An Efficient Representation of Complex Materials for Real-Time Rendering

Wan-Chun Ma[1] Sung-Hsiang Chao[1] Bing-Yu Chen[1] Chun-Fa Chang[2]
Ming Ouhyoung[1] Tomoyuki Nishita[3]

[1]National Taiwan University, Taipei, Taiwan
[2]National Tsing Hua University, Hsin-Chu, Taiwan
[3]The University of Tokyo, Tokyo, Japan

ABSTRACT

In this paper, we propose an appearance representation for general complex materials which can be applied in real-time rendering framework. By combining a single parametric shading function (such as the Phong model) and the proposed spatial-varying residual function (SRF), this representation can recover the appearance of complex materials with little loss of visual fidelity. The difference between the real data and the parametric shading is directly fitted by a specific function for easy reconstruction. It is simple, flexible and easy to be implemented on programmable graphics hardware. Experiments show that the mean square error (MSE) between the reconstructed appearance and real photographs is less than 5%.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism

General Terms

Algorithms

Keywords

Reflectance, Bi-directional Texture Function, Parametric Shading Function

1. INTRODUCTION

Homogenous materials may demonstrate simple appearance; plastics and metals are the most typical examples. Even though regarded semantically as homogeneous, real world materials often are complexes and can exhibit complicated appearance. To render this kind of materials in real-time, we need a method that is different from the present known methods, such as combination of bump, glossy, specular and diffuse maps. Before describing our ideas, we need to define what a complex material would be. The appearance of a complex material includes the following three main properties:

1. Inhomogeneity: local variations caused by contaminants, irregular interior structure, such as mineral crystallization or mix of different materials.
2. Geometry: local variations caused by surface mesostructures, such as self-shadowing and inter-reflection.
3. Transparency: the transparent effect of materials which is captured should also be considered.

Under this definition, materials such as minerals, skins and clothes all belong to complex materials.

Following the above proposition, in this paper we propose an appearance representation for general complex materials which can be applied in present real-time rendering framework. Figure 1 introduces the overall framework. The representation itself is simple, flexible and easy to be implemented on programmable graphics hardware. We use two kinds of data source for testing. One is a bi-directional texture function (BTF) database and the other is some real photographs captured by ourselves. The photographs are obtained from different materials using a homemade lighting platform and used as the data of SRF representation of each material.

2. RELATED WORK

Many studies have contributed much to rendering virtual objects in high quality. An overview of these algorithms reveals two main representations for photorealistic rendering: image-based and parametric-based.

2.1 Image-Based Methods

Image-based methods create vivid imagery without explicit knowledge of geometry or reflectance properties. Classic image-based rendering (IBR) [2] uses a large amount of 2D images of different views to generate the illusion of 3D scenes (object movies or panoramas). One may traverse the scenes by directly changing, interpolating, or warping between these images. Most of the early stage object movies are based on fixed lighting, which means it is impossible to change lighting conditions. Many attempts have been made

VRST'04, November 10-12, 2004, Hong Kong.

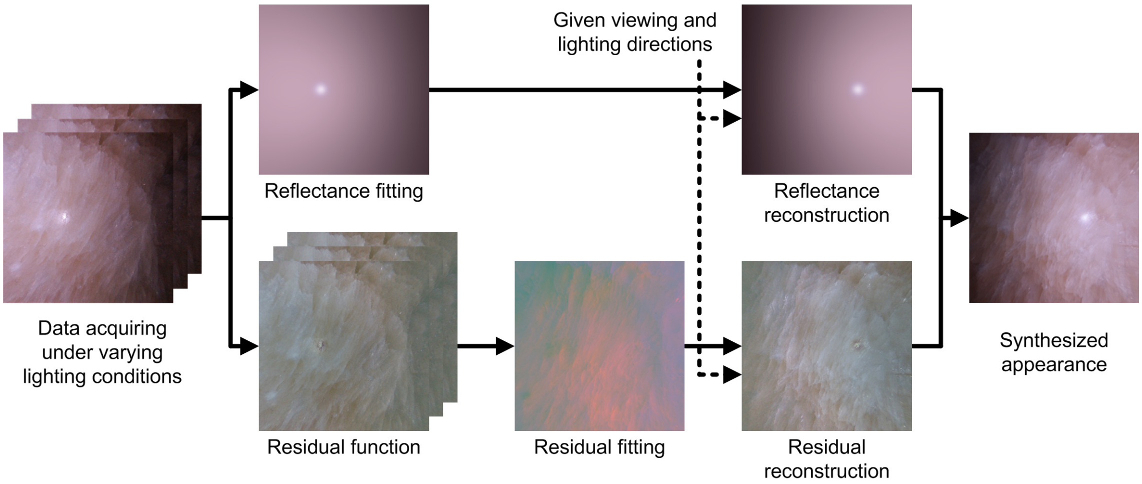

Figure 1: Overview of the framework. First we use a parametric shading function to fit the captured read data, and then the difference (residual) between the real data and the parametric shading is directly fitted by a specific function. Given viewing and lighting directions, the appearance can be synthesized with the shading function and the reconstructed residual. The value of the residual function is shifted so that it can be displayed.

to solve this problem, such as [9]. Although they may produce rendering under varying lighting condition, the viewpoint remains fixed. It is possible, though exhausting, to acquire an object movie under various lighting conditions. However, the accompanying tremendous storage need and management problems make it impractical.

In the last decade, more IBR representations are proposed. The surface light field (SLF) [3] is a function that outputs appearance color to each viewing direction from a specific surface location. The SLF can well represent the object appearance under complex (but fixed) lighting conditions. The polynomial texture map (PTM) [9] is a special case of image-based representation. A PTM approximates the sequence of input images which are captured under varying lighting condition using biquadratic polynomials, so only the fitted polynomial parameters are stored in PTM.

The BTF proposed by Dana et al. [4] is a pioneering work in representing complex surface appearance under various lighting conditions and viewpoints in a manner similar to traditional texture map. Due to the high dimensionality of BTF, it requires huge memory space for storage. Therefore, how to efficiently manipulate BTFs becomes an issue. Methods such as principle component analysis (PCA), factorization or vector quantization are frequently adopted to preprocess the data for better run-time efficiency.

2.2 Parametric-Based Methods

In contrast to image-based, parametric-based methods emphasize the use of physically-based parametric reflectance models, which are effective abstraction for describing how the light is reflected from a surface. By statistically fitting the measured data (by either dense or sparse sampling), these models generally provide fair approximations of surface appearance of arbitrary homogeneous materials. From another point of view, fitted models compress the measured data to an extremely small size (only a few parameters for each color channel). Nowadays, many parametric reflectance models are explored and widely adopted. These parametric BRDFs (bi-directional reflectance distribution function) can be either physically-based or empirical.

However, a single parametric model cannot handle appearance generated by complicated surface properties such as transparency or inhomogeneity. Thus, different methods such as bi-directional surface scattering reflectance distribution functions (BSSRDFs) [6] or spatially-varying BRDFs (SBRDF) [8] are proposed to solve these problems. Recently, photometric stereo is adopted by parametric-based methods for surface reflectance recovering [5]. These methods can handle SBRDFs and allow for rendering under different viewing and lighting conditions. Impressive synthesis images were shown in their results. Compared to image-based approaches, which need up to hundreds of images, parametric-based methods require much fewer images (mainly for the fitting process).

We propose a method by combining the advantages of the above two approaches: using a reflectance model to represent the approximated appearance of an arbitrary material and a residual function analogous to image-based methods.

3. DATA REPRESENTATION

3.1 Data Sources

The data sources we need are BTFs. We capture our own BTFs by our simple homemade platform. We also obtain accurate BTFs directly from a BTF database provided by University of Bonn[1]. BTF is a fundamental representation of complex appearance under varying lighting and viewing directions. We try to decompose the BTF into the following two terms, a parametric reflectance function f_r and a residual function δ:

$$f_{BTF}(P,V,L) = f_r(V,L) + \delta(P,V,L), \tag{1}$$

where P indicates a position on the surface and V and L are viewing and lighting directions, respectively.

---

[1] http://btf.cs.uni-bonn.de/

3.2 Reflectance Model Fitting

We choose the Phong model with Blinn's specular highlight [1] as the parametric reflectance function f_r in Equation 1:

$$f_{Phong}(V, L) = \kappa_d \times (N \cdot L) + \kappa_s \times cos^n(N \cdot H), \quad (2)$$

where V is the viewing direction, L is the lighting direction, N is the surface normal, H is the halfway vector defined as $(L + V)/\,\|L + V\|$, and n is the shininess parameter. The model itself is isotropic and only has three parameters: κ_d, κ_s, and n. The view-independent terms, such as the ambient, are combined with the diffuse. Although there exist many other BRDF that are also plausible, the main reason why we adopt the Phong model is that it is simple and can be quickly evaluated.

Similar to [10], each pixel in the input reflectance maps is treated as a reflectance measurement with different viewing and lighting directions. A non-linear optimization is used to fit the reflectance model to the data. As many previous studies [8, 7], we take the Levenberg-Marguardt algorithm as the optimization method. For each reflectance map of a specific lighting direction, a set of Phong parameters can be retrieved. Finally, we average all the sets to get a single Phong parameter set. For the appearance data from a BTF, we first calculate its BRDF by averaging the intensity of each BTF images, then the same fitting process is applied to get the Phong parameters.

3.3 Spatial-Varying Residual Function

From the Equation 1, we can get the δ function as

$$\delta(P, V, L) = f_{BTF}(P, V, L) - f_r(V, L). \quad (3)$$

We name the δ function as the spatial-varying residual function (SRF). In the implementation, the SRF is obtained by subtracting the original reflectance maps (or BTF images) from the reconstructed Phong shading.

3.4 Residual Fitting

Here, we use the δ^* function which is similar to the Blinn's specular component to fit SRF:

$$\delta^*(P, V, L) = s(P) \cdot H. \quad (4)$$

However, you may apply any kind of δ^* function to fit the residual data. The only purpose of δ^* function is to approximate δ function. Again, H is the halfway vector. To solve the unknown s, for each position P on the reflectance map, the following linear system is formed:

$$\begin{bmatrix} H_{1x} & H_{1y} & H_{1z} \\ H_{2x} & H_{2y} & H_{2z} \\ \vdots & \vdots & \vdots \\ H_{nx} & H_{ny} & H_{nz} \end{bmatrix} \begin{bmatrix} s(P)_x \\ s(P)_y \\ s(P)_z \end{bmatrix} = \begin{bmatrix} \delta(P, V_1, L_1) \\ \delta(P, V_2, L_2) \\ \vdots \\ \delta(P, V_n, L_n) \end{bmatrix}.$$

Then we can get the s by a least square operation. For each SRFs in three different color channels the same process is performed. Hence, we may retrieve s_r, s_g and s_b for each P, respectively.

The meaning of s is related to normal perturbation in a bump map but differs with following aspects:

1. One may imagine that s is a bump-like map which contains both normal perturbation and color variations. Bump map only stores normal perturbation of the surface meso-structure.
2. However, s is a fitted result from real data and is only used to reconstruct the SRF. It has no definite and strong physical meaning. The SRF may contains information about translucency, self-shadowing and inter-reflection (since all the data are captured from real scenes). If well-reconstructed, it could exhibit better appearance than the bump maps.

4. RENDERING

The reconstruction of appearance via SRF is simple. For each position P on the surface, we need to transform the lighting and viewing directions into tangent space. Finally, the color of a pixel is calculated as

$$f^*_{BTF}(P, V, L) = f_{Phong}(V, L) + \delta^*(P, V, L). \quad (5)$$

5. RESULTS

The data manipulation and rendering process are implemented in MATLAB and OpenGL (with NVIDIA Cg), respectively. The execution platform is a desktop PC with an Intel Pentium 4 2.4GHz CPU, 512MB memory, and a NVIDIA GeForce FX5600 GPU with 128MB video memory. Both of the vertex and pixel shaders are complied in OpenGL NV30 profiles. Real-time rendering results of different materials are shown in Figures 2 and 3. The window size of the rendering system is 600 × 600 and the refresh rate is averagely more than 20 frames per second (FPS). However, due to the algorithm is fill-limited, the run-time performance still varies with the pixel number that the projected image takes.

5.1 Visual Effects

The SRF representation effectively captures appearance such as self-shadowing and shading variation, which are shown in Figure 2. The major deficiency is that the use of $s(P) \cdot H$ as the δ^* function may bring blurring effect, which is visible at the shading discontinuity, such as the shadow boundaries. Despite of this problem, the SRF representation preserves the low-frequency shading variation well and reveals the feel of 3D texture successfully, especially that it is done by an inexpensive method. Notice in Figure 2, which demonstrates a sphere made by polished translucent stone, the images reveal the shading variation of the vertical crystal cracks inside the sphere object. These kinds of appearance changes cannot be modeled by traditional bump mapping techniques, which only encodes surface normal perturbation.

5.2 Comparison

This method is an extension of PTM and bump mapping. It is quite different from the eigen-based approaches. It also differs from the SBRDF representation (e.g. diffuse + specular + glossy maps [1]) which does not encode self-occurred shading effects (e.g. occlusion and shadowing). SBRDF representation needs bump mapping techniques to enhance the rendering quality. The proposed technique integrate multiple shading effects, and can be applied to most of the present graphics hardware.

6. CONCLUSIONS AND FUTURE WORK

An inexpensive but effective representation for general complex materials and how to apply it in real-time rendering framework is described in this paper. Comparing to the

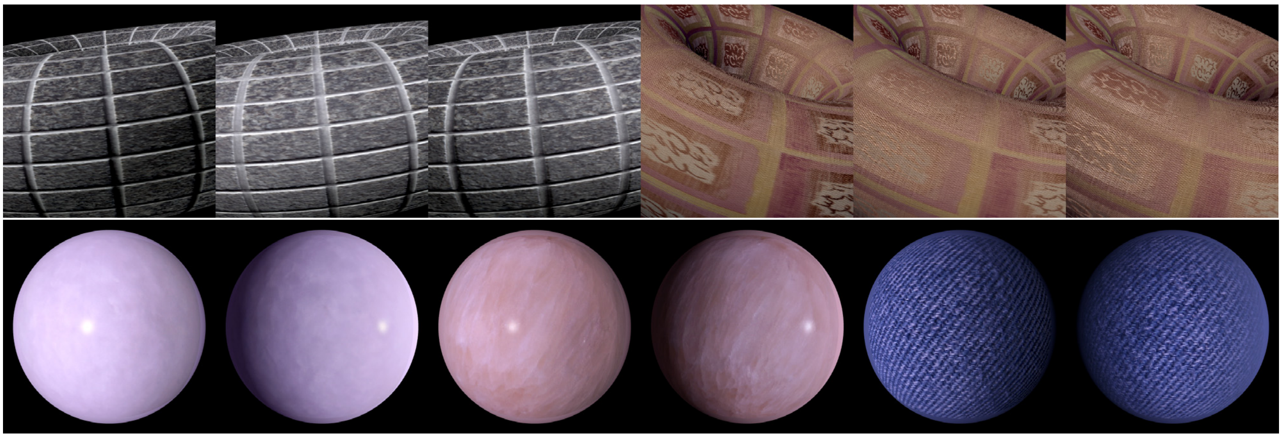

Figure 2: Upper: demonstration of self-shadowing (left) and shading variation (right) effects. Lower: Rendering results of different materials: jade, stone, and jeans (from left to right), lit in different lighting direction.

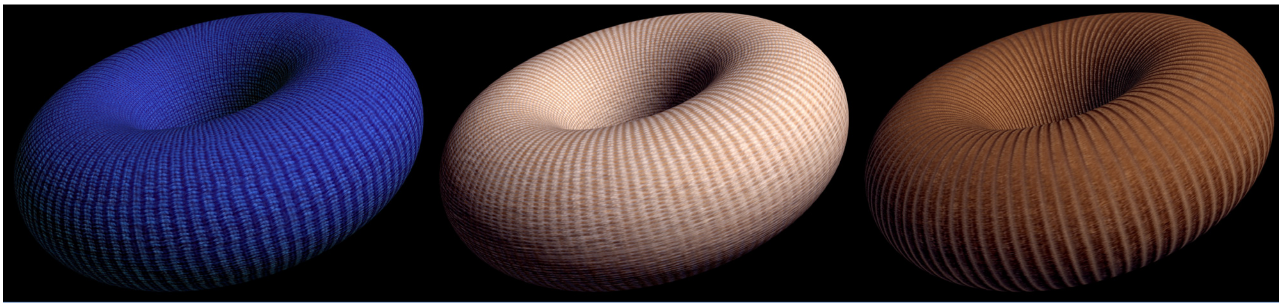

Figure 3: Rendering results of different materials from BTFs: wool, upholstery, and corduroy (from left to right).

method that directly fits a function to the real data, the use of the residual function may further reduce the reconstruction errors. The planned future improvements are included but not limited to the following items:

1. Try to make further analysis to find another δ^* function for better reconstruction.
2. Extend the specular reflectance model to an anisotropic one so that we can model the appearance of hairs, furs, CD-ROM disks, etc.

7. ACKNOWLEDGMENTS

We would like to thank the Digimax Studio[2] for providing valuable comments on Pixar RenderMan shader development. This work was partially supported by grants from National Science Council, Taiwan, R.O.C. under NSC91-2213-E-002-066, Silicon Integrated Systems (SiS) Education Foundation, and Interchange Association (Japan).

8. REFERENCES

[1] T. Akenine-Moller and E. Haines. *Real-Time Rendering, 2nd Ed.* A. K. Peters, 2002.

[2] S. E. Chen. Quicktime vr: An image-based approach to virtual environment navigation. In *SIGGRAPH 1995*, pages 29–38.

[3] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. In *SIGGRAPH 2002*, pages 447–456.

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

[5] A. Hertzmann and S. M. Seitz. Shape and materials by example: A photometric stereo approach. In *CVPR 2003*, volume 1, pages 533–540.

[6] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH 2001*, pages 511–518, 2001.

[7] E. P. Lafortune, S.-C. Foo, K. E. Torrance, , and D. P. Greenberg. Non-linear approximation of reflectance functions. In *Proc. SIGGRAPH 1997*, pages 117–126, 1997.

[8] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM TOG*, 22(2):234–257, 2003.

[9] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *SIGGRAPH 2001*, pages 519–528.

[10] W. Matusik, H. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. In *SIGGRAPH 2003*, pages 759–120.

---

[2] http://www.digimax.com.tw/