AN INVERSE PROBLEM APPROACH TO BRDF MODELING

Kei Iwasaki, Yoshinori Dobashi
Wakayama University, Hokkaido University, Japan

Fujiichi Yoshimoto and Tomoyuki Nishita
Wakayama University, The University of Tokyo, Japan

Keywords: BRDF(Bidirectional Reflectance Distribution Function), Material Editing, Inverse Problem, Image-based Lighting.

Abstract: This paper presents a BRDF modeling method, based on an inverse problem approach. Our method calculates BRDFs to match the appearance of the object specified by the user. By representing BRDFs by a linear combination of basis functions, outgoing radiances of the object surface can be represented using basis functions. The calculation of the desired BRDF results from calculating the corresponding coefficients of basis functions that minimize the sum of differences between the outgoing radiances, represented using basis functions and user specified radiances. The properties that BRDFs must satisfy are described by linear constraint conditions. This minimization problem can be solved, interactively, using a linearly constrained least squares approach. Thus, our method allows the user to design BRDFs directly, under fixed complex lighting and viewpoint, and to view the rendering results interactively, under dynamic lighting and viewpoint.

1 INTRODUCTION

The research into realistic image synthesis is one of the most important research topics in computer graphics. The illumination, incident on the objects, and the reflectance property, described as Bidirectional Reflectance Distribution Function (BRDF), are important elements in rendering realistic images of objects. For the applications such as industrial designs, commercial films, movies, and games, designers or directors often modify the illumination information and/or the reflectance properties of object surfaces by trial and error in order to obtain the desired visual effects. In order to modify the illumination information, several methods (Poulin and Fournier, 1992; Schoeneman et al., 1993; Sloan et al., 2002; Ng et al., 2003; Hasan et al., 2006; Okabe et al., 2007; Pellacini et al., 2007) have been proposed.

In recent years, methods of editing BRDFs under fixed illumination have been proposed (Ben-Artzi et al., 2006; Ben-Artzi et al., 2008). Their methods decompose the BRDFs into the products of functions that are represented by 1D parameter. By adjusting the parameter, interactive editing of BRDFs under environment illumination can be achieved. Although these methods can edit BRDFs interactively, trial and error adjustments of parameters are required to obtain the images that the user requires.

This paper presents a BRDF modeling method to permit matching of the desired appearance of objects under distant lighting represented by an environment map. Given a fixed illumination condition and a desired appearance, our method automatically calculates the desired BRDF, based on an inverse problem approach. This method represents BRDFs by a linear combination of basis functions. A color of a pixel, corresponding to a point on the object surface, can be represented using basis functions and coefficients corresponding to basis functions. The modeling of BRDFs to match the desired image results from solving an optimization problem with respect to coefficients to minimize the difference between the user specified radiances and the radiances calculated from the coefficients and the basis functions. In our method, each object possesses a unique BRDF, but does not deal with spatially varying BRDFs. The method deals with all-frequency lighting, represented by an environment map. The method does not modify the illumination conditions, (for example, adding local point light sources to add highlights), since the modification of the illumination conditions can affect

Iwasaki K., Dobashi Y., Yoshimoto F. and Nishita T. (2009).
AN INVERSE PROBLEM APPROACH TO BRDF MODELING.
In *Proceedings of the Fourth International Conference on Computer Graphics Theory and Applications*, pages 129-136
DOI: 10.5220/0001784601290136

all the synthesized objects, and therefore, it is difficult to modify the appearance of a particular object.

This paper is organized as follows. Section 2 reviews previous work. Section 3 describes our inverse BRDF modeling method. Section 4 explains the implementation details. Section 5 shows some results of the rendering process, and the conclusions to this work are brought together in Section 6.

2 PREVIOUS WORK

We first review previous methods for pre-computed radiance transfer (PRT), since our method is related to PRT methods. Then previous methods of BRDF editing are discussed.

2.1 Pre-computed Radiance Transfer (PRT)

Pre-computation-based techniques have been developed for fast re-lighting. Dobashi et al. (Dobashi et al., 1995) proposed a method that used spherical harmonics to achieve fast re-lighting for interactive lighting design. Moreover, Dobashi et al. (Dobashi et al., 1996) used Fourier series to represent the intensity distributions at the surfaces of objects, illuminated by the sky. Ramamoorthi and Hanrahan (Ramamoorthi and Hanrahan, 2002) proposed a real-time rendering method for environment illumination, using spherical harmonics. Sloan et al. (Sloan et al., 2002) proposed a PRT technique for real-time rendering under environment illumination using the spherical harmonics as the basis functions. Several methods (Kautz et al., 2002; Lehtinen and Kautz, 2003; Sloan et al., 2003a; Sloan et al., 2003b; Sloan et al., 2005) have been proposed to extend the PRT method. These PRT methods basically focus on the changes in the viewpoint and the illumination at run-time. A change in the BRDF at run-time is possible, but a large volume of pre-computed BRDF data is required.

Ng et al. (Ng et al., 2003) used wavelet basis functions for a non-linear lighting approximation and achieved interactive rendering for diffuse surfaces. Several methods (Wang et al., 2004; Liu et al., 2004; Tsai and Shih, 2006) have been proposed to render glossy surfaces. Although these methods can create realistic images in an all-frequency lighting environment, they require pre-computed BRDF data and rendering it is quite difficult to arbitrarily change BRDFs at run-time. Okabe et al. (Okabe et al., 2007) proposed an appearance-based illumination design system based on PRT techniques. This method, however, focuses on the design of illumination, not BRDFs.

In recent years, PRT methods for calculating dynamic BRDFs have been proposed. Sun et al. (Sun et al., 2007) proposed a PRT method for dynamic BRDFs. This method compresses the pre-computed BRDF data by using tensor decompositions, and calculates the pre-computed transfer tensors (PTT). Then interactive rendering with dynamic viewpoint, lighting and BRDFs can be achieved by using the PTTs. This method, however, requires the pre-computed BRDF data and it is difficult to change the BRDF that is not included in the pre-computed BRDF data.

2.2 BRDF Editing

Ben-Artzi et al. (Ben-Artzi et al., 2006) proposed a real-time BRDF editing method under an all-frequency illumination environment. This method can edit the BRDFs by changing the parameters, such as the glossiness of the Phong BRDF. Akerlund et al. (Akerlund et al., 2007) proposed a precomputed visibility cuts for relighting static scenes with dynamic BRDFs. These methods, however, require trial and error adjustments of the parameters, to obtain the desired images.

Colbert et al. (Colbert et al., 2006) proposed a method that models the BRDFs by drawing the highlights on a spherical canvas. This method approximates the highlights by using Ward BRDFs (Ward, 1992). This method, however, cannot create highlights, directly, on a user-specified region of the object surface.

Ashikhmin et al. (Ashikhmin et al., 2000) proposed a method to generate BRDFs by inputting a 2D micro-facet orientation distribution. Jaroszkiewicz and McCool (Jaroskiewicz and McCool, 2003) proposed a method to extract BRDFs and material maps from images using homomorphic factorization. Lawrence et al. (Lawrence et al., 2006) proposed an inverse shade tree framework that decomposes spatially varying BRDFs (SVBRDFs) into 1D editable functions. These methods, however, are limited to a point light source. Pellacini et al. (Pellacini and Lawrence, 2007) proposed an editing method for spatially and temporally varying BRDFs. This method, however, does not consider the complex illumination. Khan et al. (Khan et al., 2006) proposed an image-based material editing method that transforms a photograph of an object into a photograph of the object whose material is changed. Although this method can change the appearance of the object surface, this method cannot calculate the optimal BRDF to mach the images that the user requires. Ngan et al. (Ngan et al., 2006) proposed a visual navigation system of analytical BRDF models under complex lighting.

(a) initial state (car body is diffuse BRDF) | (b) user specifies regions (white strokes) to add highlights | (c) rendering result using modeled BRDF

Figure 1: These figures show the overview of our method. The left image (a) shows the car body with diffuse BRDF as the initial state. As shown in the center image (b), the user paints strokes where the user wants to add highlights or change colors. Our method models the BRDF so as to create highlights in the regions specified by the user. The right image (c) shows the result that is rendered by using the calculated BRDF. The computational time of the BRDF in (c) is 0.26 sec. As shown in (c), our method can obtain the BRDF to match the appearance the user requested interactively.

This method, however, does not model the desired BRDF. Recently, Pacanowski et al. (Pacanowski et al., 2008) proposed a sketch-based interface for highlight modeling. This method, however, is not applied to complex lighting. To address these problems, we propose a calculation method for BRDFs that fit the user-specified radiances by using an inverse problem approach.

3 PROPOSED INVERSE BRDF MODELING

3.1 Overview

Outgoing radiance, $B(x,\omega_o)$, at point, x, in direction, ω_o, under environment illumination is calculated from the following equation:

$$B(x,\omega_o) = \int_\Omega L(\omega_i)V(x,\omega_i)f_r(\omega_i,\omega_o)(\omega_i \cdot n(x))d\omega_i, \quad (1)$$

where Ω is the direction on the hemisphere, ω_i is the incident direction, $L(\omega_i)$ is the source radiance, $V(x,\omega_i)$ is the visibility function, $f_r(\omega_i,\omega_o)$ is the BRDF, and $n(x)$ is the normal vector at x (see Fig. 2). To simplify the notation, the dot product $(\omega_i \cdot n(x))$ is included in the visibility function $V(x,\omega_i)$, and $\tilde{V}(x,\omega_i)$ is defined as $\tilde{V}(x,\omega_i) = V(x,\omega_i)(\omega_i \cdot n(x))$.

In our method, BRDF f_r is represented with a sum of diffuse term f_d and specular term f_s as follows:

$$f_r(\omega_i,\omega_o) = f_d + f_s(\omega_i,\omega_o). \quad (2)$$

The diffuse term f_d is represented with $f_d = \rho_d/\pi$, where ρ_d is the diffuse albedo. The component $B_d(x)$ of the outgoing radiance corresponding to the diffuse term f_d is calculated from:

$$B_d(x) = \int_\Omega L(\omega_i)\tilde{V}(x,\omega_i)\frac{\rho_d}{\pi}d\omega_i = \rho_d E(x), \quad (3)$$

where $E(x)$ is the irradiance at x. To model the diffuse term f_d simply and efficiently, the diffuse albedo ρ_d is directly specified by the user.

Our method represents the specular term $f_s(\omega_i,\omega_o)$ by a product of a linear combination of basis functions and the static function s of the incident and outgoing directions (the static function s is described in Section 3.2):

$$f_s(\omega_i,\omega_o) = s(\omega_i,\omega_o)\sum_{j=0}^{J-1} c_j\phi_j(\omega_i,\omega_o), \quad (4)$$

where c_j is j-th coefficient, corresponding to j-th basis function, ϕ_j, and J is the number of basis functions. By substituting Eq. (4) into Eq. (1), the component $B_s(x,\omega_o)$ of the outgoing radiance corresponding to the specular term f_s is rewritten as:

$$\begin{aligned} B_s&(x,\omega_o(x)) \\ &= \int_\Omega L(\omega_i)\tilde{V}(x,\omega_i)\left(\sum_{j=0}^{J-1} c_j\phi_j(\omega_i,\omega_o)s(\omega_i,\omega_o)\right)d\omega_i \\ &= \sum_{j=0}^{J-1} c_j \int_\Omega L(\omega_i)\tilde{V}(x,\omega_i)\phi_j(\omega_i,\omega_o)s(\omega_i,\omega_o)d\omega_i. \end{aligned} \quad (5)$$

When modeling BRDFs, we assume that the viewpoint and the object surfaces are fixed [1]. Under this assumption, the viewing direction, ω_o, depends on the position, x, and is represented by $\omega_o(x)$.

In this equation, we define the transfer function $T_j(x)$ as:

$$T_j(x) = \int_\Omega L(\omega_i)\tilde{V}(x,\omega_i)\phi_j(\omega_i,\omega_o(x))s(\omega_i,\omega_o(x))d\omega_i. \quad (6)$$

Using $T_j(x)$, the component $B_s(x)$, is calculated from:

$$B_s(x) = \sum_{j=0}^{J-1} c_j T_j(x). \quad (7)$$

[1] Our method can deal with modeling the BRDFs with multiple viewpoints, by solving Eq. (7) using transfer functions of multiple viewpoints.

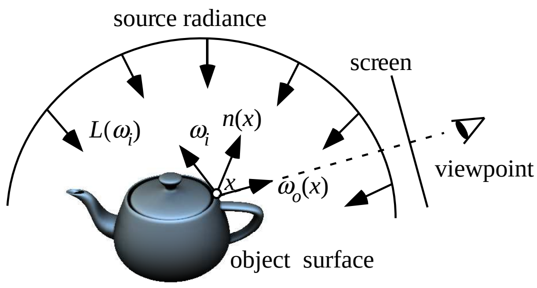


Figure 2: Outgoing radiance calculation.

Let the pixels corresponding to the object surface be $\mathbf{X} = \{x_0, x_1, \cdots, x_{N-1}\}$ (N is the number of pixels corresponding to the object surface), with the colors $\mathbf{B} = \{B_0, B_1, \cdots, B_{N-1}\}$ at these pixels. The colors are specified by the users by painting strokes, and the details are described in Section 4.3. A desired BRDF of the specular term is obtained by calculating the coefficients, c_j, for each object to minimize the following equation:

$$\arg\min_{c_j} \sum_{i=0}^{N-1} (B_i - \sum_{j=0}^{J-1} c_j T_j(x_i))^2. \quad (8)$$

In solving the optimization problem expressed by Eq. (8), we have to take into account the following BRDF properties.

- Helmholtz reciprocity
- energy conservation
- non-negativity

The Helmholtz reciprocity is represented by the following equation:

$$f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i). \quad (9)$$

The energy conservation is represented by the following equation:

$$\int_\Omega f_r(\omega_i, \omega_o)(\omega_i \cdot n) d\omega_i \leq 1 \quad \forall \omega_o. \quad (10)$$

Non-negativity states that $f_r(\omega_i, \omega_o) \geq 0$ for any ω_i and ω_o.

In the next section, basis functions that are suited to represent specular term and to satisfy the BRDF properties are described.

3.2 Basis Functions to Represent BRDFS

BRDFs are, in general, four dimensional functions (two dimensions for the incident direction and two dimensions for the outgoing direction), and require a large number of basis functions to be represented. This results in increasing the pre-computation data, thus decreasing the rendering frame rates. To represent BRDFs with a small number of basis functions, our method employs Distribution-based BRDFs (Ashikhmin, 2006). The distribution-based BRDF model is calculated from the following equation:

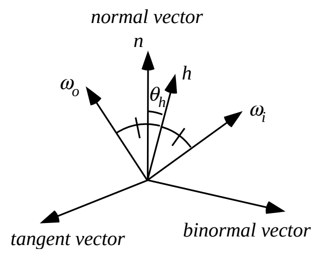


Figure 3: Half vector.

$$f_s(\omega_i, \omega_o) = \frac{p(h)F(\omega_i \cdot h)}{(\omega_i \cdot n) + (\omega_o \cdot n) - (\omega_i \cdot n)(\omega_o \cdot n)}, \quad (11)$$

where h is the half vector of ω_i and ω_o, calculated from $h = (\omega_i + \omega_o)/||(\omega_i + \omega_o)||$ (see Fig. 3). $p(h)$ is the distribution function of the half vector, $F(\omega_i \cdot h)$ is the Fresnel term, and n is the normal vector. Since h is reciprocal for any ω_i and ω_o, f_r satisfies Helmholtz reciprocity for any function p. In the distribution-based BRDF model, the reflectance property is mainly affected by the distribution function $p(h)$. Therefore, our method represents the distribution function, $p(h)$, by a linear combination of basis functions, $\phi_j(h)$, as $p(h) = \sum_{j=0} c_j \phi_j(h)$, and the other factor, $F(\omega_i \cdot h)/((\omega_i \cdot n) + (\omega_o \cdot n) - (\omega_i \cdot n)(\omega_o \cdot n))$, is the static function s in Eq. (4).

Our method represents the distribution function, $p(h)$, by basis functions, $\phi_j(h)$. In the previous PRT methods (Sloan et al., 2002; Ng et al., 2003), orthogonal basis functions, such as spherical harmonics and Haar wavelets, are used to approximate functions. In our method, orthogonality is not necessary and the method only needs a small number of basis functions to approximate the distribution function efficiently. The distribution functions for analytic BRDF models tend to have sharp peaks to represent highlights. Therefore, our method requires basis functions, $\phi_j(h)$, that can represent high-frequency functions with sharp peaks. Spherical harmonics are suited to represent low-frequency functions, but a large number of functions is required to represent high-frequency functions. We also considered Haar wavelets (Ng et al., 2003) and Daubechies wavelets (Ben-Artzi et al., 2006) that are often used to represent high-frequency functions, but Haar and Daubechies wavelets can take negative values. To satisfy the non-negativity of the BRDF, it is sufficient to satisfy condition $p(h) > 0$ for all h, since the denominator of the distribution BRDF and the Fresnel term are non-negative for any ω_i and ω_o. However, if wavelets are used to represent the distribution function, many constraint conditions for coefficients, c_j, are required to satisfy the non-negativity for any ω_i and ω_o. This results in increased calculation time of coefficients, c_j.

To represent the distribution function, $p(h)$, our method uses spherical radial basis functions (SRBF) (Tsai and Shih, 2006) as

$$p(h) \approx \sum_{j=1}^{J-1} c_j G(h \cdot \xi_j, \lambda), \tag{12}$$

where G is the Gaussian SRBF, ξ_j is the center of j-th SRBF and λ is the bandwidth parameter of SRBF.

The reasons for using SRBFs are as follows. Firstly, high-frequency functions can be represented by using a small number of SRBFs, as described in (Tsai and Shih, 2006). Secondly, the constraint conditions for the non-negativity of the distribution function can be described easily by the SRBFs. Since SRBFs are non-negative function for any half vector h, (that is for any ω_i and ω_o), the constraint condition for the non-negativity is simply described as $c_j \geq 0$.

The energy conservation condition of the sum of diffuse term f_d and specular term f_s is represented as:

$$\int_\Omega (f_d + f_s(\omega_i, \omega_o))(\omega_i \cdot n) d\omega_i \leq 1. \tag{13}$$

In Eq. (13), the integration of the diffuse term $f_d = \frac{\rho_d}{\pi}$ is computed as $\int_\Omega \frac{\rho_d}{\pi}(\omega_i \cdot n) d\omega_i = \rho_d$.

To satisfy the energy conservation condition of the specular term for any ω_o, we discretize N_l directions on the hemisphere. Then our method calculates the integration in Eq. (10) for each discretized outgoing directions, ω_o^l $(0 \leq l \leq N_l - 1)$:

$$\rho_d + \int_\Omega \sum_{j=0}^{J-1} c_j G(h \cdot \xi_j, \lambda) s(\omega_i, \omega_o^l) d\omega_i \leq 1. \tag{14}$$

In Eq. (14), we define the integrated values for each discretized outgoing direction, ω_o^l as g_j^l:

$$g_j^l = \int_\Omega G(h \cdot \xi_j, \lambda) s(\omega_i, \omega_o^l) d\omega_i. \tag{15}$$

Then the energy conservation conditions for N_l discretized directions are described as:

$$\sum_{j=0}^{J-1} g_j^l c_j \leq 1 - \rho_d \quad (0 \leq l \leq N_l - 1). \tag{16}$$

Consequently, our method calculates the coefficient vector, $\mathbf{c} = (c_0, c_1, \cdots, c_{J-1})$, that minimizes:

$$\sum_{i=0}^{N-1} (B_i - \sum_{j=0}^{J-1} c_j T_j(x_i))^2, \tag{17}$$

while satisfying the following linear constraint conditions:

$$c_j \geq 0 \; (0 \leq j \leq J-1), \tag{18}$$

$$\sum_{j=0}^{J-1} g_j^l c_j \leq 1 - \rho_d \quad (0 \leq l \leq N_l - 1). \tag{19}$$

4 IMPLEMENTATION DETAILS

4.1 Precomputation

Our method pre-computes the transfer function, $T_j(x)$, at each point, x, corresponding to each pixel of the screen. The computational time for the coefficient vector, $\mathbf{c}$, depends on the number of corresponding pixels, N (see Eq. (8)). If the screen size is 640×480, N is about 300,000 and the computational time of $\mathbf{c}$ becomes enormous. To calculate $\mathbf{c}$ interactively, our method prepares a low-resolution screen of the frame buffer. Then the least squares problem with linear constraint conditions is solved using down-sampled images. In our experiments, the size of down-sampled images is set to 160×120 for the 640×480 screen, and this works well for all our examples.

The transfer function at each pixel is then computed. To compute the transfer function described in Eq. (6), a large number of uniformly distributed directions on the sphere are prepared. Then the integration in Eq. (6) is calculated by summing the product of functions L, $\tilde{V}$, ϕ_j and s evaluated at each direction. This calculation, however, is time-consuming since the number of directions is quite large.

To compute the transfer function efficiently, our method employs the precomputed visibility cuts method proposed by (Akerlund et al., 2007). In general, the cosine-weighted visibility function $\tilde{V}$ has large coherent regions. Therefore, the directions on the sphere can be partitioned into a small number of clusters, where $\tilde{V}$ can be approximated by a piecewise constant function. The cosine-weighted visibility function in the k-th cluster C_k is approximated by the mean cluster value $\langle v_k \rangle$:

$$\langle v_k(x) \rangle = \frac{1}{|C_k|} \sum \tilde{V}(x, \omega_i), \omega_i \in C_k. \tag{20}$$

The transfer function $T_j(x)$ can be computed efficiently as:

$$T_j(x) = \sum_k |C_k| L(\omega_k) \langle v_k(x) \rangle \tilde{\phi_j}(h_k), \tag{21}$$

where ω_k is the representative direction of cluster C_k, $\tilde{\phi_j}$ is the j-th basis function multiplied with the static function, and h_k is the half vector of ω_k and viewing direction $\omega_o(x)$.

To compute the mean cluster value $\langle v_k(x) \rangle$ at surface point x corresponding to each pixel, our method precomputes each cluster value at each vertex x_v of the triangles of the object surface. Then the mean cluster value $\langle v_k(x) \rangle$ at x is computed by interpolating the precomputed cluster values at three vertices of the triangle that contains surface point x.

4.2 BRDF Modeling and Rendering

Our method calculates the coefficient vector, **c**, that minimizes Eq. (17). This coefficient vector can be obtained by solving the least squares problem with linearly constrained conditions, using Numerical Algorithm Group library (NAG, 2007). Our method also allows the users to create a BRDF that is physically impossible by ignoring the linearly constrained conditions, if the users want.

After the coefficient vector **c** is obtained, our method calculates the outgoing radiance at each vertex by using Eq. (5). To allow the users to change the viewpoint and the illumination, our method calculates Eq. (5) on the fly. To compute Eq. (5) efficiently, our method uses the precomputed visibility cuts. The outgoing radiance $B(x_v, \omega_o)$ at vertex x_v is calculated from:

$$B(x_v,\omega_o) = \sum_k L(\omega_k)\langle v_k(x_v)\rangle\left(\frac{\rho_d}{\pi} + \sum_j c_j \tilde{\phi}_j(h_k)\right). \qquad (22)$$

The calculation of the radiance at each vertex expressed in Eq. (22) is performed on the GPU as described in (Akerlund et al., 2007).

4.3 Interface

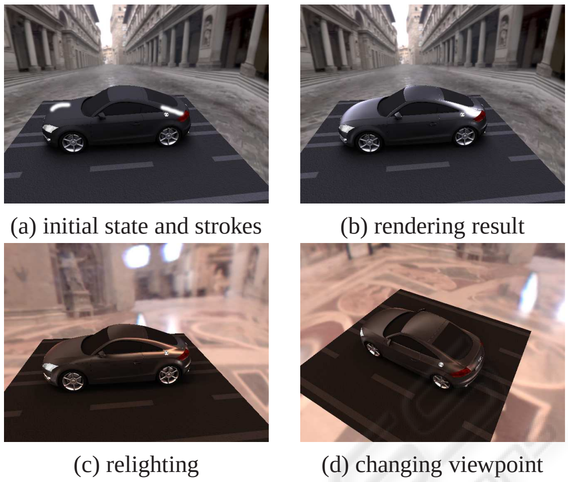
Our method specifies the colors by painting strokes on the image directly. The strokes are represented as point sequences. The brush to paint strokes has four parameters: color C, diameter D, gaussian parameter κ and scaling factor S. The specified color B_i of pixel x_i using color C is calculated from the following equation:

$$\begin{aligned} B_i &= w_i C + (1-w_i)B'_i, \\ w_i &= \begin{cases} \exp(-\kappa d_i) & d_i \leq D \\ 0 & d_i > D \end{cases}, \end{aligned} \qquad (23)$$

where d_i is the distance between x_i and the stroke, B'_i is the color of pixel x_i before painting. The users can also specify the color B_i by using the scaling factor S ($S \geq 0$) and the color B'_i as $B_i = w_i S B'_i + (1-w_i)B'$. Not only creating highlights, our method also can delete highlights by assigning negative values to C.

5 RESULTS

We have tested our algorithm on a standard PC equipped with Core2Quad 2.66GHz CPU with 3GB memory and a GeForce 8800 GTX. Fig. 4 shows images of a car model with BRDFs modeled by using our method. Fig. 4(a) shows the user specified white strokes and Fig. 4(b) shows the rendering result using the modeled BRDF that brightens the specified regions. Figs. 4(c) and (d) show the rendering results



(a) initial state and strokes (b) rendering result (c) relighting (d) changing viewpoint

Figure 4: A car model with BRDFs modeled by using our method. The precomputation time of visibility cuts is 70 minutes. The data size of precomputed visibility cuts is 123 MB. The average number of clusters is about 200..

using the BRDF in (b) under different lighting and viewpoint. Fig. 5 shows images of a bunny model. Figs. 5(a) and (c) show the initial states and strokes and Figs. 5(b) and (d) show the rendering results, respectively. The calculated BRDFs are visualized in lower left of Figs. 5(b) and (d). The yellow arrow shows the incident direction.

Fig. 6 shows images of a room scene. Highly specular BRDF is modeled in the left modern chair, and dim highlight is created in the right antique chair.

Our method sets each center of SRBF, ξ_j, to each direction corresponding to each pixel of the hemicube. That is, the number (J) of basis functions to represent the specular term is $147(3 \times 7 \times 7)$ in our examples. The bandwidth parameter λ is calculating the relationship between the variance σ and λ :$\sigma^2 = 1 - (\cosh(2\lambda) - (2\lambda)^{-1})$ (Narcowich and Ward, 1996). We set σ to $\pi/40$ and this works well in our examples. The number of the discretized direction (N_l in Eq. (16)), is set to 192.

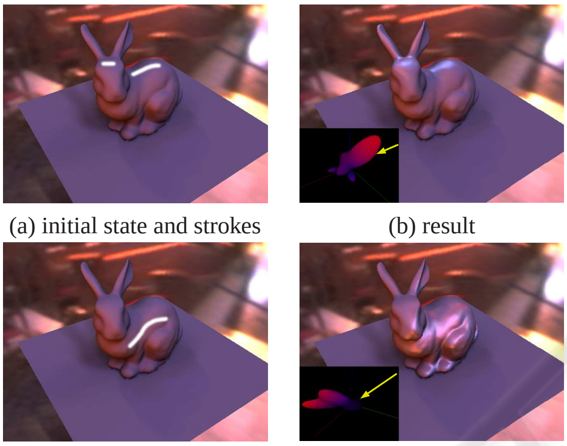
Table 1 shows the statistics of our method. As shown in Table 1, our method can show the calculated BRDF immediately. Moreover, our method allows the users to relight the scene and change the viewpoint interactively. These interactive feedbacks are helpful for the users to model BRDFs intuitively.

The limitation of our method is as follows. Although our method calculates the BRDFs as much as possible to match the appearance specified by the user, the radiances of regions where the user does not specify may also change. The regions whose transfer functions are similar to those at specified regions tend to be affected. This can be relaxed by weighting the

Table 1: Statistics of our method. T_s represents the computational time to solve a least squares problem. N is the number of pixels corresponding to the object surface in the 160×120 image. T_{trans} indicates the computational time of transfer functions at N pixels. In Fig. 6, T_s and N for sofa/left chair/cushion of the right chair are listed. T_{trans} in Fig. 6 shows the total time to compute transfer functions of three objects.

Fig. No.	Vertices	fps	T_s	N	T_{trans}
Fig. 1	77K	2.2	0.26 sec	2656	28 sec
Fig. 4	77K	2.2	0.17 sec	1797	24 sec
Fig. 5	22K	2.3	0.14 sec	1299	22 sec
Fig. 6	117K	0.7	0.22/0.11/0.04	2066/1092/460	55 sec

values of transfer functions in Eq. (17).



(a) initial state and strokes (b) result

(c) initial state and strokes (d) result

Figure 5: The bunny model with BRDFs modeled by using our method. The precomputation time of visibility cuts is 4 minutes. The data size of precomputed visibility cuts is 53 MB. The average number of clusters is about 300..

(a) initial state (b) result

Figure 6: The room scene with BRDFs modeled by using our method. The precomputation time of visibility cuts is 54 minutes. The data size of precomputed visibility cuts is 279 MB. The average number of clusters is about 332.

6 CONCLUSIONS

We have proposed an appearance-based BRDF modeling method, using an inverse problem approach. By representing BRDFs by a linear combination of basis functions, the outgoing radiances can be represented with a linear combination of transfer functions, calculated using basis functions. The method solves the optimization problem to minimize the sum of the differences between the outgoing radiances calculated from the transfer functions and the radiances specified by the user. Our method solves this problem by satisfying the BRDF properties: reciprocity, energy conservation, and non-negativity. The method allows the user to model the desired BRDF interactively and intuitively.

In future, we intend to calculate, not only the desired BRDFs, but also the optimal illumination information, to obtain desired images.

REFERENCES

Akerlund, O., Unger, M., and Wang, R. (2007). Precomputed visibility cuts for interactive relighting with dynamic BRDFs. In *Proceedings of Pacific Graphics 2007*, pages 161–170.

Ashikhmin, M. (2006). Distribution-based BRDFs. In *http://jesper.kalliope.org/blog/library/dbrdfs.pdf*.

Ashikhmin, M., Premoze, S., and Shirley, P. (2000). A microfacet-based BRDF generator. In *Proceedings of SIGGRAPH 2000*, pages 65–74.

Ben-Artzi, A., Egan, K., Ramamoorthi, R., and Durand, F. (2008). A precomputed polynomial representation for interactive BRDF editing with global illumination. *ACM Transactions on Graphics*, 27(2):13.

Ben-Artzi, A., Overbeck, R., and Ramamoorthi, R. (2006). Real-time BRDF editing in complex lighting. *ACM Transaction on Graphics*, 25(3):945–954.

Colbert, M., Pattanaik, S., and Krivanek, J. (2006). BRDF-shop: Creating physically correct bidirectional reflectance distribution functions. *IEEE Computer Graphics and Applications*, 26(1):30–36.

Dobashi, Y., Kaneda, K., Yamashita, H., and Nishita, T. (1995). A quick rendering method using basis functions for interactive lighting design. *Computer Graphics Forum*, 14(3):229–240.

Dobashi, Y., Kaneda, K., Yamashita, H., and Nishita, T. (1996). Method for calculation of sky light luminance aiming at an interactive architectural design. *Computer Graphics Forum*, 15(3):112–118.

Hasan, M., Pellacini, F., and Bala, K. (2006). Direct-to-indirect transfer for cinematic relighting. *ACM Transaction on Graphics*, 25(3):1089–1097.

Jaroskiewicz, R. and McCool, M. D. (2003). Fast extraction of BRDFs and material maps from images. In *Proceedings of Graphics Interface 2003*, pages 1–10.

Kautz, J., Sloan, P., and Snyder, J. (2002). Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proc. Eurographics Workshop on Rendering 2002*, pages 301–308.

Khan, E. A., Reinhard, E., Fleming, R. W., and Bulthoff, H. H. (2006). Image-based material editing. *ACM Transactions on Graphics*, 25(3):654–663.

Lawrence, J., Ben-Artzi, A., DeCoro, C., Matusik, W., Pfister, H., Ramamoorthi, R., and Rusinkiewicz, S. (2006). Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics*, 25(3):735–745.

Lehtinen, J. and Kautz, J. (2003). Matrix radiance transfer. In *Proc. Symposium on Interactive 3D Graphics 2003*, pages 59–64.

Liu, X., Sloan, P., Shum, H., and Snyder, J. (2004). All-frequency precomputed radiance transfer for glossy objects. In *Proc. Eurographics Symposium on Rendering 2004*, pages 337–344.

NAG (2007). Numerical algorithm group c library.

Narcowich, F. J. and Ward, J. D. (1996). Nonstationary wavelets on them-sphere for scattered data. *Applied and Computational Harmonic Analysis*, 3(4):324–336.

Ng, R., Ramamoorthi, R., and Hanrahan, P. (2003). All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics*, 22(3):376–381.

Ngan, A., Durand, F., and Matusik, W. (2006). Image-driven navigation of analytical BRDF models. In *Proceedings of Eurographics Symposium on Rendering 2006*, pages 399–408.

Okabe, M., Matsushita, Y., Shen, L., and Igarashi, T. (2007). Illumination brush: Interactive design of all-frequency lighting. In *Proceedings of Pacific Graphics 2007*, pages 171–180.

Pacanowski, R., Grainer, X., Schick, C., and Poulin, P. (2008). Sketch and paint-based interface for highlight modeling. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*.

Pellacini, F., Battaglia, F., Morley, R., and Finkelstein, A. (2007). Lighting with paint. *ACM Transactions on Graphics*, 26(2):9.

Pellacini, F. and Lawrence, J. (2007). Appwand: Editing measured materials using appearance-driven optimization. *ACM Transactions on Graphics*, 26(3):54.

Poulin, P. and Fournier, A. (1992). Lights from highlights and shadows. In *Proceedings of Symposium on Interactive 3D Graphics 1992*, pages 31–38.

Ramamoorthi, R. and Hanrahan, P. (2002). Frequency space environment map rendering. *ACM Transactions on Graphics*, 21(3):517–526.

Schoeneman, C., Dorsey, J., Smits, B., Arvo, J., and Greenberg, D. (1993). Painting with light. In *Proceedings of SIGGRAPH'93*, pages 143–146.

Sloan, P., Hall, J., Hart, J., and Snyder, J. (2003a). Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics*, 22(3):382–391.

Sloan, P., Kautz, J., and Snyder, J. (2002). Precomputed radiance transfer for real-time rendering in dynamic scenes. *ACM Transactions on Graphics*, 21(3):527–536.

Sloan, P., Liu, X., Shum, H., and Snyder, J. (2003b). Bi-scale radiance transfer. *ACM Transactions on Graphics*, 22(3):370–375.

Sloan, P., Luna, B., and Snyder, J. (2005). Local, deformable precomputed radiance transfer. *ACM Transactions on Graphics*, 24(3):1216–1224.

Sun, X., Zhou, K., Chen, Y., Lin, S., Shi, J., and Guo, B. (2007). Interactive relighting with dynamic BRDFs. *ACM Transaction on Graphics*, 26(3):27.

Tsai, Y.-T. and Shih, Z.-C. (2006). All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transaction on Graphics*, 25(3):967–976.

Wang, R., Tran, J., and Luebke, D. (2004). All-frequency relighting of non-diffuse objects using separable BRDF approximation. In *Proc. Eurographics Symposium on Rendering 2004*, pages 345–354.

Ward, G. J. (1992). Measuring and modeling anisotropic reflection. *ACM SIGGRAPH Compute Graphics*, 26(2):265–272.