Synthesis of Reflectance Function Textures from Examples

Yacov Hel-Or [1,2] Tom Malzbender [1] Dan Gelb [1]

[1] Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA., U.S.A.
[2] On sabbatical from the Interdisciplinary Center, School of Computer Science, Kanfey Nesharim St., Herzliya, Israel.

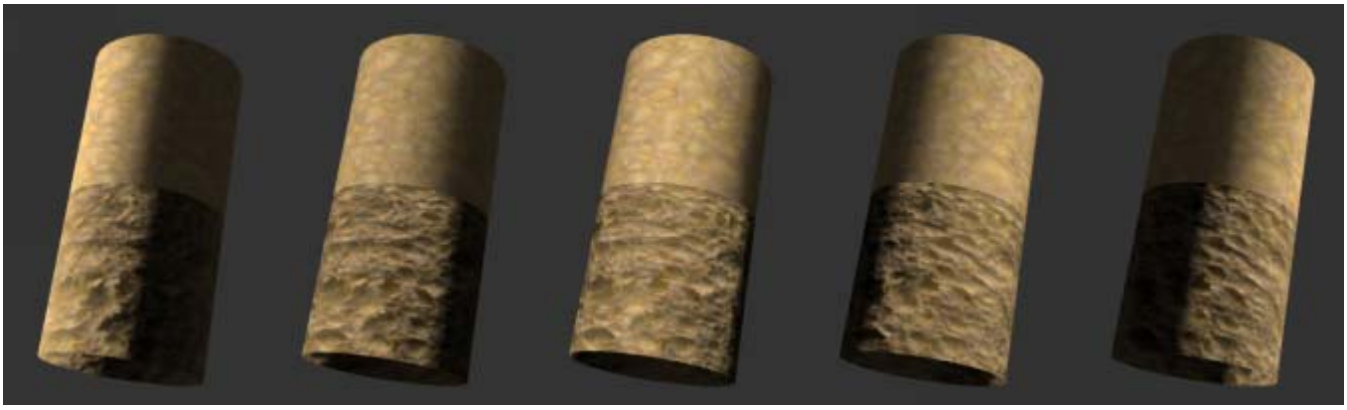

Figure 1. Bottom: Synthetic sponge material with varying reflectance function per textel.
Top: Synthetic sponge without reflectance functions (conventional texture map).

Abstract

We extend the machinery of existing texture synthesis methods to handle texture images where each pixel contains not only RGB values, but reflectance functions. Like conventional texture synthesis methods, we can use photographs of surface textures as examples to base synthesis from. However multiple photographs of the same surface are used to characterize the surface across lighting variation, and synthesis is based on these source images. Our approach performs synthesis directly in the space of reflectance functions and does not require any intermediate 3D reconstruction of the target surface. The resulting synthetic reflectance textures can be rendered in real-time with continuous control of lighting direction.

1. Introduction

The characterization of real world textures and surfaces is an important aspect of enabling photorealistic rendering. Early methods [Perlin 85, Peachy 85] relied on the ingenuity of the programmer to develop procedural models that approximate the appearance of real world materials. Unfortunately, the procedural representation did not lend itself to the automatic characterization of surface textures from photographs. This ability is especially useful when synthesis of new instantiations of textures can be performed from such a characterization. Powerful texture synthesis methods were developed in the late 80's and 90's that are able to synthesize new texture samples from photographic examples. Discussion of these methods can be found in section 2.2. These methods have practical ramifications for 3D computer graphics since they can simplify the texture mapping process in several ways. First, larger patches of a texture can be produced, yielding more source material for the texture mapping process. Second, textures may be synthesized with periodic boundary conditions, producing textures that can be seamlessly tiled, with the high image quality of photographs.

Although powerful, these texture synthesis methods have limitations that we address in this paper. Since the example photographs used as input to the texture synthesis algorithms are captured under specific lighting conditions, the synthesized textures have these same lighting conditions 'baked in'. Although the results will be convincing when those lighting conditions match the lighting conditions that the texture patch finds itself on the 3D object, in general this will not be the case. In fact, when a single texture map is wrapped around even a simple 3D object, opposite sides of the object experience completely reversed lighting directions in the local coordinate system of the texture map. Due to spatial variation in the surface mesostructure and microgeometry, the modifications in surface appearance due to these changes in local illumination are poorly approximated by attenuating the surface intensity due to the 3D object geometry. Figure 1 demonstrates this behavior, when rendering results using a light-dependent texture model are compared to the rendering of the same texture modeled without lighting dependency.

Image-based re-lighting methods [Debevec 00, Malzbender 01, Ashikhmin 02, Debevec 02] provide a solution to this quandary. In this approach, multiple photographs of a surface, person or object are taken under varying lighting conditions and static pose, and a reflectance model characterizing the surface appearance under changing lighting conditions is constructed. With enough resolution spatially and in lighting direction, very realistic renderings of the original can be produced under arbitrary lighting conditions.

In this paper we demonstrate how the image-based representation of polynomial texture maps (PTMs) [Malzbender 01] can be directly leveraged for the purposes of synthesizing textures that

have real-time continuous lighting control. These synthetic textures can be produced having the practical advantages of tileability and photorealism that texture synthesis methods provide, combined with the lighting control of image-based reflectance functions. The machinery that we develop in this paper to directly compare pixels of reflectance functions, instead of pixels of color values, can also be leveraged to extend any texture synthesis method comparing color values.

2. Surface Reflectance Characterization

The reflectance properties of a homogeneous opaque material can be exhaustively specified by its Bidirectional Reflectance Distribution Function (BRDF), introduced in [Nicodemus 77]. The BRDF measures the ratio of radiance L exiting a surface at direction (ϕ_e, θ_e), to the incidence irradiance I striking the surface in a differential solid angle from direction (ϕ_i, θ_i):

$$BRDF(\phi_i,\theta_i,\phi_e,\theta_e,\lambda) = dL(\phi_e,\theta_e) \Big/ dI(\phi_i,\theta_i) \qquad (1)$$

Note that the BRDF also depends on illumination wavelength λ, which is often integrated over r,g,b sensitivity functions.

Although BRDF functions adequately characterize homogeneous smooth materials, they fail to specify 2D texture characteristics due to the texture's spatial variations. The introduction of spatial variation to the BRDF is required to adequately model complex textures for many computer graphics applications. The Bidirectional Texture Function, BTF, introduced in [Dana 99a], provides such a framework:

$$BTF_{r,g,b}(\phi_i,\theta_i,\phi_e,\theta_e,u,v) \qquad (3)$$

where the spatial variation is indexed by (u,v). The BTF can efficiently specify 3D texture surfaces, whose characteristics arise from spatial variations of both albedo and surface normal. Thus, the BTF function can implicitly characterize appearance effects such as shading, shadows, self occlusions, inter-reflections, mutual shadowing, etc. However, storage requirements for either the BTF or the BRDF can be prohibitive for real-time computer graphics applications, due to the high dimensionality (6 and 5 respectively) of both representations.

In this paper we restrict ourselves to a less general, but more tractable representation we called the *Unidirectional Texture Function*, or UTF. Unlike the two previous reflectance representations discussed, the UTF includes a dependence on only one direction, namely that of the incident light:

$$UTF_{r,g,b}(\phi,\theta,u,v) \qquad (4)$$

One attraction of the UTF is that it is extremely easy to collect for a real world material. It requires only a stationary digital camera and a movable light source, and specifically does not require any camera calibration and geometric reasoning as one needs for acquiring a BTF. By sacrificing any dependence on view direction, we loose the ability to *capture* view dependent phenomena such as specular highlights and self-occlusions. However, since surface normals are easily calculated from a UTF representation (which is the angle giving the maximal UTF value) [Debevec 00] [Malzbender 01], specular highlights can be reintroduced into the UTF rendering process.

3. Previous BTF/UTF Texture Synthesis

There has been extensive work in the area of 2D texture synthesis [e.g. Efros 99,Heeger 95,Efros 01,Wei 00,Portilla 00,Debonet 97]. However, the synthesis of reflectance textures from examples is conceptually different from the 2D texture synthesis. A collection of images of a particular surface acquired under various lighting conditions cannot be treated as an independent collection of 2D textures. There are strong correlations between the sampled images, as all of them are instances of a unique underlying physical surface. These correlations have to be maintained while synthesizing a novel reflectance texture.

Relative to the volume of previous work in the area of 2D texture synthesis, there are only a handful of papers relevant to the synthesis of reflectance textures. The work of Leung and Malik [Leung 01] as well as Dana and Nayar [Dana 99b], deal with the synthesis of new images of an existing 3D texture viewed under novel viewing/lighting directions. However, synthesis of a totally new BTF is a more challenging problem. Note, that the synthesized and the example textures have 4D reflectance functions assigned to each pixel. Working explicitly with this data is computationally prohibited. Thus, some sort of dimensionality reduction prior to synthesis must be applied.

[Liu 01] uses a texture's height-field along with an albedo map as an intermediate representation for BTF. This representation is reconstructed from the texture examples, using shape-from-shading techniques. Then, a synthesis scheme is applied directly to the height-field, using non-parametric sampling [Efros 99], resulting in a representation of a novel texture from which the BTF can be derived. [Leung 01] suggests using the 3D texton map as a basis for generating a novel 3D texture. This approach is similar in spirit to [Liu 01] where a texton map is used as an intermediate compact representation. The texture's BTF can be derived from this representation similarly to the height-field map. [Tong 02] also uses the texton map representation as a basis for synthesis directly on a 3D object. However, the volume of data associated with a full BTF required small input images (64^2 – 128^2), limiting the range of textures that could be accommodated.

Although all previously suggested methods use an intermediate compact representation for BTF, none can perform texture rendering or relighting in real time. This difficulty arises due to the inefficiency of the chosen intermediate texture representations for rendering. This paper suggests a new technique for reflectance function texture synthesis, which enabling real-time lighting and viewing control. Our texture representation is that of a UTF modeled with polynomial texture maps (PTM) [Malzbender 01]. By sacrificing the view dependence of a BTF we gain a compact texture representation well matched to the rendering process, but also directly employed for synthesis. The PTMs, produced by our synthesis method can be then used in place of conventional texture maps and applied to 3D objects, providing interactive and realistic control of lighting effects, such as shading, self shadowing, interreflections, and surface scattering.

4. Non Parametric Texture Synthesis

Viewing a texture image as a realization of a homogenous Markovian process implies that the color distributions of a texture block W are completely characterized by its causal neighborhood N_w, and that this characterization is spatially invariant (Figure 2). Therefore, the conditional probability $P(W|N_W)$ completely determines the texture characteristics. Likewise, the probability that a given image is a realization of the texture process is given by:

$$P(W_1, W_2, \cdots) = \prod_i P(W_i \mid N_{W_i}) \qquad (5)$$

where $W_1, W_{2,\ldots}$ are the blocks composing the image.

In this paper we build on a collection of techniques, which are referred to as non-parametric texture synthesis. In these methods, the Markovian process parameters are not estimated but rather the process is emulated by sampling directly from the example texture. Hence, a realization of the conditional probability $P(W|N_W)$ is achieved by randomly choosing a block from amongst all blocks W_i in the texture example satisfying:

$$\sum_{u,v} \left\| N_W(u,v) - N_{W_i}(u,v) \right\|_2 < \delta \qquad (6)$$

where δ is a predefined threshold, as shown in Figure 2. The various non-parametric synthesis techniques differ mainly in the definitions of a block and its causal neighborhood. In the original work of [Efros 99] pixels are generated one at time in a raster scan, by searching in the source texture image for pixels whose neighborhoods are similar to that of the pixel being synthesized. This search is done in an exhaustive manner, yielding a slow run time. The method has been accelerated in the work of [Wei 00]. However, both of these approaches are somewhat unreliable, due to the feedback introduced by using recently synthesized pixels as neighborhoods for subsequent pixel synthesis. This problem in mitigated in [Efros 01] and [Liang 01] in which entire blocks of texture are copied from the source images into the working texture, again based on the similarity of their neighborhoods. Overlapping block regions are either alpha blended or optimal boundary cuts are calculated between each neighboring blocks, so that block stitching looks smooth. This method is called *"block-based"* texture synthesis, and although simple, it is surprisingly effective.

5. Reflectance Texture Synthesis

This paper extends the block-based method from working on images containing color values, to 'images' of reflectance functions. We view a UTF image as a texture of *functions* rather than a texture of *values*. Thus, a UTF image is regarded as a realization of a Markovian process in the spatial domain. However, the stochastic process is performed over functions rather than over values. According to this view, a function index $\psi = \Psi\{g(\phi,\theta)\}$ is assigned to each pixel reflectance function $g(\phi,\theta)$, and the function index, ψ, is regarded as a random variable, over which a stochastic process is defined.

A Markovian process over functions implies that the distribution of functions attached to a texture block W is characterized by the conditional probability:

$$P(\Psi\{W\} \mid \Psi\{N_W\}) \qquad (7)$$

where $\Psi\{W\}$ and $\Psi\{N_W\}$ are the indices of the function arrays attached to the block W and its neighborhood N_W respectively. The conditional distribution above characterizes a UTF process, which can be imitated using non-parametric synthesis similarly to the synthesis of conventional textures. The only difference is that we have to perform the neighborhood comparisons on function indices, and that copied blocks are composed of an array of function indices. Two questions remain open. First, how can we obtain a continuous representation of reflectance functions from a finite set of texture images, each at a specific lighting condition? Second, how can we attach an index for each possible reflectance function such that probability function can be easily calculated for each function index? Both of these questions can be resolved using the polynomial texture maps.

In the following section a brief description of the PTM texture representation is given. A sequent section will describe how we use this representation for 3D texture synthesis.

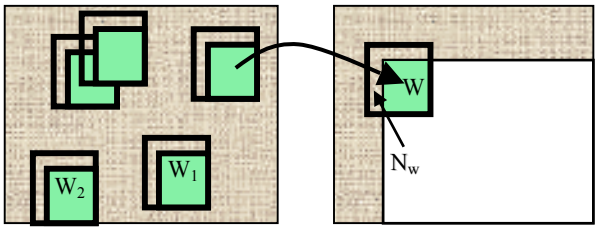

Figure 2. Copying blocks with similar causal neighborhoods

5.1 PTMs

Polynomial texture maps [Malzbender 01] provide a compact representation for reflectance functions. In this approach, a real-world surface is photographed multiple times with a fixed digital camera under varying illuminations directions $\{(l_u^i, l_v^i)\}_{i=1}^N$ providing N images: $\{L^i(u,v)\}_{i=1}^N$. Let (l_u, l_v) denotes the projection of a unit vector whose direction is (θ, ϕ) onto the (u,v) plane. The PTM representation of a texture patch describes, independently for each pixel (u,v), the luminance variation, L, as a function of (l_u, l_v). The luminance is modeled by biquadradic polynomial function in l_u, l_v:

$$\begin{aligned} L(u,v;l_u,l_v) = {} & a_0(u,v)l_u^2 + a_1(u,v)l_v^2 + \\ & a_2(u,v)l_u l_v + a_3(u,v)l_u + a_4(u,v)l_v + a_5(u,v) \end{aligned} \qquad (8)$$

where the parameters $(a_0..a_5)$ are chosen to best fit the acquired image values, thus minimizing:

$$\sum_i \left\| L(u,v;l_u^i, l_v^i) - L^i(u,v) \right\|_2$$

Color values for each pixel are recovered from this luminance model and unscaled color values $(R_n(u,v), G_n(u,v), B_n(u,v))$:

$$\begin{aligned} R(u,v) &= L(l_u, l_v, u, v) R_n(u,v); \\ G(u,v) &= L(l_u, l_v, u, v) G_n(u,v); \\ B(u,v) &= L(l_u, l_v, u, v) B_n(u,v); \end{aligned} \qquad (9)$$

Once the coefficients $(a_0..a_5, R_n, G_n, B_n)$ are estimated for each pixel, renderings of the surface for arbitrary lighting direction can be computed in real time using either pure software, or programmable graphics hardware acceleration, to map them onto 3D surfaces in a manner similar to conventional texture mapping. The biquadratic polynomial as the interpolation function is not mandatory and other bases are possible as well. Note, however, that the PTM basis functions are closely related to the low degree spherical harmonics, which are optimally spanning Lamberian surfaces illuminated under various directions [Ramamoorthi 02, Basri 03].

5.2 Reflectance Texture Synthesis using PTM

Going back to block-based 3D texture synthesis, PTM coefficients can be efficiently used as the reflectance function indices over

which texture synthesis is applied. Thus, a similar block-based synthesis scheme can be applied directly to the 9 dimensional vector field $(a_0..a_5,R_n,G_n,B_n)$. However, care must be taken: In the 2D texture synthesis, the conditional probability $P(W|N_W)$ is achieved by sampling *similar* blocks in the source texture, where similarity is defined based on the values in the causal neighborhoods (Equation 6). This scheme cannot be automatically applied to function indices in the reflectance texture case. The transformation from function space to index space does not necessarily preserve function distances, namely:

$$\iint \left\| g_1(l_u,l_v) - g_2(l_u,l_v) \right\|_2 dl_u dl_v \neq \left\| \psi\{g_1\} - \psi\{g_2\} \right\|_2$$

for any given two functions g_1,g_2. This implies also that function pdf's are not preserved, and in our case: $P(W)\neq P(\psi\{W\})$. Therefore, instead of using the original coefficients of the PTM, we linearly transform the coefficients so that an orthogonal basis is used. Since orthogonal transformation is distance preserving, distance between two functions can be measured directly in the index vectors, and consequently, function probability and its corresponding index probability can be used indistinguishably.

In our implementation we have used the 2D Legendre polynomial basis, which is orthogonal over $[-1, 1]^2$. The 6 basis functions used in the standard PTM (eq. 8), $B = \{l_u^2, l_v^2, l_u l_v, l_u, l_v, 1\}$ were transformed into the orthonormal basis B':

$$B' = \left\{ \frac{1}{2}, \frac{\sqrt{3}}{2} l_u, \frac{\sqrt{3}}{2} l_v, \frac{3}{2} l_u l_v, \frac{\sqrt{45}}{4} l_u^2 - \frac{\sqrt{45}}{12}, \frac{\sqrt{45}}{4} l_v^2 - \frac{\sqrt{45}}{12} \right\}$$

If $\boldsymbol{a}=[a_0..a_5]$ are the original PTM coefficients, the new basis coefficients, $\boldsymbol{b}=[b_0..b_5]$, are easily calculated by applying a matrix multiplication $\boldsymbol{b}=M\boldsymbol{a}$, where M is a 6x6 matrix:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & \sqrt{3}/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{3}/2 & 0 \\ 0 & 0 & 3/2 & 0 & 0 & 0 \\ \sqrt{45}/4 & 0 & 0 & 0 & 0 & -\sqrt{45}/12 \\ 0 & \sqrt{45}/4 & 0 & 0 & 0 & -\sqrt{45}/12 \end{bmatrix}$$

Using the new coefficients, each pixel (u,v) has an associated index which is constructed by the 9 dimensional vector $\psi\{L(u,v)\}=[b_0..b_5,R_n,G_n,B_n]_{u,v}$. This representation is used as described in [Efros 01]: a realization of the conditional probability $P(W|N_W)$ is achieved by randomly choosing a block from amongst all blocks W_i in the texture example satisfying:

$$\sum_{u,v} \left\| \psi\{N_W(u,v)\} - \psi\{N_{W_i}(u,v)\} \right\|_2 < \delta \qquad (10)$$

Edge handling between synthesized blocks is performed also in the orthogonal representations. In our case, the coefficients in the common boundaries were alpha blended. Since PTM functions are represented in an orthonormal space, coefficient blending is equivalent to blending the reflectance functions with similar weights. In a similar manner optimal cut along the common boundaries should be performed in the orthonormal basis as pixel comparisons are meaningful. At the final stage of the synthesis, an inverse transformation is performed to the standard PTM representation.

6. Search Strategy

The main burden of block based texture synthesis is the search required for each generated block in the synthesized PTM. For each such block, a full search is performed in the source PTM, i.e. the distance must be computed between the block neighborhood and each of the $n\times n$ neighborhoods of the source PTM image. Naively applying this search is time consuming. Several approaches have been suggested to expedite this search. Among them, multiscale search [Liang 01], tree structure vector quantization [Wei 00], and *k*-coherence search [Tong 02]. All these approaches improve run time by orders of magnitude at the expense of approximating the search results. A rejection scheme, proposed in [Hel-Or 02], can dramatically improve run time, without sacrificing the resulting accuracy. In this approach, highly dissimilar block neighborhoods in the example PTM are rejected quickly.

Each PTM block neighborhood, *represented in the orthogonal form*, is unfolded and represented as a 1D vector. Thus, if a block neighborhood is composed of p pixels, its associated 1D vector is a $9p$ dimensional vector, because each pixel includes 9 PTM coefficients. Using this notation, the error distance between two block neighborhoods N_{w1} and N_{w2} is defined as:

$$d(N_{W1},N_{W2}) = \left\| N_{W1} - N_{W2} \right\|^2 = \sum_{i=1}^{k} \left[N_{W1}(i) - N_{W2}(i) \right]^2 \qquad (11)$$

where $k=9p$. However, it is possible to reject a neighborhood before evaluating all $k=9p$ sums if the error distance already exceeds a threshold δ. This threshold can be set ahead of time or can simply be the actual error distance to the best neighborhood evaluated so far. Equation 11 is still valid when N_{W1} and N_{W2} are represented in a different orthogonal basis, $\hat{N}_{W1}, \hat{N}_{W2}$. Thus,

$$d(N_{w1},N_{w2}) \geq \sum_{i=1}^{k} \left[\hat{N}_{W1}(i) - \hat{N}_{W2}(i) \right]^2 \qquad (12)$$

for any $k \leq 9p$. If we choose a basis that concentrates vector energy in the first few entries, we can achieve a tight lower bound with very few calculations. Such a representation can be calculated by applying the Singular Value Decomposition on the entire neighborhood ensemble, or by using a basis set which is known to have energy compactness for natural images, such as the DCT basis, the Harr Wavelets or the Walsh Hadamard which can be computed very efficiently [Hel-Or 02].

Using this lower bound (eq. 12), a very fast search scheme can be applied as follows: First, each neighborhood vector in the example PTM is projected onto the first basis-vector U_1, resulting in an $n\times n$ array of scalar values. Given a new block neighborhood, N_w, from the synthesized image, we project N_w onto U_1, and calculate a lower bound on the neighborhood distance for each neighborhood in the example PTM using eq. 12. Note, that this lower bound is achieved by applying a single subtraction and a single multiplication per example neighborhood. Each neighborhood, whose lower bound is above the threshold δ, is rejected and discarded in further calculations. Typically, 90% of the neighborhoods are rejected after the first projection. For the resulting neighborhoods we continue with the second basis-vector U_2, increasing the lower bound, and rejecting additional neighborhoods. This process continues with consecutive basis-vectors. After very few projections (~3), only a few candidates remain, for which the actual distances are calculated. A typical run time for synthesis of a 512x512 patch composed of 30x30 blocks is approximately 3-5 minutes using this approach, compared to 30-60 minutes for the brute force search.

7. Results

Figures 3-5 show results of reflectance texture synthesis from photographic examples collected under 50 light directions. Figure 3

shows a single source photograph along with synthesis results under varying lighting. Figures 4 and 5 demonstrate various synthesized reflectance textures applied to 3D objects, also under a number of lighting directions. Figure 1 shows the difference between reflectance texture mapping, and the conventional texture mapping. These objects can all be illuminated and viewed in real-time as shown in the associated video.

8. Conclusion

We have presented a texture synthesis method that results in the construction of images of reflectance functions instead of simply color values. This synthesis is performed directly using the representation of polynomial texture maps, thus the resulting texture maps can be rendered in real time on modern graphics hardware with parametric control over lighting direction. At the heart of our approach is the ability to compare pixels of reflectance functions directly in place of comparing pixel RGB values. This same approach allows any texture synthesis method that compares pixel colors to be extended in the analogous manner to support the synthesis of reflectance function textures.

References

[Ashikhmin 02] Ashikhmin, M., Shirley, P. "Steerable Illumination Textures", *ACM Trans. on Graphics*, Vol. 21, No. 1, Jan. 2002, pp. 1-19.

[Basri 03] Basri, R., Jacobs, D., "Lambertian Reflectence and Linear Subspaces", IEEE T-PAMI, Vol. 25, No. 2, Feb. 2003.

[Dana 99a] Dana, K., Ginneken, B., Nayar, S., Koenderink, J., "Reflectance and Texture of Real-World Surfaces", *ACM Transactions on Graphics*, Vol 18, No. 1, January 1999, pp. 1-34.

[Dana 99b] Dana, K., Nayar, S., "3D Textured Surface Modeling", *WIAGMOR Workshop, IEEE Conference on Computer Vision and Pattern Recognition,* 1999.

[Debonet 97] Debonet, J., "Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images", *SIGGRAPH '97*, pp. 361-368.

[Debevec 00] Debevec, P., Hawkins, T., Tchou, C., Duiker, H., Sarokin, W., Sagar, M., "Acquiring the Reflectance Field of a Human Face", *SIGGRAPH 2000*, pp. 145-156.

[Debevec 02] Debevec, P., Wenger, A., Tchou, C., Gardner, A., Waese, J., Hawkins, T., "A Lighting Reproduction Approach to Live-Action Compositing", *SIGGRAPH 2002*, pp. 547-557.

[Dong 02] Dong, J., Chantler, M., "Capture and Synthesis of 3D Surface Texture", Texture 2002, *Second International Workshop on Texture Analysis and Synthesis*, June 1, 2002

[Efros 99] Efros, A., Leung, T., "Texture Synthesis by Non-parametric Sampling", *IEEE International Conference on Computer Vision*, Corfu, Greece, Sept. 1999, pp. 1033-1038.

[Efros 01] Efros, A., Freeman, W., "Image Quilting for Texture Synthesis and Transfer", *SIGGRAPH 2001*, pp. 341-346.

[Heeger 95] Heeger, D., Bergen, J., "Pyramid-based texture Analysis/Synthesis", *SIGGRAPH '95*, pp. 229-238, 1995.

[Hel-Or 02] Hel-Or Y., Hel-Or H., "Real Time Pattern Matching using Projection Kernels", *The Interdisciplinary Center Tech. Report*, CS-2002-1.

[Leung 01] Leung, T., Malik J., "Represention and Recognizing the Visual Appearance of Materials using Three-dimensional Textons", *International Journal of Computer Vision*, 43(1):29-44, 2001.

[Liang 01] Liang, L., Lieu, C., Xu, Y., Guo, B., Shum, H., "Real-Time Texture Synthesis by Patch Based Sampling", *ACM Transactions on Graphics*, Vol. 20, No. 3, July 2001, pp. 127-150.

[Liu 01] Liu, X., Yu, Y., Shum, H., "Synthesizing Bidirectional Texture Functions for Real-World Surfaces", *SIGGRAPH 2001*, pp. 97-106.

[Nicodemus 77] Nicodemus, F.E., Richmond, J.C., Hsai, J.J., "Geometrical Considerations and Nomenclature for Reflectance", *U.S. Dept. of Commerce, National Bureau of Standards*, October 1977.

[Malzbender 01] Malzbender, T., Gelb, D., Wolters, H., "Polynomial Texture Maps", *SIGGRAPH 2001*, pp. 519-528, August, 2001.

[Portilla 00] Portilla, J., Simoncelli, E., "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients", *International Journal of Computer Vision*, 40(1):49-71, December 2000.

[Ramamoorthi 02] Ramamoorthi, R., "Analytic PCA Construction for Theoretical Analysis of Lighting Variability in Images of a Lambertian Object", IEEE T-PAMI, vol. 24, No. 10, October 2002.

[Tong 02] Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B.l, Shum, H., "Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces", *SIGGRAPH 2002*, pp. 665-672.

[Wei 00] Wei, L., Levoy, M., "Fast Texture Synthesis Using Tree-Structured Vector Quantization", *SIGGRAPH 2000*, pp. 479-488, 2000.

[Zalesny 01] Zalesny, A., Van Gool, L., "Multiview Texture Models", *IEEE Conference on Computer Vision and Pattern Recognition 2001*, pp. 615-622, Dec. 2001.

Figure 3. Left: Section of original source image under one lighting condition. Right: Synthetic texture under varying lighting directions.

Figure 4. Synthesized texture (oatmeal) under varying lighting directions.

Figure 5. Synthesized sponge, popcorn kernels, cheerios and black-eyed peas texture mapped under varying lighting directions.