IEEE International Conference on Computer Vision, Corfu, Greece, September 1999

Recognizing surfaces using three-dimensional textons

Thomas Leung and Jitendra Malik
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720
{leungt,malik}@cs.berkeley.edu

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.*

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. Some representative techniques include Markov random fields [2] and filter responses [6, 16]. In all these work, surface normal variations are ignored. However, nature shows an abundance of such relief textures. Examples are shown in Figure 1. Notice

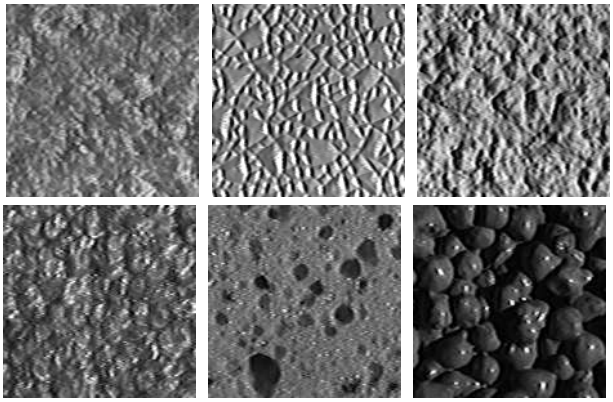


Figure 1. Some natural 3D textures from the Columbia-Utrecht database [4]. Left to right: "Terrycloth", "Rough Plastic", "Plaster-b", "Rug-a", "Sponge", "Painted Spheres". These textures illustrate the problems caused by the 3D nature of the material: *specularities*, *shadows* and *occlusions*.

in particular the effect 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 views. 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. 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 problems: two regions will have the same intensity under one illumination; while the shadowed region will be darker in another.

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 of 3D textures [3, 4, 13, 14, 20]. However, the

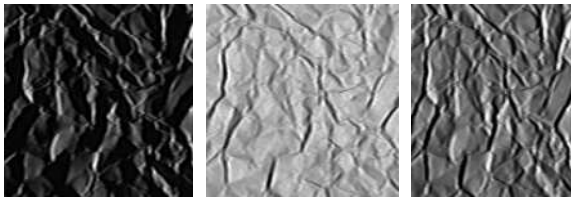


Figure 2. The same patch of the material "Crumpled Paper" imaged under three different lighting and viewing conditions. The aspect ratio of the figure is determined by the slant of the surface. Even though the three images are corresponding patches from the same material, the appearances are drastically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

problem of texture recognition under varying lighting and viewing conditions has not yet been addressed.

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 more than 20 years ago. Julesz's textons [12] — orientation elements, crossings and terminators — fell 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, 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 viewing 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. The problem of 3D texture recognition is presented in Section 6 and results are shown for classifying materials under novel viewing and lighting conditions. In Section 7, 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 8.

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 [6, 16], synthesis [10], as well as recognition [18].

Though the representation of textures using filter responses is extremely versatile, one might say that it is overly redundant (each pixel values 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 Figure 3. There are a total of 48 filters. Each pixel is now transformed to a $N_{fil} = 48$ dimensional vector. These vectors are clustered using a K-means algorithm [5]. 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 will achieve a local minimum of this criterion.

K-means is a vector quantization algorithm [8]. A useful way to evaluate such algorithms is to compare the original image with the quantized image. Figure 4 shows such comparisons. The original image is shown in (a). The cluster centers are visualized in terms of the original filter kernels in (b). This is done by premultiplying the vectors representing the cluster centers by the pseudoinverse of the filterbank [11]. The reconstructed image after quantization is shown in (c). The close resemblance between (a) and (c) suggests that the quantization does not introduce much error perceptually.

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 are referred to [15], where we applied the idea of textons to the problem of image segmentation using multiple cues.

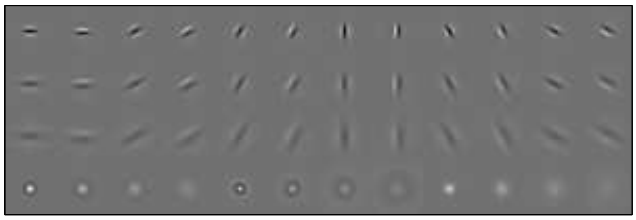
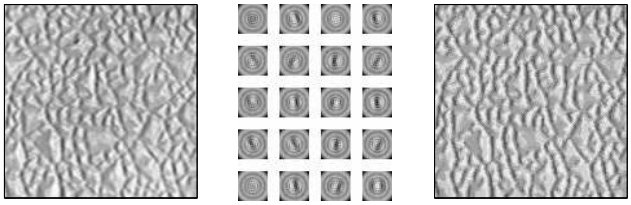


Figure 3. The filter bank used in our analysis. Total of 48 filters: 36 oriented filters, with 6 orientations, 3 scales and 2 phases; 8 center surround derivative filters and 4 low-pass Gaussian filters.



(a) (b) (c)

Figure 4. Illustration of K-means clustering and reconstruction from filter responses with $K = 20$. (a) Original image. (b) the K-means centers reconstructed as a local filter. These centers mainly 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.

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 (Figure 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. 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 $N_{vl} \gg 1$[1]. 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 [1, 19]. 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 light 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 all the images. 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, grooves of some width, etc.. Similarly, the K-means centers will also encode albedo change vs. geometric 3D features, as well as materials of different 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 filter response vector, the appearance vector.

4 Constructing the Vocabulary of 3D Textons

Our goal in this paper is to use images from a set of materials (the training materials) to learn a vocabulary which characterizes all 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 bound-

[1] $N_{vl} = 20$ in our experiments.

aries 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 [4][2]. There are 60 different materials, each with 205 images at different viewing and lighting angles[3]. 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[4].
2. For each of the 20 materials individually, the K-means clustering algorithm is applied to the data vectors. The number of centers, denoted as K, for this step is 400. The K-means algorithm is initialized by random samples from the image.
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 those centers with too few data assigned to them[5].
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.

A comparison of the texton vocabularies of different sizes are shown in Figure 5. 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. The SSD error between the reconstructed image and the original image is plotted in the figure[6]. The first diagram is the error for new samples of the training materials[7]. The lower diagram is for novel materials. Notice three points: (1) there is no significant difference in the reconstruction error between training materials and novel

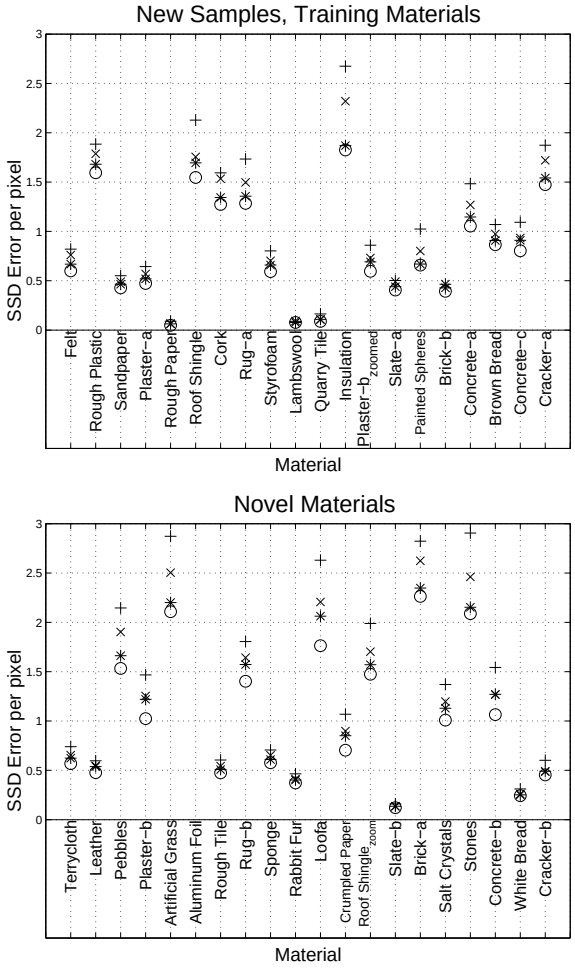


Figure 5. SSD reconstruction error for different materials. Top: the 20 materials used to create the texton vocabulary. Bottom: 20 novel materials. Several vocabularies of different sizes are created: "o" for $K = 800$; "*" for $K = 400$; "+" for $K = 200$ and "+" for $K = 100$.

materials. In other words, our texton vocabulary is encoding generic features, rather than material-specific properties. This is due to the small number of 3D textons allowed in our vocabulary. (2) The SSD errors are small for almost all materials. 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. (3) 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.

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

[2] `http://www.cs.columbia.edu/CAVE/curet/`

[3] More images if the material is anisotropic.

[4] In our experiments, $N_{fil} = 48$.

[5] For the comparisons in Figure 5, $K = 800, 400$ and 200 as well.

[6] We recognize that the SSD error is by no means perceptually correct, but it is a convenient way of comparing two images.

[7] Note that these pixels are not in the training set, though they are from the materials used to construct the texton vocabulary.

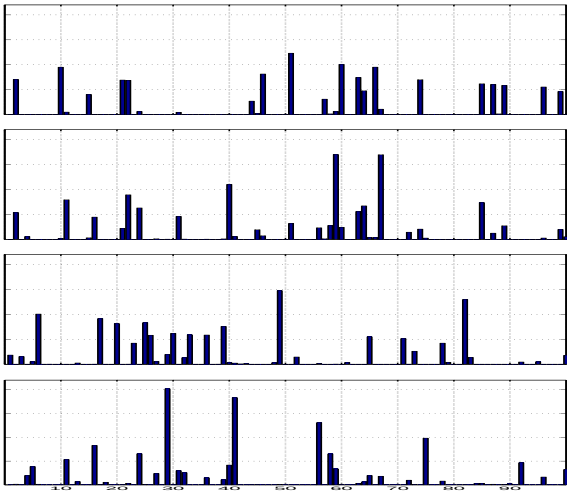


Figure 6. Top to bottom: the histograms of labels for the materials: "Rough Plastic", "Plaster-a", "Pebbles" and "Terrycloth" respectively. These histograms are the representation for recognizing the different textures.

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 Model Acquisition

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. In this paper, 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 Figure 6. 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):

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

The significance for a certain chi-square distance is given by the chi-square probability function: $P(\chi^2|\nu)$. $P(\chi^2|\nu)$ is the probability that two histograms from the same model will have a distance larger than χ^2 by chance; and $\nu = \#bins - 1$. $P(\chi^2|\nu)$ is given by the incomplete gamma function [17]:

$$\begin{aligned} P(\chi^2|\nu) &= Q(\nu/2, \chi^2/2) \qquad (2) \\ \text{and} \quad Q(a,x) &= \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt \end{aligned}$$

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

6 Texture recognition

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

6.1 3D Texture Recognition from Multiple Viewpoint/Lighting Images

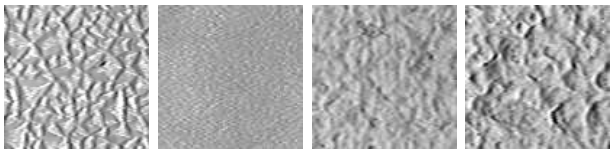
We first investigate 3D texture recognition when multiple images of each sample are given. Every time we get a sample of the material, 20 images of different lighting and viewing directions are provided. From these images, a texton labeling is computed. Then the sample is classified to be the material with the smallest chi-square distance between the sample histogram and the model histogram. For each material, 4 disjoint samples of size 100×100 are to be classified. The overall recognition rate is 95.6%[8].

Another way to demonstrate the result is to use the classification matrix in Figure 7. Each element in the matrix e_{ij} is given by the chi-square probability function (Equation 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.

"Receiver Operation Characteristics" (ROC) curves are also good indications of the preformance . The ROC 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 on the chi-square distance (τ). 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. Figure 8 shows the ROC curve for our recognition problem. The top-left corner represents perfect recognition. Our algorithm performs very well.

[8] Recognition rate is 95.0% for new samples of materials 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 is encoding generic local features, rather than retaining material-specific information.

	Felt	Terrycloth	Rough Plastic	Leather	Sandpaper	Pebbles	Plaster−a	Plaster−b	Rough Paper	Artificial Grass	Rough 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	0.2	**1.0**	0.1	0.0	0.0	0.0
Aluminum Foil	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	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 7. Classification matrix for 14 materials. Each entry e_{ij} is given by the chi-square probability function (Equation 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.

6.2 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* light/view condition. 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. However, in the context of texture recognition, this problem is rarely studied.

A problem now arises in 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, trying to assign a texton label to the pixels from just one image is ambiguous. Doing it by simply assigning to the label with the smallest distance can result in a texton histogram that has no resemblance to that of the target material.

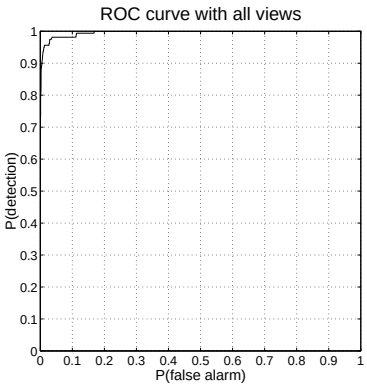


Figure 8. Receiver operation characteristics (ROC) curve for a very simple texture recognition problem. The top-left corner represents perfect recognition performance. The diagonal line refers to chance. The performance for our algorithm is very good.

We solve this problem of finding a labelling using a Markov chain Monte Carlo (MCMC) algorithm. Instead of giving each pixel a single label in the texton vocabulary, we allow each pixel i to have N_i possibilities at first. The MCMC algorithm will try to find the best labelling given the possibilties 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 Equation 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 $P(x')$ with Equation 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.

What the MCMC algorithm does is to draw samples from the following distribution: $P(\text{labelling}|\text{material } n)$ or $P(x|h_n)$ where x is in the space of possible labellings. $P(x|h_n)$ is given by the chi-square probability function in Equation 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 $\max_t P(x^{(t)}|h_n)$.

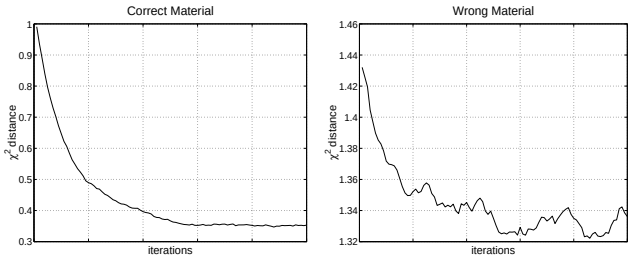


Figure 9. Left: the decay of the χ^2 distance between the histogram of the state $x^{(t)}$ and the histogram of the correct material. Right: same for a wrong material. For the correct material, the decay of the distance is much faster and the minimum distance is much smaller. Notice that the y-axes are at different ranges.

MCMC algorithms have been applied to computer vision for a long time, most well-known in the paper by Geman and Geman [7], 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 [9].

In our experiments, each pixel is allowed to have 5 possible labels, chosen from the closest 5 textons. In other words, $N_i = 5\ \forall i$. Each iteration, we are allowed to change 5% of the size of the image (M in step 3)[9]. Figure 9 shows typical behavior of the MCMC algorithm. Shown on the left is the chi-square distance between the histogram of the state $x^{(t)}$ and h_n where material n is the correct material, while that for a wrong material is shown on the right. For the correct material, the decay of the distance is much faster and the minimum distance is much smaller (the y-axes are at different ranges).

The recognition performance is shown in the ROC curves in Figure 10. The 5 different curves represent 5 randomly chosen novel viewing and lighting directions for the samples to be classified. 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. The performance of our algorithm is excellent. Given the difficulty of the task, one interesting comparison to make will be to contrast the performance of our algorithm and that of a human.

7 Novel View/Light Prediction

The universal 3D texton vocabulary can also be used to predict the appearance of materials at a novel viewing and lighting conditions. This application is of primary interest in computer graphics.

[9] A cooling schedule can definitely be employed here. At first, more sites are allowed to change to speed up exploration of the space. When the distribution is close to convergence, fewer sites are allowed to alter to "fine-tune" the distribution.

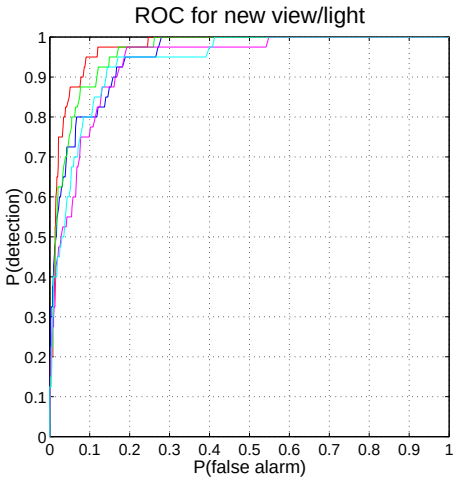


Figure 10. Texture recognition under novel lighting and viewing conditions. The 5 different curves represent 5 randomly chosen novel viewing and lighting directions for the samples to be classified. The model histogram for each material is obtained using images from 4 different view/light settings. The performance of our algorithm is excellent.

Consider one image of a novel texture at a particular lighting and viewing angle. We label the filter response vector at each pixel to one of the K elements in the texton vocabulary. In other words, each pixel in the input texture is labelled as being one of the K textons. Since we know exactly how each texton changes its appearance under a new lighting and viewing direction through the appearance vector, the appearance of the input image at a different viewing and lighting arrangement can be computed readily. If we have more images of the incoming texture (say 4), the labels can be computed using all these images.

Results for novel view/light prediction are shown in Figure 11. 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. The results shown are for novel materials—those not used to construct the texton vocabulary. The first column shows images obtained using traditional texture mapping; middle column shows the ground truth and the last column displays 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.

8 Discussion

In this paper, we have presented a framework for recognizing textures made up of both reflectance and surface normal variations. The basic idea is to build a universal tex-

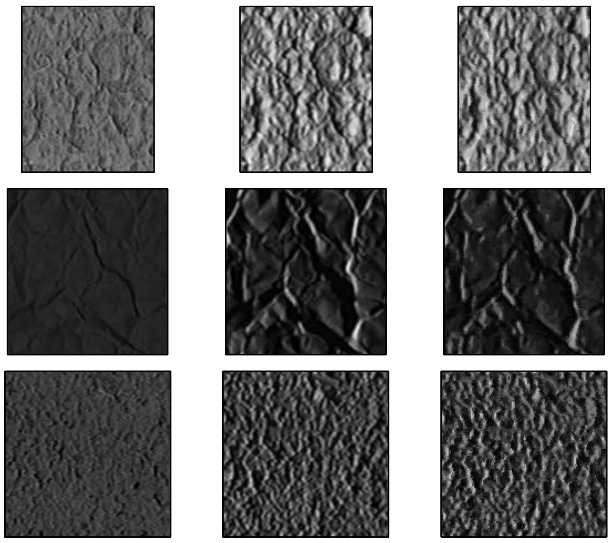


Figure 11. Predicting appearance of novel materials at various lighting and viewing conditions. First column: traditional texture mapping; middle column: ground truth; last column: results using texton vocabulary. Our algorithm correctly captures the highlights, shadows and occlusions while traditional texture mapping produces images that look "flat".

ton 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.

Our model also enables us to recognize curved texture surfaces. The curved surface essentially provide multiple views and light directions in one image. Since our model for each material is invariant to light source direction or viewpoint, such curved surfaces can be handled the same way.

Acknowledgement

This research was supported by (ARO) DAAH04-96-1-0341, the Digital Library Grant IRI-9411334, and a Berkeley Fellowship to TL.

References

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

[2] R. Chellappa and S. Chatterjee. Classification of textures using gaussian markov random fields. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-33, 1985.

[3] K. Dana and S. Nayar. Histogram model for 3d textures. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 618–24, Santa Barbara, CA, June 1998.

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

[5] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[6] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–13, 1989.

[7] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pat. Ana. Mach. Int.*, 6:721–41, 1984.

[8] A. Gersho and R. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, 1992.

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

[10] D. Heeger and J. Bergen. Pyramid-based texture analysis/synthesis. In *Computer Graphics Proceedings, SIGGRAPH 95*, pages 229–38, Los Angeles, CA, Aug. 1995.

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

[12] B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–7, March 1981.

[13] J. Koenderink and A. van Doorn. Illuminance texture due to surface mesostructure. *J. Optical Soc. Am. A*, 13(3):452–63, March 1996.

[14] T. Leung and J. Malik. On perpendicular texture or: Why do we see more flowers in the distance? In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 807–13, San Juan, Puerto Rico, June 1997.

[15] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: cue integration in image segmentation. In *Proc. IEEE Intl. Conf. Computer Vision*, Corfu, Greece, Sept. 1999.

[16] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. America A*, 7(5):923–32, 1990.

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

[18] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 267–72, San Juan, Puerto Rico, 1997.

[19] A. Shashua. On photometric issues in 3d visual recognition from a single 2d image. *Int. J. Computer Vision*, 21(1-2), 1997.

[20] B. van Ginneken, M. Stavridi, and J. Koenderink. Diffuse and specular reflectance from rough surfaces. *Applied Optics*, 37(1):130–9, January 1998.