Interactive Bi-scale Editing of Highly Glossy Materials

Kei Iwasaki∗
Wakayama University
Yoshinori Dobashi†
Hokkaido University
Tomoyuki Nishita‡
The University of Tokyo

Figure 1: Interactive (1.1-7.7 fps) bi-scale material design and real-time rendering (87-155 fps) under environment lighting. The small-scale geometries are shown in the insets of top left corner. (a) Rendering result using a small-scale diffuse BRDF. (b) Rendering result using a highly glossy BRDF. (c) Our method can handle spatially-varying bi-scale materials calculated from different small-scale geometries and BRDFs (upper left is used to render the SIGGRAPH logo). (d) is zoomed in image of (c). Two-hue appearance (green and yellow) of specular reflection can be rendered, which is difficult to render by using single-scale BRDFs.

Abstract

We present a new technique for bi-scale material editing using Spherical Gaussians (SGs). To represent large-scale appearances, an effective BRDF that is the average reflectance of small-scale details is used. The effective BRDF is calculated from the integral of the product of the Bidirectional Visible Normal Distribution (BVNDF) and BRDFs of small-scale geometry. Our method represents the BVNDF with a sum of SGs, which can be calculated on-the-fly, enabling interactive editing of small-scale geometry. By representing small-scale BRDFs with a sum of SGs, effective BRDFs can be calculated analytically by convolving the SGs for BVNDF and BRDF. We propose a new SG representation based on convolution of two SGs, which allows real-time rendering of effective BRDFs under all-frequency environment lighting and real-time editing of small-scale BRDFs. In contrast to the previous method, our method does not require extensive precomputation time and large volume of precomputed data per single BRDF, which makes it possible to implement our method on a GPU, resulting in real-time rendering.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture;

Keywords: material editing, reflectance filtering, normal distribution function, spherical Gaussian

1 Introduction

Interactive editing of visual appearances under real-world complex lighting is beneficial in design applications. Recent advances in computer graphics enable us to edit surface materials in real-time [Ben-Artzi et al. 2006; Sun et al. 2007; Cheslack-Postava et al. 2008; Wang et al. 2009]. These methods mainly focus on editing surface materials on a single scale. The visual appearance, however, depends on the scale of the geometry. For example, small-scale bumps on metal surfaces can be seen if the viewer is very close to the surface, while anisotropic reflection of light can be seen as the viewer zooms out from the surface.

In recent years, a physically-based editing method for visual appearance at the bi-scale level (large/small-scale) has been proposed [Wu et al. 2011]. Their method introduced a bidirectional visible normal distribution function (BVNDVF), which describes the normal distribution function of small-scale geometry taking into account shadowing and masking effects. The large-scale appearance is described by using the effective BRDF [Han et al. 2007; Wu et al. 2011], which is calculated by integrating the product of BVNDVF and the large-scale BRDFs. Their method calculates the effective BRDF by the product of two matrices that store a densely sampled small-scale BRDF and BVNDVF. To accelerate the computation time and to compress the matrices, a random projection method is employed [Vempala 2004]. Unfortunately, since their method is not a closed-form solution and hence based on numerical integration, it requires dense sampling of different directions for highly glossy BRDFs, resulting in the requirement for a huge volume of precomputed data and extensive precomputational time for BVNDVF and BRDFs.

To address these problems, we propose a closed-form solution to calculate effective BRDFs by using spherical Gaussians (SGs). The key insight of our method is to introduce a unified framework of SG representation for small-scale BRDFs, BVNDVF, and effective BRDFs. Our unified SG framework has many advantages in terms of achieving efficient rendering with effective BRDFs and editing of small-scale geometries and small-scale BRDFs. Our method rep-
resents both the BVNDF and the small-scale BRDFs using a linear combination of SGs, which makes it possible to integrate the product analytically and accurately. In addition, our SG representation for BVNDF and small-scale BRDFs enables us to render highly-glossy effective BRDFs. Although the product integral of BVNDF and BRDF can be calculated analytically by the convolutions of two SGs, which can be evaluated on-the-fly from SG representations of BVNDF and small-scale BRDFs. Our SG representation of effective BRDF can be easily incorporated into SG-based rendering for all-frequency environment lighting. Our GPU implementation demonstrates real-time rendering and interactive editing of small-scale geometries.

The contributions of our method are as follows.

- SG representation of BVNDF is introduced, which can be calculated on-the-fly, enabling interactive editing of small-scale geometries and real-time editing of highly glossy small-scale BRDFs.

- A simple SG form for the convolution of SGs is derived, making it easy for it to be incorporated into an SG-based rendering framework.

- The memory footprint of our method is small enough to comfortably fit in the GPU memory, which enables interactive editing of small-scale geometries and BRDFs, and consequently real-time rendering.

2 Previous Work

Filtering-based Method: To render surface details at different scales, normal map filtering methods have been proposed [Fournier 1992; Schilling 1997; Toksöy 2005; Han et al. 2007; Tan et al. 2008]. Toksöy et al. [2005] proposed a GPU-based mip-mapping method to filter normal maps. Han et al. [2007] proposed an accurate filtering method for normal maps by using spherical harmonics (SH) and von Mises-Fisher (vMF) distributions that are the same as the normalized SG. These methods, however, do not take into account shadowing and masking effects. Tan et al. [2008] proposed an efficient filtering method for resolution-dependent reflectance models. Although this method can render resolution-dependent reflectance models in real-time, it does not handle editing of small-scale geometry and BRDFs.

Geometry-based BRDF estimation Method: Westin et al. [1992] proposed a method to predict BRDFs from small-scale geometry. Heidrich et al. [2000] proposed a calculation method for large-scale BRDFs from height fields by using precomputed visibility, taking into account indirect illumination. Ashikhmin et al. [2000] presented a calculation method for BRDF from a 2D micro-facet orientation distribution. Wu et al. [2009] developed a characteristic point method that calculates the filtered reflectance of small-scale geometries taking into account shadowing and masking effects. These methods, however, require long precomputation times and therefore do not achieve interactive editing of small-scale geometries and BRDFs. Wu et al. [2011] extended their method and proposed the first physically-based bi-scale material editing method. This method [Wu et al. 2011] can calculate BRDFs in a few seconds by editing small-scale geometry and selecting precomputed BRDFs. However, it also requires a large volume of precomputed BRDF data, even if it is compressed, and cannot change BRDF parameters.

Material Editing Method under Environment Lighting: Benamzi et al. [2006; 2008] developed a BRDF editing method under complex lighting. Sun et al. [2007] proposed a BRDF editing method with dynamic viewpoints and lighting taking into account indirect illumination. Cheslack-Postava et al. [2008] presented a lighting and material design system using a non-linear cut approximation. Wang et al. [2009] proposed a real-time method for rendering dynamic, spatially-varying BRDFs under all-frequency environment lighting. Interactive editing methods for translucent materials [Xu et al. 2007; Wang et al. 2008] and hairs [Xu et al. 2011] also have been proposed. Although these methods can edit large-scale appearances under complex lighting efficiently, they do not handle editing of small-scale geometry and BRDFs.

3 Preliminaries

3.1 Effective BRDF

The appearance of objects on a large-scale can be described by using effective BRDFs $f(\omega_i, \omega_o)$ [Han et al. 2007; Wu et al. 2011]. The effective BRDF $f(\omega_i, \omega_o)$ is calculated by integrating the product of BVNDF $\gamma(\hat{n}, \omega_i, \omega_o)$ and BRDF $f(\hat{n}, \omega_i, \omega_o)$ in a small-scale as follows:

$$f(\omega_i, \omega_o) = \int_{S^2} f(\hat{n}, \omega_i, \omega_o) \gamma(\hat{n}, \omega_i, \omega_o) (\hat{n} \cdot \omega_i) (\hat{n} \cdot \omega_o) d\hat{n}$$

where $\omega_i$ is the incident direction, $\omega_o$ is the outgoing direction, $S^2$ is a set of directions on a unit sphere, $f$ is the BRDF on a small-scale, while $\gamma$ is the BVNDF for an infinitesimal area and is calculated as follows:

$$\gamma(\hat{n}, \omega_i, \omega_o) = \int_A V_s(x_s, \omega) V_s(x_s, \omega_o) \delta(n(x_s) - \hat{n}) dA(x_s),$$

where $V_s(x_s, \omega)$ is the visibility function, which returns a value of 0 if a ray from position $x_s$ in the $\omega$ direction intersects the small-scale geometry, and otherwise returns a value of 1. $\delta$ is a Dirac delta function, $\hat{n}$ is a parameter of the BVNDF $\gamma$, and we distinguish $\hat{n}$ from $n(x_s)$, which is a normal at $x_s$. $a_0(\omega_o) = \int_A V_s(x_s, \omega_o) (n(x_s) \cdot \omega_o) dA(x_s)$.

3.2 Spherical Gaussian

A spherical Gaussian (SG) is a type of spherical function and is represented by the following equation:

$$G(\omega, \xi, \eta, \mu) = \mu \exp(\eta(\omega - \xi - 1)),$$

where unit vector $\xi$ is the lobe axis, $\eta$ is the lobe sharpness, and $\mu$ is the lobe amplitude, which consists of RGB components. SG can represent all-frequency signals by adjusting the lobe sharpness (sharp signals can be represented by setting lobe sharpness $\eta$ large). The convolution of two SGs, $G(\omega, \xi_1, \eta_1, \mu_1)$ and $G(\omega, \xi_2, \eta_2, \mu_2)$, can be calculated analytically [Tsai and Shih 2006] as:

$$\int_{S^2} G(\omega, \xi_1, \eta_1, \mu_1) G(\omega, \xi_2, \eta_2, \mu_2) d\omega = 4\pi \mu_1 \mu_2 \sinh(\eta_1 \xi_1 + \eta_2 \xi_2) e^{\eta_1 + \eta_2} / (|\eta_1 \xi_1 + \eta_2 \xi_2|)^2$$

4 BVNDF Representation using SGs

To calculate effective BRDFs efficiently, our method defines $\bar{\gamma}(\hat{n}, \omega_i, \omega_o)$ as:

$$\bar{\gamma}(\hat{n}, \omega_i, \omega_o) = \frac{\gamma(\hat{n}, \omega_i, \omega_o) (\hat{n} \cdot \omega_i) (\hat{n} \cdot \omega_o)}{a_0(\omega_o)}.$$
Figure 2: SG representation of BVNDFs. (a) Our method first calculates NDF and estimates parameters of lobe axes and sharpness. Each colored arrow corresponds to each SG lobe representing NDF. (b) Shadowing and masking effects are considered by changing each lobe amplitude.

Our method represents $\bar{\gamma}$ with a linear combination of SGs $\sum_{j=1}^{J} G(\mathbf{n}, \xi_j, \eta_j, \mu_j)$, where $J$ is the number of SGs. However, fitting a linear combination of SGs to $\bar{\gamma}$ is computationally expensive, since fitting SGs usually requires a non-linear optimization problem. Moreover, since $\gamma$ is a 6D function (2D for each parameter $\hat{n}, \omega_s, \omega_a$), the number of fitting parameters is quite large (i.e. $3J$), and since three of the parameters ($\xi_j, \eta_j$, and $\mu_j$) are correlated, it is quite difficult to fit SGs to $\bar{\gamma}$ for each incident direction and outgoing direction interactively.

To address this, our method first fits the normal distribution function (NDF) $\Gamma(\mathbf{n})$ with a linear combination of SGs as follows:

$$\Gamma(\mathbf{n}) = \int_{A} \delta(\mathbf{n}(\mathbf{x}) - \mathbf{n})dA(\mathbf{x}) \approx \sum_{j=1}^{J} G(\mathbf{n}, \xi_j, \eta_j, \mu_j).$$

Fitting of NDF, $\Gamma(\mathbf{n})$, using SGs is performed by using the spherical Expectation Maximization (EM) algorithm proposed by Han et al. [2007]. As shown in Fig. 2, our method assumes that each lobe amplitude $\mu_j$ varies according to the incident direction $\omega_s$ and the outgoing direction $\omega_a$, while the lobe axis $\xi_j$ and the lobe sharpness $\eta_j$ do not change. Thus, $\bar{\gamma}(\mathbf{n}, \omega_s, \omega_a)$ is represented by:

$$\bar{\gamma}(\mathbf{n}, \omega_s, \omega_a) \approx \sum_{j=1}^{J} G(\mathbf{n}, \xi_j, \eta_j, \mu_j(\omega_s, \omega_a)).$$

Each amplitude $\mu_j(\omega_s, \omega_a)$ is calculated to minimize the squared difference $E_k$ for each discretized direction $(\omega_s^k, \omega_a^k)$:

$$E_k = \int_{S^2} \left[ \bar{\gamma}(\mathbf{n}, \omega_s^k, \omega_a^k) - \sum_{j=1}^{J} \mu_j(\omega_s^k, \omega_a^k)G(\mathbf{n}, \xi_j, \eta_j, 1) \right]^2 d\mathbf{n}.$$

Our method calculates the amplitude $\mu_j(\omega_s, \omega_a)$ to minimize the squared error $E_k$ in Eq. (8) between $\gamma$ and the SGs. This equates to finding a value for $\mu_j(\omega_s^k, \omega_a^k)$ that satisfies $\frac{\partial E_k}{\partial \mu_j(\omega_s^k, \omega_a^k)} = 0$, resulting in solving a linear equation $A \cdot \mathbf{b}_k = \mathbf{b}_k$, where $A$ is a symmetric matrix whose elements $a_{i,j} = \int_{S^2} G(\mathbf{n}, \xi_j, \eta_j, 1)G(\mathbf{n}, \xi_i, \eta_i, 1)d\mathbf{n}$, $\mathbf{b}_k$ and $\mathbf{b}_k$ are column vectors whose $j$-th elements are $\mu_j(\omega_s^k, \omega_a^k)$ and $\int_{S^2} \bar{\gamma}(\mathbf{n}, \omega_s^k, \omega_a^k)G(\mathbf{n}, \xi_j, \eta_j, 1)d\mathbf{n}$, respectively. In our experiments, we observe that $\mu_j(\omega_s, \omega_a)$ varies smoothly.

Figs. 3 and 4 show rendering results using our SG representation of BVNDFs for various small-scale geometries. A Blinn-Phong BRDF with an exponent parameter of 512 is used as the small-scale BRDF. Fig. 3 shows the specular components of large-scale appearance generated from the small-scale geometries shown in the bottom row. The highly specular reflection alters due to changes in the small-scale geometries.

Fig. 4 shows a comparison of different numbers of SGs to approximate $\bar{\gamma}$. Due to the low-frequency normal distributions of the height field in top left corner of Fig. 4(a), the specular reflections are dimmed and diffused compared to Fig. 3. As shown in Fig. 4, RMS errors between $\gamma$ and our SG representations are small for various small-scale geometries, and a small number of SGs (from 4 to 8) are sufficient to represent $\gamma$ accurately.

4.1 Effective BRDF Representation using SGs

The effective BRDF $\bar{f}(\omega_s, \omega_a)$ is calculated by using Eq. (1). To compute Eq. (1), the previous method [Wu et al. 2011] discretized many directions for the numerical integration, resulting in the requirement for a huge volume of precomputed data for $\gamma$ and BRDF. Although the previous method compresses the data by using the random projection method, it requires extensive precomputation time and the volume of the compressed data is still large for highly glossy BRDFs. Therefore, it is quite difficult to change the parameters of BRDFs arbitrarily (e.g. tuning of the exponent parameters of Blinn-Phong BRDFs).

Our method represents the small-scale BRDF $f$ as the sum of a diffuse term $k_d$ and a specular term $k_s f_s(\mathbf{n}, \omega_s, \omega_a)$ as $f(\mathbf{n}, \omega_s, \omega_a) = k_d + k_s f_s(\mathbf{n}, \omega_s, \omega_a)$, where $k_s$ is the coefficient of the specular term. The diffuse component $f_d(\omega_s, \omega_a)$ of the effective BRDF can be calculated analytically as follows:

$$f_d(\omega_s, \omega_a) = k_d \frac{\int_{S^2} G(\mathbf{n}, \xi_j, \eta_j, \mu_j(\omega_s, \omega_a))d\mathbf{n}}{2\pi k_d \sum_{j=1}^{J} \mu_j(\omega_s, \omega_a) \frac{1 - e^{-2\eta_j}}{\eta_j}}.$$

To compute Eq. (1) efficiently and accurately, our method represents the specular component $f_s(\mathbf{n}, \omega_s, \omega_a)$ using SGs with half vector $\mathbf{h} = \frac{\mathbf{n}}{|\mathbf{n}|}$, similar to the previous methods [Han et al. 2007; Wang et al. 2009]. The isotropic specular component $f_s$ can be approximated by using SG $G(\mathbf{n}, h, \eta, \mu)$. For example, the specular component of the Blinn-Phong BRDF $f_s(\mathbf{n}, \omega_s, \omega_a) = \frac{h^2}{1 - h^2} (\mathbf{n} \cdot \mathbf{h})^8$ can be approximated with the SG as $G(\mathbf{n}, h, n, n^2/2)$. Other isotropic BRDF models, such as the Cook-Torrance BRDF model and the Ward BRDF model, also can be represented by using SGs [Wang et al. 2009]. As shown in Eq. (4), the specular component $f_s(\omega_s, \omega_a)$ of the effective BRDF can be calculated an-
Although this method can achieve good results for large lobe sharpness. For small lobe sharpness that violates our assumption, the minimum lobe sharpness is over 40.

4.2 SG Form of SG Convolution

Real-time rendering with effective BRDFs under environment lighting is difficult due to the large number of directional lights. One solution to address this is to use importance sampling of the environment lighting, which approximates the environment lighting with a small number of directional lights as employed in the previous method [Wu et al. 2011]. Although this method can achieve real-time performance, importance sampling based only on lighting is problematic for highly-glossy BRDFs. Another solution is to approximate environment lighting with a linear combination of SGs, similar to the previous methods [Tsai and Shih 2006; Wang et al. 2009]. Unfortunately, the specular component $f_s$ in Eq. (10) cannot be directly incorporated into the SG-based lighting, since the convolution of two SGs is not closed in SG basis. Therefore, an SG representation of the convolution of two SGs is essential to achieve real-time rendering under environment lighting. In the following, we derive the SG form of the convolution of two SGs $G_1 = G(\omega, \xi_1, \eta_1, \omega)$ and $G_2 = G(\omega, \xi_2, \eta_2, \omega)$. The convolution of $G_1$ and $G_2$ is calculated as follows:

$$\int_{S^2} G_1 G_2 d\omega = \frac{4\pi \mu_1 \mu_2 \sinh(\|\eta_1 \xi_1 + \eta_2 \xi_2\|)}{\|\eta_1 \xi_1 + \eta_2 \xi_2\|} = \frac{2\pi \mu_1 \mu_2}{\|\xi\|}.$$ (11)

where $\zeta = \eta_1 \xi_1 + \eta_2 \xi_2$. Here, we assume that both lobe sharpness values $\eta_1$ and $\eta_2$ are not too small. In practice, this assumption is valid, since a large value is used for the lobe sharpness to represent the specular component $f_s(\hat{n}, \omega, \omega_0)$ (for example, the lobe sharpness $\eta$ is equal to the exponent $n$ for a Blinn-Phong BRDF). That is, the lobe sharpness $\eta$ for a Blinn-Phong BRDF is an exponent of 128). Then, we can assume that $e^{-\|\zeta\|-\eta_1-\eta_2} \approx 0$. Next, $\|\zeta\|$ can be represented as follows:

$$\|\zeta\| = \sqrt{\eta_1^2 \|\xi_1\|^2 + 2\eta_1 \eta_2 \xi_1 \cdot \xi_2 + \eta_2^2 \|\xi_2\|^2}$$

$$= (\eta_1 + \eta_2) \sqrt{1 + \frac{2\eta_1 \eta_2 (\xi_1 \cdot \xi_2 - 1)}{(\eta_1 + \eta_2)^2}}.$$ (12)

By using this representation, the numerator in Eq. (11) can be calculated as follows:

$$e^{\|\zeta\|-\eta_1-\eta_2} = e^{(\eta_1 + \eta_2) \sqrt{1 + \frac{2\eta_1 \eta_2 (\xi_1 \cdot \xi_2 - 1)}{(\eta_1 + \eta_2)^2}}}$$

$$\approx e^{(\eta_1 + \eta_2) \sqrt{1 + \frac{2\eta_1 \eta_2 (\xi_1 \cdot \xi_2 - 1)}{(\eta_1 + \eta_2)^2}}}$$

$$= e^{(\eta_1 + \eta_2) \sqrt{1 + \frac{2\eta_1 \eta_2 (\xi_1 \cdot \xi_2 - 1)}}},$$ (13)

where a linear approximation of the Taylor expansion $\sqrt{1 + x} \approx 1 + x/2$ is used. The denominator $|\zeta|$ can be approximated with $\eta_1 + \eta_2$, since $\frac{2\eta_1 \eta_2 (\xi_1 \cdot \xi_2 - 1)}{(\eta_1 + \eta_2)^2}$ can be considered as negligible. Finally, the convolution of two SGs can be represented by a single SG as:

$$\int_{S^2} G_1 G_2 d\omega \approx G(\xi_1, \xi_2, \frac{\eta_1 \eta_2}{\eta_1 + \eta_2}, \frac{2\pi \mu_1 \mu_2}{\eta_1 + \eta_2})$$ (14)

By using this representation, the specular component of the effective BRDF can be represented as follows:

$$f_s(\omega_1, \omega_0) \approx \sum_{j=1}^{J} G(\hat{n}, \xi_j, \frac{\eta_1 \eta_2}{\eta_1 + \eta_2}, \frac{2\pi \mu_1 \mu_2}{\eta_1 + \eta_2} \mu_j),$$ (15)

Discussion: Fig. 5 shows the plots of our SG form (Eq. (14)) compared to the analytical representation (Eq. (11)). As shown in Fig. 5, our SG form matches well with the ground truth for large lobe sharpness. For small lobe sharpness that violates our assumption, we analyzed the errors between the ground truth and our SG form by setting $\eta_1$ to 1 and changing $\eta_2$ from 1 to 10. The maximum errors decreases rapidly when $\eta_2$ is larger than 2. The maximum errors for $\eta_2 = 4, 6, 8$ are 0.070, 0.034, and 0.020, respectively. As described before, lobe sharpness $\eta_1$ and $\eta_2$ are both less than 10 are very rare. One of two SGs in the convolution represents the specular component of small-scale BRDFs, and BRDFs represented by SGs whose lobe sharpness is less than 10 are almost diffuse and therefore it would be better to be treated as diffuse component. Another SG in the convolution represents the NDF, and the NDF represented by SG with lobe sharpness 1 is almost random distribution over the unit sphere. Even for rough surfaces as shown in Fig. 4, the minimum lobe sharpness is over 40.

Recently, Jakob proposed a SG convolution closure approximation method [Jakob 2012]. This method calculates the SG convolution approximation iteratively by using the Newton’s method, while our method can calculate it directly. We analyzed the maximum errors of this method in four pairs of lobe sharpness in Fig. 5. The maximum errors are (a) $8.7 \times 10^{-3}$, (b) $1.2 \times 10^{-6}$, (c) $3.6 \times 10^{-8}$, and (d) $3.0 \times 10^{-6}$, respectively. That is, our SG form can obtain good results for large lobe sharpness.
5 Implementation Details

To achieve interactive editing of small-scale geometries, we implemented our method using CUDA. The calculation of BVNDF $\gamma(n, \omega_i, \omega_o)$ is similar to the previous method [Wu et al. 2011] except that they only use the GPU in the calculation of the visibility functions, while our method uses the GPU in the whole process. $\gamma(n, \omega_i, \omega_o)$ is sampled for each discretized direction $\omega_i^k$, $\omega_o^k$, and the discretized normal $n_i$ and $\omega_o$ are discretized on a hemisphere by using [Shirely and Chiu 1997] and the normal is discretized by using a cube-map.

First, our method randomly samples the small-scale geometry, which consists of triangular meshes, using a probability proportional to the area of each mesh. Our method calculates the positions $x_n$ and normals $n(x_n)$ of the sampled points and approximates the small-scale geometry as a set of planar facets whose centers are each sample point. Each facet is assigned to the same area $\Delta A$. The value of the visibility function $V_f(x_n, \omega_o^k)$ is calculated by using a traditional shadow-mapping method. Then, the value of BVNDF $\gamma(n, \omega_i^k, \omega_o^k)$ is calculated by discretizing Eq. (2) as $\gamma(n_i, \omega_i^k, \omega_o^k) = \frac{1}{N_f} \sum_{n=1}^{N_f} V_f(x_n, \omega_i^k) V_s(x_n, \omega_o^k) \delta(n(n_i) - n_i) \Delta A$, where $N_f$ is the number of facets, $\delta$ is a discretized delta function which returns a value of 1 if $n(x_n)$ is closest to the $i$-th discretized normal $n_i$ among all the discretized normal and which returns 0 otherwise. $\alpha_s(\omega_i^k)$ and $\Gamma(n_i)$ are also calculated in the same way. The summations in these calculations are performed efficiently by using a parallel reduction technique. Fitting of NDF $\Gamma(n)$ using SGs is also performed on the GPU by implementing the spherical EM algorithm [Han et al. 2007] using CUDA.

Rendering : Our rendering method is based on the previous method [Wang et al. 2009]. Our method focuses on static scenes under all-frequency environment lighting using the effective BRDF described in Eqs. (9) and (15), while our method can be applied to dynamic scenes by using the previous method [Iwasaki et al. 2012] and can also be applied to local light sources [Wang et al. 2009]. The outgoing radiance $L(x, \omega_o)$ at $x$ due to direct illumination is calculated by using the following equation $^2$.

$$L(x, \omega_o) = \int_{S^2} L(\omega_i) V(x, \omega_i) f_{ds}(\omega_i, \omega_o) d\omega_i,$$

where $L(\omega_i)$ is the distant lighting represented by environment maps, and $V(x, \omega_i)$ is the visibility function due to large-scale geometry. For simplicity, we omit $x$ in the following. As described before, our method represents the diffuse component $f_d$ and the specular component $f_s$ of the effective BRDFs separately. Therefore, our method calculates the diffuse component $L_d(\omega_o) = \int_{S^2} L(\omega_i) V(\omega_i) f_d(\omega_i, \omega_o) d\omega_i$, and the specular component $L_s(\omega_o) = \int_{S^2} L(\omega_i) V(\omega_i)f_s(\omega_i, \omega_o) d\omega_i$ of the outgoing radiance, and finally sums the two components.

To compute $L_d$, our method represents $L_d(\omega_i)$ with a linear combination of SGs as $L_d(\omega_o) \approx \sum_{k=1}^K G(\omega_i, \xi_k, \eta_k, \mu_k)$, where $K$ is the number of SGs and is set to 9, as in [Wang et al. 2009]. As described previously, since $f_d(\omega_i, \omega_o)$ is a smooth function, $f_d(\omega_i, \omega_o)$ also varies smoothly. Therefore, $L_d$ can be calculated by using the following equation.

$$L_d(\omega_o) \approx \sum_{k=1}^K f_d(\xi_k, \eta_k, \mu_k) \int_{S^2} G(\omega_i, \xi_k, \eta_k, \mu_k) V(\omega_i) d\omega_i.$$

The integral of the product of SG and the visibility function can be calculated by using spherical signed distance functions (SSDFs) [Wang et al. 2009].

$^2$the cosine term is included in the effective BRDF as Eq. (1).

To calculate the specular component $L_s$, our method first converts the parameter $h$ of SG into the integration variable $\omega_i$ using the spherical warp $\Psi$ [Wang et al. 2009], where $\omega_i = \Psi(h) = 2(\omega_o \cdot \hat{h})\hat{h} - \omega_o$. By using the spherical warp, $L_s$ can be calculated as follows:

$$L_s(\omega_o) = \sum_{i=1}^j \int_{S^2} L(\omega_i) V(\omega_i) G(\omega_i, \xi_j, \eta_j, \mu_j(\omega_i, \omega_o)) d\omega_i,$$

where $\xi_j = 2(\omega_o \cdot \hat{h})\hat{h} - \omega_o$, $\omega_o = \frac{\eta_j}{\eta_j + \eta_o}$, and $\mu_j(\omega_i, \omega_o) = \frac{\eta_j + \eta_o}{\eta_j - \eta_o} \mu_j(\omega_i, \omega_o)$. Since $\mu_j(\omega_i, \omega_o)$ varies smoothly, our method approximates that $\mu_j(\omega_i, \omega_o)$ is constant across the support of each SG. By utilizing this approximation, $L_s(\omega_o)$ can be calculated by using the following equation:

$$L_s(\omega_o) \approx \sum_{i=1}^j \mu_j(\xi_j, \eta_j, 1) \int_{S^2} L(\omega_i) V(\omega_i) G(\omega_i, \xi_j, 1) d\omega_i.$$

The integral of the two product of the lighting, the visibility function and the SG can be calculated efficiently by using pre-filtered environment maps [Green et al. 2006; Green et al. 2007] and SSDF [Wang et al. 2009].

Our method calculates $L_d$ and $L_s$ per pixel by using a GLSL fragment program in a single pass. To achieve real-time rendering, $\mu_j(\omega_i, \omega_o)$ as calculated by using CUDA is transferred to OpenGL and stored as a 2D texture array $f_s(\omega_i, \omega_o)$ is calculated by using Eq. (9) and is stored as a 2D texture. The data required to compute SSDF is stored as a texture buffer object and is fetched in the fragment program.

6 Results

We have implemented our algorithm on a standard PC with an Intel Core i7-2700K CPU and a GeForce GTX 580 GPU. The image resolutions are 640x480. The rendering frame rates for all examples are over 150 fps (frames per second) except for Figs. 1(c)(d) and Fig. 6 (87 fps). The computational time to edit the small-scale geometry ranges from 1.1 to 7.7 fps. The bidirectional domain $\omega^d$ is discretized in 144 x 144 and the normal domain is discretized in 6 x 128 x 128 to compute BVNDF $\gamma$. Our method can change dynamic viewpoints, lighting, and small-scale BRDFs in real-time, and can edit small-scale geometries interactively.

Fig. 6 shows a cloth model whose small-scale geometries are weave fabric. The insets of the top left corner show the small-scale weave fabric that consists of red warp threads and blue weft threads. From Figs. 6(a) to (c), the appearances of the cloth alter due to changes in the width of red warp threads. The average computational time to edit small-scale geometries is about 1.28 fps. The breakdown of the computational time is 651 msec for calculating the BVNDF $\gamma$, 41 msec for computing visibility, 28 msec for estimating NDF, 15 msec for calculating amplitudes $\mu_j(\omega_i^d, \omega_o^d)$ and rest for other
computations. The computational time to edit the small-scale geometry depends on its complexity, mainly depends on the number of discretized normals whose $\gamma(\hat{n})$ are non-zero, since our method calculates the BVNDF $\gamma$ for the non-zero normals. In Fig. 6, the number of non-zero discretized normals ranges from 105 to 226.

Fig. 7 shows the rendering results of chess scene with different small-scale BRDFs. The Cook-Torrance BRDF models are used as small-scale BRDFs. By editing small-scale geometries, the appearances of chess pieces alter from metallic-like surfaces to plastic-like surfaces, while small-scale BRDFs are unchanged. The computational time to edit small-scale geometries is about 5.5 fps. The breakdown of the computational time is 67 msecs for calculating the BVNDF $\gamma$, 42 msecs for estimating NDF, 41 msecs for computing visibility, and rest for other computations. The number of non-zero normals ranges from 5 to 9.

Fig. 8 shows the rendering results with measured BRDFs [Matusik et al. 2003]. Our method represents the measured BRDFs using SGs similar to the previous method [Wang et al. 2009]. The small-scale BRDFs used to render the pieces of pawn, knight, bishop, queen, and king are metallic-silver, metallic-blue, metallic-gold, ch-ball-blue-metallic, and alum-bronze, respectively.

Fig. 9 shows comparisons between our rendering results, the previous method [Wu et al. 2011] and the reference solution. Figs. 9(a), (b), (e), and (f) are rendered by using effective BRDFs calculated from the previous method [Wu et al. 2011]. Figs. 9(a-c) and (e-g) are rendered by summing up all the contributions of $6 \times 32 \times 32$ directional lights of the environment map, and the computational time is 72 sec. In Fig. 9, a Blinn-Phong BRDF with an exponent of 512 and measured alum-bronze BRDF are used. The size of the precomputed BRDF matrix is $1.7M \times 6.1K$, and the volume of the compressed data by using the random projection method [Vempala 2004] is 460MB. The number of singular values is set to 70. Figs. 9(b) and (f) show the rendering result by increasing the number of singular values to 120 when the volume of precomputed data increases to 782MB. We have tried to increase the number of singular values even further, but failed due to the huge precomputed data volume that is then required. Figs. 9(c) and (g) are rendered by using the effective BRDF directly evaluated using Eq. (1). $\gamma(h, \omega_1, \omega_2)$ and $f(h, \omega_1, \omega_2)$ are densely sampled for $3.4M$ directions for the bidirectional domain $(\omega_1, \omega_2)$ and $6.1K$ for $h$.

As shown in Fig 9, it is quite difficult to reproduce highly specular reflections by using the previous method, while our method can render plausible results that match to the reference well. The computational times to achieve effective BRDFs for Figs. 9(a) and (b) are 17 sec and 25 sec, respectively with our non-optimized code. The precomputational time for the BRDF matrix is about 40min, which can be accelerated by using our GPU implementation while that listed in [Wu et al. 2011] is more than 24 hours using a CPU. In contrast to the previous method, our method does not require huge volumes of precomputed data nor extended precomputational time for small-scale BRDFs. Our method does not require precomp-

7 Conclusions and Future Work

We have proposed an efficient calculation method for effective BRDFs calculated from small-scale geometries and small-scale BRDFs. We have proposed a unified framework of SG representations for BVNDF, small-scale BRDF, and effective BRDFs. Our method can calculate the SG representation of BVNDF from small-scale geometries in parallel, enabling interactive editing of small-scale geometries and the calculation of effective BRDFs. We have derived a novel SG representation of the convolution of two SGs. By using this method, an effective BRDF that is calculated from the convolution of SGs corresponding to a BVNDF and small-scale BRDFs, can be represented by using SGs and directly combined with SG-based rendering. This results in real-time rendering using effective BRDFs under all-frequency environment lighting.

Although our method can render convincing results of highly glossy bi-scale materials, it also has several limitations. Our method handles only direct illumination and isotropic BRDFs as small-scale materials, while these limitations are the same as the previous method [Wu et al. 2011]. In future work, we would like to take into account indirect illumination in small-scale geometries to calculate effective BRDFs using the previous method [Laurijssen et al. 2010]. Moreover, we would like to extend our method to multi-scale BRDF editing.

Acknowledgements

We would like to thank the reviewers for their valuable comments. This research was partially supported by JSPS KAKENHI Grant Number 24700093.

References


Figure 9: Comparisons between previous method using SVD (a,b,c,f), reference image (c,g), and our method (d,h). The Blinn-Phong BRDF with an exponent 512 (top) and the measured alum-bronze BRDF (bottom) are used. The numbers of truncating singular values are 70 (a,e) and 120 (b,f), respectively.


TSAI, Y.-T., AND SIHH, Z.-C. 2006. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. ACM Trans. Graph. 25, 3, 967–976.


