Locally Adaptive Products for Genuine Spherical Harmonic Lighting

Joo Ho Lee* Min H. Kim†

KAIST, Korea

ABSTRACT

Precomputed radiance transfer techniques have been broadly used for supporting complex illumination effects on diffuse and glossy objects. Although working with the wavelet domain is efficient in handling all-frequency illumination, the spherical harmonics domain is more convenient for interactively changing lights and views on the fly due to the rotational invariant nature of the spherical harmonic domain. For interactive lighting, however, the number of coefficients must be limited and the high orders of coefficients have to be eliminated. Therefore spherical harmonic lighting has been preferred and practiced only for interactive soft-diffuse lighting. In this paper, we propose a simple but practical filtering solution using locally adaptive products of high-order harmonic coefficients within the genuine spherical harmonic lighting framework. Our approach works out on the fly in two folds. We first conduct multi-level filtering on vertices in order to determine regions of interests, where the high orders of harmonics are necessary for high frequency lighting. The initially determined regions of interests are then refined through filling in the incomplete regions by traveling the neighboring vertices. Even not relying on graphics hardware, the proposed method allows to compute high order products of spherical harmonic lighting for both diffuse and specular lighting.

Keywords

global illumination, spherical harmonic lighting

1 INTRODUCTION

Precomputed radiance transfer (PRT) techniques have been widely used to render global illumination effects in interactive graphics applications such as video games. General PRT techniques precompute and approximate the visibility and the cosine term between a point (vertex or pixel), so-called *ambient occlusion*, and incoming hemispherical illumination, so that we can compute such expensive shading on the fly. This precomputed ambient occlusion is often modulated together with other illumination properties such as an environment map and bidirectional reflectance functions (BRDF). Monte Carlo sampling on the hemispherical illumination has been practiced commonly to compute ambient occlusion.

The PRT techniques are based on a frequency vector space of lighting, which uses spherical harmonics (SH) [1, 2], Haar wavelets [3, 4] or discrete cosine transform (DCT) [5]. For instance, the SH-based techniques project an environment light map to a set of SH basis over the sphere, which consists of harmonic basis functions and the lighting coefficients [1]. The main benefit of the SH-based approaches is that we can decompose the light transport to the different frequency bands so that we can omit exclusively high-frequency illumination in the hierarchical frequency layers for interactively lighting applications. Only several layers of harmonic coefficients can render diffuse surface plausibly [6].

However, if we want to render a higher-frequency

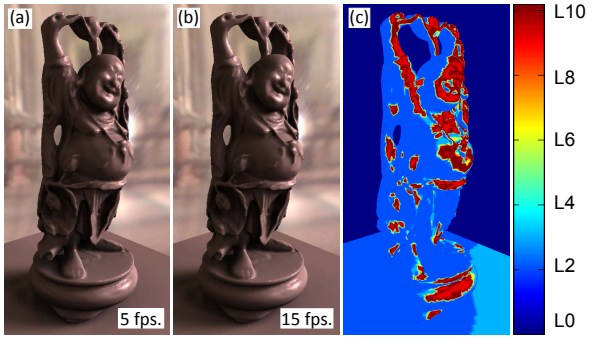

Figure 1: *Comparison of rendering between full SH lighting (a) and our locally adaptive approach (b). (c) shows the adaptive local filtering. Our approach achieves real-time performance (15 fps.) with genuine SH lighting with level 10 (121 coefficients). The rendered image (b) is plausibly comparable to the calculation of full-order coefficients (a). The peak signal-to-noise ratio between the two images is 41.73.*

* Author e-mail: jhlee@vclab.kaist.ac.kr

† Corresponding author e-mail: minhkim@vclab.kaist.ac.kr

shadow within the SH framework, we need a higher order of harmonics basis despite of sacrificing the computational costs. For this reason, all frequency rendering in interactive applications with a small number of SH coefficients has been thought to be impossible. Therefore, the wavelet domain has been preferred for all-frequency lighting.

Wavelet decomposition includes the mean and local deviation basis functions of each frequency layer. Analogous to an image gradient including spatially varying local frequency information, these local deviation bases operate as wavelet coefficients. Hence, the wavelet-based techniques are capable of rendering all-frequency illumination effects with sharper shadow and BRDFs with fidelity using a relatively small number of coefficients [3, 7, 8, 4]. However, owing to the mathematical nature of the wavelet domain, the wavelet coefficients cannot be rotated directly. Implementing the rotation of wavelet coefficients for real time is cumbersome, requiring more efforts on programming with graphics hardware. Recently, hybrid approaches such as [9] have been proposed where the low-order spherical harmonics is used for indirect illumination and the wavelets are used for direct lighting.

In this paper, we propose a simple but practical lighting solution using locally adaptive products of SH coefficients only within the SH lighting framework. Our fundamental insight is based on the observation that the lighting changes along the level of harmonics. When we increase the number coefficients of harmonics, the change of light intensity forges near the edges of shadow, i.e., the number of vertices, which require the high frequency lighting, becomes smaller as the harmonics level increases. We take the insight from this observation and propose a novel multi-level filtering approach for determining the regions of interest (ROIs) for high frequency lighting against the regions, where the high order of harmonics makes no changes in lighting. This allows us to compute the high-order products of diffuse and specular lighting on given limited computational resources, rendering higher frequency lighting exclusively. Refer to Figure 1 for the comparison of our rendering and classical spherical harmonic rendering. Our contributions are:

- A novel multi-level filtering approach for locally adaptive products and
- A refining method of the vertex ROIs using the breadth-first search method.

2 RELATED WORK

Precomputed radiance techniques have been developed intensively for more than a decade since Sloan et al. [1]. This section overviews the related work of the PRT techniques very briefly. Refer to Ramamoorthi [10] for more detailed review of the latest PRT technologies.

Low-Frequency PRT. Ramamoorthi and Hanrahan [6] proposed that the 25 coefficients (zero to four levels of harmonic bases) could be enough to render diffuse reflectance with general environmental maps. However, high-frequency shadowing and specular representation require a lager set of basis functions in these traditional SH lighting techniques. Sloan et al. [1] then introduced a seminal precomputation framework to render the diffuse and glossy materials with environmental maps decomposed in spherical harmonics. Their method can support various illumination effects such as interreflections and caustics, assuming that the scene is static and that the environmental maps are in low frequency. The frame rate of the proposed method was less than four frames per second with 50K vertices using five harmonics layers (in 2002). Kautz et al. [11] enabled to render arbitrary BRDFs and Sloan et al. [12] applied the principal component analysis (PCA) clustering method to extend their previous work [1] for interactive applications.

All-Frequency PRT. Traditional SH lighting techniques employ a limited number of coefficients to provide a real-time response rate. High-frequency lighting such as sharp shadows cannot be rendered with the number of coefficients in SH lighting. Sloan et al. [13] proposed a radiance transfer technique that achieves higher-frequency lighting by combining global illumination effects at two different scales: a coarsely sampled macro-scale through the PRT and a finely sampled meso-scale on bidirectional texture function (BTF) on graphics hardware. Ng et al. [3] introduced a seminal wavelet-based representation for all-frequency lighting using a nonlinear approximation. This method can render non-diffuse scenes at a fixed viewing and scenes at arbitrary viewing directions. Ng et al. [4] extended their work [3] to triple products of wavelet integrals for rendering all-frequency direct illumination, allowing to change the viewpoint and the light direction. However, the rendering took three to five seconds per frame [4]. Liu et al. [7] and Wang et al. [8] proposed a precomputed radiance transfer method dedicated for rendering glossy objects. These approaches separate 4D BRDFs into the viewing (2D) and lighting (2D) direction, respectively. These functions are factored in a form of Haar wavelets and clustered to principals components [7]. Kautz et al. [14] proposed a hemispherical rasterization method that recomputes visibility for self-shadowing of dynamic objects. Inger et al. [5] recently proposed a locally adaptive product approach using discrete cosine transformation to achieve a real-time frame rate with high vertex counts.

Other Basis PRT. Sloan et al. [15] decompose the light transfer functions as the integral of zonal harmonic (ZH) basis functions, rotated about different directions. This zonal harmonics approximation is so efficient that it enables fast local transformation of lighting on the fly.

Tsai and Shih [16] decompose light transfer functions into radial basis functions, which are undemanding to rotate and handle high frequency efficiently. To achieve a real-time performance, they compress the light transfer matrix by applying a tensor approximation with clusterization, where only direct illumination is handled. Nowrouzezahrai [2] recently introduced a sparse ZH factorization method for rotating harmonic coefficients efficiently.

In this paper, we explore a simple but powerful modification of the traditional SH lighting techniques. Our objective is to render high-frequency lighting within the genuine spherical harmonics framework. Our modification renders static diffuse and specular materials, allowing the dynamic changes of the lighting and the view.

3 PRELIMINARIES

3.1 Foundations of Spherical Harmonics

This section briefly overviews the foundations of SH lighting on which our system is based [1].

Spherical Harmonics. Spherical harmonics are defined over a spherical space S, where a point on the space (x,y,z) is parameterized as $(sin\theta cos\phi, sin\theta sin\phi, cos\theta)$. The angular portion of this system is called the spherical harmonics $Y_l^m(\theta,\phi)$ as a Laplace's equation, where $l \in \mathbf{N}$ (the total number of layers) is the coefficient layer index, and m is an integer index between $-l$ and l. The complex basis is transformed and defined to the real basis $y_l^m(\theta,\phi)$ in practice using associated Legendre polynomials P_l^m as follows:

$$y_l^m = \begin{cases} \sqrt{2}K_l^m cos(m\phi)P_l^m(cos\theta), & m>0 \\ \sqrt{2}K_l^m sin(-m\phi)P_l^{-m}(cos\theta), & m<0 \\ K_l^0 P_l^0(cos\theta), & m=0, \end{cases}$$

where K_l^m is the normalized coefficients.

A spherical function f of a band over the sphere can be defined as a coefficient f_l^m of the SH band, i.e., f_l^m is the integral of the SH vectors of the band $\int_S f(s)y_l^m(s)ds$. The discrete sum of the n-th order bands can approximate the original function $f(s)$ as below:

$$\tilde{f}(s) = \sum_{l=0}^{n-1}\sum_{m=-l}^{l} f_l^m y_l^m(s), \tag{1}$$

where the n-th order in the traditional SH basis includes n^2 coefficients.

Integration in SH. We compute diffuse lighting as the products of ambient occlusion and a light map. In spherical harmonics, the integration I of the products of functions $\tilde{f}$ and $\tilde{g}$ can be projected as a dot product as follows:

$$\begin{aligned} I &= \int_S \tilde{f}(s)\tilde{g}(s)ds = \int_S \sum_i F_i y_i(s) \sum_j G_j y_j(s) ds \\ &= \sum_i \sum_j F_i G_j \int_S y_i(s)y_j(s)ds = \sum_i F_i G_i = F \cdot G, \end{aligned}$$

where F is the SH vector of f; G is the SH vector of g; $\int_S y_i(s)y_j(s)ds$ is the integral of the orthonormal products, which turns out to be one only when i and j are equal. Otherwise the integral results in zero due to the nature of orthonormality.

Production Projection in SH. Our method computes incoming radiance as the product of ambient occlusion and a light map. The product f of two spherical functions g and h can be projected in spherical harmonics:

$$\begin{aligned} F_i &= \int_S (\sum_j G_j y_j(s) \sum_k H_k y_k(s)) y_i(s) ds \\ &= \sum_j \sum_k G_j H_k \int_S y_i(s)y_j(s)y_k(s)ds \\ &= \sum_j \sum_k G_j H_k \hat{Y}_{ijk} = \sum_j G_j \hat{H}_{ij}, \end{aligned} \tag{2}$$

where $\hat{Y}$ is a three-dimensional product-projection matrix that can be defined as $\hat{Y}_{ijk} = \int_S y_i(s)y_j(s)y_k(s)ds$; G is the SH vector of g; H is the SH vector of h; F is the SH vector of the product; $\hat{H}$ is the SH product-projection matrix of H.

Convolution in SH. To evaluate specular reflection, we convolve the incoming radiance function with the z-aligned kernel [1]. The convolution of the circular symmetric function g, which can be decomposed into zonal harmonic coefficients, with a spherical function f is an element-wise multiplication in the SH domain:

$$(g * f)_l^m = \alpha_l g_l^0 f_l^m, \tag{3}$$

where α_l is $\sqrt{4\pi/(2l+1)}$.

3.2 Diffuse SH Lighting

In our system, we approximate the diffuse SH reflection lighting L_d at each vertex as follow:

$$L_d(x,\vec{w}_o) = \int_\Omega L(w_i)O(w_i)\rho_d dw_i = \rho_d L \cdot O, \tag{4}$$

where L is a rotated environment light map; O is an ambient occlusion which is the product of visibility and the cosine term; ρ_d is a diffuse coefficient. We divide this reflection equation into three factors: the environment light map, the ambient occlusion and the reflectance. These are decomposed to spherical harmonic coefficients via Monte Carlo integration at the precomputation stage, where we prebuilt L_i and O as SH coefficient matrices. Diffuse reflection with shadows can be computed on the fly using the dot products of the SH vectors of the ambient occlusion, lighting and diffuse reflectance.

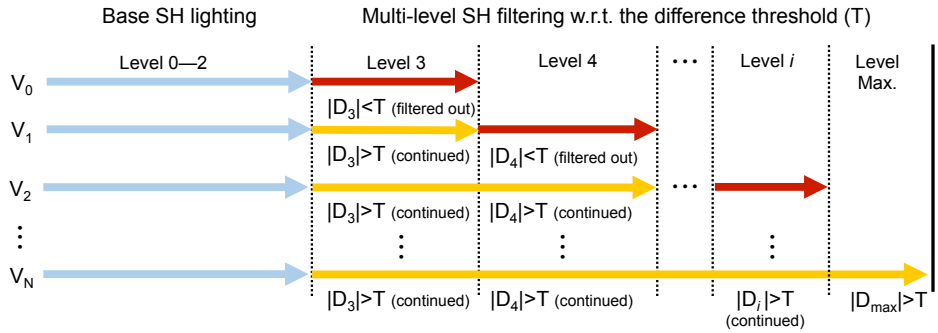

Figure 2: *Multi-level SH filtering. We first compute the base SH lighting for all vertices. Then at each level, we compute the difference D of lighting at each vertex. If the difference of the vertex is lower than the threshold T, we do not calculate the SH light of the vertex in the higher levels. Otherwise, we repeat the computation in the next levels.*

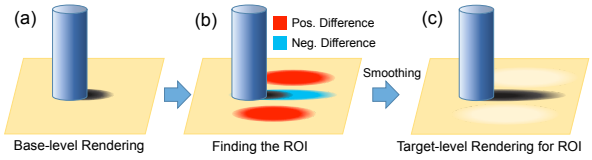

Figure 3: *Schematic overview of our multi-level SH lighting framework. (a) We compute the reference SH lighting initially and (b) compute the difference between the current and the reference level in SH lighting. We define the narrow ROIs for the next level computation and (c) compute the next level.*

3.3 Specular SH Lighting

For specular lighting, we separate reflectance into diffuse (Eq. (4)) and specular reflection terms (Eq. (5)) [17]. We then compute isotropic specular reflection L_s by convolving the specular reflectance function along the incoming radiance [1]:

$$L_s(\vec{N},\vec{w}_o) = \int_\Omega L_i(\vec{w}_i) O(\vec{w}_i) \rho_s(\vec{w}_i,\vec{w}_o) dw_i\,, \quad (5)$$

where we employ the isotropic Phong model, assuming the specular lobe ρ_s as a single symmetric lobe with a specular parameter k: $\rho_s(w_i,w_o) = (B(\vec{w}_i)\cdot\vec{w}_o)^k$ [18], where $B()$ is a bounced light.

In order to implement interactive lighting of specular reflection, we convert the expensive convolutions into element-wise multiplications using the SH basis. We simplify Eq. (5) as below [1]:

$$\begin{aligned} L_s(x,\vec{w}_o) &= \int_\Omega R(B(\vec{w}_o)-\vec{w}_i)\rho_z(\vec{w}_i)dw_i \\ &= R * \rho_z(B(\vec{w}_o))\,, \end{aligned} \quad (6)$$

where R is an incoming radiance on ρ_z; $B(\vec{w}_o)$ is a reflection of $\vec{w}_o$; ρ_z is a z-aligned specular lobe.

Our precomputation therefore includes the SH decomposition of the Phong reflectance with predetermined parameters of shininess. To implement specular lighting, we first compute the reflected ray $B(\vec{w}_o)$ and the SH basis value $y_l^m(B(\vec{w}_o))$. We also need to compute the incoming radiance R in the SH domain as the product of the ambient occlusion O and the light L in the intensity domain. We then conduct element-wise multiplications of the z-aligned specular lobe ρ_z and the light R in the SH domain, equivalent to the convolutions of the specular lobe in the intensity domain.

Specular lighting is the most expensive computation within the SH lighting framework. Compared to other steps, which are computed in linear time, the product projection requires N^2 matrix multiplications, where N is the number of SH coefficients. Note that in this paper we focus on reducing the computational costs in specular lighting within the genuine spherical harmonic framework.

4 MULTI-LEVEL SH LIGHTING

We specify our scope to optimizing the genuine SH lighting framework without relying on any other domain such as wavelet or zonal harmonics. In particular, our objective is to enhance the frame rate of SH lighting with specularity and high-frequency shadows.

4.1 Multi-Level Filtering

We desire to decrease the computational costs in SH rendering by reducing the number of the SH levels adaptively for each vertex. Our method first computes the reference base harmonics of lighting for each vertex. To define the ROIs of vertices for the higher levels in harmonics, we compute the SH lighting at the current level and filter out the vertices in the next level by comparing the lighting differences of the current against the reference (under a certain threshold T). However, the selected ROIs of the vertices could be discontinued arbitrarily due to the gradient variation of the harmonic levels. We therefore fill in the holes of the surrounding ROIs after the initial selection of vertices. We then

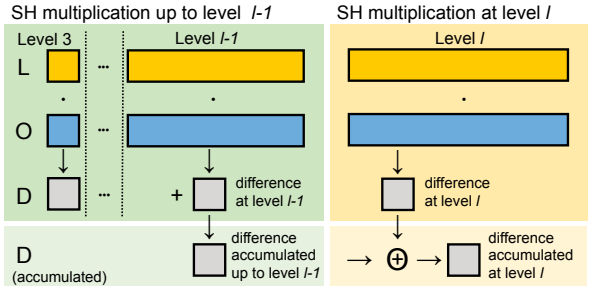

Figure 4: *Computation of SH diffuse lighting. We determine the diffuse ROI of the vertices at level l.*

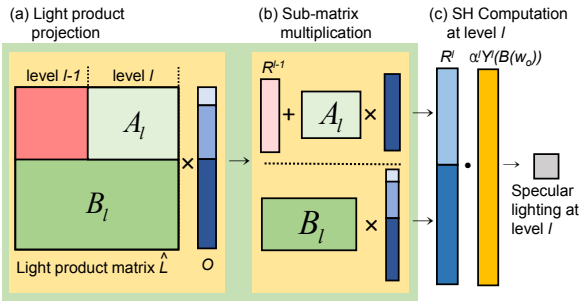

Figure 5: *Computation of SH specular lighting. First, we compute the incoming radiance R at level l and compute two sub-matrix multiplications and a vector addition for R_l, avoiding the multiply duplication of light product projection at lower levels. Finally, we compute specular lighting S at level l via the dot product of R_l and $\alpha Y(B(w_o))$.*

compute the next level SH lighting exclusively for the vertices within the ROIs, followed by smoothing of the edges around the ROIs. Figure 3 shows the schematic overview of our method. Figure 2 shows the workflow to determine the ROI of vertices at each level according to the level-by-level difference of lighting. Algorithm 1 describes our multi-level lighting method in the pseudo codes.

4.2 Diffuse ROI

Our method calculates the specular and the diffuse lighting separately adopting the Phong reflectance model. Our method first determines the reference base-level diffuse lighting for every vertex. The reference base level for diffuse reflection is limited up to the second level (the first top three layers) as the three SH layers are fundamentally important to render diffuse reflection [10]. Then, we find a region of interest, which includes the significant lighting changes between the base and the target level.

In order to reckon the level-by-level changes of SH diffuse lighting per vertex, we multiply the ambient occlusion and a light to compute diffuse lighting at level l per vertex x. We then calculate the difference D of SH lightings between the current and the base levels and accumulate it as the lighting difference D (see Fig. 4):

$$L^l_{diffuse} = \sum_{i=l^2}^{(l+1)^2-1} L_i O_i \,,$$

where $L^l_{diffuse}$ is a diffuse reflection at a level l for a vertex x; L_i is an i-th SH coefficient for a environment light map; O_i is an i-th SH coefficient for the ambient occlusion at a vertex x.

For instance, the inside of shadows is the result of negative lighting difference while the outside of shadows is the result of positive light difference, see Fig. 8. Thus, to capture the change of shadows, we check whether the absolute value of the diffuse lighting difference $|D|$ is larger than the threshold T as the level grows. Algorithm 2 presents the pseudo code of the computation of diffuse lighting at each level.

4.3 Specular ROI filtering

The inspiration on our specular calculation is based on our observation that the specular region shrinks as the SH order of lighting increases. Based on this observation, we obtain the specular ROIs of vertices. We first compute the reference base level of specular SH lighting same as the original diffuse lighting (up to the second level). Then we compute specular lighting at the current level with Eq. (6) and check whether the vertices should be included in the current ROI specular. We repeat this vertex filtering level by level, yielding smaller ROIs with higher orders. To this end, we compute the product R of the ambient occlusion at a level l (see Fig. 5(b)):

$$R^l_i = \begin{cases} \sum_{j=0}^{(l+1)^2-1} \widehat{L}_{ij} O_j \,, & i \geq l^2 \\ R^{l-1}_i + \sum_{j=0}^{l^2-1} \widehat{L}_{ij} O_j \,, & i < l^2 \end{cases},$$

where R^l_i is the i-th SH coefficient for the product of the ambient occlusion O and the light L at the level l; where $\widehat{L}_{ij}$ is the product-projection matrix of the light L defined as the product of the light L and the product projection matrix $\hat{Y}$ (see Eq. (2)). Note that the l^2-th element is the first element at a level l while the $((l+1)^2-1)$-th element is the last element at the level l.

We then convolve the incoming radiance with the z-aligned specular lobe and compute the specular lighting S at the level l as the value of the convolution at a direction w_o (see Fig. 5(c)):

$$S_l = \sum_{i=0}^{(l+1)^2-1} R_i \cdot \alpha_i y_i(B(w_o)) \,,$$

where S_l is the specular lighting at the level l; α_i is the i-th convolution coefficient (Eq. (3)); $y_i(B(w_o))$ is the value of i-th SH basis function at a direction $B(w_o)$. Similar to the diffuse filtering, we stop further computation if specular reflection S is lower than the threshold T. Algorithm 3 presents the pseudo code of the computation of specular lighting at each level.

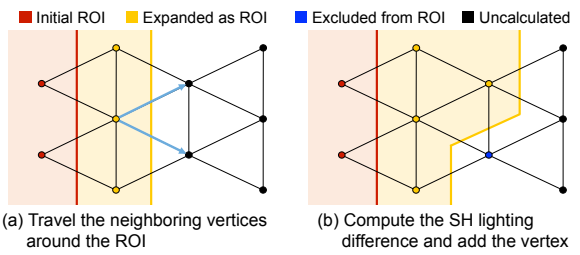

Figure 6: *Filling in the incomplete ROIs. (a) Traveling the neighboring vertices around the ROI. (b) Computing the SH lighting difference of the neighbors exclusively and adaptively modifying (expanding) the ROI according to the difference.*

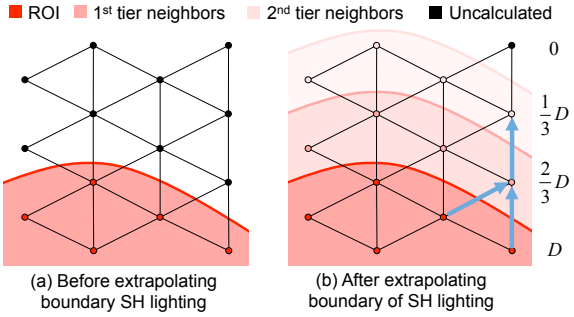

Figure 7: *Smoothing the boundaries. (a) We extrapolate the calculated SH lighting to neighboring vertices in the breadth-first order. (b) For the neighbors, we average the SH lighting and extrapolate it up to the second tier neighbors. D is an SH lighting on the boundary of the ROI.*

4.4 Refinements

Refining the ROIs. Some vertices could be filtered out from the ROI at an early level although they might have significant lighting changes in higher orders. We therefore refine the ROIs of diffuse and specular lighting by traveling the spatial coherence around the neighboring ROIs in the breadth-first order. We first compute the difference of SH lighting exclusively among neighboring vertices around the initial ROI (up to the maximum order of harmonics). If the lighting on the vertex varies significantly, the vertex is added to the initial ROI. We repeat this process until no more significant lighting changes are detected among the neighboring vertices (see Fig. 6).

Smoothing the Boundaries. We render high-order SH lighting exclusively within the ROIs, yielding sharp-edge artifacts often around the ROIs. This edge artifacts occur as the computational trade-off between the speed and the accuracy while comparing the SH lighting differences. For instance, when we use a very low level of threshold T, most of the vertices are classified to the ROI. There is no improvement in computational efficiency. When we use a high level of T, our naive calculation might suffer from sharp edges around the boundary of the ROIs. Hence, we empirically chose a level of threshold (we use $T = 0.02$ for specular reflection and $T = 0.01$ for diffuse reflection) first.

We travel the neighboring vertices from the boundary of the ROIs using the breadth-first search (BFS) method, and linearly extrapolate the computed SH lighting of the ROIs toward the second level of the outer boundary of the ROIs in the breadth-first order (see Fig. 7). Note that we take all the neighboring SH lightings of the connected vertices into account by averaging them.

5 RESULTS

This section describes the implementation detail of our method and demonstrate results.

Implementation. We implemented our locally adaptive SH lighting method in C++ on a machine with an Intel i7 3770 CPU with 3.4 GHz (4 cores; L1/L2/L3 cache: 4×32KB, 4×256KB and 8MB), 16GB DRAM, and an Nvidia GeForce GTX 670 (2GB RAM). Our implementation is genuinely CPU-based through multi-threading using OpenMP. Note that the GPU is only utilized to compute triangle interpolation.

We first parallelize the incoherent vertex-dependent computations such as computing the bounce vectors of lighting per view and multi-level filtering (see Sec. 4.1). In addition, we parallelize our BFS method for refining the ROIs of SH lighting. We define an array of outer vertices from the boundary of the initial ROIs of the base level, storing the array as a parallel queue. We test the SH lighting differences of the vertices between the current and higher SH order simultaneously within the queue, yielding a selection of the vertices. We then create another parallel queue of neighboring outward vertices from the selected vertices, preparing the next-level traveling. The tests of lighting differences follow in the same manner. We repeat this process recursively until no vertex is selected.

Results. Here we demonstrate the performance of our method. Refer to our supplemental video for the comparison of our rendering performance with reference computation. Table 1 compares the performance of our method with three objects and various environment maps.

Figure 8 compares the SH diffuse lighting, the accumulated difference map and the initial ROI. Note that the diffuse shadows become sharper as the SH level increases. This shadow frequency changes can be regarded as the accumulated SH light differences. Our initial ROI are filtered level-by-level in respect to the accumulated SH lighting differences.

Figure 9 presents our overall results with respect to frames per second (FPS) and peak signal-to-noise ratio (PSNR). Our method is to render the base lighting for each vertex and add the lighting changes for the ver-

Environment map	Model	# of vertices	SH level	# of SH coeffs.	FPS (ours)	FPS (full cal.)	Filtering ratio (diffuse)	Filtering ratio (specular)	Speed gain
Peter's basilica	Armadillo	52k	10	121	13.7	5.4	13%	80%	2.53
Euclayptus Grove	Armadillo	52k	10	121	12.1	5.5	18%	73%	2.20
Galileo tomb	Armadillo	52k	10	121	13.2	5.4	9%	78%	2.44
Galileo tomb	Max Planck	58k	10	121	13.9	5.1	19%	83%	2.72
Grace Catjedral	Max Planck	58k	10	121	12.3	5.1	9%	79%	2.41
Grace Cathedral	Happy Buddha	59k	9	100	12.7	5.7	10%	78%	2.23
Uffizi Gallery	Happy Buddha	59k	10	121	14.2	4.2	14%	91%	3.38
Peter's Basilisca	Happy Buddha	59k	9	100	14.3	5.7	16%	85%	2.50

Table 1: *Quantitative measurements of our method with four objects with various environmental maps.*

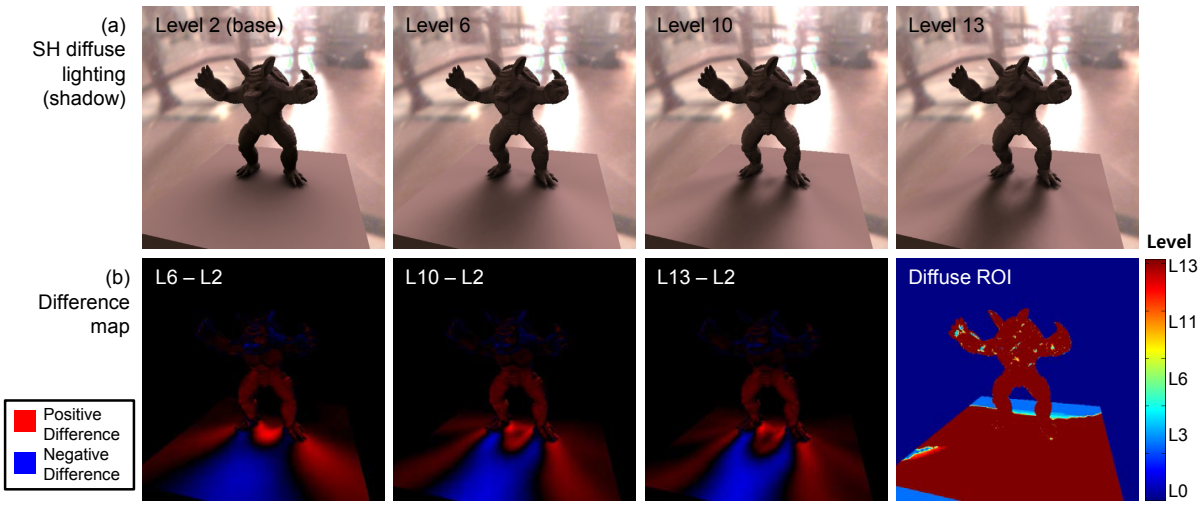

Figure 8: *Level-by-level difference of our SH diffuse shadow lighting. (a) shows the SH diffuse lighting at each level and (b) shows the difference map of the levels against the reference base level.*

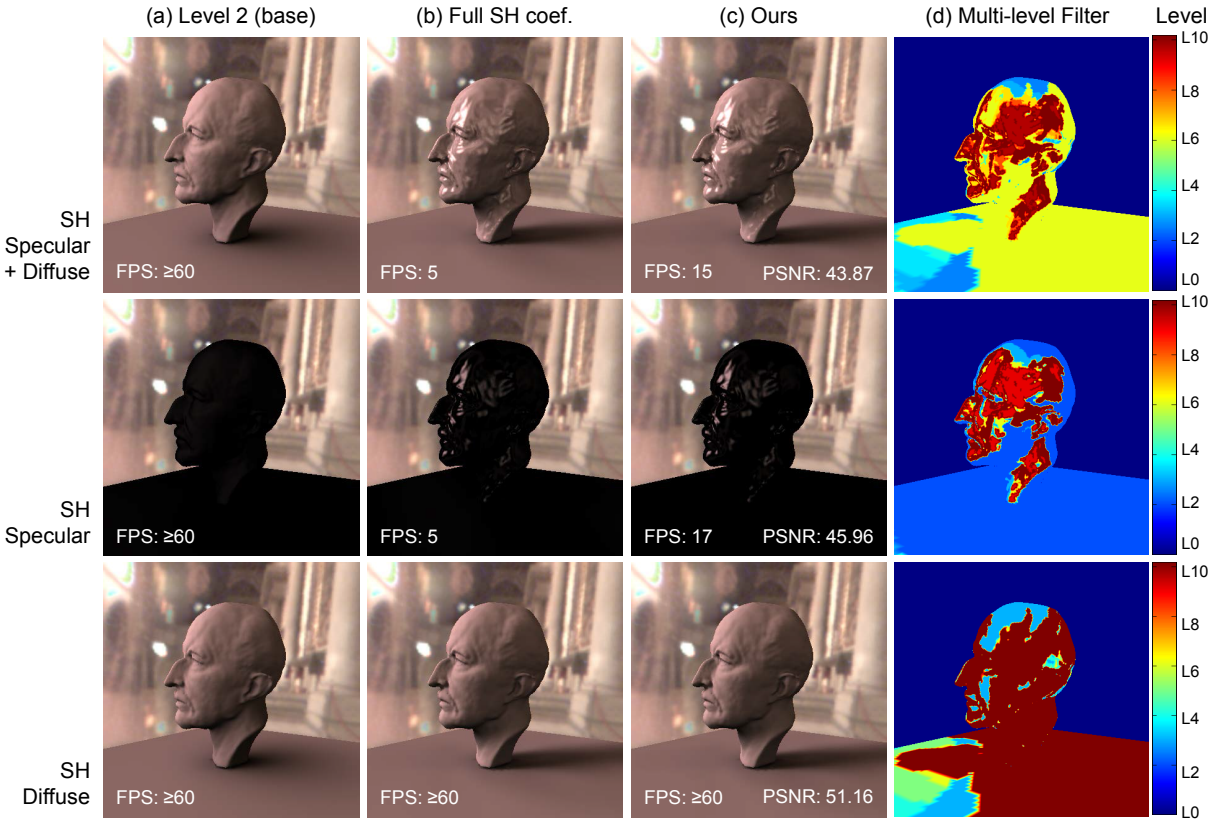

Figure 9: *SH specular and diffuse lighting. (a) shows the reference base level. (b) shows the full calculation of the entire SH coefficients. (c) presents our specular and diffuse lighting. (d) illustrates our multi-level filter for ROIs.*

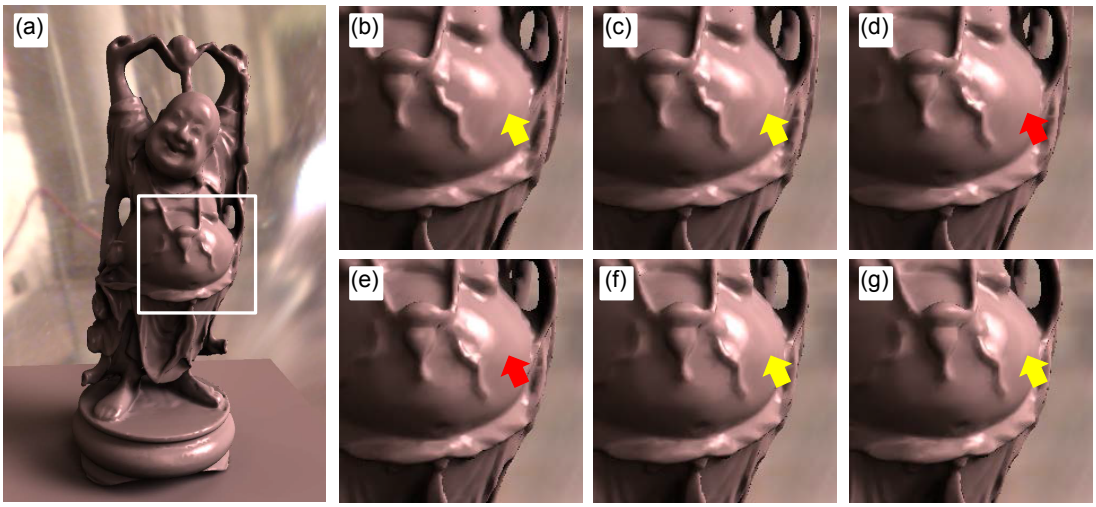

Figure 10: *Evaluation of visual differences between frames. (a) shows our rendering results of SH diffuse/specular lighting of the "Happy Buddha" model that spins about its object axis. (b)–(g) present the close-up views around the belly part of the model (the white box area in (a)) in 0.2 seconds intervals. (b)–(c) show smoother transition from the specular highlight to the diffuse color. However, the frame (d) and (e) bluntly show the sudden changes of the ROI, lacking middle-level specularity while the object spins (compare the regions with the red and the yellow arrows). (f) and (g) start to present the missed specularity suddenly again. Although the visual differences between the frames are subtle in this figure, the difference in the video becomes more noticeable. Refer to the supplemental video for more example.*

tices only within the ROI (the red region of Figure 9(d)). Our specular and diffuse lighting speeds up the rendering time by a factor of 3.0 with a high PSNR as 43.87, which looks virtually identical. It is worth noting that the diffuse ROIs cover overall vertices in the scene while the specular ROIs are distributed in the small regions.

6 DISCUSSION

Our method achieves lower FPSs than the full calculation up to the fourth SH level because the adaptive SH computation with the overhead for checking the lighting differences is more expensive than the raw SH computation. However, as the SH level increases, the cost of SH computation grows faster than the overhead. Therefore, our algorithm can achieve interactive SH rendering in higher than the fourth level. Figure 11 and Table 2 compare the impact of the level of SH coefficients on the rendering speed.

As shown in Table 1, the adaptive filtering performance of our method with specular SH lighting becomes more efficient than that with diffuse SH lighting. Diffuse SH lighting is determined by the hemispherical integral of the product of the ambient occlusion and lighting, while specular SH lighting is affected primarily by the convolution of the incident light vectors around the bounce vector. Most of the vertices are visible with each other while computing the hemispherical integrals of diffuse lighting so that the filtering ratio of our diffuse ROIs is relatively low. In contrast, we need only dominant light information around the specular lobe for rendering specular SH lighting so that our adaptive filtering method become more efficient in particular with specular.

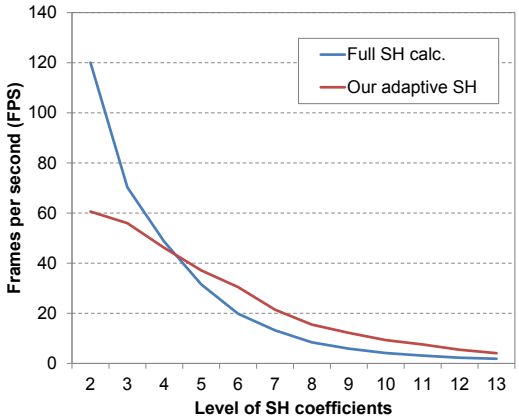

Figure 11: *The level of SH coefficients impacts on rendering speed of our method (model: Happy Buddha).*

We currently implemented our algorithm in the CPU environments. Our method is based on dynamic filtering on SH lighting components for each vertex so that the array size of the SH lighting computation is inconsistent. Therefore, our method might not be efficient in the GPU environments as it is in the CPU environments.

Fig. 10 presents the visual differences between frames in the video. Our adaptive lighting method often filters out some orders of SH coefficients, depending on the threshold level of light difference. When the threshold level is too high, the size of ROIs shrinks rapidly, yielding such flickering artifacts. In other words, the filtering performance could be sacrificed when the threshold level is too

SH Level	# of Coeffs.	FPS (full SH calc.)	FPS (ours)	PSNR
2	9	120.0	60.0	46.72
3	16	70.3	56.0	50.72
4	25	48.6	46.1	45.85
5	36	31.5	37.1	44.29
6	49	19.8	30.5	42.52
7	64	13.2	21.4	43.27
8	81	8.4	15.5	42.86
9	100	5.9	12.2	44.22
10	121	4.2	9.3	44.27
11	144	3.1	7.6	42.80
12	169	2.3	5.4	42.49
13	196	1.8	4.1	42.00

Table 2: *Frame-rate comparison between the full SH coefficients computation and our adaptive method. The PSNR compares the image differences between them.*

low. We could trace back that the typical ringing artifact of spherical harmonic lighting impacts the traveling algorithm of our method (described in Sec. 4), yielding such visual artifacts between frames. We will remove this by attenuating high-frequency lighting coefficients [19] in our future work.

7 CONCLUSION

We propose a simple but practical multi-level filtering solution for genuine spherical harmonic lighting. Our approach evaluates the local variance of SH lighting level by level to filter out vertices that do not require high order computation of SH coefficients. The coarsely determined regions of interests are refined by filling in the incomplete regions by traveling the neighboring vertices from the outer boundary of the ROIs in the breadth-first order. Our method allows to compute high order products of spherical harmonic lighting for both diffuse and specular lighting with a real-time rate by changing the lighting and view.

ACKNOWLEDGMENTS

Min H. Kim gratefully acknowledges support from the National Research Foundation (NRF) of Korea (2013R1A1A1010165 and 2013M3A6A6073718) and additional support from Microsoft Research Asia.

APPENDIX A

This appendix provides pseudo-codes for our multi-level SH filtering and the computation of diffuse and specular SH lighting at each level in our spherical harmonic lighting.

Algorithm 1 Multi-Level SH Filtering

procedure MULTILEVELFILTERING
  for vertex x in scene **do**
    for level i from 0 to *BASE* **do**
      $base[x] = \text{COMPUTESHLIGHT}(i,x)$
    end for
    $diff[x] = 0$
    for level l from $BASE+1$ to *TARGET* **do**
      $diff[x] += \text{COMPUTESHLIGHT}(l,x)$
      if $diff[x] < T$ **then**
        REMOVEFROMROI(x)
      end if
    end for
  end for
  REFINEROI()
end procedure

Algorithm 2 Compute diffuse lighting at level l

procedure COMPUTEDIFFUSE(L,X)
  $levelDiffuse = 0$
  for i from l^2 to $(l+1)^2-1$ **do**
    $levelDiffuse += L(i)O(x,i)$
  end for
end procedure

Algorithm 3 Compute specular lighting at level l

procedure COMPUTESPECULAR(L,X,R)
  $levelDiffuse = 0$
  for i from l^2 to $(l+1)^2-1$ **do**
    $R(i) = 0$
    for j from 0 to $(l+1)^2-1$ **do**
      $R(i) += \hat{Y}(i,j)O(x,j)$
    end for
    $levelSpecular += R(i)\alpha^i Y^i(B(w_o))$
  end for
  for i from 0 to l^2-1 **do**
    for j from l^2 to $(l+1)^2-1$ **do**
      $R(i) += \hat{Y}(i,j)O(x,j)$
    end for
    $levelSpecular += R(i)\alpha^i Y^i(B(w_o))$
  end for
end procedure

REFERENCES

[1] P.-P. Sloan, J. Kautz, and J. Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments," *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2002)*, vol. 21, no. 3, pp. 527–536, 2002.

[2] D. Nowrouzezahrai, P. Simari, and E. Fiume, "Sparse zonal harmonic factorization for efficient sh rotation," *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2012)*, vol. 31, no. 3, pp. 23:1–23:9, 2012.

[3] R. Ng, R. Ramamoorthi, and P. Hanrahan, "All-frequency shadows using non-linear wavelet lighting approximation," *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2003)*, vol. 22, no. 3, pp. 376–381, 2003.

[4] R. Ng, R. Ramamoorthi, and . P. Hanrahan, "Triple

product wavelet integrals for all-frequency relighting," *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2004)*, vol. 23, no. 3, pp. 477–487, 2004.

[5] Y. Inger, Z. Farbman, and D. Lischinski, "Locally adaptive products for all-frequency relighting," *Comput. Graph. Forum*, vol. 32, no. 2, pp. 73–82, 2013.

[6] R. Ramamoorthi and P. Hanrahan, "An efficient representation for irradiance environment maps," in *Proc. the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 497–500.

[7] X. Liu, P.-P. Sloan, H.-Y. Shum, and J. Snyder, "All-frequency precomputed radiance transfer for glossy objects," in *Proceedings of Eurographics Symposium on Rendering 2004*. Eurographics Association, 2004, pp. 337–344.

[8] R. Wang, J. Tran, and D. Luebke, "All-frequency relighting of non-diffuse objects using separable BRDF approximation," in *Proceedings of Eurographics Symposium on Rendering 2004*. Eurographics Association, 2004, pp. 345–354.

[9] Y. Yue, K. Iwasaki, B.-Y. Chen, Y. Dobashi, and T. Nishita, "Interactive rendering of interior scenes with dynamic environment illumination," in *Computer Graphics Forum*, vol. 28, no. 7. Eurographics Association, 2009, pp. 1935–1944.

[10] R. Ramamoorthi, *Precomputation-Based Rendering*. NOW Publishers Inc., 2009.

[11] J. Kautz, P.-P. Sloan, and J. Snyder, "Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics," in *Proceedings of the 13th Eurographics Workshop on Rendering*, ser. EGRW '02. Eurographics Association, 2002, pp. 291–296.

[12] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder, "Clustered principal components for precomputed radiance transfer," *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2003)*, vol. 22, no. 3, pp. 382–391, Jul. 2003.

[13] P.-P. Sloan, X. Liu, H.-Y. Shum, and J. Snyder, "Bi-scale radiance transfer," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 370–375, Jul. 2003.

[14] J. Kautz, J. Lehtinen, and T. Aila, "Hemispherical rasterization for self-shadowing of dynamic objects," in *Eurographics Symposium on Rendering*. Eurographics Association, 2004, pp. 179–184.

[15] P.-P. Sloan, B. Luna, and J. Snyder, "Local, deformable precomputed radiance transfer," *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2005)*, vol. 24, no. 3, pp. 1216–1224, 2005.

[16] Y.-T. Tsai and Z.-C. Shih, "All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation," in *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2006)*, vol. 25, no. 3. ACM, 2006, pp. 967–976.

[17] J. Kautz and M. D. McCool, "Interactive rendering with arbitrary brdfs using separable approximations," in *Rendering Techniques 99*. Springer, 1999, pp. 247–260.

[18] B. T. Phong, "Illumination for computer generated pictures," *Commun. ACM*, vol. 18, no. 6, pp. 311–317, Jun. 1975.

[19] P.-P. Sloan, "Stupid spherical harmonics tricks," in *Proceedings of Game Developers Conference 2008*, San Francisco, 2008.