse decomposition by overcomplete basis and the symbolic sketch.

Experiments I. Figure 2 shows some results of sparse decomposition with overcomplete basis, and more results are shown in Figures 3 and 4. Figure 2.a are observed images, Figure 2.b are reconstructed images with a small number of bases whose coefficients are larger than 5, $|c_j| > 5$, the rate of compression is usually about 100 to 150 folds, e.g., for a 200× 200 image, we only need about 300 bases. Figure 2.c is a finer reconstruction with all bases whose $|c_j| > 2$, the rate of compression is about 30. For each selected base $\mathbf{b}_{(x,y,l,\theta,e)}$, we illustrate it symbolically by a linelet of the same length l and orientation θ, while ignoring the brightness $c_{(x,y,l,\theta,e)}$ and the odd/even indicator e. All the linelets are of the same width. Figure 2.d shows the sketches of the images.

3 Modeling the Image Sketch

Now we shall improve the simple prior model (1) by a sophisticated sketch model that accounts for the spatial arrangements of image bases.

We may use the following two representations interchangeably for the sketch of a texture image $\mathbf{I}$, and let's denote the sketch by $\mathbf{S}$.

1. A list: let n be the number of active bases, then we have
$$\mathbf{S} = \{s_t = (x_t, y_t, c_t, l_t, \theta_t, e_t), t = 1, ..., n\}.$$
2. A bit-map: let $\delta_{x,y}$ be the indicator of whether there is an active base at (x, y) or not, then
$$\mathbf{S} = \{s_{x,y} = (\delta_{x,y}, c_{x,y}, l_{x,y}, \theta_{x,y}, e_{x,y}), \forall (x, y)\}.$$
If $\delta_{x,y} = 0$, i.e., there is no active base, then all the (c, l, θ, e) take null values.

Using the first representation, the two-level top-down model is of the following form:

$$\mathbf{S} = \{s_t = (x_t, y_t, c_t, l_t, \theta_t, e_t), t = 1, ..., n\} \sim \text{sketch model}(\Lambda),$$
$$\mathbf{I} = \sum_{s_t \in \mathbf{S}} c_t \mathbf{b}_{x_t, y_t, l_t, \theta_t, e_t} + \epsilon.$$

Clearly, the sparse and independent component analysis is a special case of the above model, where at the top level, (x_t, y_t) follow Poisson point process, (l_t, θ_t, e_t) follow independent uniform distributions, and c_t follows a normal distribution with a large variance. The sketch $\mathbf{S}$ is the latent variable or missing data, and Λ is the unknown parameter. So the overall model should be fitted by the EM algorithm or its stochastic versions. The general form of such a model fitting procedure is to iterate the following two steps.

1. Scene reconstruction: Estimate $\mathbf{S}$ from $\mathbf{I}$, conditional on Λ.
2. Scene understanding: Estimate Λ from $\mathbf{S}$.

In our current work, we isolate the problem of modeling $\mathbf{S}$, assuming that $\mathbf{S}$ can be inferred from $\mathbf{I}$ by matching pursuit algorithm and its MCMC version with the simple sparse and independent prior model on $\mathbf{S}$. To further simplify the problem, we ignore c and e, i.e., the photometrical aspects, and only concentrate on the modeling of (l, θ), i.e., the geometrical aspects of the sketch $\mathbf{S}$.

3.1 Previous Models on Textures

1. MRF and FRAME models. The FRAME model by Zhu, et al. (1997) incorporates the idea of large filters and histogram[10] into Markov random fields.

$$p(\mathbf{I} \mid \Lambda) = \frac{1}{Z(\Lambda)} \exp\{\sum_{l,\theta,e} \sum_{x,y} \lambda_{l,\theta,e}(< \mathbf{I}, \mathbf{b}_{(x,y,l,\theta,e)} >)\}, \tag{3}$$

where $\Lambda = \{\lambda_{l,\theta,e}(\ ), \forall (l, \theta, e)\}$ is a collection of one-dimensional functions, and $Z(\Lambda)$ is the normalizing constant. This model is of the Gibbs form. The λ functions can be quantitized into step functions. This model is derived and justified

as the maximum entropy distribution that reproduces the marginal distributions (or histograms in the quantitized case) of $< \mathbf{I}_{\text{obs}}, \mathbf{b}_{(x,y,l,\theta,e)} >$ where $\mathbf{I}_{\text{obs}}$ is the observed image.

If we compare model (3) with model (1), we can see that in FRAME model (3), the $\mathbf{b}_{(x,y,l,\theta,e)}$ are used for bottom-up operation, and the model is based on features $< \mathbf{I}, \mathbf{b}_{(x,y,l,\theta,e)} >$. In contrast, in model (1), the $\mathbf{b}_{(x,y,l,\theta,e)}$ are used for top-down representation, and the model is based on latent variables $c_{(x,y,l,\theta,e)}$. We believe that the feature-based models and latent-variable models are two major classes of models, and the former can be modified into the latter if we replace features by latent variables. Of course, this is in general not a trivial step.

As to the spatial model for point process $\mathbf{S} = \{s_t = (x_t, y_t, l_t, \theta_t), t = 1, ..., n\}$, Guo, et al. (2001) proposed a Gestalt model of the Gibbs form

$$p(\mathbf{S} \mid \Lambda) = \frac{1}{Z(\Lambda)} \exp\{\lambda_0 n + \sum_t \lambda_1(s_t) + \sum_{t_1 \sim t_2} \lambda_2(s_{t_1}, s_{t_2})\}, \tag{4}$$

where $t_1 \sim t_2$ means that s_{t_1} and s_{t_2} are neighbors. So this model is a pair-potential Gibbs point process model (e.g., Stoyan, Kendall, Mecke, 1985). Guo et al. (2000) further parameterized the model by introducing a small set of Gestalt features (Koffka, 1935) for spatial arrangement. Again, the model can be justified by the maximum entropy principle.

The Gibbs models (3) and (4) are analytically intractable because of the intractability of the normalizing constant $Z(\Lambda)$. Sampling and maximum likelihood estimation (MLE) have to be done by MCMC algorithms.

2. Causal Markov models. One way to get around this difficulty is to use a causal Markov model. The causal methods have been used extensively in early work on texture generation, most notably, the causal model of Popat and Picard (1993). In the causal model, the joint distribution of the image $\mathbf{I}$ is factorized into a sequence of conditional distribution by imposing a linear order on all the pixels (x, y), e.g.,

$$p(\mathbf{I}) = \prod_{x=1}^{m} \prod_{y=1}^{n} p(\mathbf{I}_{x,y} \mid \mathbf{I}_{\mathcal{N}(x,y)}),$$

where x, y index the pixel, and $\mathcal{N}(x, y)$ is the neighboring pixels within a certain spatial distance that are scanned before (x, y) The causal model is analytically tractable because of the factorization form.

The causal plan has been successfully incorporated in example-based texture synthesis by Efros and Leung (1999), Liang et al. (2001), Efros and Freeman (2001). Hundred of realistic textures have been synthesized.

3.2 Modeling the Sketch Patterns

With the above preparation, we are ready to describe our model for image sketch. Let $\mathbf{S}$ be the sketch of $\mathbf{I}$, and let $\mathbf{S}_{\mathcal{N}(x,y)}$ be the sketch of the causal neighborhood

of (x, y). Recall that both $\mathbf{S}$ and $\mathbf{S}_{\mathcal{N}(x,y)}$ have two representations. Our model is of the following causal form

$$p(\mathbf{S}) = \prod_{x=1}^{m} \prod_{y=1}^{n} p(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)}).$$

The conditional distribution is

$$p(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)}) = \frac{1}{Z(\Lambda \mid \mathbf{S}_{\mathcal{N}(x,y)})} \exp\{\lambda_0 \delta_{x,y} + \lambda_1(l_{x,y}, \theta_{x,y}) + \sum_{s_t \in \mathbf{S}_{\mathcal{N}(x,y)}} \lambda_2(l_{x,y}, \theta_{x,y}; l_t, \theta_t; x_t - x, y_t - y)\},$$

where Z is the normalizing constant, and if $l_{x,y}$ and $\theta_{x,y}$ take on the null values when $\delta_{x,y} = 0$, $\lambda_1(\ )$ and $\lambda_2(\ )$ are 0. If $\lambda_1(\ )$ and $\lambda_2(\ )$ are always 0, then the model reduces to the simple Poisson model (1). Like in the FRAME model (Zhu, et al. 1997) and the Gestalt model (Guo, et al. 2001), this conditional distribution can be derived or justified as the maximum entropy distribution that reproduces the probability that there exists a linelet, the distribution of the orientation and length of the linelet, and the pair-wise configuration made up by this linelet and a nearby existing linelet.

In this model, the probability that we sketch a linelet at (x, y) depends on the attributes of this linelet, and more important, how this linelet lines up with existing linelets nearby, for instance, whether the proposed linelet connects with a nearby existing linelet, or whether the proposed linelet is parallel with a nearly existing linelet, etc. One can envisage this conditional model as modeling the way an artist sketches a picture by adding one stroke at a time. Similar maximum entropy models are also used in language[3].

We can also write the conditional model in a log-additive form

$$\log \frac{p(\delta_{x,y} = 1, l_{x,y}, \theta_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)})}{p(\delta_{x,y} = 0 \mid \mathbf{S}_{\mathcal{N}(x,y)})} = \lambda_0 + \lambda_1(l_{x,y}, \theta_{x,y}) + \sum_{s_t \in \mathbf{S}_{N(x,y)}} \lambda_2(l_{x,y}, \theta_{x,y}; l_t, \theta_t; x_t - x, y_t - y).$$

One may argue that a causal model for spatial point process is very contrived. We agree. A causal order is physically nonsensical. But for the purpose of representing visual knowledge, it has some conceptual advantages because of its analytical tractability. The situation is very similar to view-based methods for objective recognition. Moreover, the model is also suitable for the purpose of coding and compression. Mathematically, one may view this model as a causal (or factorization) approximation to the Gestalt model of Guo et al. (2001). It is expected that the causal approximation loses some of the expressive power of the non-causal model, but this may be compensated by making the causal model more redundant.

The current form of the model only accounts for pair-wise configurations of the linelets. We can easily extend the model to account for configurations that involves more than two linelets.

3.3 Feature Statistics, Model Fitting, and Feature Pursuit

Because the length and orientation has been taken care of by $\lambda_1(l_{s,y}, \theta_{x,y})$, we choose to parameterize pair-wise configuration function $\lambda_2(\ )$ as $\lambda_2(D(s_{x,y}, s_t), A(s_{x,y}, s_t))$, for $s_t \in S_{\mathcal{N}(x,y)}$, where $D(\ )$ is the smallest distance between the two linelets, and $A(\ )$ the angle between the two linelets. Note that there is some loss of information by this parameterization, in particular, it cannot distinguish between two crossing linelets and two touching linelets if the angles are the same, because in both cases, the smallest distance is 0. But still, we would like to see how far we can go with this simple parameterization.

We could further express $\lambda_1(l, \theta) = \lambda_{11}(l) + \lambda_{12}(\theta) + \lambda_{13}(l, \theta)$, i.e., decompose λ_1 into the main effects for l and θ respectively, and the interaction or combination between l and θ. This is the ANOVA (analysis of variance) decomposition, which is redundant unless we add some constraints. We do not need to worry about this issue because we use this ANOVA decomposition for the sake of feature selection to be described later, and the feature selection method will not select redundant feature statistics. Similarly, we can decompose $\lambda_2(\ )$ by $\lambda_2(D, A) = \lambda_{21}(D) + \lambda_{22}(A) + \lambda_{23}(D, A)$.

After that we choose to quantize l, θ, D, and A, i.e., each of these variables can only take a finite set of values. Then the functions can be reduced to vectors of function values. For instance, for a function $\lambda(x)$, if $x \in \{x_1, ..., x_n\}$, we can represent $\lambda(\ )$ by $\lambda = (\lambda_1, ..., \lambda_n)$, so that $\lambda_j = \lambda(x_j)$. Therefore $\lambda(x) = \sum_{j=1}^{n} \lambda_j 1_{x=x_j} =< \lambda, H(x) >$, where $H(x)$ is the vector of indicators $(1_{x=x_j}, j = 1, ..., n)$. Therefore, we can write the conditional distribution in our sketch model as

$$
\begin{aligned}
p(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)}) &= \frac{1}{Z(\Lambda \mid \mathbf{S}_{\mathcal{N}(x,y)})} \exp\{\lambda_0 \delta_{x,y} + < \lambda_{11}, H_{11}(l_{x,y}) > \\
&+ < \lambda_{12}, H_{12}(\theta_{x,y}) > + < \lambda_{13}, H_{13}(l_{x,y}, \theta_{x,y}) > \\
&+ \sum_{s_t \in \mathbf{S}_{\mathcal{N}(x,y)}} < \lambda_{21}, H_{21}(D(s_{x,y}, s_t)) > + < \lambda_{22}, H_{22}(A(s_{x,y}, s_t)) > \\
&+ < \lambda_{23}, H_{23}(D(s_{x,y}, s_t), A(s_{x,y}, s_t)) >\} \\
&= \frac{1}{Z(\Lambda \mid \mathbf{S}_{\mathcal{N}(x,y)})} \exp\{< \Lambda, H(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)}) >\},
\end{aligned}
$$

where $\Lambda = (\lambda_0, \lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{21}, \lambda_{22}, \lambda_{23})$, and

$$
\begin{aligned}
H(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)}) = (&\delta_{x,y}, H_{11}(l_{x,y}), H_{12}(\theta_{x,y}), H_{13}(l_{x,y}, \theta_{x,y}), \\
&\sum_{\mathbf{S}_{\mathcal{N}(x,y)}} H_{21}(D(s_{x,y}, s_t)), \sum_{\mathbf{S}_{\mathcal{N}(x,y)}} H_{22}(A(s_{x,y}, s_t)), \\
&\sum_{\mathbf{S}_{\mathcal{N}(x,y)}} H_{23}(D(s_{x,y}, s_t), A(s_{x,y}, s_t))).
\end{aligned}
$$

In our work, $H(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)})$ have one to two hundred components. It can be easily shown that $p(s_{x,y} \mid S_{\mathcal{N}(x,y)})$ achieves maximum entropy under the constraints on $< H(s_{x,y} \mid S_{\mathcal{N}(x,y)}) >_{p(s_{x,y} \mid S_{\mathcal{N}(x,y)})}$, where $< >_p$ means expectation with respect to distribution p. The full model is

$$p(S \mid \Lambda) = \prod_{x=1}^{m}\prod_{y=1}^{n} p(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)})$$
$$= \{\prod_{x=1}^{m}\prod_{y=1}^{n} \frac{1}{Z(\Lambda \mid \mathbf{S}_{\mathcal{N}(x,y)})}\} \times \exp\{< \Lambda, \sum_{x=1}^{m}\sum_{y=1}^{n} H(s_{x,y} \mid \mathbf{S}_{\mathcal{N}(x,y)}) >\}.$$

Now, let's consider model fitting. Let $\mathbf{S}_{\text{obs}}$ be the observed sketch of an image. Then we can estimate Λ by maximizing the log-likelihood

$$l(\Lambda \mid \mathbf{S}_{\text{obs}}) = \sum_{x,y}\{< \Lambda, H(\mathbf{S}^{\text{obs}}_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}) > - \log Z(\Lambda \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)})\}.$$

Statistical theories of exponential family models tell us that

$$\frac{\partial}{\partial \Lambda} l(\Lambda \mid \mathbf{S}_{\text{obs}}) = \sum_{x,y}\{H(s^{\text{obs}}_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}) - < H(s_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}) >_{p(s_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}, \Lambda)}\},$$

and

$$\frac{\partial^2}{\partial \Lambda^2} l(\Lambda \mid \mathbf{S}_{\text{obs}}) = -\sum_{x,y}\{\text{Var}_{p(s_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}, \Lambda)}[H(s_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)})]\}.$$

Therefore, the log-likelihood is concave, and the model can be fitted by the Newton-Raphson or equivalently in this case, the Fisher scoring algorithm,

$$\Lambda_{t+1} = \Lambda_t - [\frac{\partial^2}{\partial \Lambda^2} l(\Lambda \mid \mathbf{S}_{\text{obs}})] \mid^{-1}_{\Lambda_t} \frac{\partial}{\partial \Lambda} l(\Lambda \mid \mathbf{S}_{\text{obs}}) \mid_{\Lambda_t} .$$

The convergence of Newton-Raphson is very fast; usually 5 iterations can already give very good fit.

Both the first and second derivatives of the log-likelihood are of the form $\sum_{x,y} g(x,y)$. For each pixel (x,y), we need to evaluate the probabilities of all possible $s_{x,y}$. So the computation is still quite costly, although much more manageable compared to MCMC type of algorithms. To further increase the efficiency, we choose to sample a small number of pixels instead of going through all of them. More specifically, for each (x,y), let $\pi_{x,y} \in [0,1]$ be the probability that pixel (x,y) will be included in the sample. Then after we collect a sample of (x,y) by independent coin-flipping according to $\pi_{x,y}$, we can approximate

$$\sum_{x,y} g(x,y) \approx \sum_{(x,y) \in \text{sample}} g(x,y)/\pi_{x,y},$$

where the right hand side is the Hovitz-Thompson unbiased estimator of the left hand side. As to the choice of $\pi_{x,y}$, if there is a linelet at (x,y) on $\mathbf{S}_{\text{obs}}$, then we always let $\pi_{x,y} = 1$. For other empty pixels (x,y), we can set $\pi_{x,y}$ according to our need for speed. Usually, even if we take $\pi_{x,y} = .01$ for empty pixels, the algorithm can still give satisfactory fit.

It is often the case that some components of $\sum_{x,y} H(s^{\text{obs}}_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)})$ are 0, and if we implement the usual Newton-Raphson procedure, then the corresponding components of Λ will go to $-\infty$, thereby creating hard constraints. While this is theoretically the right result, it can make the algorithm unstable. We choose to stop fitting such components as long as the corresponding components of $\sum_{x,y} < H(s_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}) >_{p(s_{x,y}|\mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)})}$ drop below a threshold, e.g., .5.

For a specific observed image, we do not want to use all the one to two hundred dimensions in our model. In fact, we can just select a small number of components of $H(s_{x,y} \mid S_{\mathcal{N}(x,y)})$ using some model selection methods such as Akaike information criterion (AIC). While best-set selection is too time-consuming, we can consider a feature pursuit scheme studied by Zhu, et al. (1997), i.e., we start with only λ_0 and $\delta_{x,y}$ in our model. Then we repeatedly add one component of H at a time, so that the added component leads to the maximum increase in log-likelihood. Although the log-likelihood is analytically tractable for the causal model, the computation of the increase of log-likelihood for each candidate component of H is still quite costly. So as an approximation, we choose the component H_k so that

$$g_k = \frac{\{\sum_{x,y} H_k(s^{\text{obs}}_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}) - < H_k(s_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)}) >_{\hat{p}}\}^2}{\sum_{x,y} \text{Var}_{\hat{p}}[H_k(s_{x,y} \mid \mathbf{S}^{\text{obs}}_{\mathcal{N}(x,y)})]},$$

is the largest, where $\hat{p}$ is the currently fitted model. Intuitively, we choose a component that is worst fitted by the current model. Then we stop after a number of steps. Again, the Horvitz-Thompson estimator can be used for faster computation of g_k.

Experiments II. Figures 3 and 4 show some experiments. For each input image (left), we first compute its sketch (middle), and then a sketch model is learned with 40-60 feature statistics selected by feature pursuit. The angles as well as orientations are quantitized into 12 bins, and the distances are quantitized into 12 bins too. Of course, the model fails to capture some features that human vision is sensitive, which is expected of a pair-wise model. More sophisticated features should be used, and we leave this to future work.

4 Discussion

There are two major loose ends in our work. One is that the coefficients of the active bases are not modeled. The other is that the model fitting is not rigorous. The model can be further extended to incorporate more sophisticated local structures, such as local shapes and lighting, as well as more sophisticated organizations such as flows and graphs. The key is that these concepts should be understood in the context of a top-down generative model. For some stochastic textures, sparse decomposition may not be achievable, and therefore, we might have to stay with models built on pixel values such as the FRAME model.

We would like to stress that our goal in this work is to find a top-down generative model for textures. We are not merely trying to come up with a line-drawing version of the texture image by some edge detector, and then synthesize

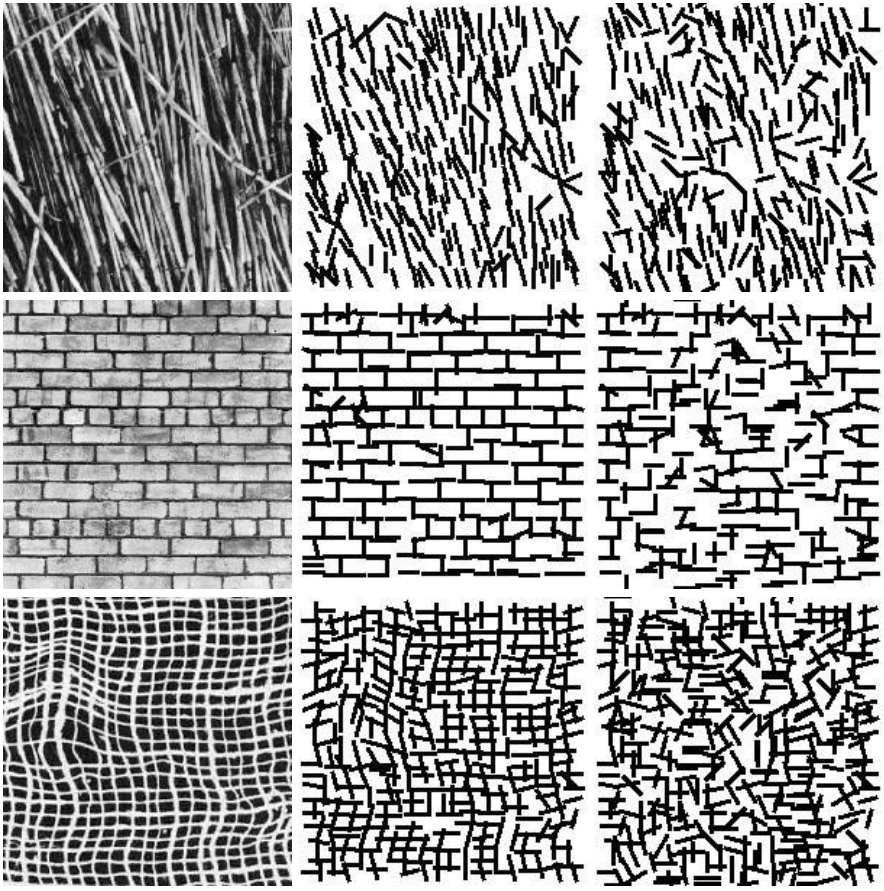

Fig. 3. Examples of learning sketch model. a) is an input image, b) is its computed sketch, and c) is the synthesized sketch as a random sample from the sketch model learned from b).

similar line-drawings. We would also like to point out that our work is inspired by Marr's primal sketch (Marr, 1982). Marr's method is bottom-up procedure-based, that is, there does not exist a top-down model to guide the bottom-up procedure.

Our eventual goal is to find the top-down model as a visual system's conception of primal sketch, so that the largely bottom-up procedure will be model-based. The hope is that the model is unified and explicit like a language, with rich vocabularies for local image structures as well as their spatial organizations. When fitted to a particular image, an automatic model selection procedure will identify a low-dimensional sub-model as the most meaningful words. The model should lie between physics-based models (that are not explicit and unified) and

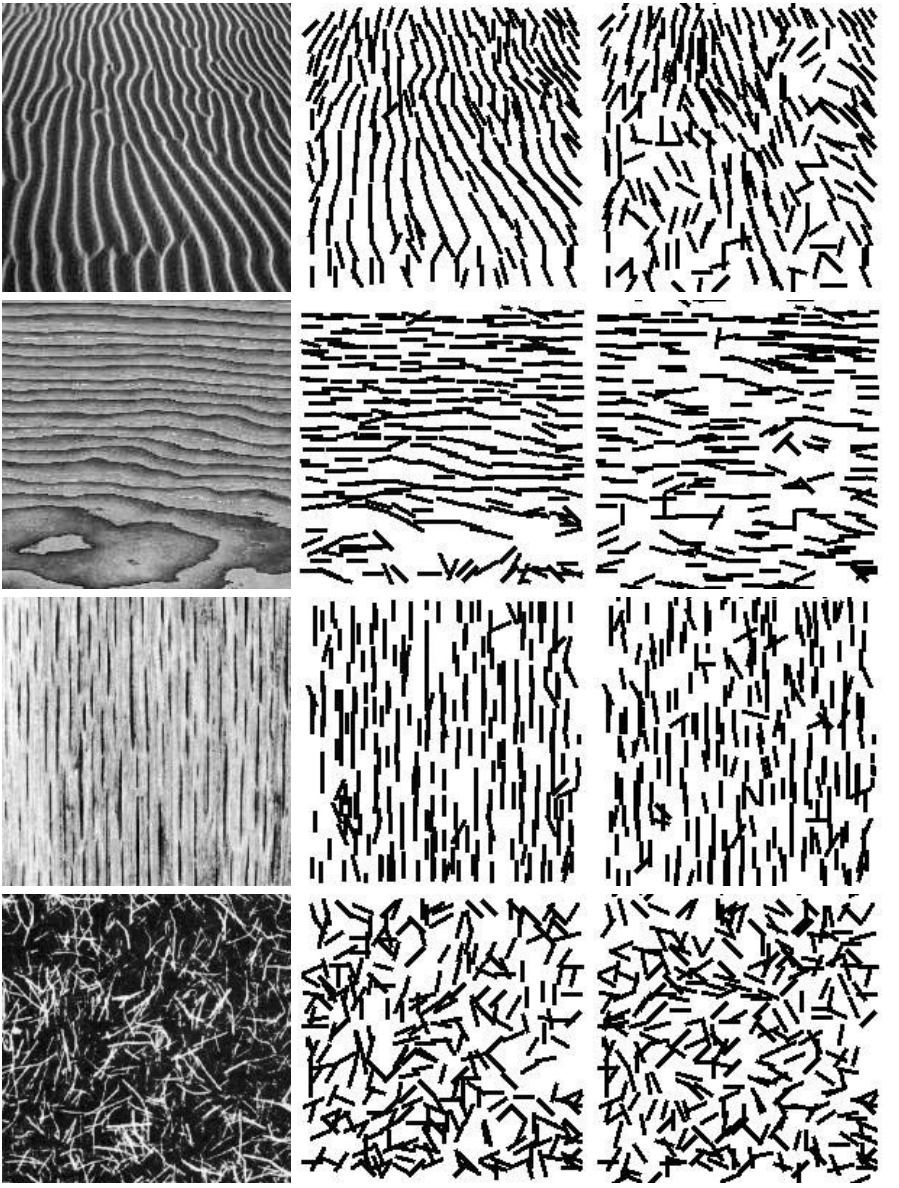

Fig. 4. Examples of learning sketch model. a) is an input image, b) is its computed sketch, and c) is the synthesized sketch as a random sample from the sketch model learned from b).

image-based synthesis (that does not involve dimension reduction). Needless to say, our current effort is merely a modest first step towards this goal.

References

1. A. J. Bell and T.J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution", *Neural Computation*, 7(6): 1129-1159, 1995.
2. F. Bergeaud and S. Mallat, "Matching pursuit: adaptive representation of images and sounds." *Comp. Appl. Math.*, 15, 97-109. 1996.
3. A. Berger, V. Della Pietra, and S. Della Pietra, "A maximum entropy approach to natural language processing", *Computational Linguistics*, vol.22, no. 1 1996.
4. E. J. Candès and D. L. Donoho, "Curvelets - A Surprisingly Effective Nonadaptive Representation for Objects with Edges", *Curves and Surfaces*, L. L. Schumaker et al. (eds), Vanderbilt University Press, Nashville, TN.
5. A. P. Dempster, N.M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society series B*, 39:1-38, 1977.
6. A. A. Efros and T. Leung, "Texture synthesis by non-parametric sampling", *ICCV*, Corfu, Greece, 1999.
7. A. A. Efros and W. T. Freeman, "Image Quilting for Texture Synthesis and Transfer", SIGGRAPH 2001.
8. S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images". *IEE Trans. PAMI* **6**. pp 721-741. 1984.
9. C. E. Guo, S. C. Zhu, and Y. N. Wu, "Visual learning by integrating descriptive and generative methods", *ICCV*, Vancouver, CA, July, 2001.
10. D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis", *SIGGRAPHS*, 1995.
11. M. S. Lewicki and B. A. Olshausen, "A probabilistic framework for the adaptation and comparison of image codes", *JOSA*, A. 16(7): 1587-1601, 1999.
12. L. Liang, C. Liu, Y. Xu, B.N. Guo, H.Y. Shum, "Real-Time Texture Synthesis By Patch-Based Sampling", MSR-TR-2001-40, March 2001.
13. J. Malik, and P. Perona, "Preattentive texture discrimination with early vision mechanisms", *J. of Optical Society of America A*, vol 7. no.5, May, 1990.
14. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. on PAMI*, vol.11, no.7, 674-693, 1989.
15. S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary", *IEEE trans. on Signal Processing*, vol.41, pp3397-3415, 1993.
16. D. Marr, *Vision*, W.H. Freeman and Company, 1982.
17. D. B. Mumford "The Statistical Description of Visual Signals" in *ICIAM 95*, edited by K.Kirchgassner, O.Mahrenholtz and R.Mennicken, Akademie Verlag, 1996.
18. B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images" *Nature*, 381, 607-609, 1996.
19. K. Popat and R. W. Picard, "Novel Cluster-Based Probability Model for Texture Synthesis, Classification, and Compression." *Proc. of the SPIE Visual Comm. and Image Proc.*, Boston, MA, pp. 756-768, 1993.
20. J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients", *IJCV*, 40(1), 2000.
21. Y. Wu, S. C. Zhu, and X. Liu, (2000)"Equivalence of Julesz texture ensembles and FRAME models", *IJCV*, 38(3), 247-265.
22. S. C. Zhu, Y. N. Wu and D. B. Mumford, "Minimax entropy principle and its application to texture modeling", *Neural Computation* Vol. 9, no 8, Nov. 1997.