Texture for Appearance Models in Computer Vision and Graphics

Oana G. Cula
Kristin J. Dana

Rutgers University, Piscataway, New Jersey, USA

1current address: Johnson & Johnson, Skillman, New Jersey, USA
Abstract. Appearance modeling is fundamental to the goals of computer vision and computer graphics. Traditionally, appearance was modeled with simple shading models (e.g. Lambertian or specular) applied to known or estimated surface geometry. However, real world surfaces such as hair, skin, fur, gravel, scratched or weathered surfaces, are difficult to model with this approach for a variety of reasons. In some cases it’s not practical to obtain geometry because the variation is so complex and fine-scale. The geometric detail is not resolved with laser scanning devices or with stereo vision. Simple reflectance models assume that all light is reflected from the point where it hits the surface, i.e. no light is transmitted into the surface. But in many real surfaces, a portion of the light incident on one surface point is scattered beneath the surface and exits at other surface points. This subsurface scattering causes difficulties in accurately modeling a surface such as frosted glass or skin with a simple geometry plus shading model. So even when a precise geometric profile is attainable, applying a pointwise shading model is not sufficient. Because of these issues, image-based modeling has become a popular alternative to modeling with geometry and point-wise shading.

Real world surfaces are often textured with a variation in color (as in a paisley print or leopard spots) or a fine-scale surface height variation (e.g. crumpled paper, rough plaster, sand). Surface texture complicates appearance prediction because local shading, shadowing, foreshortening and occlusions change the observed appearance when lighting or viewing directions and changed. As an example, consider a globally planar surface of wrinkled leather where large local shadows appear when the surface is obliquely illuminated and disappear when the surface is frontally illuminated. Accounting for the variation of appearance due to changes in imaging parameters is a key issue in developing accurate models. The terms BRDF and BTF have been used to describe surface appearance. The BRDF (bidirectional reflectance distribution function) describes surface reflectance as a function of viewing and illumination angles. Since surface reflectance varies spatially for textured surfaces, the BTF was introduced to add a spatial variation. More specifically, the bidirectional texture function (BTF) is observed image texture as a function of viewing and illumination directions. In this chapter, topics in BRDF and BTF modeling for vision and graphics are presented. Two methods for recognition are described in detail: (1) bidirectional feature histograms and (2) symbolic primitives that are more useful for recognizing subtle differences in texture.
Texture for Appearance Models in Computer Vision and Graphics

1. Introduction

The visual appearance of an object or person, is a seemingly simple concept. In everyday life, we see the visual appearance of objects, surfaces and scenes. We remember what we see, and store some type of cognitive representation of the visual world around us. So what are the important attributes of appearance? Size, shape and color are clearly at the top of the list. But for accurate computational descriptions of appearance, as needed for recognition and rendering algorithms, attributes of size, shape and color are not sufficient. The need for a more comprehensive description of appearance is the motivation behind the study and appreciation of texture. The scenes and surfaces of our world are filled with textures: rocks, sand, trees, skin, velvet, burlap, foliage, screen, crystals. In this chapter, we concentrate on textured objects or surfaces which have a fine-scale geometric variation as depicted in Figure 1. By fine-scale, we mean geometric details (height changes) that are small compared to the viewing distance and are typically hard to measure such as fine-scale wrinkles in leather, venation of leaves, fibers of textiles, fuzziness of a peach, weave of a fabric, and roughness of plaster. These textures have also been termed relief textures or 3D textures and can be accompanied by color variations.

![Figure 1. Surface appearance or texture is the reflected light in a spatial region. We are interested in the case when the local geometry is not smooth and has some roughness. In general this fine-scale geometry is difficult to measure or model so image-based modeling techniques are useful.](image-url)
In natural environments, most surfaces exhibit some amount of fine-scale geometry (tactile texture) or roughness. Because of this non-smooth surface geometry, appearance is affected by local occlusion, shadowing and foreshortening, as shown in Figure 2. Here, the rough surface of the plaster is viewed under different surface tilts and light source directions, so appearance changes significantly. More examples are shown in Figure 3 which shows hair and fabric texture. Figure 4 shows an interesting demonstration of unwrapping the texture of a ball to visualize the constituent image. This unwrapped texture image is the result of an operation that is essentially the inverse of texture mapping. With texture mapping, a single image is mapped onto the geometry of the ball. The unwrapping requires multiple stages and knowledge of the object geometry and camera parameters. The important point for the purposes of this discussion is that the unwrapped image is not uniform in appearance. The lighting and viewpoint variations around the ball cause a change in appearance of the fine-scale texture. Therefore instead a single texture image, cannot sufficiently capture appearance.

1.1. Geometry-based vs. Image-based. Knowing that a single image is not sufficient to capture appearance, the pertinent question is: what additional information is needed? There are essentially two options. The first is geometry-based:
Figure 4. Unwrapped texture of a ball. Since the ball geometry is known, the section of the image can be unwrapped (inverse of texture mapping) and the local appearance of the texture section is shown. Notice that local foreshortening, shadowing and occlusions change across the texture because of the differences in global surface orientation and illumination direction.

measure fine-scale geometry explicitly so that an extremely detailed mesh is created. The second option is image-based: sample the space of imaging parameters, i.e. choose a finite set of illumination and viewing directions, and record the image of the surface. In general, the geometry-based approach is not favored for several reasons. First, fine-scale geometry is often very hard to measure. Consider the hair texture of Figure 3, a laser scanner or stereo method would have great difficulty because of the large amount of occlusions. Consider also the very fine details on skin texture such as individual pores. Each scanning system has a finite resolution so there is an inevitable loss of detail. Also, the translucency of materials such as skin make scanning very difficult. Laser scanners work best for white opaque objects that do not exhibit internal light scattering.

But even if we could measure the fine-scale geometry perfectly, geometry is not appearance. In order to render the object the reflectance must be known. A typical computer graphics shader uses very simple shading models that are only an approximation of the actual reflectance. For highly accurate modeling, the bidirectional reflectance distribution function (BRDF) for the surface material is needed. The BRDF gives the surface reflectance for any combination of viewing direction and incident illumination direction. Additionally, many real world surfaces are not spatially homogeneous, so the BRDF changes across the surface. To measure the BRDF at each point, the reflectance is measured from all exitance angles and for
all incident angles. But for a rough surface, some angles are occluded by the neighbors, i.e. the peaks and valleys of the surface create occlusions making a pointwise BRDF is difficult to measure.

Additionally, BRDF models assume that all light is reflected from the point where it hits the surface, i.e. no light is transmitted into the surface. But in many real surfaces, a portion of the light incident on one surface point is scattered beneath the surface and exits at other surface points. This subsurface scattering causes difficulties in accurately modeling a surface such as frosted glass or skin with a simple geometry plus shading model. So even when a precise geometric profile is attainable, applying a pointwise shading model is not sufficient. Because of these issues, image-based modeling has become a popular alternative to modeling with geometry and point-wise shading. The BTF (bidirectional texture function) is the nomenclature introduced in [14, 15] for an image-based characterization of texture appearance.

2. BRDF and BTF: a historical perspective

The BRDF has been a standard term in computer vision for decades. It’s formal definition is the ratio of the radiance exiting a surface point to the irradiance incident on the surface point. Informally it’s the ratio of the amount of input light to the output light. The units of the BRDF can seem formidable at first glance, \( \text{watts per steridian per meter}^2 \). To parse the units, consider that the amount of input light is the light power in watts measured per unit area, so input light has the units watts per meter\(^2\). The amount of output light is slightly more complicated. The total output light is watts per meter\(^2\) but it radiates in many directions. For the BRDF we are interested in the amount of output light in a particular direction. The units of the solid angle in a particular direction are steridians. Hence, the output light is measured in watts per meter\(^2\) per steridian. The BRDF was first defined by Nicodemus in 1970 [41]. Since it is a function of viewing and illumination angles it can be expressed as \( f(\theta_i, \phi_i, \theta_v, \phi_v) \).

Real world surfaces typically do not have a uniform BRDF due to both surface markings and surface texture. The bidirectional texture function (BTF) extends the BRDF in order to characterize surfaces reflectance that varies spatially. The early concept of BTF was introduced with the Columbia-Utrecht Texture and Reflectance (CUReT) database in 1996 [14, 15] and used for numerous texture modeling and recognition studies. The BTF is expressed as \( f(x, y, \theta_i, \phi_i, \theta_v, \phi_v) \), but there is an important subtlety in the definition. As discussed, in 1, the BTF is not simply the BRDF at each surface point. The BTF concept is best expressed when considering a flat piece of rough material. Instead of modeling the exact fine-scale surface geometry and then applying a measured BRDF on the bumpy mesh, we assume the geometry is locally flat and that appearance changes with viewing and illumination direction. The model ignores the fine-scale geometry when defining viewpoint and illumination directions. That is, the imaging directions are defined with respect to the reference plane. Appearance is captured by obtaining images from multiple viewing and illumination directions. The fine-scale shadowing, occlusions, shading and foreshortening that affect the pixel intensities of the recorded images become part of the appearance model, explicitly, without regard for the height profile of the surface. In effect, the fine-scale geometric variations and any additional color
variations are modeled as a spatially varying BRDF $f(x, y, \theta_i, \phi_i, \theta_v, \phi_v)$. Specifically, the reflectance at each point is not a typical reflectance function but instead contains the nonlinearities of the shadowing and occlusions of fine-scale geometry. A surface point at $x, y$ may be shadowed as the illumination direction changes from $\theta_i$ to $\theta_i + \delta$ for some small angle $\delta$, causing an abrupt change in the BTF $f(x, y, \theta_i, \phi_i, \theta_v, \phi_v)$ to near zero reflectance.

Of course, the BTF extends to non-flat surfaces as well. The conceptual model is that the object can be characterized by a geometric mesh that is “texture-mapped” not with a single image but with a BTF. Typical texture mapping maps each point of the 3D vertex into a 2D texture image parameterized by the texture coordinates $u, v$ which vary from 0 to 1. But recall that a single image is not sufficient for authentic replications of appearance. Instead the imaging parameters for that vertex must be part of the mapping. That is, the vertex in object space is
mapped to \( f(u, v, I, V) \) which is the BTF with standard texture coordinates \((u,v)\). Here the vector \( I \) is used for the illumination direction instead of the polar and azimuth angles \((\theta_i, \phi_i)\). Similarly the viewing direction is specified by \( V \). A BTF sample is an image \( f(x, y, I, V) \) with the \( x, y \) coordinates which are now scaled to \( u, v \) coordinates which vary from 0 to 1. A comparison of the appearance of standard texture mapping and BTF mapping for simple cylinders is shown in Figure 5.

Ongoing research seeks to address the following important questions: 1) How many samples are necessary to appropriately capture appearance? Since the space of parameters is a 4 dimensional space with variations of illumination and viewing angles, even a sparse sampling of the space gives a large number of image. For example, 30 illumination angles and 30 viewing angles for each illumination direction is 900 images. 2) Where should the samples be positioned in the space of imaging parameters to best represent the surface? Should the viewing and illumination angles chosen be a uniform sampling of the imaging space? Are there some directions that should be sampled more densely? 3) How can in-between samples be obtained, i.e. how to interpolate a full continuous BTF from the finite number of measured samples.

One of the difficulties in answering these questions is that the answer depends on the surface itself. In an empirical study of a set from the Curet database\[10], it was shown that a very important sample for recognizing the surface was the sample which was viewed at a 45 degree angle from the global surface normal and illuminated from the opposite 45 degree angle. This empirical result is consistent with intuition because shadows and occlusions accentuate details but too many shadows obscure the surface. Another important issue in evaluating the effectiveness of the BTF is to consider the perceptual issues in replacing geometry with texture. An important contribution in this area is the work of [46].

3. Recent Developments in BTF Measurements and Models

Surface appearance has been a popular topic in computer vision and computer graphics in the last decade. The research can be categorized into the following topics: (1) recognition, (2) representation, (3) rendering and (4) measurement.

Recognition methods have been developed which learn the appearance of textured surface through a training stage using example images from surfaces with varying imaging parameters, i.e. varying illumination and viewing directions. The 3D texton method [28, 29], uses textons from registered training images to build an appearance vector which is the observed appearance under multiple imaging parameters. Histograms of appearance vectors characterize the texture and can be used to recognize novel image sets under the same imaging parameters. The bidirectional feature histogram (BFH) method [10, 11, 12] uses an image texton vocabulary from arbitrary unregistered input images. Once the image texton library is learned, histograms of texton labels characterize surfaces. Histograms from multiple images using different imaging parameters characterize surface appearance under multiple viewing and illumination. This model is used for recognizing a surface using a single image under unknown imaging parameters that was not part of the training set. More details of the BFH recognition method are provided in Section 4.1. Another method which uses histograms of learned image features is [53], and this method clusters using rotationally invariance filter responses in order to learn local image features.
Representations for surface BTF’s are computational models built from measurements that can be used to synthesize appearance from novel viewing and illumination directions. These representations provide a means of interpolating between samples and storing surface information in a concise format. Representations methods that have been employed thus far for BTF’s include principle components analysis (PCA) [42, 59, 39, 19], spherical harmonics [47], basis functions of recovered BRDF’s [27], tensor factorization [54], and steerable basis textures [2].

Rendering BTF’s in an efficient manner for realistic surface appearance in graphics has received significant attention in the literature. Early pioneering work included view dependent texture in image-based rendering of architecture [18]. More recent work on texture rendering that enables efficient BTF rendering includes [52, 32, 34, 50, 35]. A variation of BTF rendering which models surface geometry as a displacement mapping includes [55, 58, 43].

Measurements of surface appearance are particularly important in creating models for recognition, rendering and representations. Sampling the appearance space is the first step to most of the current example-based methods. In addition to the Curet database, there have been several more recent texture databases including the Bonn BTF database [7, 40], Oulu Texture Database [44], and the Heriot-Watt Photex database [45]. The Photex database has the advantage of having registered data amenable to photometric stereo. The latest in texture databases characterizing the time varying aspect of surface appearance including surfaces whose appearance changes as the dry (wood paint), decay (fruit), and corrode (metals) [22].

Specialized devices are needed to measure appearance. The measurement apparatus is often a gonioreflectometer with lighting and cameras at multiple positions over a hemisphere or dome [17, 57]. For object or face appearance, dome-based imaging apparatus is necessary. However, texture surfaces can often be characterized by a small locally flat sample that make smaller devices a reasonable option. The fundamental difficulty in changing the imaging parameters to obtain appearance measurements has led to several novel devices to measure texture appearance. These include a texture camera [13, 16], kaleidoscope for BTF measurements [23], and a photometric stereo desktop scanner [8].

Measurements of surface appearance have been incorporated in digital archiving work in order to create an accurate digital representation of the appearance of historical sites and artwork. Important work in this area include digitizing the Florentine Pieta [4, 5] and the digital Michelangelo project [31]. In these projects the goal is to measure both global shape and local surface appearance. The main goal is a digital representation that can simulate the physical presence of the archived object.

Some of the fascinating papers in the literature on appearance are those that explain specific phenomena in real world surfaces. Models have been developed for weathered materials [20], the appearance of finished wood [36], velvet [26, 33], granite [49], and plant leaves [56]. These models consider the physics of the surface and how light interacts at the material boundaries to create an accurate prediction of appearance. The specific methods demonstrate the complexities and the variety of natural surfaces.

Modeling texture appearance with images instead of geometry has a consistent foundation with general image-based rendering approaches in graphics [9, 21, 30, 37, 48], and appearance-based modeling in vision [51, 1, 38, 6, 3]. Image-based
rendering caused a convergence of computer graphics and computer vision. Prior work in graphics concentrated on modeling object geometry and applying shading models. Image-based rendering allowed rendering without ever knowing the object geometry. Similarly, with the BTF, rendering is done with no knowledge of the surface geometry.

4. Appearance models for recognition

In this section we detail one method for recognition based on texture appearance called the bidirectional feature histogram. This method is but one of many modeling and recognition papers in the field. However, it has the advantage that the actual viewing and illumination parameters do not have to be known for the test and for the training images. Application of the model in recognizing skin textures is discussed in Section 5.

4.1. Bidirectional Feature Histogram. One model for the BTF is the Bidirectional Feature Histogram [11]. A statistical representation is a useful tool for modeling texture for recognition purposes. The standard framework for texture recognition consists of a primitive and a statistical distribution (histogram) of this primitive over space. So how does one account for changes with imaging parameters (view/illumination direction)? Either the primitive or the statistical distribution should be a function of the imaging parameters. Using this framework, the comparison of our approach with the 3D texton method [29] is straightforward. The 3D texton method uses a primitive that is a function of imaging parameters, while our method uses a statistical distribution that is a function of imaging parameters. In our approach the histogram of features representing the texture appearance is called bidirectional because it is a function of viewing and illumination directions. The advantage of our approach is that we don’t have to align the images obtained under different imaging parameters.

The primitive used in our BTF model is obtained as follows. We start by taking a large set of surfaces, filter these surfaces by oriented multiscale filters and then cluster the output. The hypothesis is that locally there are a finite number of intensity configurations so the filter outputs will form clusters (representing canonical structures like bumps, edges, grooves pits). The clustering of filter outputs are textons. A particular texture sample is processed using several images obtained under different imaging parameters (i.e., different light source directions and camera directions). The local structures are given a texton label from an image texton library (set up in preprocessing). For each image, the texton histograms are computed. Because these histograms are a function of two directions (light source and viewing direction), they’re called bidirectional feature histograms or BFH. The recognition is done in two stages: (1) a training stage where a BFH is created for each class using example images and (2) a classification stage. In the classification stage we only need a single image and the light and camera direction is unknown and arbitrary. Therefore we can train with one set of imaging conditions but recognize under a completely different set of imaging conditions.

Within a texture image there are generic structures such as edges, bumps and ridges. Figure 6 illustrates the pre-processing step of constructing the image texton library. We use a multiresolution filter bank $F$, with size denoted by $3 \times f$, and consisting of oriented derivatives of Gaussian filters and center surround derivatives of Gaussian filters on three scales as in [29]. Each pixel of a texture image
K-Means Clustering on size \( f \) Feature Vectors

\( \text{Texton Labels} \)

\[ \text{Image Texton Library} \]

Figure 6. Creation of the image texton library. The set of \( q \) unregistered texture images from the BTF of each of the \( Q \) samples are filtered with the filter bank \( F \) consisting of \( 3 \times f \) filters, i.e. \( f \) filters for each of the three scales. The filter responses for each pixel are concatenated over scale to form feature vectors of size \( f \). The feature space is clustered via k-means to determine the collection of key features, i.e the image texton library.

is characterized by a set of three multi-dimensional feature vectors obtained by concatenating the corresponding filter responses over scale. K-means clustering is used on these concatenated filter outputs to get image textons. By using a large set of images in creating the set of image textons, the resulting library is generic enough to represent the local features in novel texture images that were not used in creating the library.

The histogram of image textons is used to encode the global distribution of the local structural attribute over the texture image. This representation, denoted by \( H(l) \), is a discrete function of the labels \( l \) induced by the image texton library, and it is computed as described in Figure 7. Each texture image is filtered using the same filter bank \( F \) as the one used for creating the texton library. Each pixel within the texture image is represented by a multidimensional feature vector obtained by concatenating the corresponding filter responses over scale. In the feature space populated by both the feature vectors and the image textons, each feature vector is labeled by determining the closest image texton. The spatial distribution of the representative local structural features over the image is approximated by computing the texton histogram. Given the complex height variation of the 3D textured sample, the texture image is strongly influenced by both the viewing direction and
Figure 7. 3D texture representation. Each texture image $I_j$, $j = 1 \ldots n$, is filtered with filter bank $F$, and filter responses for each pixel are concatenated over scale to form feature vectors. The feature vectors are projected onto the space spanned by the elements of the image texton library, then labeled by determining the closest texton. The distributions of labels over the images are approximated by the texton histograms $H_j(l), j = 1 \ldots n$. The set of texton histograms, as a function of the imaging parameters, forms the 3D texture representation, referred to as the bidirectional feature histogram (BFH).

The illumination direction under which the image is captured. Accordingly, the corresponding image texton histogram is a function of the imaging conditions.

Note that in our approach, neither the image texton nor the texton histogram encode the change in local appearance of texture with the imaging conditions. These quantities are local to a single texture image. We represent the surface using a collection of image texton histograms, acquired as a function of viewing and illumination directions. This surface representation is described by the term bidirectional feature histogram. It is worthwhile to explicitly note the difference between the bidirectional feature histogram and the BTF. While the BTF is the set of measured images as a function of viewing and illumination, the bidirectional feature histogram is a representation of the BTF suitable for use in classification or recognition.

The dimensionality of histogram space is given by the number of textons in the image texton library. Therefore the histogram space is high dimensional, and a compression of this representation to a lower-dimensional one is suitable, providing that the statistical properties of the bidirectional feature histograms are still preserved. To accomplish dimensionality reduction we employ PCA, which finds an optimal new orthogonal basis in the space, while best describing the data. This
approach has been inspired by [38], where a similar problem is treated, specifically an object is represented by set of images taken from various poses, and PCA is used to obtain a compact lower-dimensional representation.

In the classification stage, the subset of testing texture images is disjoint from the subset used for training. Again, each image is filtered by $F$, the resulting feature vectors are projected in the image texton space and labeled according to the texton library. The texton distribution over the texture image is approximated by the texton histogram. The classification is based on a single novel texture image, and it is accomplished by projecting the corresponding texton histogram onto the universal eigenspace created during training, and by determining the closest point in the eigenspace. The 3D texture sample corresponding to the manifold onto which the closest point lies is reported as the surface class of the testing texture image.

5. Application: Human Skin Texture Recognition

5.1. Hand Texture Recognition. Many texture recognition experiments are done with textures from very distinct classes, like many of the textures of the CUREt database. However, it is particularly interesting to show recognition of textured surfaces that are not very different in composition. An illustrative example is the recognition of different samples of skin texture. Human skin has fine-scale details as shown in Figure 8 including skin glyphs, skin imperfections, skin dryness, scars, etc. These images are from the Rutgers Skin Texture Database [12].

Consider the task of recognizing which section of the hand is depicted in a particular skin texture image. Different regions of the hand have distinct textural features, although the distinction is more subtle than with other textured surfaces, e.g. pebbles vs. grass. We summarize a simple experiment for hand texture recognition that was described in more detail in [12]. For this experiment, the bidirectional feature histogram model is used. The skin regions correspond to three distinct regions of a finger: bottom segment on palm side, fingertip, and bottom segment on the back of the hand, as illustrated in Figure 9. Test images are from two subjects: for subject 1 both the index and middle fingers of left hand have been imaged, for subject 2 the index finger of left hand has been measured. For each of 9 combinations of finger region, finger type and subject, 30 images are captured, corresponding to 3 camera poses, and 10 light source positions for each camera pose. As a result the dataset for the hand texture experiments contains 270 skin texture images. Figure 10 illustrates few examples of texture images in this dataset. During preprocessing each image is converted to gray scale, and is manually segmented to isolate the largest approximately planar skin surface is used in the experiments.
For constructing the image texton library, we consider a set of skin texture images from all three classes, however only from index finger of subject 1. This reduced subset of images is used because we assume that the representative features for a texture surface are generic. This assumption is particularly applicable to skin textures, given the local structural similarities between various skin texture classes.

Each texture image is filtered by employing a filter bank consisting of 18 oriented Gaussian derivative filters with six orientations corresponding to three distinct scales as in [28]. The filter outputs corresponding to a certain scale are grouped to form six-dimensional feature vectors. The resulting three sets of feature vectors are used each to populate a feature space, where clustering via k-means is performed to determine the representatives among the population. We empirically choose to employ in our experiments a texton library consisting of 50 textons for each scale.

During the first set of experiments, the training and testing image sets for each class are disjoint, corresponding to different imaging conditions or being obtained from different surfaces belonging to the same class (e.g. fingertip surface from different fingers). For each of the classes we consider all available data, that is, each texture class is characterized by 90 images. We vary the size of the training set for each class from 45 to 60, and, consequently the test set size is varied from 45 to 30. For a fixed dimensionality of the universal eigenspace, i.e. 30, the profiles of individual recognition rates for each class, as well as the profile of the global recognition rate indexed by the size of the training set are illustrated in Figure 11 (a). As the training set for each class is enlarged, the recognition rate improves, attaining the value 100% for the case of 60 texture images for training and the rest of 30 for testing. To emphasize the strength of this result consider that the classification is based on either: a single texture image captured under different imaging conditions than the training set; or a single texture image captured under the same imaging conditions, but from a different skin surface. The variation of recognition rate as a function of the dimensionality of the universal eigenspace, when the size of the training set is fixed to 60, is depicted in Figure 11 (b). As expected, the performance improves as the dimensionality of the universal eigenspace is increased.

Figure 9. Illustration of the hand locations imaged during the experiments described in Section 5.1.
5. APPLICATION: HUMAN SKIN TEXTURE RECOGNITION

Figure 10. Examples of hand skin texture images for each location, and for each of the three fingers imaged during our experiments. In each of the pictures first row depicts skin texture corresponding to class 1 (bottom segment, palm side), second row presents texture images from class 2 (fingertip), and third row consists of texture images from class 3 (bottom segment, back of palm). In (a) images are obtained from index finger of subject 1, in (b) from middle finger of subject 1, and in (c) from index finger of subject 2.

In training and testing, images are from spatially disjoint image regions. We divide each skin texture image into two non-overlapping subimages, denoted as lower half subimage, and upper half subimage. This results in a set of 60 texture subimages, two for each of the 30 combinations of imaging parameters. For this experiment we consider data obtained from index finger of subject 1. The training
Figure 11. Recognition rate as a function of the size of the training set (a) (when dimensionality of the universal eigenspace is fixed to 30), and as a function of the dimensionality of the universal eigenspace (b) (when the training set of each class has cardinality 60), both corresponding to first set of recognition experiments reported in Section 5.1. (c) Profile of recognition rate as a function of the dimensionality of the universal eigenspace, corresponding to second recognition experiment, described in Section 5.1.

set is constructed by alternatively choosing lower half and upper half subimages, which correspond to all 30 imaging conditions. The testing set is the complement of training set relative to the set of 60 subimages for each class. The recognition rate indexed by the dimensionality of the universal eigenspace is plotted in Figure 11 (c). For the case of a 30-dimensional eigenspace, the global recognition rate is about 95%, when for class 1 is attained a recognition rate of 100%, class 3 is classified with an error smaller than 4%, and for class 2 the recognition rate is about 87%. Class 2 is the most problematic to be classified, due in part to the non-planarity of the fingertip.

6. Image Texton Alternative

Although the image texton method works well when inter-class separation is large, there are several drawbacks to this approach. Clustering the feature vectors (filter outputs) in a high dimensional space is difficult and the results are highly dependent on the prechosen number of clusters. Furthermore, pixels which have very different filter responses are often part of the same perceptual texture primitive.

Consider the texture primitive needed to characterize structure in a textured region such as skin pores (see Figure 8). For this task, the local geometric arrangement of intensity edges is important. However, the exact magnitude of the edge pixels is not of particular significance. In the clustering approach, two horizontal edges with different gradient magnitude may be given different labels and this negatively affects the quality of the texture classification.

One solution to this issue is a representation that is tuned to common edges regardless of the magnitude of the filter response. Specifically, the index of the filter with the maximal response is retained as the feature for each pixel. The local configuration of these nonlinear features is a simple and useful texture primitive. The dimensionality of the texture primitive depends on the number of pixels in the local configuration and can be kept quite low. No clustering is necessary as the
7. Face Texture Recognition

Face recognition has numerous applications for user interfaces and surveillance. Many face recognition systems use overall structure of the face and key facial features such as the configuration and shape of the eyes, nose and mouth. However, fine-scale facial details provide an interesting basis for recognition. Twins will have different skin imperfections and markings. These details that humans may not consciously use in recognition become an additional fingerprint for identification. We summarize the facial recognition experiment in [12] here.

For face texture recognition, we use features that are the maximal response texture primitive. For this experiment, skin texture images from all 20 subjects are used. The imaged face locations are the forehead, chin, cheek and nose. Each location on each subject is imaged with a set of 32 combinations of imaging angles, therefore the total number of skin images employed during the experiments is 2496 (18 subjects with 4 locations on the face, 2 subjects with 3 locations on the face, 32 imaging conditions per location).

Color is not used as a cue for recognition because we are specifically studying the performance of texture models. The filter bank consists of five filters: 4 oriented Gaussian derivative filters, and one Laplacian of Gaussian filter. These filters are chosen to efficiently identify the local oriented patterns evident in skin structures. The filter bank is illustrated in Figure 13 (a). Each filter has size 15x15. Define several types of texture primitives by grouping maximal response indices corresponding to nine neighboring pixels. Specifically, define five types of local configurations, denoted by Pi, i=1..5, and illustrated in Figure 13 (b). Featureless regions are assigned a separate index F0, which corresponds to pixels in the image where the filter response are weak, that is, where the maximum filter response is not larger than a threshold. Therefore the texture primitive can be viewed as a string of nine features, where each feature can have values in the set {0,...,5}. A comprehensive set of primitives can be quite extensive, therefore the dimensionality of the primitive histogram can be very large. Hence the need to prune the initial set of primitives to a subset consisting of primitives with high probability of occurrence in the image. Also, this reasoning is consistent with the repetitiveness of the local structure that is characteristic property of texture images.
Creating the set of symbolic primitives

Training

Classification

Figure 12. The recognition method based on symbolic primitives. First, a set of representative symbolic primitives is created. During training a skin primitive histogram is created for each image, while the recognition is based on a single novel texture image of unknown imaging conditions.
We construct the set of representative symbolic primitives by using 384 images from 3 randomly selected subjects and for all four locations per subject. We first construct the set of all symbolic primitives from all 384 skin images, then we eliminate the ones with low probability. The resulting set of representative symbolic primitives is further employed for labeling the images, and consequently to construct the primitive histogram for each image. Figure 14 exemplifies five instances of images labeled with various symbolic primitives. The left column illustrates the original images, while the right column presents the images with pixels labeled by certain symbolic primitives (white spots). The first row shows pixels in the image labeled by primitives of type P1, where all filters are horizontally oriented. The second row shows images labeled with primitives of type P3, where the filters are oriented at -45°. Notice that the symbolic primitives successfully capture the local structure in the image.

We use skin images from forehead, cheek, chin and nose to recognize 20 subjects. For each subject, the images are acquired within the same day and do not incorporate changes with aging. A human subject is characterized by a set of 32 texture images for each face location, i.e. 128 images per subject.

Classification is achieved based on a set of four testing images, one for each location (forehead, cheek, chin and nose). Each of the four test images is labeled by a tentative human identification, then the final decision is obtained by taking the majority of classes. The training and testing set are disjoint with respect to imaging parameters (the training images are obtained from different viewing and illumination direction from the test images). Knowledge of the actual viewpoint and illumination direction is never needed in the recognition. In practice this is important because the test (and training images) can be from arbitrary lighting direction and viewpoint which is far more convenient than trying to precisely align the light source and human subject.

To test recognition performance, the number of images in the training set for each subject and location is varied. The remainder of the images is used for the test set. Specifically, the training set is varied from 16 to 26 texture images for each subject and face location, and the final testing is on the remaining four subsets of 16 to 6 texture images per subject and location. The global recognition rate for this experiment reaches 73%. This result suggests that human identification can be aided by including skin texture recognition as a biometric in addition. This experiment uses skin texture alone, which is quite difficult and not typically necessary. In face recognition applications, a combination of recognition based on overall face structure with the addition of facial texture recognition is desirable.
Figure 14. Two instances of images labeled with various texture primitives. The left column illustrates the original images, while the right column presents the image with pixels labeled by certain symbolic primitives (white spots). Specifically, the first row shows pixels in the image labeled by primitives of type P1, where all filters are horizontally oriented; the second row shows images labeled with primitives of type P3, where the filters are oriented at $-45^\circ$. Notice that indeed the symbolic primitives successfully capture the local structure in the image.

8. Summary

Surface appearance is often not well described by geometry and simple shading models, especially when the surface exhibits fine-scale height variation or relief texture. When detailed appearance is needed but geometry/shading does not provide sufficient accuracy, the bidirectional texture function is an appropriate surface descriptor. Since the BTF is image-based, fine-scale surface geometry is not captured. Effects like shadowing, occlusions and foreshortening are encapsulated as part of an “effective reflectance” when using the BTF. Consequently, the BTF is not the same representation as a BRDF applied to an exact geometric surface profile.

The set of images that comprise a sampled BTF can be used to build texture models such as the bidirectional feature histogram for recognition. Learned vocabularies of local intensity variations, i.e. image textons, are built by clustering feature outputs. An alternative is to look at a local geometric configuration of maximal filter responses. The BTF representation has been used in recognition, rendering and representation of surfaces. Because of the large amount of data required for densely sampling the BTF, efficiency in algorithms and conciseness in representation remains an ongoing research effort.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0092491 and Grant No. 0085864. The unwrapped texture image Figure 4 was generated in collaboration with Dongsheng Wang and Dinesh Pai.
Bibliography


[44] University of oulu texture database.

[45] Heriot watt photometric texture database.


