3D Textured Surface Modeling

Kristin J. Dana and Shree K. Nayar
Computer Science Department
Columbia University
New York, NY 10027

Abstract

Because object geometry varies at many scales, it is often convenient to distinguish shape from texture. While shape is a deterministic macroscopic description, texture is a finer scale geometric description with some repetitive or random component. The distinction between texture and shape is important when developing object recognition systems. Acquiring fine scale geometry is difficult due to local occlusions and limited resolution of imaging systems. Also, the complexities in geometric detail make geometric modeling of texture particularly challenging. Consequently, appearance is a convenient description for surfaces. A useful integration may entail geometry-based recognition for shape and appearance-based recognition for surface detail. We discuss our work on investigating and modeling surface detail using the framework of the BRDF (bidirectional reflectance distribution function) and the BTF (bidirectional texture function).

1 Introduction

In real world scenes, the interplay of light with the complexities of object geometry produce our visual experience. Object geometry varies at many scales, so it is often convenient to distinguish shape from texture. While shape is a deterministic macroscopic description, texture is a finer scale geometric description with some repetitive or random component. For example, a tree trunk has an approximately cylindrical shape, but the trunk's surface has a complex fine-scale geometry from the covering bark. We use the word texture to describe the surface's fine-scale geometry. To be more precise, since the word texture has several meanings, we can make the distinction between 2D texture and 3D texture. A color or albedo variation on a smooth surface is termed 2D texture, while a fine-scale surface height variation is termed 3D texture. Examples of 2D texture are a checkerboard pattern or a leopard print; examples of 3D texture are grass, foliage, gravel and any rough surface.

The distinction between fine-scale geometry and macroscopic geometry, i.e. the distinction between 3D-texture and shape is important when developing object recognition systems. Acquiring fine scale geometry is difficult due to local occlusions and the resolution of the imaging system. Also, the complexities in geometric detail make geometric modeling of 3D texture particularly challenging. Consequently, appearance is a convenient description for surfaces. The appearance of 3D-textured surfaces varies dramatically with viewing and illumination direction as exemplified in Figure 1, so a robust recognition system should incorporate models of surface detail. A useful integration may entail geometry-based recognition for shape and appearance-based recognition for surface detail.

In this paper, we discuss our work on investigating 3D-textured surfaces. The work characterizes surface appearance using the taxonomy of the BRDF (bidirectional reflectance distribution function) and the BTF (bidirectional texture function). In Section 2, we discuss our measurements of the BRDF and BTF which comprise the CUReT database [6][7]. In Section 3, we discuss existing models of the BRDF and model fits using the CUReT data. Models of the BTF are relatively new to the literature, and in Section 4 we discuss our recent work in BTF modeling. Application of these models in texture synthesis and recognition are discussed in Section 5. We conclude with implications for combining geometry-based and appearance-based recognition systems.

2 CUReT Database

Characterizing the appearance of real-world textured surfaces is a fundamental problem in computer vision and computer graphics. Appearance depends on view, illumination and the scale at which the texture

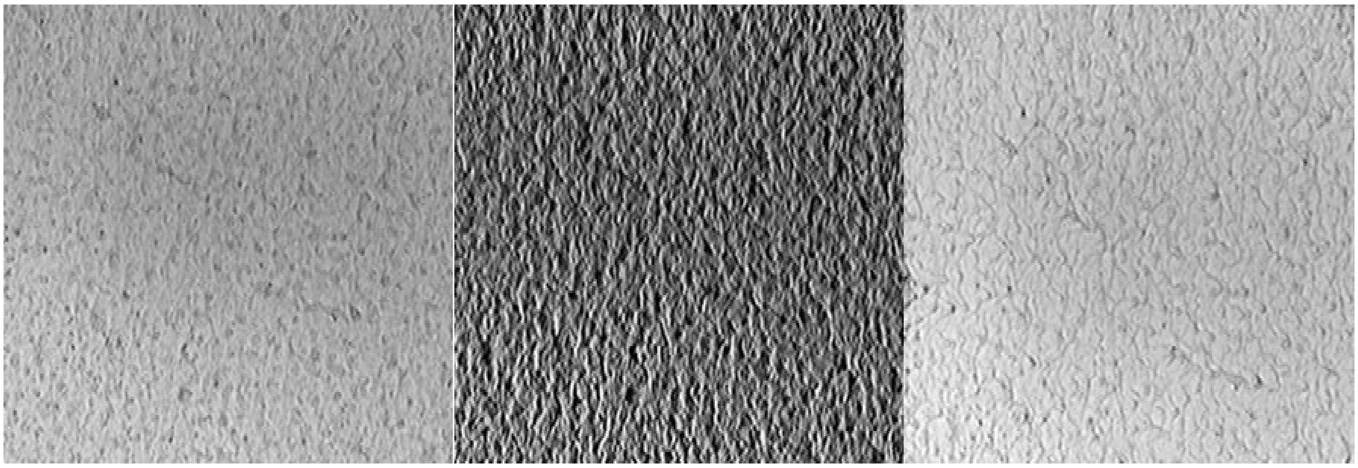

Figure 1. Three images of the same plaster surface under different illumination and viewing directions.

is observed. At coarse scale, where local surface variations are subpixel and local intensity is uniform, appearance is characterized by the BRDF (bidirectional reflectance distribution function). At fine scale, where the surface variations give rise to local intensity variations, appearance can be characterized by the BTF (bidirectional texture function). As a direct analogy to the BRDF, we introduce the term BTF to describe the appearance of texture under varying illumination and viewing directions. Specifically, the BTF is image texture as a function of the illumination and viewing angles.

The goal of the CUReT database is to provide a description both empirically and parametrically of a large and diverse set of common surfaces. We present a BTF database with image textures from over 60 different samples (see Figure 2), each observed with over 200 different combinations of viewing and illumination directions. The measurement methods involved in the BTF database are conducive to a simultaneous measurement of the BRDF. Accordingly, we also present a BRDF database with reflectance measurements for over 60 different samples, each observed with over 200 different combinations of viewing and illumination directions. A robotic manipulator and CCD camera are employed in the measurement procedure to allow simultaneous measurement of the BTF and the BRDF of large samples (about 10x10cm). Both of these unique databases are publicly available (www.cs.columbia.edu/CAVE/curet) and have important implications for computer vision and graphics. Studying the dependence of 3D texture on viewing and illumination directions is fairly new in texture research [13][27][6][7] and with the availability of this database there is an increased attention to the topic in the recent literature [4][28][32][5].

3 BRDF Models

Although BRDF models have been widely discussed and used in computer vision and computer graphics [35][31][25][34][17][30][36][13][21], the BRDFs of a large and diverse collection of real-world surfaces have never before been available. Our measurements comprise a comprehensive BTF/BRDF database (the first of its kind) that is publicly available. Exactly how well the BRDFs of real-world surfaces fit existing models has remained unknown as each model is typically verified using a small number (2 to 6) of surfaces. Our large database allows researchers to evaluate the performance of existing and future BRDF models and representations. As part of the CUReT project we fit the BRDF measurements to two existing analytical representations: the Oren-Nayar model [21][19] for surfaces with isotropic roughness and the Koenderink et al. decomposition [13] for both anisotropic and isotropic surfaces. The Oren-Nayar model is a physical model which represents the surface as a collection of randomly oriented V-grooves. The Koenderink et al. representation is a decomposition of the BRDF onto a set of basis functions on the unit hemisphere. The fitting results provide a concise description that is required for functional utility of the measurements. These BRDF parameters are publicly available and can be used directly and conveniently in a variety of algorithms where accurate, concise and analytical reflectance descriptions are needed. Accordingly, the results are pertinent to a variety of areas including remote-sensing, photogrammetry, image understanding and scene rendering.

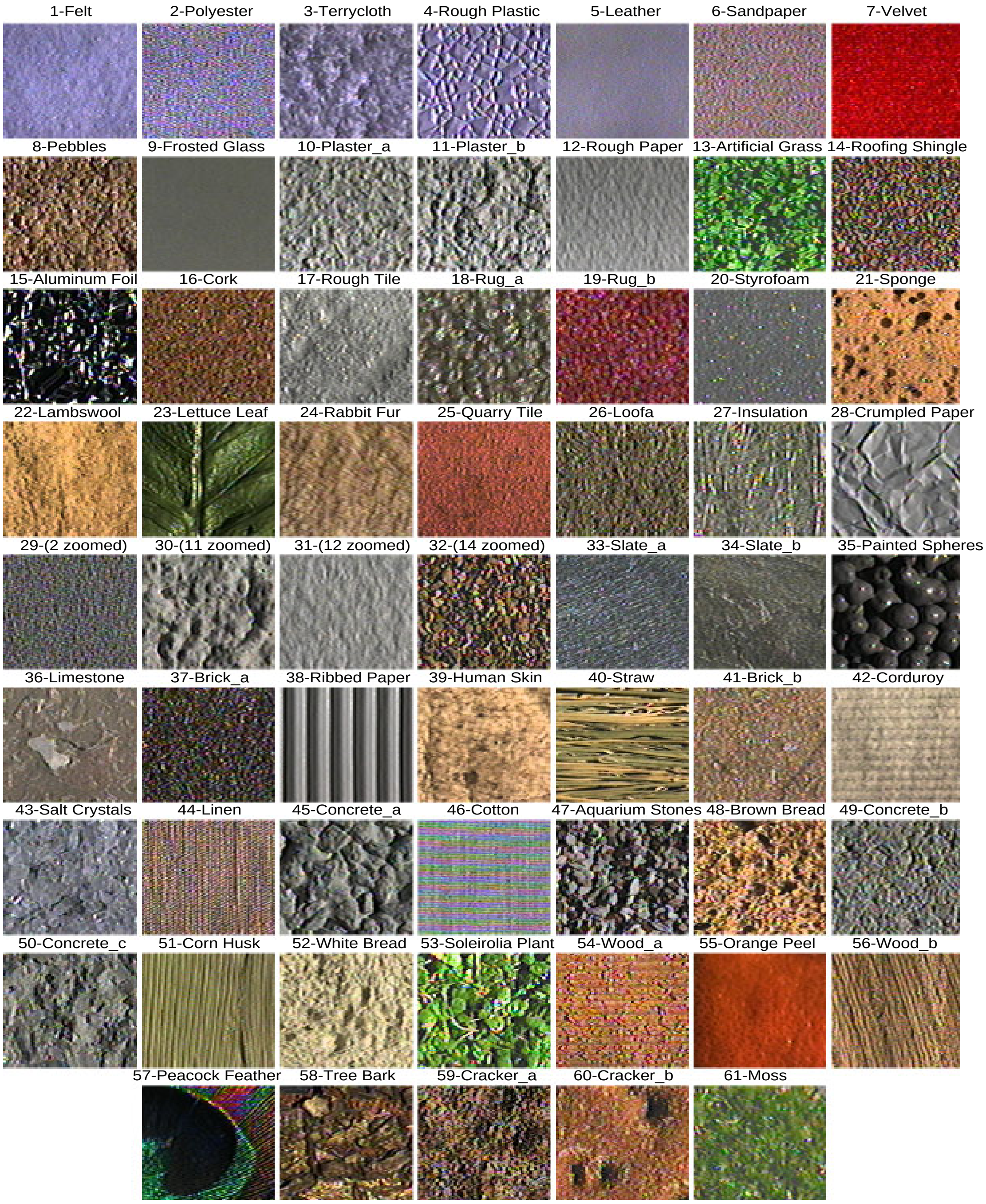

Figure 2. The collection of 61 real-world surfaces used in the measurements. The name and number of each sample is indicated above its image. The samples were chosen to span a wide range of geometric and photometric properties. The categories include specular surfaces (aluminum foil, artificial grass), diffuse surfaces (plaster, concrete), isotropic surfaces (cork, leather, styrofoam), anisotropic surfaces (straw, corduroy, corn husk), surfaces with large height variations (crumpled paper, terrycloth, pebbles), surfaces with small height variations (sandpaper, quarry tile, brick), pastel surfaces (paper, cotton), colored surfaces (velvet, rug), natural surfaces (moss, lettuce, fur) and man-made surfaces (sponge, terrycloth, velvet). Different samples of the same type of surfaces are denoted by letters, e.g. Brick_a and Brick_b. Samples 29, 30, 31 and 32 are close-up views of samples 2, 11, 12 and 14, respectively.

4 BTF Models

4.1 Histogram Model

Modeling and synthesizing the bidirectional texture function is key to achieving robust texture recognition and segmentation as well as realistic texture rendering. A fundamental representation of texture is the histogram of pixel intensities. For 3D texture, just as image texture is bidirectional, the histogram is a bidirectional histogram. Changes in the histogram of 3D texture with illumination and viewing directions are indicative of the surface structure. The work of [32] also addresses histograms of 3D texture by investigating the physical mechanisms underlying bidirectional histograms from a large variety of surfaces and by using statistical simulations to generate histograms of gaussian rough surfaces. In our recent work [4], we develop an analytical model of the bidirectional histogram of image texture. For arbitrary surfaces, developing such a model is extremely difficult. So, for tractability, we assume the imaged surface has an isotropic random-slope profile and constant-albedo Lambertian reflectance. This model proves to be a good approximation for a variety of natural and manmade surfaces found in ordinary scenes. Our model is based on a geometric/photometric analysis of the interaction of light with the surface. We show the accuracy of the model by fitting to the histograms of real 3D textures from the CUReT texture database. The model can be used in applications for both computer vision and computer graphics. The parameters obtained from the model fits are roughness measures which can be used in texture recognition schemes. In Section 5, we show how this histogram model can be applied in texture synthesis.

There are two main tasks involved with the development of the histogram model. The first task is the conversion of the surface slope probability density function (pdf) to the image intensity pdf. This conversion requires a careful analysis of surface masking, shadowing and shading that results in an expression for the image intensity pdf in integral form. This integral is rather complicated and does not lend itself to an analytical solution. The second task of the model development is a suitable approximation of the integral that leads to a concise parametric histogram representation. This approximation is done using coordinate transformation and a basis decomposition using the Discrete Cosine Transform. The resulting expression for the probability of a particular intensity $p\left(I_0\right) dI_o$ is

$$p_I\left(I_o\right) dI_o = \frac{q}{q_v F} \sum_{j=1}^{L} \alpha_j \sum_i \kappa_j^i \mu^i, \qquad (1)$$

where L is the number of basis polynomials, α_j are the coefficients obtained when projecting the integrand onto the basis polynomials, and $\kappa_j^i \mu^i$ are the functions resulting from the integration of the basis polynomials. The details are described in [4].

In matrix notation this becomes

$$p_I\left(I_o\right) dI_o = \frac{q}{q_v F} \left(\kappa\mu\right)^T \alpha. \qquad (2)$$

where κ is an $L \times (2L-1)$ matrix which depends on S (source direction); μ is a $(2L-1) \times 1$ vector which depends on V (viewing direction) and S; α is an $L \times 1$ vector which depends on σ (surface roughness); and $\left(\frac{q}{q_v F}\right)$ is a constant which depends on V, S and the surface roughness σ.

Note that $p_I\left(I_o\right) dI_o$ is the value of the histogram for a single intensity bin I_o. To obtain an expression for the entire histogram, we construct the matrix Ω so that the rows are the individual $\kappa\mu$ vectors for each intensity $I_o \in [0,1]$. For example, if there are 256 discrete intensity values Ω is a $256 \times L$ matrix. Define $\tilde{h}$ as the predicted histogram vector. Then,

$$\tilde{h} = \frac{q}{q_v F} \Omega\alpha. \qquad (3)$$

This equation gives a simple matrix formula for the complete bidirectional histogram $\tilde{h}$ of a Lambertian, isotropic, randomly rough surface.

The measured histograms were obtained from the Columbia-Utrecht texture database for the following samples: Sample 11 (plaster), Sample 10 (plaster), Sample 49 (concrete), Sample 50 (concrete) and Sample 8 (pebbles). For each sample, 19 histograms from images obtained with different viewing and illumination directions were used to represent the measured bidirectional histogram. These histograms correspond to the plane-of-incidence measurements from the database. Let $h_j(i)$ denote the ith element of the jth measured histogram where $i = 0, 1, ..255$ and $j = 0, 1, ...18$. Similarly let $\tilde{h}_j(i)$ denote the ith element of the jth estimated histogram as given by Equation 3. Since our histogram model is appropriate for constant-albedo samples, only gray-scale image information was used. In the model, shadows are assumed to be zero intensity; however, in the actual images shadows are usually non-zero. To account for this discrepancy, all image pixels with intensity lower than a manually chosen shadow threshold are counted as zero-intensity shadows.

The camera response for the measurements was approximately linear so each recorded intensity in related to the actual intensity by a gain factor and an offset. Three parameters were estimated for each sample: gain ζ, offset χ, and roughness parameter σ. The actual albedo of the sample is implicitly included in the estimation of camera gain ζ. The value for L in Equation 1 was chosen as 16, for a compact yet accurate representation. The Levenberg-Marquardt algorithm (implemented in Matlab) was used to estimate the parameters ζ, χ and σ which minimize the error E, taken over the collection of histograms, where

$$E = \sum_{j=0}^{18} \sum_{i=0}^{255} (h_j(i) - \tilde{h_j}(\zeta i + \chi))^2. \tag{4}$$

The fits were improved by applying gaussian blurring to the modeled histogram. The fitting results for five different texture samples are shown in Figure 3. These results indicate a good match between the model and measurements even in the shadow regions that correspond to zero intensity.

4.2 Correlation Model

To extend the histogram model, we've developed a correlation model for the same class of 3D textures to characterize the spatial relationship among pixels and the change of the spatial relationship with viewing direction. In [5] we present a model which uses surface statistical parameters to predict the change in the correlation length with viewing direction. Many texture algorithms have been developed for 2D texture analysis such as shape from texture [22][29][15], texture recognition and texture segmentation [2][33][14][24]. Most of these algorithms are based implicitly or explicitly on the power spectrum or equivalently on the correlation of image texture. For 3D texture, the correlation function of image texture changes in a complicated manner with viewing direction because of local foreshortening effects that depend on the varying local surface normal. Consider the texture images in Figure 5. This figure shows three oblique views of two surfaces at increasingly oblique viewing angles. The surfaces shown have the same image texture viewed frontally, but one surface is rough (3D texture) and the other surface is smooth (2D texture). The images of the smooth texture are simply warped versions of the frontally viewed 3D texture. Notice oblique views of the 2D and 3D texture are quite different. In particular, the oblique views of 2D texture show higher spatial frequencies and therefore a smaller correlation length than the oblique views of 3D texture. For algorithms which rely on spectral characteristics of texture a computational model

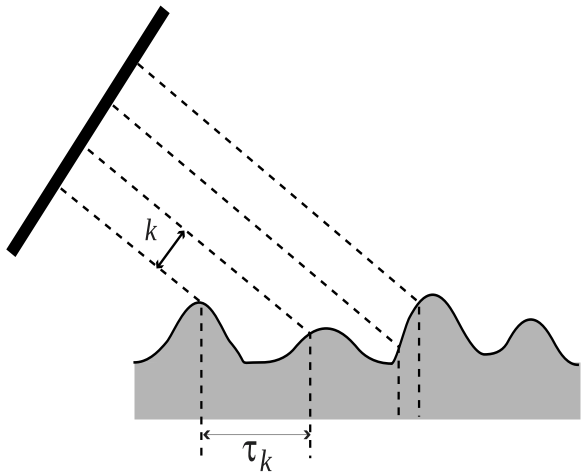

Figure 4. For a fixed distance k in the image, the corresponding surface distance is a random variable τ_k.

which quantifies the change in correlation length with viewing direction is clearly important.

We assume that the 3D texture of interest is Lambertian and has a random height profile that can be modeled as a gaussian distribution with variance σ_h^2. We further assume that two surface points are jointly normal and the autocorrelation of the surface height process is a guassian with variance β^2. The image of this surface gives rise to an image texture. In this work, we are interested in finding the correlation length of the image texture for an arbitrary illumination and viewing direction. A fixed distance k in the image corresponds to a random distance τ_k on the surface due to the varying surface profile as shown in Figure 4. Because τ_k is a random variable denoting the surface sampling, the correlation function can be written as

$$E\left(I\left[j\right], I\left[j-k\right]\right) = E\left\{E\left(I(t), I(t-\tau_k)|\tau_k\right)\right\}, \tag{5}$$

where E denotes the expected value, $I[j]$ is the intensity for image pixel j and $I\left(t\right)$ is the intensity for the surface point at t. Note that the image intensity is written as a one-dimensional quantity for notational simplicity. To further simplify the notation let $I(t)$ and $I(t-\tau)$ be denoted by I_0 and I_τ respectively. Then, by the definition of the expected value,

$$E\left(I\left[j\right], I\left[j-k\right]\right) = \int_0^\infty E\left(I_0, I_\tau|\tau_k = \tau\right) p_{\tau_k}\left(\tau\right) d\tau, \tag{6}$$

where $p_{\tau_k}\left(\tau\right)$ is the probability density function (pdf) of the random variable τ_k. The development in [5] derives an expression for $p_{\tau_k}\left(\tau\right)$ and uses Equation 6 to

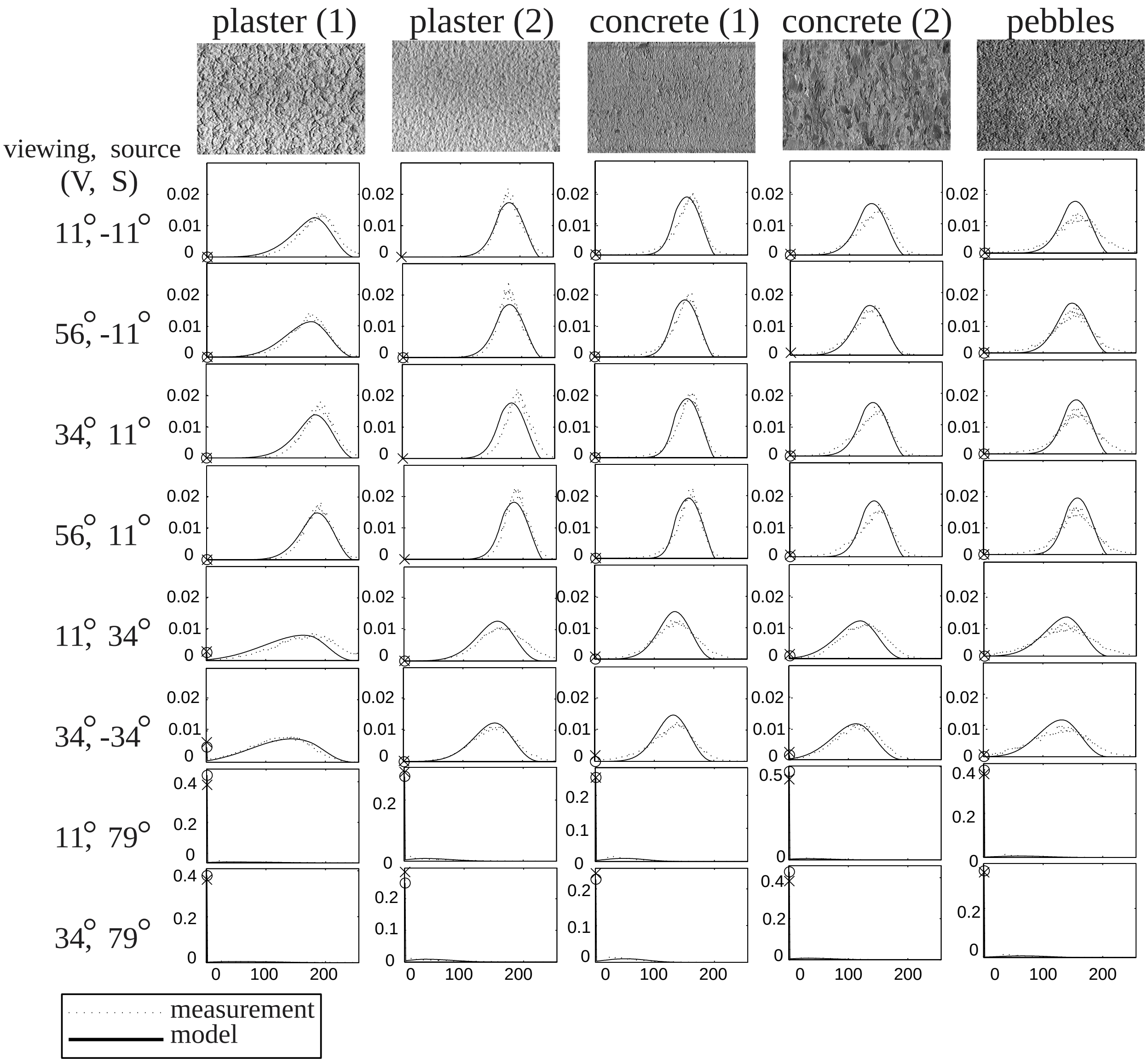

Figure 3. Each column corresponds to the histogram model fit for a sample from the Columbia-Utrecht texture database. From left to right the samples are Sample 11 (plaster), Sample 10 (plaster), Sample 49 (concrete), Sample 50 (concrete) and Sample 8 (pebbles). In each panel the model fit is shown by the solid line while the measured histogram is shown by the dotted line. The zero intensity bin is shown with an 'x' for the measured histogram and an 'o' for the modeled histogram. The estimated roughness parameter for each sample, from left to right, is 0.41, 0.23, 0.24, 0.31, 0.36, respectively. For each row, the polar angle of the viewing direction V and the illumination direction S are given on the left in degrees (negative polar angles correspond to a 180° azimuth). The model was fit using 19 histograms per sample, but for c onciseness, 8 histograms per sample are shown in this figure.

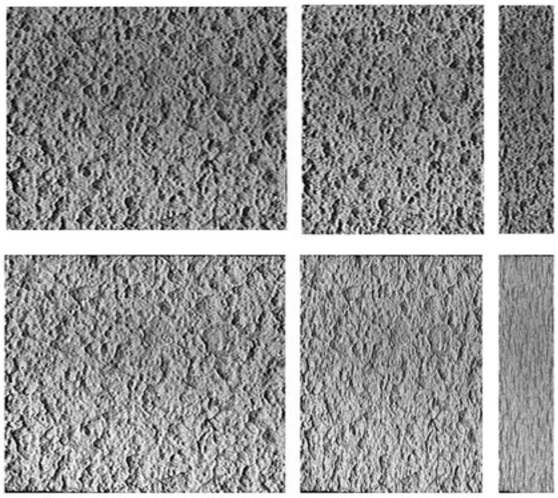

Figure 5. (Top Row) Oblique views of a 3D texture. From left to right, the associated viewing angle θ_v is $33.75^{\circ\circ}$, $56.25^{\circ\circ}$ and $78.75^{\circ\circ}$. These images were obtained from a rough plaster sample of the texture database described in [6]. (Bottom Row) Oblique views of a 2D texture with θ_v varying as in top row. These views were generated by warping the frontal view of the same plaster surface. This contrived 2D texture has the same appearance in the frontal view as the rough plaster sample.

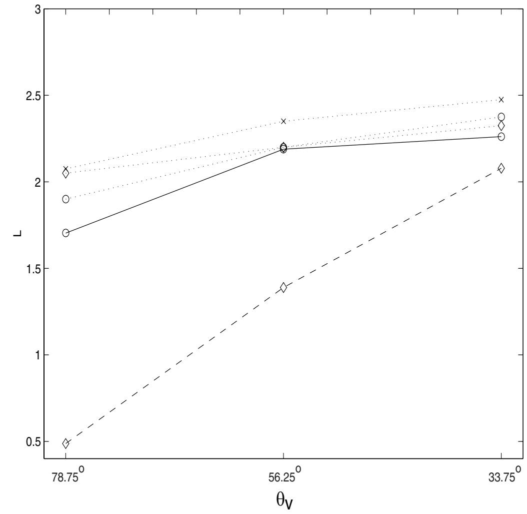

Figure 6. Measured and modeled correlation lengths. The dotted lines show the measured correlation length L as a function of $\theta_v = 33.75^\circ, 56.25^\circ$ and 78.75°. Each dotted line corresponds to a different illumination direction $\theta_s = 11.25^\circ, 33.75^\circ, -11.25^\circ$.The solid line corresponds to the model of the correlation length using the parameters that best match the measurements: $\sigma_h = 0.57$ and $\beta = 1.16$. The dashed line shows the correlation length that would be predicted if we assume the texture is a 2D texture.

predict the correlation length as a function of viewing direction.

We employ the 9 texture images obtained from the CUReT database for Sample 11 (rough plaster) corresponding to viewing angles $\theta_v = 33.75^\circ, 56.25^\circ$ and 78.75° and illumination angles $\theta_s = 11.25^\circ, 33.75^\circ$ and -11.35°. The correlation lengths were computed and plotted as a function of θ_v in Figure 6 (dashed lines). Using the correlation lengths we estimated the value of σ_h and β for this surface as $\sigma_h = 0.57$ and $\beta = 1.16$. The corresponding estimate of the correlation length L is shown in Figure 6. Also shown in this figure is the correlation length for a presumed planar surface (2D texture). There are two important things to notice here. First, the measured correlation length as a function of viewing direction is similar for all three illumination directions considered. Second, the model does a good job predicting the correct value of the correlation length especially when compared to the prediction obtained by assuming 2D texture.

4.3 PCA Models

Physical models have the advantage of capturing the salient texture properties in a small number of parameters. However these descriptions often come at the cost of imposing assumptions which limits the model applicability. Texture models derived from principal component analysis have a larger number of parameters but make no assumptions about the type of texture or surface process. Such models are useful when a large and diverse set must be analyzed and physical models are not sufficiently capturing the variation of the set. For appearance-based matching, the eigenspace decomposition can be performed directly on images. However for texture, the images themselves are merely examples of texture. Two images can depict the same type of texture but each could depict a different instance. For example, two spatially adjacent images of a spatially invariant textured region will depict the same textures even though the image pixels differ. So the eigenspace analysis should be performed not on the images themselves, but rather on a texture representation that is invariant over different texture instances. One choice of texture representation for the eigenspace analysis is image correlation in and between color bands as described in [28]. We've chosen a 3D texture representation based on the multiscale conditional histograms [11][2] that have been previously applied to 2D texture with excellent results for both synthesis and recognition. By using principal components analysis on the representation as a function of viewing and illumination direction, the non-parametric multiscale histogram can be used to characterize 3D texture. Use of this texture representation in recognition is described in Section 5.2.

5 Application of BTF Models

The BTF models discussed in the previous sections can be applied in a variety of domains. We discuss two applications in this section. The first application is that of texture synthesis, using the texture histogram model and a technique called texture morphing. The second application is texture recognition using a principal component analysis on the multiscale histogram texture representation.

5.1 Texture Synthesis

Texture injects realism into rendering, transforming dull synthetic scenes into digital replicas of reality. Often, however, current techniques of texture rendering fall short of this ideal. Consider standard texture-mapping [10] where texture is treated as 2D patterns "painted on" the surface. While this texture model is appropriate for certain cases, it is far too restrictive to enable photorealistic rendering of general scenes. Indeed many of the textures that occur in real-world scenes are 3D in their geometry; the texture is due to a height variation, rather than a color or albedo variation. Examples of 3D-textured surfaces encountered in practice include: foliage, soil, and sand in natural scenes; concrete, plaster, brick and pavement in urban scenes; rugs, upholstery and textured walls in domestic scenes; hair, skin, and clothing of people in portrait scenes. Rendering these textures is particularly challenging because changes in viewing and illumination direction dramatically alters texture appearance as shown in Figure 1.

Currently, methods to render 3D-textured surfaces fall into two categories. The first is standard texture-mapping techniques where a single image of the texture is warped around the object being rendered. This approach often suffers from a lack of realism in the rendered object . The second category are those methods that rely on precise models of the surface geometry [8][12][23]. While these approaches often yield convincing results, they require a fairly accurate geometric model and therefore are useful only in certain situations. In this work, we introduce a technique for rendering 3D textured surfaces that surpasses the performance of standard texture-mapping without requiring detailed knowledge of the underlying surface geometry. Our approach is based on having a few reference images of the textured surface and knowledge of the histogram behavior as the illumination and viewing direction changes.

To demonstrate the need to account for the 3D nature of textures, we've rendered cylinders using both standard 2D texture-mapping and a simple 3D texture-mapping (see Figure 7). The 3D texture-mapping is simply a "cut and paste" method using a piecewise-planar cylinder model and a set of 13 texture images obtained with the appropriate viewing and illumination directions from the CUReT database. Notice the realism of the cylinders rendered from a set of images when compared to the cylinders rendered with a single image. Even though the height variations on these surfaces are quite small (on the order of a millimeter), the visual effects of foreshortening, masking, shading and shadowing caused by the surface roughness are significant. By ignoring these effects, standard texture-mapping destroys the impression of roughness and adds an artificially smooth appearance to the cylinders.

The cut-and-paste method of 3D texture rendering used to generate the cylinder is not a viable rendering solution. In practice, it's unreasonable to assume that images of the surface from all desired viewing and illumination directions are available. Instead an algorithm is needed to accomplish similar renderings with significantly less prior knowledge about the texture. Our 3D texture morphing is designed for this purpose. There are two main questions in determining the prior knowledge required for the 3D texture morphing algorithm. First, what is the minimum set of texture images that contains all the necessary information about that texture? Second, what information is required to drive the transformation of a particular image of the texture into another image (with different viewing and/or illumination directions) of the same texture?

To answer the first question we assume that the textured surface is Lambertian and can be described mathematically as a height function where each point on the surface has only one height value. The frontal view of such a surface provides a large portion of the necessary information since all surface features are visible. Oblique views under the same illumination can be derived from the frontal view by applying a resampling function. For illumination changes, we follow the work in [1][26][9][16][18]. A singular value decomposition (SVD) of a set of images of the same scene under varying illumination conditions results in a set of three basis images. Any image of the scene (with fixed viewing direction) under varying illumination can be described as a linear combination of these basis images. While the work described in [1] is proven for convex Lambertian objects, good results are also reported when applying the method to non-convex objects. We've also found reasonable empirical results using textured surfaces that are not convex. Therefore we employ a set of frontal images of the texture under varied illumination conditions as a sufficient set to capture all the necessary texture information.

At this point the question is: What information will drive the transformation of the given information, the set of frontally viewed texture images, into an image of the texture from an arbitrary view and illumination? We use the gray-scale image histogram (i.e. histogram of the texture image) as a function of viewing direction V and illumination direction S. We assume that this histogram function $H(V, S)$ is known, either through modeling or measurement. There are advantages in using the histogram function to facilitate texture morphing. First, the histogram function is a considerably more concise representation than the entire texture image as a function of viewing and illumination direction. Furthermore, the histogram function is invariant over all instances of texture in a particular class.

The transformation method is a two part process. A desired illumination direction S, viewing direction V, and the set of basis images for the frontally viewed texture are given. We first synthesize a texture image $I_{F,S}$ which has a frontal view F and illumination direction S, by finding the correct combination of the basis images to match the histogram $H(F, S)$. Then we estimate the resampling function which will morph $I_{F,S}$ into $I_{V,S}$. The estimation of the resampling function is driven by minimizing the difference between the desired histogram $H(V, S)$ and the histogram of the resampled image. An example rendering using this texture morphing method is shown in Figure 7.

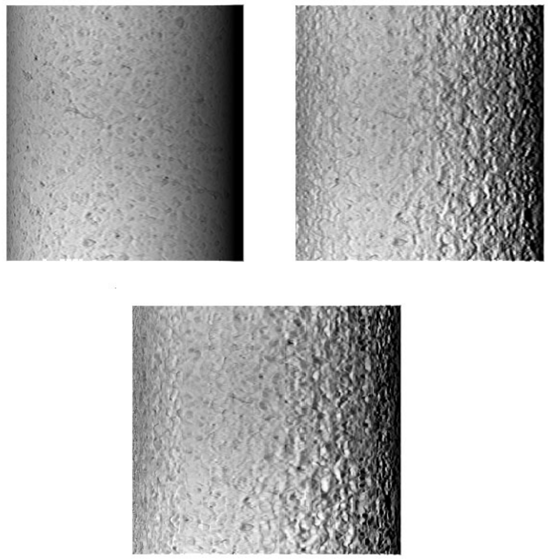

Figure 7. (Top-left) A cylinder of plaster rendered using standard texture-mapping. (Top-right) The "cut and paste" 3D texture mapping using 13 images. (Bottom) A cylinder of rough plaster rendered using 3D texture morphing. Notice the realistic appearance of roughness, similar to the plaster cylinder generated with the "cut and paste" method shown above. Unlike the "cut and paste" method, 3D texture morphing does not need 13 images from the correct viewing and illumination directions. The 3D texture morphing algorithm successfully creates the same effect.

5.2 Texture Recognition

In Section 4.3, we discuss a representation for 3D texture obtained from the principal components of a series of texture images as a function of viewing and illumination direction. Each texture image is represented by a conditional multiscale histogram similar to that described in [11] [2]. This multiscale histogram has been shown to yield excellent results for 2D texture recognition as well as 2D texture synthesis. Principal component analysis provides a convenient way to extend the representation to 3D texture. This 3D texture representation is based solely on appearance and has no underlying surface model. We employ the SLAM [20] software library to perform the eigenspace analysis in recognition experiments using CUReT database texture images. For each texture image, a gaussian image pyramid [3] is constructed. For each level of the pyramid a conditional histogram $h_j(i, i')$ is computed as the number of pixels in intensity bin i with a parent in intensity bin i'. The parent of pixel at image location x, y is the pixel at the next coarser pyramid level at image location $x/2, y/2$. The histograms are normalized so that $\sum_i h_j(i, i') = 1$ for all i'. For our experiments, we use 3 levels of the gaussian pyramid and a bin size equal to 20 with an intensity range of $(0, 255)$ which gives 13 intensity bins. The size of the histogram representation for intensity is $3 \times 13 \times 13 = 507$. To increase the robustness of the recognition the image gradient in the x and y directions is also used to form a gradient multiscale histogram. Therefore the total size of each input vector is $3 \times 507 = 1521$. Color information is not included; only the green component of the RGB image is retained. The choice not to use color ensures that the recognition results indicate a recognition of structural appearance.

Twenty samples from the CUReT database were used in the recognition experiments. These samples are Sample 2 (polyester), 3 (terrycloth), 6 (sandpaper), 8 (pebbles), 10 (plaster), 13 (artificial grass), 14 (roofing tile), 15 (aluminum foil), 18 (rug_a), 19 (rug_b), 22 (lambswool), 23 (lettuce leaf), 26 (loofa), 27 (insulation), 28 (crumpled paper), 30 (plaster_b), 35 (painted stones), 38 (ribbed paper), 40 (straw), and 42 (corduroy). The images were manually segmented prior to processing to ensure that only texture information is included in the analysis. For each sample, a subset of the images in the database were used as training images and a different subset were used as test images. In this manner, the test images have illumination and viewing directions that are different from the training images. Recognition results were obtained under 3 different ranges of viewing polar angle θ_v and illumination polar angle θ_s. Range A is $\{\theta_s > 45°, \theta_v > 45°\}$; Range B is $\{\theta_s > 57°, \theta_v > 57°\}$; Range C is $\{\theta_s > 70°, \theta_v > 70°\}$. For Range A, there were 400 training images and 140 test images over all samples. For Range B, there were 1100 training images and 200 test images. For Range C, there were 1880 training images and 360 test images. The recognition rates as a function of the number of dimensions in the eigenspace are shown in Figure 8. As expected and consistent with experiments in [28], the recognition rates improve when the range of θ_s and θ_v are reduced. The high recognition rates shown in Figure 8 indicate that this representation does a good job capturing the varying surface appearance of these samples. The results are especially encouraging when one considers that the training images were taken from different viewing and illumination direction than the test images.

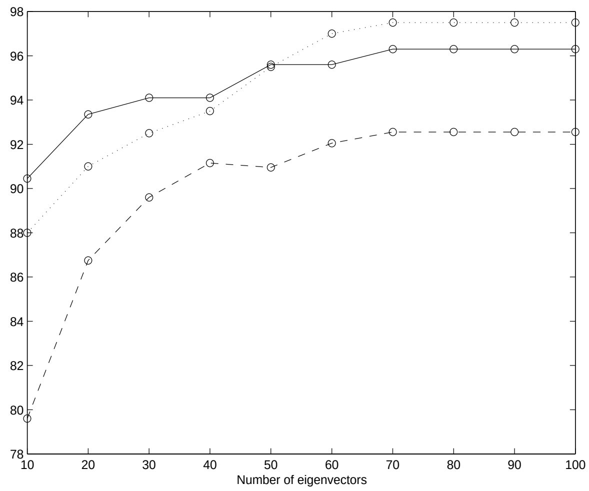

Figure 8. Recognition rates (percentage of images correctly recognized) as a function of the number of eigenvectors for 20 textured samples. Three ranges of viewing angle θ_v and illumination angle θ_s are tested. Range A (solid line) is $\{\theta_s > 45°, \theta_v > 45°\}$; Range B (dotted line) is $\{\theta_s > 57°, \theta_v > 57°\}$; Range C (dashed line) is $\{\theta_s > 70°, \theta_v > 70°\}$

6 Conclusion

Clearly it is desirable to use all the information attainable from a scene to aid the recognition process. This implies an integration of geometry and appearance-based methods for recognition. Fine scale surface geometry is difficult to acquire and cumbersome to model geometrically. Ignoring the effects of fine-scale geometric variations on surfaces is not appropriate because while surface height variations are small, the effects on appearance of varying surface normal and local shadowing/occlusions are large. As a result of these considerations, a natural division of labor in an integrated recognition system is to handle surface geometry with appearance-based methods and object geometry using geometry-based methods. One integration method may have geometry-based systems and appearance-based systems run independently, each casting a vote in the recognition process. In another method, the appearance-based surface analysis can also a assist object geometry acquisition via algorithms for shape-from-shading with accurate BRDF models and shape-from-texture with accurate BTF models. There are clearly many open issues in developing such systems.

We have described measurements and models of 3D textured surface appearance in the framework of the BRDF and the BTF. This work can be useful in developing and testing recognition systems that account for surface appearance due to fine scale surface geometry.

References

[1] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions. *Proceedings of the IEEE Conference on Com-*

puter Vision and Pattern Recognition, pages 270–277, June 1996.

[2] J. D. Bonet and P. Viola. Texture recognition using a non-parametric multi-scale statistical model. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 641–647, 1998.

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

[4] K. Dana and S. Nayar. Histogram model for 3d textures. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 618–624, June 1998.

[5] K. Dana and S. Nayar. Correlation model for 3d texture. *International Conference on Computer Vision*, page to appear, September 1999.

[6] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real world surfaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 151–157, June 1997.

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

[8] K. Fleischer, D. Laidlaw, B. Currin, and A. Barr. Cellular texture generation. *ACM SIGGRAPH*, 1995.

[9] P. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 995–999, 1994.

[10] P. Heckbert. A survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, 1986.

[11] D. Heeger and J. Bergen. Pyramid-based texture analysis/synthesis. *ACM SIGGRAPH*, pages 229–38, 1995.

[12] J. Kajiya and T. Kay. Rendering fur with 3d textures. *ACM SIGGRAPH*, 23(3):271–80, 1989.

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

[14] J. Krumm and S. Shafer. Texture segmentation and shape in the same image. *International Conference on Computer Vision*, pages 121–127, 1995.

[15] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23:149–168, 1997.

[16] H. Murase and S. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, pages 5–24, 1995.

[17] S. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):611–634, July 1991.

[18] S. Nayar and H. Murase. Dimensionality of illumination in appearance matching. *IEEE Conf. on Robotics and Automation*, 1996.

[19] S. Nayar and M. Oren. Visual appearance of matte surfaces. *Science*, 267:1153–1156, Feb. 1995.

[20] S. Nene, S. Nayar, and H. Murase. Slam: A software library for appearance matching. *Technical Report CUCS-019-94*, Proceedings of ARPA Image Understanding Workshop, November 1994.

[21] M. Oren and S. Nayar. Generalization of the lambertian model and implications for machine vision. *International Journal of Computer Vision*, 14:227–251, 1995.

[22] M. Patel and F. Cohen. Shape from texture using markov random field models and stereo-windows. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 290–5, 1992.

[23] K. Perlin. Hypertexture. *ACM SIGGRAPH*, 23(3):253–62, 1989.

[24] R. Picard, T. Kabir, and F.Liu. Real-time recognition with the entire brodatz texture database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–639, 1993.

[25] P. Poulin and A. Fournier. A model for anisotropic reflection. *ACM SIGGRAPH*, 24(4):273–282, August 1990.

[26] A. Shashua. Geometry and photometry in 3d visual recognition. *PhD thesis, MIT*, 1992.

[27] M. Stavridi, B. van Ginneken, and J. Koenderink. Surface bidirectional reflection distribution function and the texture of bricks and tiles. *Applied Optics*, 36(16), June 1997.

[28] P. Suen and G. Healey. Analyzing the bidirectional texture function. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 753–758, June 1998.

[29] B. Super and A. Bovik. Shape from texture using local spectral moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):333–43, April 1995.

[30] H. Tagare and R. DeFigueiredo. A framework for the construction of reflectance maps for machine vision. *Computer Vision, Graphics and Image Processing*, 57(3):265–282, May 1993.

[31] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, 1967.

[32] B. van Ginneken, J. Koenderink, and K. J. Dana. Texture histograms as a function of irradiation and viewing direction. *International Journal of Computer Vision*, page to appear, 1999.

[33] L. Wang and G. Healey. Illumination and geometry invariant recognition of texture in color images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 419–424, 1996.

[34] G. Ward. Measuring and modeling anisotropic reflection. *ACM SIGGRAPH*, 26(2):265–272, 1992.

[35] S. Westin, J. Arvo, and K. Torrance. Predicting reflectance functions from complex surfaces. *ACM SIGGRAPH*, 26(2):255–263, 1992.

[36] L. Wolff. A diffuse reflectance model for smooth dielectrics. *Journal of the Optical Society of America A*, 11:2956–2968, November 1994.