Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons

THOMAS LEUNG * AND JITENDRA MALIK
Computer Science Division, University of California at Berkeley, Berkeley, CA 94720-1776, USA

Abstract.   We study the recognition of surfaces made from different materials such as concrete, rug, marble, or leather on the basis of their textural appearance. Such natural textures arise from spatial variation of two surface attributes: (1) reflectance and (2) surface normal. In this paper, we provide a unified model to address both these aspects of natural texture. The main idea is to construct a vocabulary of prototype tiny surface patches with associated local geometric and photometric properties. We call these *3D textons*. Examples might be ridges, grooves, spots or stripes or combinations thereof. Associated with each texton is an *appearance vector*, which characterizes the local irradiance distribution, represented as a set of linear Gaussian derivative filter outputs, under different lighting and viewing conditions.

Given a large collection of images of different materials, a clustering approach is used to acquire a small (on the order of 100) 3D texton vocabulary. Given a few (1 to 4) images of *any* material, it can be characterized using these textons. We demonstrate the application of this representation for recognition of the material viewed under novel lighting and viewing conditions. We also illustrate how the 3D texton model can be used to predict the appearance of materials under novel conditions.

Keywords: 3D texture, texture recognition, texture synthesis, natural material recognition

1.   Introduction

We study the recognition of surfaces made from different materials such as concrete, rug, marble, or leather on the basis of their textural appearance. Such natural textures arise from spatial variation of two surface attributes: (1) reflectance and (2) surface normal. In this paper, we provide a unified model to address both of these aspects of natural texture.

In the past, texture recognition/discrimination has been posed primarily as a 2D problem. Viewpoint and illumination are assumed constant. Two of the representative techniques are Markov random fields (Chellappa and Chatterjee, 1985; Cross and Jain, 1983; Mao and Jain, 1992; Yuan and Rao, 1993; Zhu et al., 1998) and filter responses (Fogel and Sagi, 1989; Jain

and Farrokhsia, 1991; Malik and Perona, 1990; Puzicha et al., 1997; Rubner and Tomasi, 1999). In all these work, surface normal variations are ignored. However, nature shows an abundance of such relief textures. Examples are shown in Fig. 1. Notice in particular the effects of surface normal variations: *specularities*, *shadows*, and *occlusions*. Figure 2 shows samples of the same material under different viewpoint/lighting settings. The appearance looks drastically different in the 3 images. Recognizing that they belong to the same material is a challenging task.

Variations due to surface relief cannot be dealt with by simple brightness normalization or intensity transforms (such as histogram matching). For example, if the surface structure is a ridge, a dark-light transition in one image under one illumination will become a light-dark transition when the light source is moved to the other side of the ridge. Shadows also cause significant

---

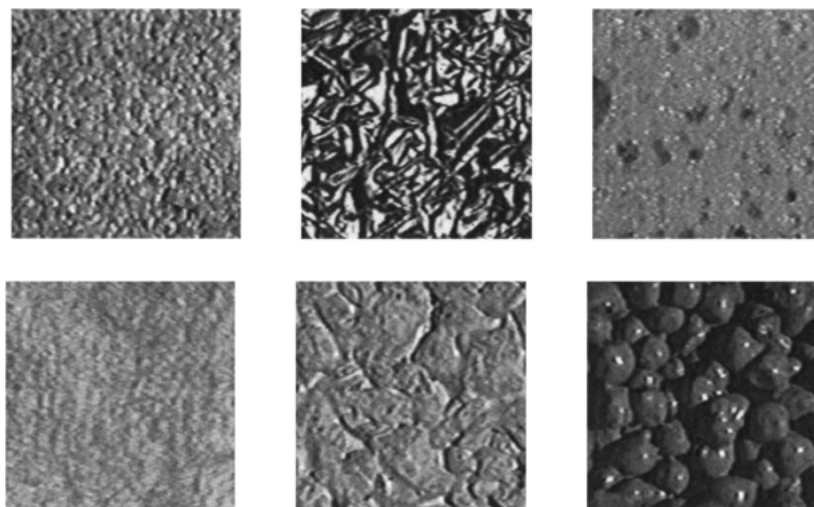*Present address: Compaq Cambridge Research Laboratory.

Figure 1. Some natural 3D textures from the Columbia-Utrecht database (Dana et al., 1999). Left to right: "Pebbles", "Aluminum Foil", "Sponge", "Rabbit Fur", "Concrete" and "Painted Spheres". These textures illustrate the effects caused by the 3D nature of the material: *specularities*, *shadows*, and *occlusions*.
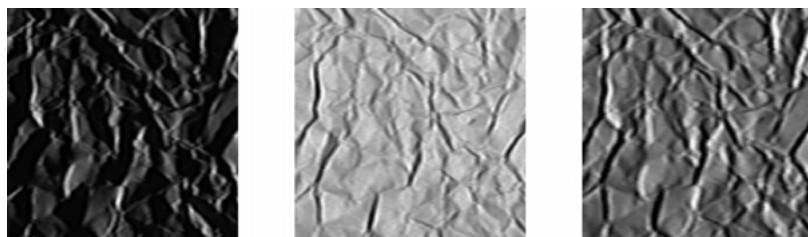

Figure 2. The same patch of the material "Crumpled Paper" imaged under three different lighting and viewing conditions. The aspect ratio of the figure is determined by the slant of the surface. Even though the three images are corresponding patches from the same material, the appearances are drastically different.

problems: two regions will have the same brightness under one illumination; while the shadowed region will be darker in another. These two problem cases are illustrated in Fig. 3.

The complexity in the relationship between the image intensity values to the viewing/lighting settings and the properties of 3D textures led to recent interest in building explicit models for 3D textures (Chantler, 1994; Chantler and McGunnigle, 1995; Dana and Nayar, 1998; Dana and Nayar, 1999b; Dana et al., 1999; Koenderink and van Doorn, 1996; Koenderink et al., 1999; Leung and Malik, 1997; van Ginneken et al., 1998). From these analytical models, such as Gaussian distributed height variation, or cylindrical models, low-order statistical quantities, e.g. brightness distribution or correlation length, are derived. However, these models are rather simple and they lack the expressiveness to solve the general problems of natural material rep-

resentation, recognition, and synthesis under varying lighting and viewing conditions.

The main idea of this paper is the following—at the local scale, there are only a small number of perceptually distinguishable micro-structures on the surface. For example, the local surface relief $\hat{n}(x, y)$ might correspond to ridges, grooves, bumps, hollows, etc. These could occur at a continuum of orientations and heights, but perceptually we can only distinguish them up to an equivalence class. Similarly, reflectance variations fall into prototypes like stripes, spots, etc. Of course one can have the product of these two sources of variation.

Our goal is to build a small, finite vocabulary of micro-structures, which we call 3D textons. This term is by analogy to 2D textons, the putative units of preattentive human texture perception proposed by Julesz nearly 20 years ago. Julesz's textons (Julesz, 1981)—orientation elements, crossings and terminators—fell
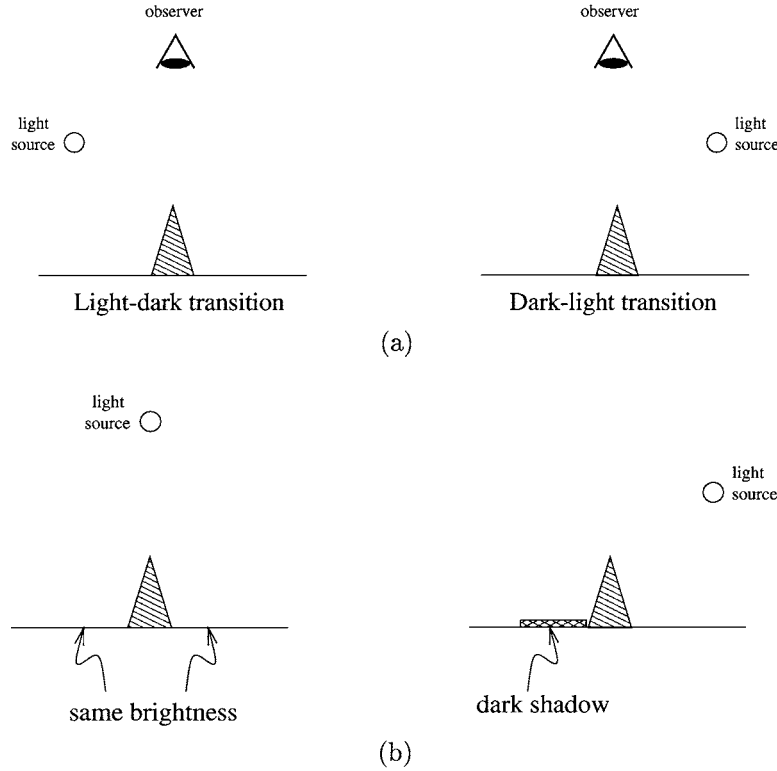
Figure 3. (a) A ridge: under one illumination, the ridge appears as a light-dark transition, while it appears as a dark-light transition under another. (b) Shadows: under one illumination, two regions have the same brightness, while under another, the brightness is different.

into disuse as they did not have a precise definition for gray level images. In this paper, we re-invent the concept and operationalize it in terms of learned co-occurences of outputs of linear oriented Gaussian derivative filters. In the case of 3D textons, we look at the concatenation of filter response vectors corresponding to different lighting and viewing directions.

Once we have built such a universal vocabulary of 3D textons, the surface of any material such as marble, concrete, leather, or rug can be represented as a spatial arrangement (perhaps stochastic) of symbols from this vocabulary. Only a small number of views are needed for this. Suppose we have learned these representations for some materials, and then we are presented with a single image of a patch from one of these materials under a novel illumination or viewpoint, the objective is to recognize which one. We have developed a recognition algorithm using a Markov Chain Monte Carlo (MCMC) sampling method.

The structure of this paper is as follows. In Section 2, we show an operationalization of finding 2D textons from images. We analyze images of different view-

ing and lighting conditions together and extend the notion of textons to 3D textons in Section 3. The algorithm for computing a 3D texton vocabulary is given in Section 4. How a material is represented in terms of the learned textons is discussed in Section 5. We contrast our 3D texton model with principle component analysis in Section 6. The problem of 3D texture recognition is presented in Section 7. Results are shown for classifying materials under novel viewing and lighting conditions. In Section 8, we present an application of the 3D texton vocabulary to predict the appearance of textures under novel viewing and lighting conditions. We conclude in Section 9. Some of the results presented here were published in Leung and Malik (1999).

2. 2D Textons

We will characterize a texture by its responses to a set of orientation and spatial-frequency selective linear filters (a filter bank). This approach has proved to be useful for segmentation (Fogel and Sagi, 1989; Malik and Perona, 1990), recognition (Puzicha et al., 1997;
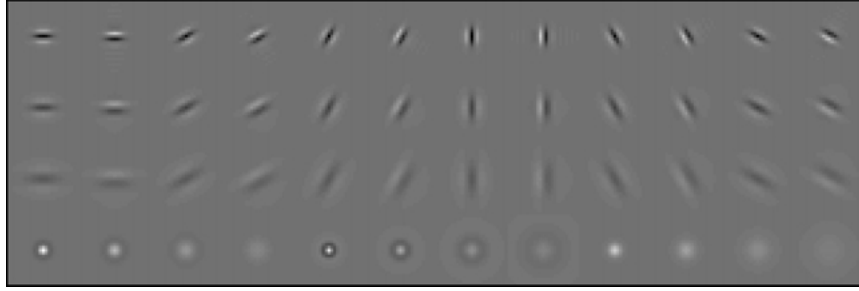
Figure 4. The filter bank used in our analysis. Total of 48 filters: 36 oriented filters, with 6 orientations, 3 scales, and 2 phases, 8 center-surround derivative filters and 4 low-pass Gaussian filters.

Rubner and Tomasi, 1999), as well as synthesis (de Bonet and Viola, 1998; Heeger and Bergen, 1995; Zhu et al., 1998).

The representation of textures using filter responses is extremely versatile, however, one might say that it is overly redundant (each pixel is represented by N_{fil} filter responses, where N_{fil} is usually around 50). Moreover, it should be noted that we are characterizing textures, entities with some spatially repeating properties by definition. Therefore, we do not expect the filter responses to be totally different at each pixel over the texture. Thus, there should be several distinct filter response vectors and all others are noisy variations of them.

This intuition leads to our proposal of clustering the filter responses into a small set of prototype response vectors. We call these prototypes *textons*. Algorithmically, each texture is analyzed using the filter bank shown in Fig. 4. There are a total of 48 filters (36 elongated filters at 6 orientations, 3 scales, and 2 phases, 8 center-surround difference of Gaussian filters, and 4 low-pass Gaussian filters.). Each pixel is now transformed to a $N_{fil} = 48$ dimensional vector. These vectors are clustered using a vector quantization algorithm, in particular *K*-means (Ball and Hall, 1967; Duda and Hart, 1973; Gersho and Gray, 1992; MacQueen, 1967; Ripley, 1996; Sebestyen, 1962). The criterion for this algorithm is to find *K centers* such that after assigning each data vector to the nearest center, the sum of the squared distance from the centers are minimized. *K*-means is a greedy algorithm which iteratively performs the following two operations: (1) assign data vectors to the nearest of the *K* centers; (2) update each of the *K* centers to the *mean* of the data vectors assigned to it. These two steps are continued until the algorithm converges and a local minimum of the criterion is achieved. These centers are the

textons. The associated filter response vectors are called the *appearance vectors*, $c_k \in \mathbb{R}^{N_{fil}}$, $k = 1, \ldots, K$.

What the textons encode can be visualized by reconstructing local image patches from the appearance vectors c_k. These local image patches, P_k, can be considered as *filters* detecting particular image structures. The reconstruction task is trivial for orthogonal or self-inverting filter banks (Burt and Adelson, 1983; Heeger and Bergen, 1995; Vaidyanathan, 1993). For non-orothogonal, and non-self-inverting filter banks, reconstructing the image from filter responses can be set up as a least-square problem. First construct the *filter matrix F* as follows: arrange each filter as a row vector and concatenate them to form a matrix. In this representation, convolution with the filter bank is equivalent to multiplying each local image patch with *F*. Each of the P_k can be obtained by multiplying the appearance vector c_k with the *pseudo-inverse* of *F* (Jones and Malik, 1992). This is illustrated in Fig. 5. The original image is shown in (a). The textons (*K*-means cluster centers) are reconstructed and shown in (b). Notice that they correspond to the dominant local structures in the image. We can quantize the filter responses at each pixel *x* of the image to the *K* textons. To reconstruct the whole image from this array of filter responses, we first independently reconstruct the intensity at each pixel from the quantized filter responses. Then, we iterate to make sure that the filter responses of the reconstructed image agrees with the raw filter responses. The result of this process for Fig. 5(a) is shown in (c). The close resemblance between (a) and (c) suggests that the quantization does not introduce much error perceptually and that the reconstruction algorithm is doing a good job.

In the next section, we will extend the texton theory to 3D textures—texture with significant local surface relief. For more discussions on 2D textons, the readers
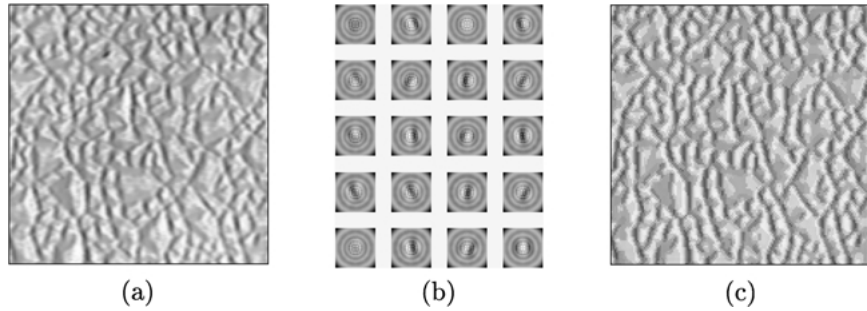
Figure 5. Illustration of K-means clustering and reconstruction from filter responses with $K = 20$. (a) Original image. (b) the K-means centers reconstructed as local filters. These centers correspond to the dominant features in the image: bars and edges at various orientations and phases; (c) Reconstruction of the quantized image. Close resemblance between (a) and (c) suggests that quantization does not introduce much error perceptually.

are referred to Malik et al. (1999), where we applied the idea of textons to the problem of image segmentation.

3. 3D Textons

For painted textures with Lambertian material, characterizing one image is equivalent to characterizing all the images under all lighting and viewing directions. However, for 3D textures, this is not the case. The effects of masking, shadowing, specularity, and mutual illumination will make the appearance of the texture look drastically different according to the lighting and viewing directions (Fig. 2). The presence of albedo variations on a lot of natural textures only makes the problem more difficult.

Let us first consider what the problems are if we try to characterize a 3D texture with only 1 image using the K-means clustering algorithm on filter outputs described in Section 2. Suppose the image of the texture consists of thin dark-light bars arising from 3 causes: (1) albedo change; (2) shadows; and (3) a deep groove. Despite the different underlying causes, all these events produce the same appearance in this particular lighting and viewing setting. Quite naturally, the K-means algorithm will cluster them together. What this means is that pixels with the same label will look different under different lighting and viewing conditions: (1) the albedo change varies according to the cosine of the lighting angle (assuming a Lambertian surface); (2) the location of the shadow boundary changes according to the direction of the light; and (3) the deep groove remains the same for a wide range of lighting and viewing conditions (Haddon and Forsyth, 1998; Koenderink and van Doorn, 1980). Thus, we will pay a significant price for quantizing these events to the same texton.

To characterize 3D textures, many images at different lighting and viewing directions will be needed. Let the number of images be N_{vl}, with $N_{vl} \gg 1$ ($N_{vl} = 20$ in our experiments). The argument is that if any two local texture structures are equivalent under N_{vl} different lighting and viewing conditions, we can safely assume that the two structures will look the same under all lighting and viewing conditions. Notice that work in the literature have attempted to show that 3–6 images will be able to completely characterize a structure in all lighting and viewing conditions (Belhumeur and Kriegman, 1998; Shashua, 1997). These results are not applicable because of the very restrictive assumptions they made: Lambertian surface model and the absence of occlusion, shadows, mutual illumination, and specularity. Indeed, deviations from these assumptions are the defining properties of most, if not all, natural 3D textures.

What this means is that the co-occurrence of filter responses across different lighting and viewing conditions specifies the local geometric and photometric properties of the surface. If we concatenate the filter responses of the N_{vl} images together and cluster these long $N_{fil}N_{vl}$ data vectors, the resulting textons will encode the appearances of dominant features in the image under *all* lighting and viewing conditions. Let us first understand what these textons correspond to. Consider the following two geometric features: a groove and a ridge. In one image, they may look the same, however, at many lighting and viewing angles, their appearances are going to differ considerably. With the filter response vectors from all the images, we can tell the difference between these two features. In other words, each of the K-means centers encodes geometric features such as ridges at particular orientations, bumps of certain sizes,
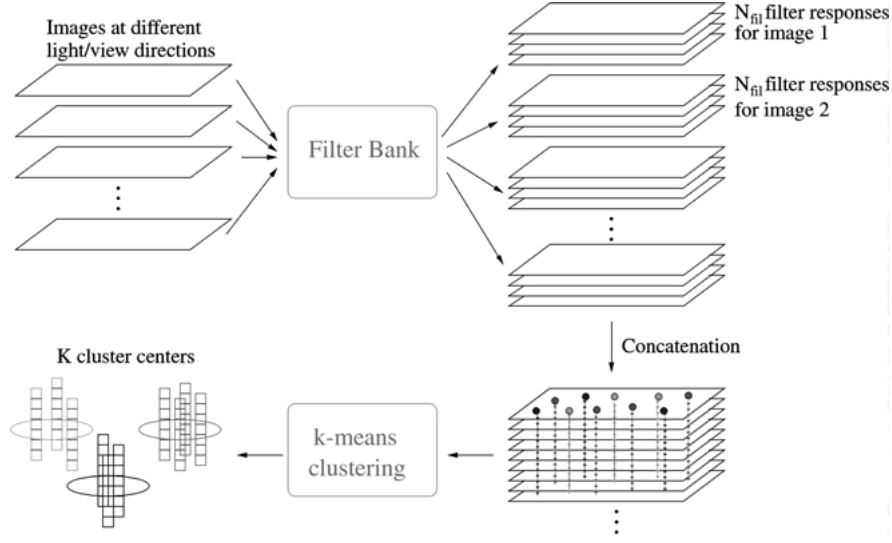
Figure 6.   Each image at different lighting and viewing directions is filtered using the filter bank. The response vectors are concatenated together to form data vectors of length $N_{fil}N_{vl}$. These data vectors are clustered using the *K*-means algorithm. The resulting centers are the 3D textons and the associated filter response vectors are called the *appearance vectors*.

grooves of some width, etc.. Similarly, the *K*-means centers will also encode albedo change vs. geometric 3D features, as well as reflectance properties (e.g. shiny vs. dull). The appearances of different features and different materials at various lighting and viewing angles are captured by the filter responses. Thus, we call these *K*-means centers 3D textons, and the corresponding $N_{fil}N_{vl}$ filter response vectors, the appearance vectors. A schematic diagram illustrating the steps of filtering, concatenating filter responses, and *K*-means clustering is shown in Fig. 6.

4.   Constructing the Vocabulary of 3D Textons

Our goal in this paper is to use images from a set of *training* materials to learn a vocabulary which can characterize *all* natural materials. This is a realistic goal because, as we have noted, the textons in the vocabulary are going to encode the appearances of local geometric and photometric features, e.g. grooves, ridges, bumps, reflectance boundaries etc. All natural materials are made up of these features. In this section, we will describe the exact steps taken to construct this universal 3D texton vocabulary.

All the images used in this paper are taken from the Columbia-Utrecht dataset (Dana et al., 1999) (`http://www.cs.columbia.edu/CAVE/curet/`). There are 60 different materials, each with 205 images

at different viewing and lighting angles.[1] 20 materials are taken randomly as the training set. For each material, 20 images of different lighting and viewing directions are used to build the texton vocabulary. The 20 images for each material are registered using the standard area-based sum-of-square-differences (SSD) algorithm.

To compute the universal vocabulary, the following steps are taken:

1. For each of the 20 training materials, the filter bank is applied to each of the $N_{vl} = 20$ images under different viewing and lighting conditions. The response vectors at every pixel are concatenated together to form a $N_{fil}N_{vl}$ vector.

2. For each of the 20 materials individually, the *K*-means clustering algorithm is applied to the data vectors. The number of centers, denoted by *K*, is 400. The *K*-means algorithm finds a local minimum of the following sum-of-square distance error:

$$Err = \sum_{i=1}^{N} \sum_{k=1}^{K} q_{ik} \|\boldsymbol{x}_i - \boldsymbol{c}_k\|^2$$

where

$$q_{ik} = 1 \quad \text{if } \|\boldsymbol{x}_i - \boldsymbol{c}_k\|^2 < \|\boldsymbol{x}_i - \boldsymbol{c}_j\|^2$$
$$\forall j = 1, \dots, K \text{ and } j \neq k$$
$$q_{ik} = 0 \quad \text{otherwise}$$

N denotes the number of pixels; $\boldsymbol{x}_i$ is the concatenated filter response vector of the ith pixel and $\boldsymbol{c}_k$ is the appearance vector for the kth center. The K-means algorithm is initialized by random samples from all the data vectors.

3. The centers for all the materials are merged together to produce a universal alphabet of size $K = 8000$.
4. The codebook is pruned down to $K = 100$ by merging centers too close together or getting rid of those centers with too few data assigned to them.
5. The K-means algorithm is applied again on samples from all the images to achieve a local minimum.

Steps 2 to 4 can be viewed as finding an initialization for the final K-means step in 5.

The learned vocabulary should possess two very important properties:

1. Expressiveness: the vocabulary learned should be able to characterize each of the materials well to allow for the discrimination between them.
2. Generalization: it should generalize well beyond the training materials. In other words, it should be as expressive for novel materials as for training materials.

An evaluation of these two properties is shown in Fig. 7. For each material, the filter responses from a frontal-parallel image of each material is quantized into the 3D texton vocabulary—filter responses at each pixel are replaced by the appearance vector of the 3D texton labeled at the pixel. An image is reconstructed from the quantized filter response vectors using the algorithm described in Section 2. The SSD error between the reconstructed image and the original image is plotted in the figure.[2] The errors from vocabularies of different sizes are also plotted for comparison. The upper diagram is the SSD error for the training materials. The lower diagram is the one for novel materials. Notice two points: (1) there is no significant difference in average reconstruction error between training materials and novel materials. In other words, our texton vocabulary is encoding generic features, rather than material-specific properties. This is an indication of good generalization. (2) The SSD errors are small for almost all materials.[3] The 3D texton vocabulary is doing a very good job encoding the properties of the materials. This reconfirms our intuition that textures are made of a small set of features. Moreover, the differences between reconstruction errors from vocabularies of different sizes are not significant. In all the

texture recognition results in this paper, the same texton vocabulary of size 100 is used. Of course, comparing reconstruction error is not the best way to evaluate the vocabulary. The real test is to use the vocabulary for the recognition and synthesis of natural materials, which we will show in Sections 7 and 8.

In our studies here, only 20 ($N_{vl} = 20$) different viewing and lighting directions are used. 20 images form a very sparse sampling of the viewing and illumination spheres. When more images are available, we should take advantage of them. However, this does not mean that we need to run the clustering algorithm on a formidably large dimensional space. We argue that 20 images are enough to make sure that each 3D texton represents different local geometric/photometric structures. Therefore, to enlarge the appearance vector of each texton, we can simply append to the vectors the average of filter responses at pixels with the corresponding label.

5. Representing Visual Appearance of Materials using 3D Textons

Once we have built such a vocabulary of 3D textons, we can acquire a model for each material to be classified. Using all the images (under different viewing and lighting conditions) available for each material, each point on the surface is assigned one of the 100 texton labels by finding the minimum distance between the texton appearance vectors to the filter responses at the point. The surface of any material such as marble, concrete, leather, or rug can now be represented as a spatial arrangement of symbols from this vocabulary. For the problem of material recognition, we ignore the precise spatial relationship of the symbols and use a histogram representation for each material. Sample histograms for 4 materials are shown in Fig. 8. Notice that these histograms are very different from each other, thus allowing good discrimination. The chi-square significance test is used to provide a measure between the similarity of two histograms (h_1 and h_2):

$$\chi^2(h_1, h_2) = \frac{1}{2} \sum_{n=1}^{\#bins} \frac{(h_1(n) - h_2(n))^2}{h_1(n) + h_2(n)} \qquad (1)$$

The significance for a certain chi-square distance is given by the chi-square probability function: $P(\chi^2 \mid \nu)$. $P(\chi^2 \mid \nu)$ is the probability that two histograms from the same model will have a distance larger than χ^2 by

New Samples, Training Materials
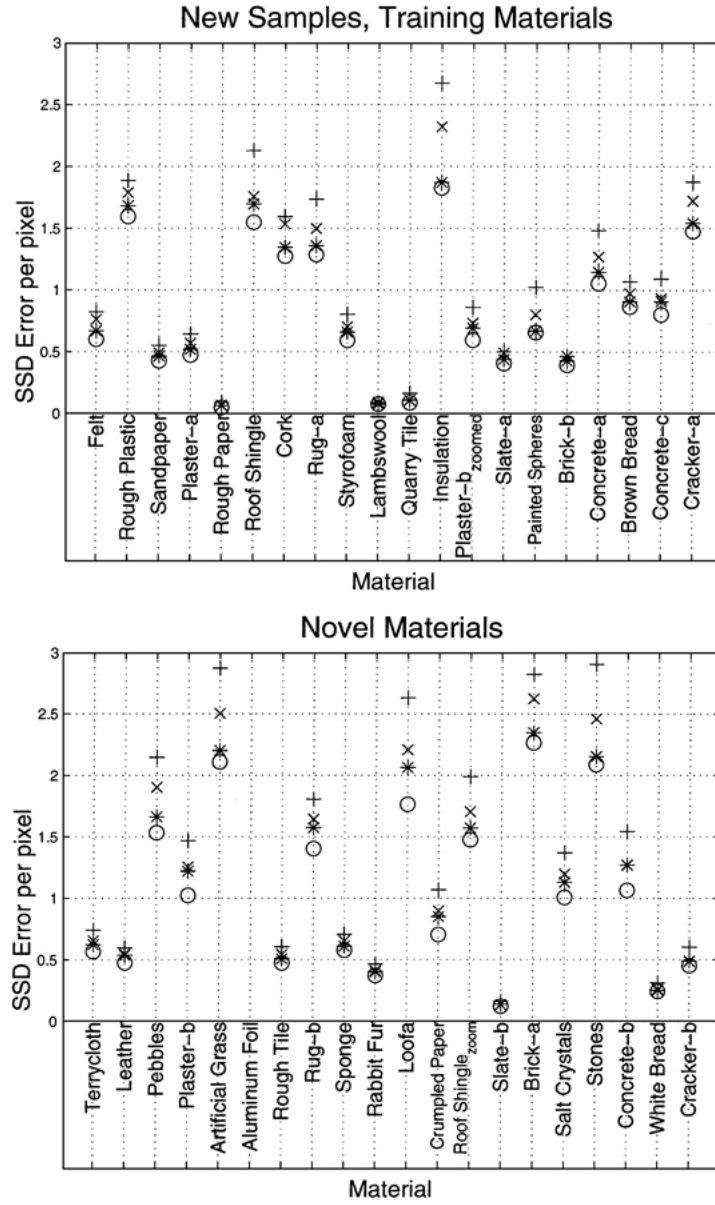

Novel Materials

Figure 7. SSD reconstruction error for different materials. Top: the 20 training materials used to create the texton vocabulary. Bottom: 20 novel materials. Several vocabularies of different sizes are created: "○" for $K = 800$; "∗" for $K = 400$; "×" for $K = 200$; and "+" for $K = 100$. Notice two points about the texton vocabulary: (1) there is no significant difference in average reconstruction error between training and novel materials—good generalization; (2) SSD errors are small for almost all materials—high descriptive power.

chance; and $\nu = \#bins - 1$. $P(\chi^2 \mid \nu)$ is given by the incomplete gamma function (Press et al., 1988):

$$P(\chi^2 \mid \nu) = Q(\nu/2, \chi^2/2) \quad \text{and}$$

$$Q(a, x) = \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} \, dt \tag{2}$$

where $\Gamma(a)$ is the gamma function.

6. Textons versus Principal Component Analysis

The texton representation can be considered as a form of data compression. It is not the only way for compressing data. Principal component analysis (PCA) is one of the most common ones. PCA is in fact a very popular approach for object and texture
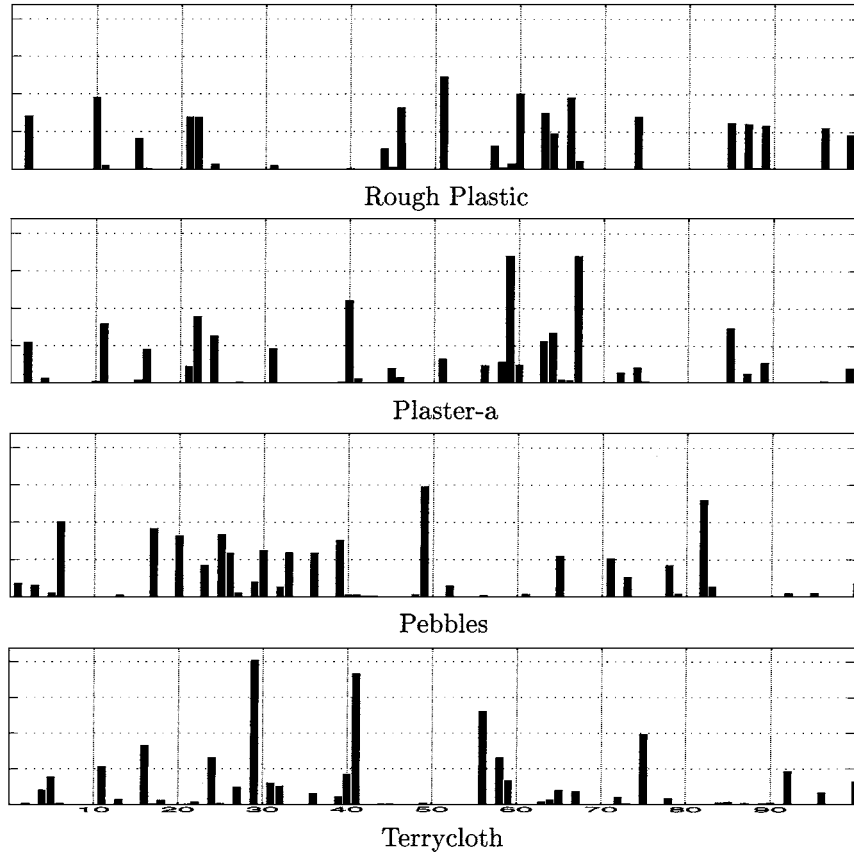
Figure 8. Top to bottom: the histograms of labels for the materials: "Rough Plastic", "Plaster-a", "Pebbles," and "Terrycloth" respectively. These histograms are used as the material representation for the task of texture recognition. The histograms are very different from each other, thus allowing good discrimination.

recognition (Belhumeur and Kriegman, 1998; Dana and Nayar,1999a; Georghiades et al., 1998; Murase and Nayar, 1995; Sirovitch and Kirby, 1987; Turk and Pentland, 1991). In PCA approaches, each image is represented as an element in $\mathbb{R}^K$, a *linear* subspace of the pixel space: $\mathbb{R}^N$ (N is the number of pixels and $K \ll N$). Object models are represented by a collection of example training images. In other words, each object model is a subset of $\mathbb{R}^K$. Classification is done using nearest neighbor, or other more sophisticated classifiers, like support vector machines.

The main problem of applying PCA to represent 3D texture is that PCA is intrinsically linear. The major effects caused by the surface relief of natural materials—shadows, occlusion, specularities, mutual illumination, . . . etc, are non-linear properties. Moreover, PCA is applicable only if the surface reflectance is Lambertian. However, most, if not all, materials are highly non-

Lambertian. Because of all these, we argue that the texton representation, which is not based on any linearity assumption, is more appropriate.

7. Texture Recognition

In this section, we will demonstrate algorithms and results on texture recognition.

7.1. 3D Texture Recognition from Multiple Viewpoint/Lighting Images

We first investigate 3D texture recognition when multiple images of each sample are given. Every time we get a sample of the material, 20 images of different lighting and viewing directions are provided. From these images, a texton labeling is computed. Then the sample is classified to be the material with the smallest

chi-square distance between the sample histogram and the model histogram. In this experiment, 20 training materials are used to construct the texton vocabulary of size 100. 40 different materials are to be classified. The models are obtained from random 100×100 patches from the images. For each material, 3 novel samples of size 100×100 are to be classified. The overall recognition rate is 95.6%.[4]

Another way to demonstrate the result is to use the similarity matrix in Fig. 9. Each element in the matrix e_{ij} is given by the chi-square probability function

(Eq. (2)) that samples of material j will be classified as material i. Here, we only show the probability for 14 materials because of space limitations. As shown in the figure, for example, "Leather" and "Rough Plastic" are likely to be classified correctly; while "Plaster-a" and "Plaster-b" are likely to be mistaken between each other. This is reasonable because the two different types of plaster indeed look very similar, as shown from the images in the bottom of the figure.

"Receiver Operation Characteristics" (ROC) curves are also good indications of the preformance. The ROC

	Felt	Terrycloth	Rough Plastic	Leather	Sandpaper	Pebbles	Plaster-a	Plaster-b	Rough Paper	Artificial Grass	Roof Shingle	Aluminum Foil	Cork	Rough Tile
Felt	**0.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Terrycloth	0.0	**1.0**	0.0	0.0	0.3	0.0	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0
Rough Plastic	0.0	0.0	**0.9**	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Leather	0.2	0.0	0.0	**1.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sandpaper	0.0	0.1	0.0	0.0	**1.0**	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pebbles	0.0	0.0	0.0	0.0	0.0	**1.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
Plaster-a	0.0	0.1	0.2	0.0	0.1	0.0	**1.0**	0.7	0.0	0.0	0.0	0.0	0.0	0.0
Plaster-b	0.0	0.2	0.1	0.0	0.0	0.0	0.8	**1.0**	0.0	0.0	0.0	0.0	0.0	0.0
Rough Paper	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	**0.5**	0.0	0.0	0.0	0.0	0.0
Artificial Grass	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	0.1	0.1	0.0	0.0
Roof Shingle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	**1.0**	0.1	0.0	0.0
Aluminum Foil	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	**1.0**	0.0	0.0
Cork	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	0.2
Rough Tile	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**

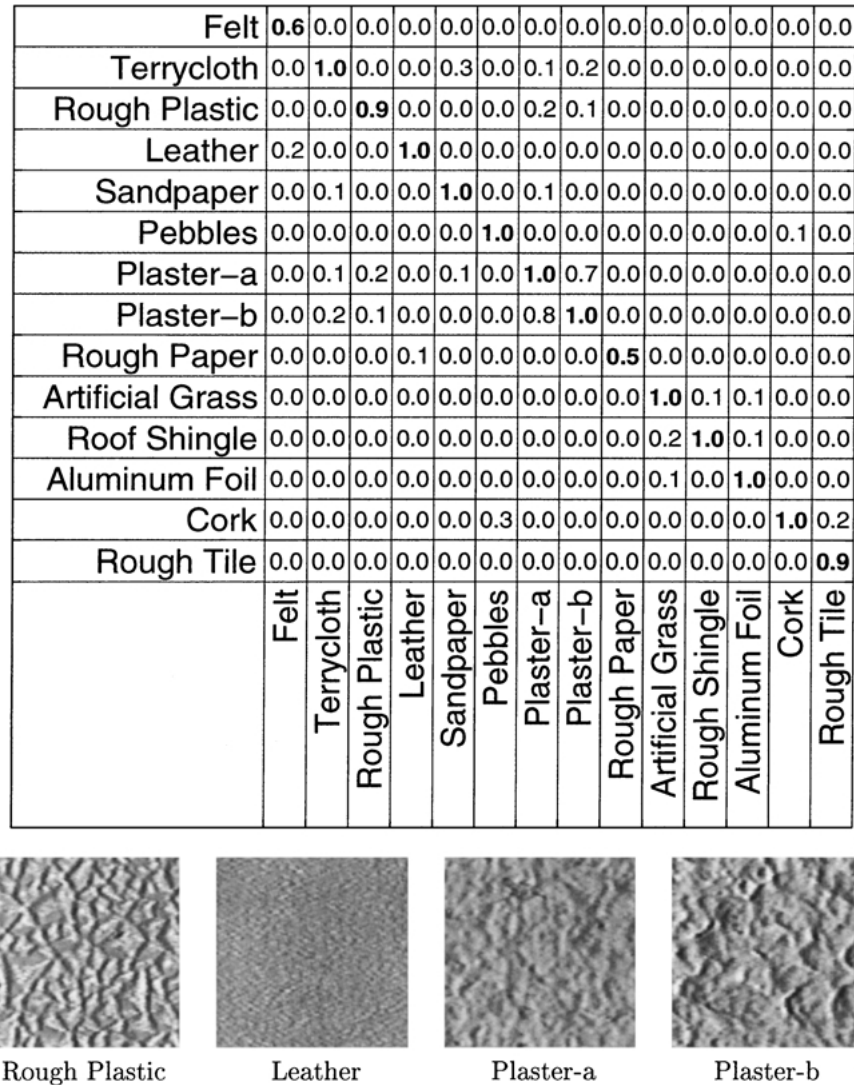Rough Plastic   Leather   Plaster-a   Plaster-b

Figure 9. Similarity matrix for 14 materials. Each entry e_{ij} is given by the chi-square probability function (Eq. (2)) that samples of material j will be classified as material i. As shown in this figure, for example, "Leather" and "Rough Plastic" are likely to be classified correctly; while "Plaster-a" and "Plaster-b" are likely to be mistaken between them. Sample images from these four materials are shown as well.
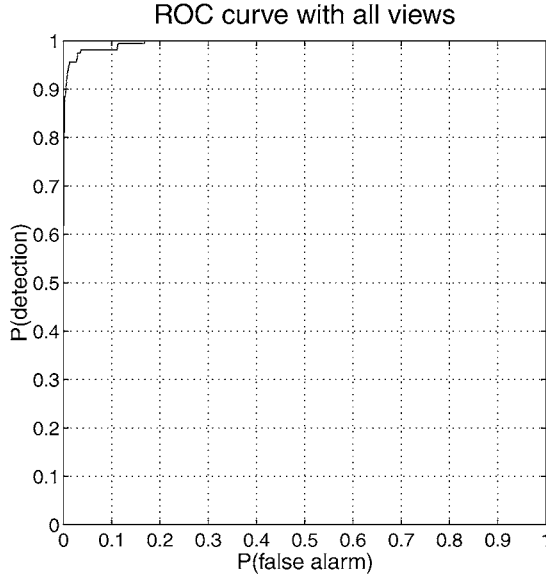
ROC curve with all views

Figure 10. Receiver operation characteristics (ROC) curve for a very simple texture recognition problem. The top-left corner represents perfect recognition performance. The diagonal line refers to chance. The performance for our algorithm is very good. The recognition achieves a 97% detection rate with only a 3% false alarm rate.

curve is a plot of the probability of detection versus the probability of false alarms. It is parametrized by a *detection threshold*. In our case, it is a threshold τ on the chi-square distance. For any incoming sample, we declare that it is the same as material n if the chi-square distance between their histograms is smaller than τ. If the sample is indeed material n, we have a detection, otherwise, it is a false alarm.[5] Figure 10 shows the ROC curve for our recognition problem. The top-left corner represents perfect recognition. Our algorithm performs very well. The recognition performance achieve a 97% detection rate with only a 3% false alarm rate.

The requirement that any material is presented with multiple images at different lighting and viewing conditions may seem unreasonable. However, if the material is on a curved surface, it is essentially equivalent to having multiple images of the same material illuminated and viewed differently.

7.2. *3D Texture Recognition from a Single Image*

Let us now consider the much more difficult problem of 3D texture recognition: for each material, the histogram model is built from 4 different light/view conditions; and for each sample to be classified, we only have a

single image under *any* known illumination condition and viewing geometry. This problem is very similar to the problem formulation of object recognition—given a *small* number of instances of the object, try to recognize it under all poses and illumination.[6] However, in the context of texture recognition, this problem is rarely studied.

Given the illumation and viewing conditions for the novel image, we know to which portion of the appearance vector the filter outputs of the incoming image is to be compared. However, a problem arises from the fact that given only 1 image, finding the texton label for each pixel is very difficult. As noted before, in just one single viewing and lighting condition, physically different features may have the same appearance. Thus, texton assignment to the pixels is ambiguous. Simply commiting to the label with the smallest distance can result in a texton histogram that has no resemblance to that of the target material.

The intuition of our approach is the following: if the texton labeling of the incoming image is known, the material identity can be assigned to the model with the minimum chi-square distance between the incoming texton histogram and the histogram of the model material. On the other hand, if the material identity is known, a texton labeling of the image can be estimated by matching the histograms of the labeling to that of the material. We solve this chicken-and-egg problem using a Markov chain Monte Carlo (MCMC) algorithm. First, each pixel i is allowed N_i possible texton labelings. The MCMC algorithm will try to find the best labelling given the possibilities and the material type.

An MCMC algorithm with metropolis sampling for finding texton labelling is shown below. For each material n and the corresponding model histogram h_n, do:

1. Randomly assign a label to each pixel i among the N_i possibilities. Call this assignment the initial state $x^{(t)}$ with $t = 0$;
2. Compute the probability of the current state $P(x^{(t)})$ using Eq. (2) with h_n as the model histogram;
3. Obtain a tentative new state x' by randomly changing M labels of the current state;
4. Compute $P(x')$ using Eq. (2);
5. Compute $\alpha = \frac{P(x')}{P(x^{(t)})}$;
6. If $\alpha \geq 1$, the new state is accepted, otherwise, accept the new state with probability α;
7. Goto step 2 until the states converge to a stable distribution.
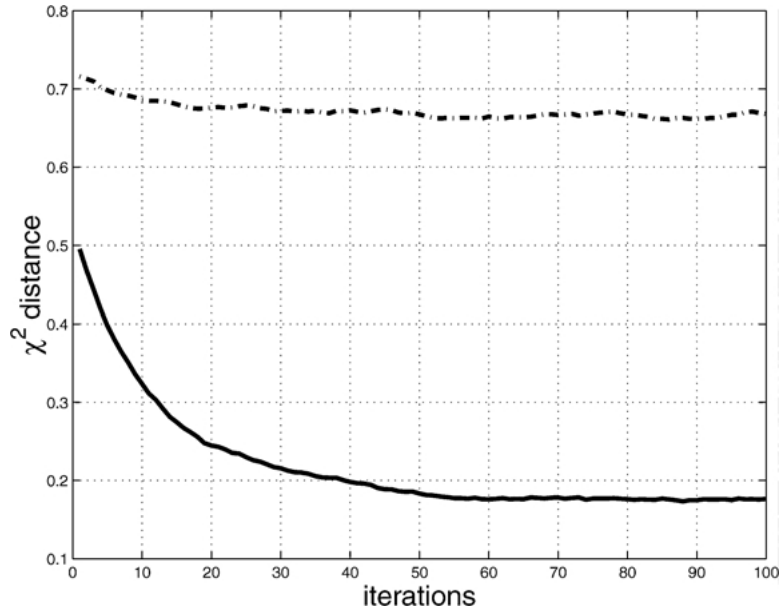
Figure 11. The decay of the χ^2 distance between the histogram of the state $x^{(t)}$ and the histogram of a model material. Solid line: correct material. Dashed line: wrong material. The decay of the distance is much faster and the minimum much smaller for the correct material.

What the MCMC algorithm does is to draw samples from the following distribution: $P(\text{labelling}|\text{material } n)$ or $P(x \mid h_n)$ where x is in the space of possible labellings. $P(x \mid h_n)$ is given by the chi-square probability function in Eq. (2). Once the states settle in a stable distribution, we can compute the probability that the incoming image sample is drawn from material n by computing $\max_t P(x^{(t)} \mid h_n)$.

MCMC algorithms have been applied to computer vision for a long time, most well-known in the paper by Geman and Geman (Geman and Geman, 1984), where the problem of image restoration is studied. For details about variations in MCMC algorithms, convergence properties, and methods to speed up convergence, please consult (Gilks et al., 1996).

In our experiments, each pixel is allowed to have 5 possible labels, chosen from the closest 5 textons. In other words, $N_i = 5 \; \forall i$. For each iteration, we are allowed to change the labels of 5% of the pixels (M in step 3).[7] Figure 11 shows typical behavior of the MCMC algorithm. The solid line is the decay of the χ^2 distance between the histogram of the state $x^{(t)}$ and h_n where material n is the correct material while the dashed line is that for a wrong material.

The recognition performance is shown in the ROC curves in Fig. 12. (These ROC curves are obtained the same way as the one in Section 7.1.) The 5 different
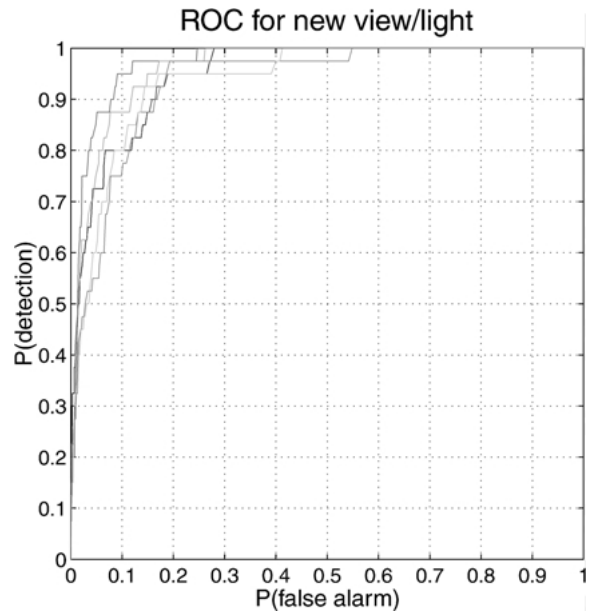

Figure 12. Texture recognition under novel lighting and viewing conditions. The 5 different curves represent 5 randomly chosen novel viewing and lighting directions for the samples to be classified. Each curve is the average performance for 40 materials. The model histogram for each material is obtained using images from 4 different view/light settings. The performance of our algorithm is excellent—a 87% detection rate with 13% false alarm.
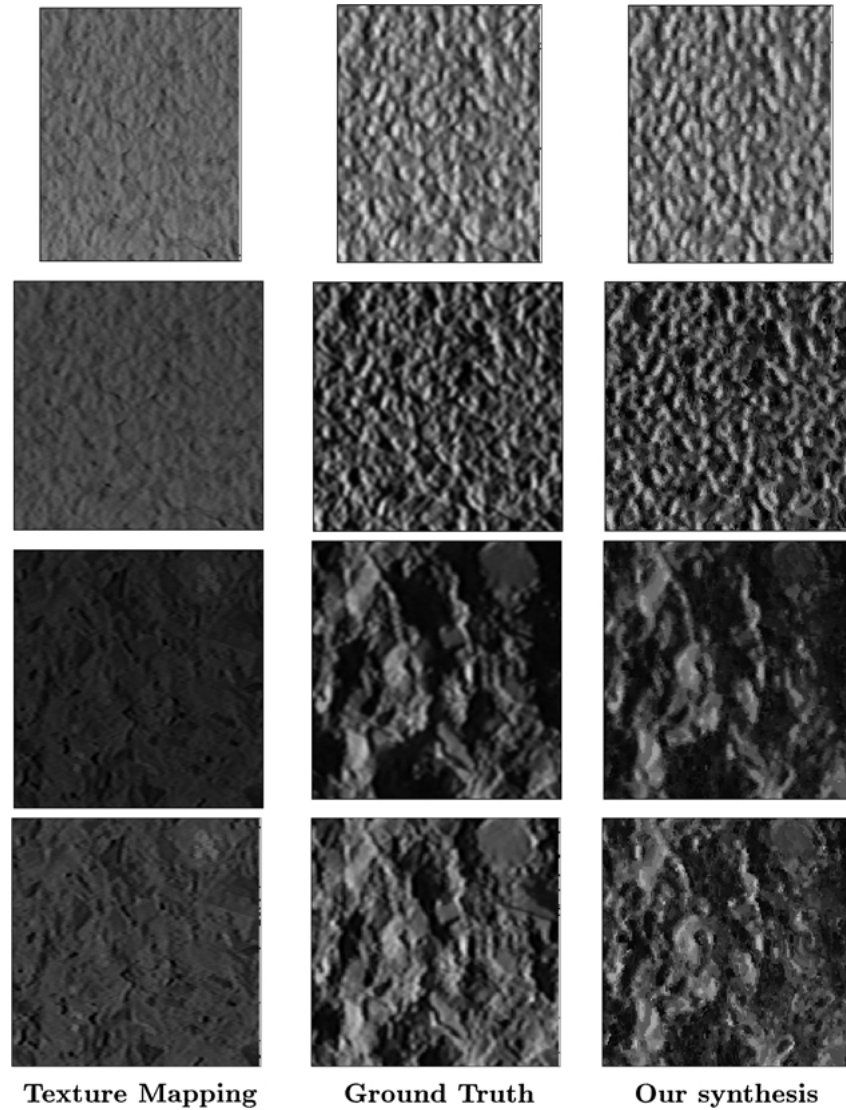
Figure 13. Texture synthesis of training materials used to create the 3D texton vocabulary. The materials are "Plaster-a" for the first two rows and "Concrete-c" for the last two. First column: texture mapping; middle column: ground truth; last column: synthesized results. Texture mapping produces images that look "flat", while our algorithm correctly captures the highlights, shadows, and occlusions.

curves correspond to 5 randomly chosen novel viewing and lighting directions for the samples to be classified. The curves are showing the average performance for 40 materials. The model histogram for each material is obtained using images from 4 different view/light settings. The top-left corner of the plot stands for perfect performance. Given the difficulty of the task, the performance of our algorithm is very good. The algorithm achieves a 87% detection rate with a 13% false alarm rate. One interesting comparison to make will be to contrast the performance of our algorithm with that of a human.

8. Novel View/Light Prediction

The universal 3D texton vocabulary can also be used to predict the appearance of materials at novel viewing and lighting conditions. This application is of primary interest in computer graphics.
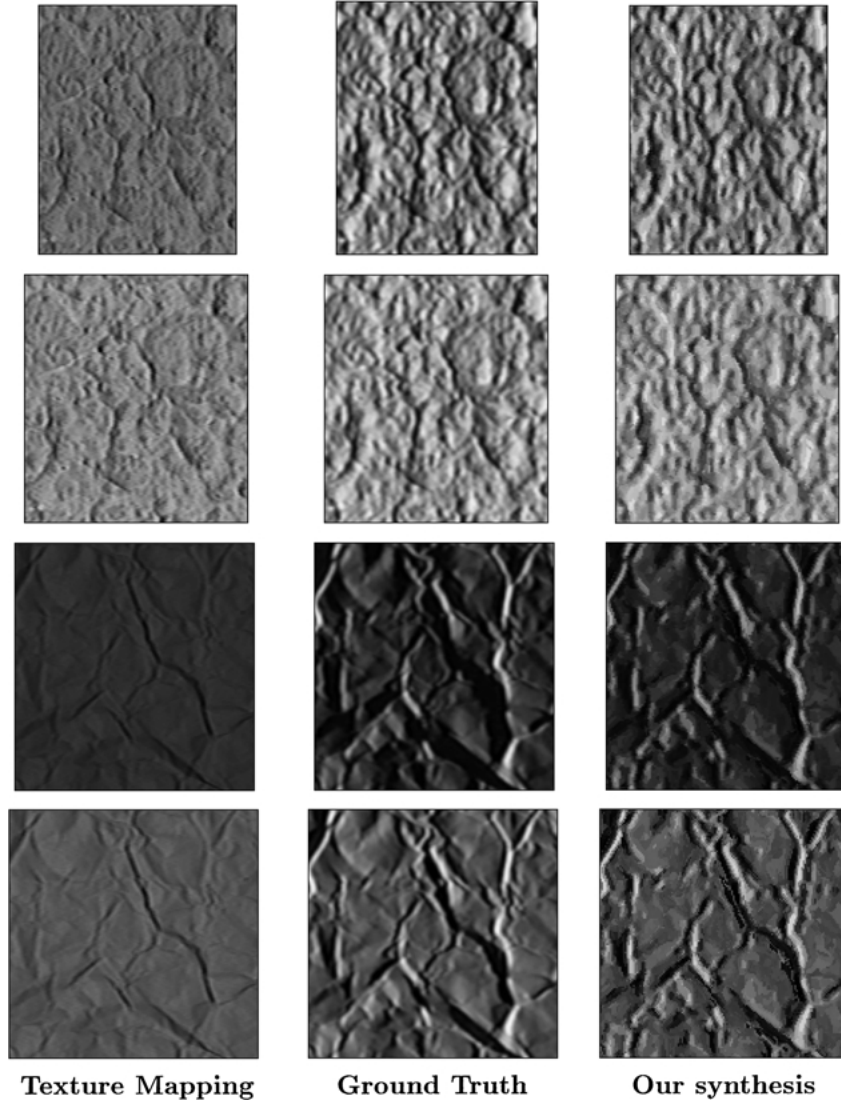
| Texture Mapping | Ground Truth | Our synthesis |

Figure 14. Predicting appearance of novel materials at various lighting and viewing conditions. The materials are "Plaster-a" for the first two rows and "Crumpled Paper" for the last two. First column: traditional texture mapping; middle column: ground truth; last column: results using texton vocabulary. Our algorithm correctly ćaptures the highlights, shadows and occlusions while traditional texture mapping produces images that look "flat".

Suppose we are given n images of a novel texture taken at n different *known* illumination and viewing directions. We compute the filter responses of the images and concatenate them to form a nN_{fil} data vector. These data vector can be labeled to one of the K elements in the texton vocabulary by matching to the corresponding sections of the $N_{vl}N_{fil}$ dimensional appearance vectors. In other words, each pixel in the input texture is labelled to one of the K 3D textons. Recall that the appearance vectors of the 3D textons encode precisely how each texton changes its appearance when illumination or viewing directions are changed. Therefore, we can predict exactly how the image is transformed under a novel illumination direction or viewing geometry.

Results for novel view/light prediction are shown in Figs. 13 and 14. In these examples, 4 images of the material under different light/view arrangements are given. We then predict the appearance of the material at other lighting and viewing conditions using the texton

vocabulary.[8] The novel lighting and viewing configurations are up to $30°$ away from the 4 example conditions. The results shown in Fig. 13 are for training materials (those used to compute the texton vocabulary). Figure 14 shows the results for novel materials. The first columns show images obtained using traditional texture mapping from a frontal parallel image; middle columns show the ground truth and the third columns display our results. Because traditional texture mapping assumes the surface is painted and Lambertian, it produces images that look "flat". Our method, on the other hand, correctly captures the 3D nature of the surface—highlights, shadows, and occlusions.

9. Discussion

In this paper, we have presented a framework for representing textures made up of both reflectance and surface normal variations. The basic idea is to build a universal texton vocabulary that decribes generic local features of texture surfaces. Using the texton vocabulary and an MCMC algorithm, we have demonstrated excellent results for recognizing 3D textures from a single image under any lighting and viewing directions. We also demonstrated how our model can be used to predict the appearance of natural materials under novel illumination condition and viewing geometry.

The current work can be combined with a texture synthesis algorithm to generate *new samples* of materials under all viewing and illumination conditions. The algorithm proposed in Efros and Leung (1999) is particularly promising. The basic idea of Efros and Leung (1999) is to synthesize texture by growing pixels. The pixel value to be grown is obtained by sampling from an example texture image. In our 3D texton model, we can grow a array of textons instead of pixel values. Given the array of textons, we can synthesize an image at *any* illumination and viewing condition. This is done by picking a section of the appearance vectors of the 3D textons and reconstructing an image from them.

Acknowledgment

Notes

1. More images if the material is anisotropic.
2. We recognize that the SSD error is by no means perceptually correct, but it is a convenient way of comparing two images.
3. Error is large for aluminum, which is very specular.
4. Recognition rate is 95.0% for training materials (those used to create the texton vocabulary) and 96.3% for novel materials. There is no significant difference between the performance for the training materials and that of the novel materials in all our experiments. Therefore, we will report only the overall recognition performance. The main reason for this indifference in performance is that the texton vocabulary attains good generalization, thus is encoding generic local features, rather than retaining material-specific information.
5. This is equivalent to making 40 2-class decisions. For example, if the threshold τ is too small, we will have 1 detection and 39 false positives. On the other hand, if τ is too large, we will have 0 detection and 0 false positive.
6. However, most object recognition algorithms require a *large* number of training examples.
7. A cooling schedule can definitely be employed here. At first, more sites are allowed to change to speed up the exploration of the space. When the distribution is close to convergence, fewer sites are allowed to alter to "fine-tune" the distribution.
8. In these results, to achieve best quality and to reduce quantization error, a texton vocabulary of size 2000 is computed.

References

Ball, G. and Hall, D. 1967. A clustering technique for summarizing multi-variate data. *Behavioral Science*, 12:153–155.

Belhumeur, P. and Kriegman, D. 1998. What is the set of images of an object under all possible illumination conditions?. *International Journal of Computer Vision*, 28(3):245–260.

Burt, P. and Adelson, E. 1983. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540.

Chantler, M. 1994. Towards illuminant invariant texture classification. In *Proc. IEE Coll. on Texture Classification: Theory and Applications*.

Chantler, M. and McGunnigle, G. 1995. Compensation of illuminant tilt variation for texture classification. In *Proceedings Fifth International Conference on Image Processing and its Applications*, pp. 767–771.

Chellappa, R. and Chatterjee, S. 1985. Classification of textures using Gaussian Markov random fields. *IEEE Transactions on Acoustics, Speech, Signal Processing*, 33(4):959–963.

Cross, G. and Jain, A. 1983. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39.

Dana, K. and Nayar, S. 1998. Histogram model for 3D textures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pp. 618–624.

Dana, K. and Nayar, S. 1999a. 3D textured surface modelling. In *Proceedings Workshop on the Integration of Appearance and Geometric Methods in Object Recognition*, pp. 46–56.

Dana, K. and Nayar, S. 1999b. Correlation model for 3D texture. In *Proceedings IEEE 7th International Conference on Computer Vision*, Vol. 2. Corfu, Greece, pp. 1061–1066.

Dana, K., van Ginneken, B., Nayar, S., and Koenderink, J. 1999. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34.

de Bonet, J. and Viola, P. 1998. Texture recognition using a non-parametric multi-scale statistical model. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pp. 641–647.

Duda, R. and Hart, P. 1973. *Pattern Classification and Scene Analysis*, John Wiley & Sons. New York, N.Y.

Efros, A. and Leung, T. 1999. Texture synthesis by non-parametric sampling. In *Proceedings IEEE 7th International Conference on Computer Vision*, Vol. 2. Corfu, Greece, pp. 1033–1038.

Fogel, I. and Sagi, D. 1989. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–113.

Geman, S. and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Georghiades, A., Kriegman, D., and Belhumeur, P. 1998. Illumination cones for recognition under variable lightin: Faces. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pp. 52–58.

Gersho, A. and Gray, R. 1992. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers: Boston, MA.

Gilks, W., Richardson, S., and Spiegelhalter, D. 1996. *Markov Chain Monte Carlo in Practice*, Chapman and Hall.

Haddon, J. and Forsyth, D. 1998. Shading primitives: Finding folds and shallow grooves. In *Proceedings IEEE 6th International Conference on Computer Vision*, Bombay, India, pp. 236–241.

Heeger, D. and Bergen, J. 1995. Pyramid-based texture analysis/synthesis. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, Los Angeles, CA, pp. 229–238.

Jain, A. and Farrokhsia, F. 1991. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24:1167–1186.

Jones, D. and Malik, J. 1992. Computational framework to determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699–708.

Julesz, B. 1981. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97.

Koenderink, J. and van Doorn, A. 1980. Photometric invariants related to solid shape. *Optica Acta*, 27(7):981–996.

Koenderink, J. and van Doorn, A. 1996. Illuminance texture due to surface mesostructure. *Journal of the Optical Society America A*, 13(3):452–463.

Koenderink, J., van Doorn, A. Dana, K. and Nayar, S. 1999. Bidirectional reflection distribution function of thoroughly pitted surfaces. *International Journal of Computer Vision*, 31(2/3):129–144.

Leung, T. and Malik, J. 1997. On perpendicular texture or: Why do we see more flowers in the distance?. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 807–813.

Leung, T. and Malik, J. 1999. Recognizing surfaces using three dimensional textons. In *Proc. IEEE International Conference on Computer Vision*, Corfu, Greece.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symposium on Math. Stat. and Prob.*, Vol. I. pp. 281–297.

Malik, J., Belongie, S., Shi, J., and Leung, T. 1999. Textons, contours and regions: Cue integration in image segmentation. In *Proceedings IEEE 7th International Conference on Computer Vision*, Corfu, Greece, pp. 918–925.

Malik, J. and Perona, P. 1990. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7(5):923–932.

Mao, J. and Jain, A. 1992. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188.

Murase, H. and Nayar, S. 1995. Visual learning and recognition of 3-D objects from appearance. *International Journal on Computer Vision*, 14(1):5–24.

Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. 1988. *Numerical Recipes in C*, Cambridge University Press.

Puzicha, J., Hofmann, T., and Buhmann, J. 1997. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 267–272.

Ripley, B. 1996. *Pattern Recognition and Neural Networks*, Cambridge University Press.

Rubner, Y. and Tomasi, C. 1999. Texture-based image retrieval without segmentation. In *Proceedings IEEE 7th International Conference on Computer Vision*, Vol. 2. Corfu, Greece, pp. 1018–1024.

Sebestyen, G. 1962. Pattern recognition by an adaptive process of sample set construction. *IRE Trans. Info. Theory*, 8:S82–S91.

Shashua, A. 1997. On photometric issues in 3D visual recognition from a single 2D image. *International Journal on Computer Vision*, 21(1/2).

Sirovitch, L. and Kirby, M. 1987. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 2:519–524.

Turk, M. and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Vaidyanathan, P. 1993. *Multirate Systems and Filter Banks*, Prentice-Hall: Englewood Cliffs, N.J.

van Ginneken, B., Stavridi, M., and Koenderink, J. 1998. Diffuse and specular reflectance from rough surfaces. *Applied Optics*, 37(1):130–139.

Yuan, J. and Rao, S. 1993. Spectral estimation for random fields with applications to Markov modeling and texture classification. In *Markov Random Fields: Theory and Application*, R. Chellappa and A. Jain (Eds.). Academic Press.

Zhu, S., Wu, Y., and Mumford, D. 1998. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126.