

석사 학위논문  
Master's Thesis

# 구면조화 조명 기법을 위한 국부 적응적 계산

Locally Adaptive Products for Genuine Spherical Harmonic Lighting

이 주 호 (李 周 浩 Lee, Joo Ho)  
전산학과  
Department of Computer Science

KAIST

2014

# 구면조화 조명 기법을 위한 국부 적응적 계산

Locally Adaptive Products for Genuine Spherical Harmonic Lighting

# Locally Adaptive Products for Genuine Spherical Harmonic Lighting

Advisor : Professor Kim, Min Hyuk

by

Lee, Joo Ho

Department of Computer Science

KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Computer Science. The study was conducted in accordance with Code of Research Ethics<sup>1</sup>.

2014. 6. 17.

Approved by

Professor Kim, Min Hyuk

[Advisor]

---

<sup>1</sup>Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

# 구면조화 조명 기법을 위한 국부 적응적 계산

이 주 호

위 논문은 한국과학기술원 석사학위논문으로  
학위논문심사위원회에서 심사 통과하였음.

2014년 6월 17일

심사위원장 김 민 혁 (인)

심사위원 박 진 아 (인)

심사위원 윤 성 의 (인)

MCS  
20124508

이 주 호. Lee, Joo Ho. Locally Adaptive Products for Genuine Spherical Harmonic Lighting.  
구면조화 조명 기법을 위한 국부 적응적 계산. Department of Computer Science. 2014.  
37p. Advisor Prof. Kim, Min Hyuk. Text in English.

### 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 thesis, 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.

# Contents

Abstract . . . . .	i
Contents . . . . .	ii
List of Tables . . . . .	iv
List of Figures . . . . .	v
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Scope . . . . .	2
1.3 Contributions . . . . .	3
1.4 Thesis Outline . . . . .	3
<b>Chapter 2. Background</b>	<b>4</b>
2.1 Spherical Harmonics . . . . .	4
2.1.1 Integration in SH . . . . .	5
2.1.2 Production Projection in SH . . . . .	5
2.1.3 Convolution in SH . . . . .	6
2.1.4 Rotation Property in SH . . . . .	6
2.1.5 Zonal Harmonics . . . . .	7
2.1.6 Rotation in Zonal Harmonics . . . . .	7
2.2 Wavelet Transform . . . . .	8
2.2.1 Wavelet Transform in 1D . . . . .	8
2.2.2 Wavelet Transform in 2D . . . . .	9
2.3 Light Transport . . . . .	10
2.3.1 Rendering Equation . . . . .	10
2.3.2 Direct Illumination . . . . .	10
2.3.3 Indirect Illumination . . . . .	11
2.3.4 Environment Light Map . . . . .	11
2.3.5 Monte-Carlo Integration . . . . .	12
2.4 Material Appearance . . . . .	12
2.4.1 Diffuse Reflectance Model . . . . .	13
2.4.2 Phong Reflectance Model . . . . .	13
2.4.3 Ward Reflectance Model . . . . .	13
2.4.4 Cook-Torrance Reflectance Model . . . . .	13

<b>Chapter 3.</b>	<b>Previous Work</b>	<b>15</b>
3.1	Spherical Harmonic-based PRTs . . . . .	15
3.1.1	Spherical Harmonic Prefiltering . . . . .	15
3.1.2	The Precomputed Radiance Transfer . . . . .	16
3.1.3	Arbitrary BRDF Shading using Spherical Harmonics . .	16
3.2	Wavelet-based PRTs . . . . .	17
3.2.1	Non-linear Wavelet Lighting Approximation . . . . .	17
3.3	All-Frequency PRTs . . . . .	17
3.4	Other Basis PRTs . . . . .	18
<b>Chapter 4.</b>	<b>Rendering Implementation using SH</b>	<b>19</b>
4.1	Precomputation . . . . .	19
4.1.1	Light Environment Map Integration . . . . .	19
4.1.2	Ambient Occlusion SH Projection . . . . .	20
4.1.3	BRDF SH Projection . . . . .	20
4.2	SH Lighting . . . . .	20
4.2.1	Diffuse SH Lighting . . . . .	20
4.2.2	Specular SH Lighting . . . . .	21
<b>Chapter 5.</b>	<b>Multi-Level SH Lighting</b>	<b>22</b>
5.1	Multi-Level Filtering . . . . .	22
5.1.1	Diffuse ROI . . . . .	23
5.1.2	Specular ROI . . . . .	24
5.2	Refinements . . . . .	26
5.2.1	Refining the ROIs . . . . .	26
5.2.2	Smoothing the Boundaries . . . . .	26
<b>Chapter 6.</b>	<b>Results</b>	<b>28</b>
6.1	Implementation . . . . .	28
6.2	Experiment Measurements . . . . .	28
6.3	Results . . . . .	29
<b>Chapter 7.</b>	<b>Discussion and Future Work</b>	<b>34</b>
<b>Chapter 8.</b>	<b>Conclusion</b>	<b>35</b>
<b>References</b>		<b>36</b>
<b>Summary (in Korean)</b>		<b>38</b>

## List of Tables

6.1	Quantitative Measurements of Our Method . . . . .	29
6.2	Frame-Rate Comparison between the Full Computation and Our Adaptive Methods . . .	29

# List of Figures

2.1	Basis Functions of Spherical Harmonics . . . . .	5
2.2	Wavlet Decomposition . . . . .	9
2.3	Coordinate Conversion from an Image Space to a Spherical Space . . . . .	12
5.1	Schematic Overview of Our Multi-Level SH Lighting Framework . . . . .	22
5.2	Multi-level SH Filtering . . . . .	23
5.3	Computation of SH Diffuse Lighting . . . . .	24
5.4	Computation of SH Specular Lighting . . . . .	25
5.5	Filling in the Incomplete ROIs . . . . .	27
5.6	Smoothing the Boundaries . . . . .	27
6.1	Rendering Speed in resepct to the Number of Coefficients . . . . .	30
6.2	Level-by-level Difference of Our SH Diffuse Shadow Lighting. . . . .	31
6.3	SH Specular and Diffuse Lighting . . . . .	32
6.4	Global Illumination . . . . .	33
6.5	Evaluation of Visual Differences between Frames . . . . .	33

# Chapter 1. Introduction

## 1.1 Motivation

Recently, while computer graphics was mostly used in the film industry in the past, needs for interactive interfaces such as computer games and human computer interfaces is increasing fast. Billions of people playing the video games for their happiness and examining goods such as jewels and cars, whether it is fancy or not through a computer. These needs for the interactive interfaces turns out the significant expectation of interactive graphics since high image quality and the high frame-rate make the interactive contents rich and dynamic.

There are many researches to render the realistic scene in computer graphics since the fundamental goal of computer graphics is to generate a virtual scene from the digitized virtual objects and lights, which cannot be distinguishable with the real scene. The researchers propose the sampling lighting methods such as Monte-Carlo ray tracing, virtual point lights, photon mapping, and path tracing to catch the complex lighting effects like shadows, caustic and inter-reflection. However, catching the complex light transfer is much challenging in interactive graphics.

For these reasons, precomputed radiance transfer (PRT) methods are proposed. General precomputed radiance transfer techniques compute light impacts for an outgoing direction with respect to an incoming light direction for each vertex before a rendering step. In other words, significant lighting factors such as visibility, light, and a bidirectional reflection distribution function, are decomposed into vectors of the orthonormal basis coefficients. Then, contrary to Monte-Carlo sampling integration, we compute the rendering equation as the linear matrix operation among the vectors of light factors for each vertex. The merit of the precomputed radiance transfer techniques is an interactive rendering with global illumination since the rendering equation can be computed with simple linear operation. In the other hand, even though updating the minor scene information like object translation, it causes the re-computation for packing the light factors which takes a lot of times.

There are two majors of precomputed radiance transfer techniques: spherical harmonic based precomputed radiance transfer techniques and the wavelet-based precomputed radiance transfer techniques. Spherical harmonic based precomputed radiance transfer methods decompose the light factors into the spherical harmonic coefficients and compute the color as the linear sum of the spherical harmonic coefficients. Spherical harmonics are a spherical version of Fourier transform, so have many similar properties with Fourier transform over a circle. The benefit of spherical harmonics is the lighting factors

can be decomposed in respect to a frequency band. It means we can omit high-frequency lighting to achieve interactive frame rates. Also, the other benefit of spherical harmonics is that, with only a set of all spherical harmonic basis functions in a band, we can describe any rotation of the spherical function, which is the weighted sum of the basis functions in the set. Compared to other basis based precomputed radiance transfer methods, spherical harmonic based precomputed radiance transfer methods can render the scene with free rotation for light, view, or object orientation, due to this property, known as rotation invariance. However, the rendering rate drops down dramatically to depict the high-frequency detail. The reason is that the number of coefficients increases as the quadratic of the level. It causes the significant computation cost. Therefore, people use the spherical harmonic based precomputed radiance transfer techniques for low-frequency lighting and the wavelet-based precomputed radiance transfer techniques came out for high-frequency lighting.

The wavelet-based precomputed radiance transfer techniques deal with all-frequency lighting. It decomposes the light factors into wavelet coefficients and sorts out the significant wavelet coefficients from them. Only sorted-out coefficients are computed for rendering the scene. Compared to Fourier transform, which computes global frequency over the defined region, the wavelet transform computes the gradient varying the local frequency information. The benefit of wavelets is that the wavelet coefficients for light factors can be compressed efficiently because most wavelet coefficients are zero. With the same number of coefficients, wavelets can present higher frequency of the scene information than spherical harmonics, keeping the low-frequency lighting. However, the computation of rotation is complicated due to the nature of wavelets. The more efforts on programming with graphic hardware are required if we compute the rotation on the wavelet domain.

## 1.2 Scope

This thesis is started from the following motivation. Rotating the scene or light, possible in a spherical harmonic lighting framework, is a fascinating function for the visualization for goods. We try to keep this rotation-free properties and avoid speed drop. In this thesis, we propose a simple but practical lighting solution using locally adaptive products of SH coefficients only within the SH lighting framework. Our method is designed for a vertex shading framework and uses CPU parallelization for speedup.

We reduce the computation cost by cutting the high-frequency lighting computation off for each vertex. To this end, we define the importance for frequency lighting and compute it. We then find the region of interest (ROI) for high-frequency lighting through multi-level filtering and compute the high-frequency lighting only for them.

We propose a complement solution for the multi-level filtering method. We remove the discontinuity of the ROIs by traveling the neighbors of the ROIs. Also, we remove the edge artifacts, caused by a

threshold mechanism. We disperse the gradient at the boundary of the ROIs attenuated with respect to the level of tiers. These operations are performed over a vertex graph.

### 1.3 Contributions

In this thesis, we compute the high-order products of diffuse and specular lighting on given limited computational resources and rendering higher frequency lighting exclusively. The following contributions have been made:

- **A novel multi-level filtering approach for locally adaptive products.** We propose the multi-level filtering method to reduce the computation costs on the rendering time. For each vertex, we compute the importance for high-frequency lighting and determine whether high-frequency lighting is significant or not.
- **A refining method of the vertex ROIs using the breadth-first search method.** We remove the discontinuities of the ROIs and the edge artifact at the boundary of the region of interests. To this end, we travel the neighbor of the ROIs in the breadth-first order and disperse the contrast in respect of the tier of neighbors.

### 1.4 Thesis Outline

Chapter 2 introduces the background for the spherical harmonics and the precomputed radiance transfer techniques. Chapter 3 presents the related works on the precomputed radiance transfer techniques. Chapter 4 describes the overview for our spherical harmonic lighting framework. Chapter 5 proposes our locally adaptive spherical harmonic method. Chapter 6 demonstrates our results. Chapter 7 discusses the future work and the limitation. Chapter 8 concludes this thesis.

## Chapter 2. Background

This chapter overviews the preliminaries of this thesis. It first explains two orthonormal basis functions, spherical harmonics and wavelets, which are used for decomposing the lighting factors. Then, we summarize the foundations of the global illumination rendering: the rendering equation, popular reflection model, and lighting techniques.

### 2.1 Spherical Harmonics

Spherical harmonics can be considered as the spherical version of the Fourier transform. 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 basis functions of this system are 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 mode 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:

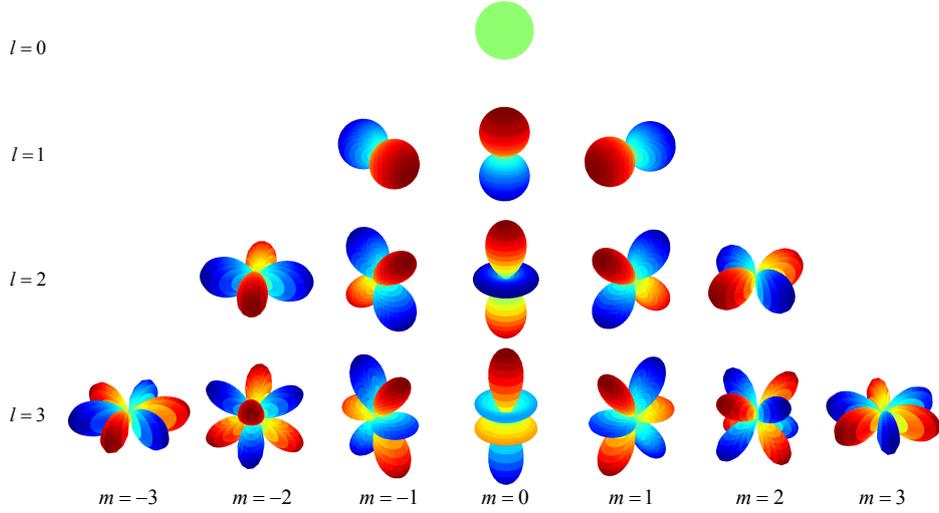
$$y_l^m = \begin{cases} \sqrt{2}K_l^m \cos(m\phi)P_l^m(\cos\theta), & m > 0 \\ \sqrt{2}K_l^m \sin(-m\phi)P_l^{-m}(\cos\theta), & m < 0 \\ K_l^0 P_l^0(\cos\theta), & m = 0, \end{cases}$$

where  $K_l^m$  is the normalized coefficients.

Figure 2.1 shows the spherical harmonic basis functions within four levels. The frequency of the spherical harmonic basis function increases as the spherical harmonic level  $l$  increases. The mode  $m$  represents the form of the basis functions. If  $m$  is zero, the basis function oscillates only in the polar domain. If  $m$  is the absolute value of the spherical harmonic level, it fluctuates only in the azimuthal domain. The functions of negative  $m$  and positive  $m$  have  $90/m$  degree of the phase difference on the polar domain.

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:

$$\tilde{f}(s) = \sum_{l=0}^{n-1} \sum_{m=-l}^l f_l^m y_l^m(s) = \sum_{i=0}^{n^2-1} f_i y_i(s), \quad (2.1)$$



**Figure 2.1:** Basis functions of spherical harmonics. The row index  $l$  represents the spherical harmonic frequency. The column index  $m$  decides the frequency distribution.

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

### 2.1.1 Integration in SH

We compute diffuse lighting as the integration of 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.

### 2.1.2 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:

$$\begin{aligned}
 F_i &= \int_S \left( \sum_j G_j y_j(s) \sum_k H_k y_k(s) \right) y_i(s) ds \\
 &= \sum_j \sum_k G_j H_k \int_S y_i(s) y_j(s) y_k(s) ds \\
 &= \sum_j \sum_k G_j H_k \hat{Y}_{ijk} = \sum_j G_j \hat{H}_{ij},
 \end{aligned} \tag{2.2}$$





## 2.2 Wavelet Transform

The wavelet transform is another decomposition technique for signals. The researchers decompose an image into wavelet coefficients to analyze the local frequency of the image. The wavelet-based precomputed radiance transfer methods decompose each image of cubes for visibilities, the BRDF, and the environment light map into the wavelet coefficients. After that, they pick out the significant coefficients to compute the rendering equation. In this section, we explain the Haar wavelets which Ng et al. apply in the PRT [4].

### 2.2.1 Wavelet Transform in 1D

Harr wavelets have two kinds of the orthonormal basis functions: a scaling function and a wavelet function defined with a scale index  $j = 0, 1, 2, 3, \dots$  and a location index  $k = 0, \dots, 2^j - 1$ . The scaling function  $\varphi_{j,k}$  is defined as:

$$\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k), \quad (2.4)$$

where  $\varphi(x)$  is a basic scaling function:

$$\varphi(x) = \begin{cases} 1, & 0 \leq x < 1 \\ 0, & \textit{otherwise} . \end{cases} \quad (2.5)$$

A local intensity of a function  $f$ , defined over  $[0, 1)$ , can be measured as the scaling coefficient  $w_{\varphi_{j,k}}$  in respect to a scale  $j$ , which is the integration of the intensity  $\int f(x)\varphi_{j,k}(x)dx$ .

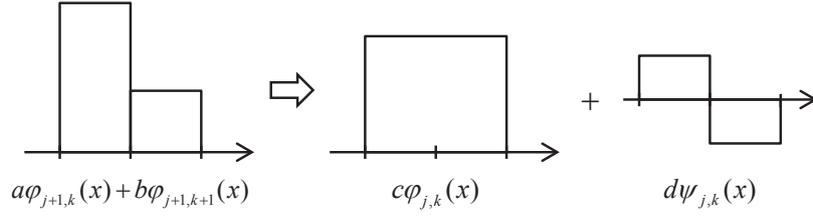
The wavelet function  $\psi_{j,k}$  is defined as:

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k), \quad (2.6)$$

where  $\psi(x)$  is a basic wavelet function:

$$\psi(x) = \begin{cases} -1, & 0 \leq x < \frac{1}{2} \\ 1, & \frac{1}{2} \leq x < 1 \\ 0, & \textit{otherwise} . \end{cases} \quad (2.7)$$

The wavelet coefficient  $w_{\psi_{j,k}}$ , defined as  $\int f(x)\psi_{j,k}(x)dx$ , represents a local gradient of a function  $f$  related to scale  $j$ . The sum of wavelet functions and scaling functions at scale  $j$  can represent every function which is the sum of scaling functions at scale  $j + 1$  as Figure 2.2 shows.



**Figure 2.2:** *Wavelet decomposition. The sum of two scaling wavelet functions at scale  $j + 1$  can be represented as the sum of the scaling function and the wavelet function at scale  $j$ , where  $a$ ,  $b$ ,  $c$ , and  $d$  are coefficients for the basis functions.*

A function  $f$  can be recovered via 1D wavelet transform:

$$f(x) = \sum_k w_{\varphi_{j_s, k}} \varphi_{j_s, k}(x) + \sum_{j=j_s}^{\infty} \sum_k w_{\psi_{j, k}(x)} \psi_{j, k}(x), \quad (2.8)$$

where  $j_s$  is the minimum scale.

### 2.2.2 Wavelet Transform in 2D

2D wavelets are the products of 1D wavelets at different axes  $x$  and  $y$ . 2D wavelets have four basis functions, one scaling function and three wavelet functions. The 2D scaling function  $\varphi_{j, m, n}$  and wavelet functions  $\psi_{j, m, n}^i$  are defined as:

$$\begin{aligned} \varphi_{j, m, n}(x, y) &= 2^{j/2} \varphi(2^j x - m, 2^j y - n) \\ \psi_{j, m, n}^i(x, y) &= 2^{j/2} \psi^i(2^j x - m, 2^j y - n), i = \{H, V, D\}, \end{aligned} \quad (2.9)$$

where  $j$  is a scale index;  $m$  is a translation index for  $x$  axis;  $n$  is a translation index for  $y$  axis;  $\varphi(x, y)$  is a 2D scaling basic function;  $\psi^i(x, y)$  is a 2D wavelet basic function;  $\{H, V, D\}$  represents a direction of the wavelet function. The 2D basic functions are the products of the 1D basic functions:

$$\begin{aligned} \varphi(x, y) &= \varphi(x)\varphi(y), \\ \psi^H(x, y) &= \psi(x)\varphi(y), \\ \psi^V(x, y) &= \varphi(x)\psi(y), \\ \psi^D(x, y) &= \psi(x)\psi(y). \end{aligned} \quad (2.10)$$

A 2D function  $f(x, y)$ , defined over  $[0, 1) \times [0, 1)$ , can be decomposed into the 2D wavelet coefficients:

$$\begin{aligned} w_{\varphi_{j, m, n}} &= \int \int \varphi_{j, m, n}(x, y) f(x, y) dx dy, \\ w_{\psi_{j, m, n}^i} &= \int \int \psi_{j, m, n}^i(x, y) f(x, y) dx dy, \end{aligned} \quad (2.11)$$

where  $w_{\varphi_{j,m,n}}$  is the scaling coefficient;  $w_{\psi_{j,m,n}}$  is the wavelet coefficient. The function  $f$  can be obtained from the 2D wavelet coefficients via 2D wavelet transform:

$$f(x, y) = \sum_m \sum_n w_{\varphi_{j_s, m, n}} \varphi_{j_s, m, n}(x, y) + \sum_{i=H, V, D} \sum_{j=j_s}^{\infty} \sum_m \sum_n w_{\psi_{j, m, n}^i} \psi_{j, m, n}^i(x, y). \quad (2.12)$$

## 2.3 Light Transport

The light transports describe the realistic light effects such as color bleeding and caustic. Section 2.3.1 shows how we define the light transport into a mathematic equation. Section 2.3.2 and Section 2.3.3 discuss the direct and indirect illumination. We describe the coordinate conversion of the environment light map in Section 2.3.4. Finally, we introduce the Monte-Carlo integration to compute the rendering equation.

### 2.3.1 Rendering Equation

In the real world, light comes into our eyes through physical phenomena such as reflection, refraction, scattering, absorption, and emission. The rendering equation is an equation which represents these light transfer at point  $x$ . The outgoing radiance in direction  $w_o$  at point  $x$  comprises the emitted radiance in the same direction and the reflected radiance for the incoming radiance from every direction  $w_i$ . For reflection, the reflection ratio varies in respect to incoming direction and outgoing direction. As the incoming radiance is getting near the surface, the impact of the incoming radiance is getting small. We can define the rendering equation at point  $x$  as below:

$$L(x \rightarrow w_o) = L_e(x \rightarrow w_o) + \int_{\Omega} L(w_i \rightarrow x) f_r(w_i, w_o) V(x, w_i) (w_i \cdot N_x) dw_i, \quad (2.13)$$

where  $w_i$  is the incoming vector;  $w_o$  is the outcoming vector;  $N_x$  is the normal of the point  $x$ ;  $L_e$  is the emitted radiance;  $L_i$  is the incoming radiance;  $f_r$  is the bidirectional radiance distribution function;  $V$  is the visibility function;  $\rightarrow$  notation describes whether the direction is incoming or outcoming. In our thesis, we only take the direct diffuse reflection and the direct specular reflection into account and compute it as the linear operation among the SH coefficients of lighting components.

### 2.3.2 Direct Illumination

We can arrange the light transport in terms of the number of reflection. The direct illumination is to illuminate objects from light directly. It considers only effects of unoccluded incoming radiances from emitted light at each vertex. Since we use the light environment map in this thesis. The rendering

equation for direct illumination is defined as below:

$$L_{direct}(x \rightarrow w_o) = \int_{\Omega} L_{env}(w_i) f_r(w_i, w_o) V(x, w_i) (w_i \cdot N_x) dw_i, \quad (2.14)$$

where  $L_{env}(w_i)$  is the incoming radiance from the environment map at direct  $w_i$ .

### 2.3.3 Indirect Illumination

Indirect illumination is to illuminate the scene with the light with at least one bounce. In order to compute indirect illumination, we compute the reflected radiance bounce-by-bounce by integrating the previous reflected radiance:

$$L_i(x \rightarrow w_o) = \int_{\Omega} L_{i-1}(w_i \rightarrow x) f_r(w_i, w_o) (1 - V(x, w_i)) (w_i \cdot N_x) dw_i,$$

where  $L_i$  is the radiance with  $i$  bounces.  $L_0$  will be  $L_{env}$  in this thesis. The indirect illumination is an infinite sum of the bounced radiances  $L_{indirect} = \sum_{i=1}^{\infty} L_i$ .

### 2.3.4 Environment Light Map

The environment light map stores radiances from all incoming direction. The environment light map enables rendering the scene with area lighting. This thesis uses Debevec’s light probe images [5]. In order to compute the corresponding 3D direction of pixel  $(i, j)$ , we first change the coordinate space to  $uv$  space. In  $uv$  space, the center of an image is origin and the top-right point is  $(1, 1)$ :

$$(u, v) = \left( \frac{2j}{W} - 1, 1 - \frac{2i}{H} \right), \quad (2.15)$$

where  $W$  is the width of the image;  $H$  is the height of the image. We compute a spherical coordinate  $(\phi, \theta)$  of a pixel (see Figure 2.3(a)):

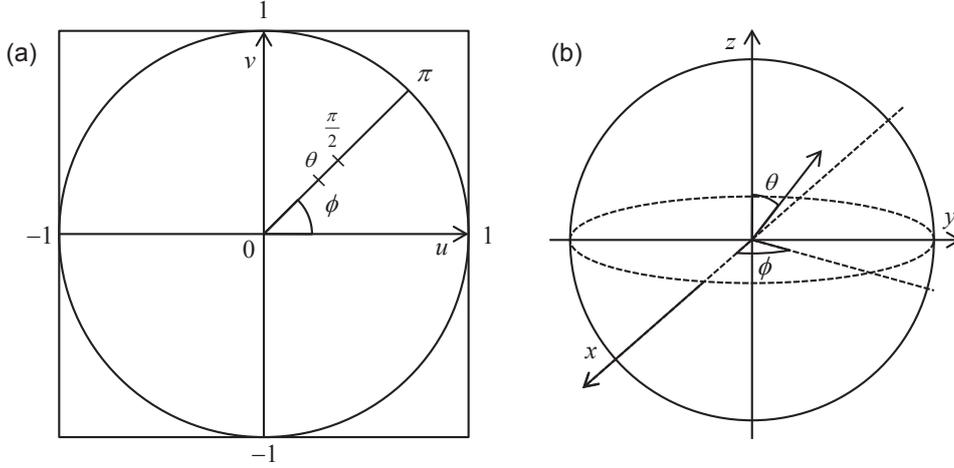
$$(\phi, \theta) = (atan2(u, v), r\pi), \quad (2.16)$$

where  $r$  is the distance from a origin  $u^2 + v^2$ ;  $atan2(u, v)$  is a inverse tangent function of two parameters whose range is  $(-\pi, \pi]$ . The 3D direction of pixel  $(i, j)$  is computed as (see Figure 2.3(b)):

$$(x, y, z) = (\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta). \quad (2.17)$$

Note that the center of the light probe images has  $z$ -direction  $(0, 0, 1)$ . Suppose that we wrap the sphere with the light probe image, while aligning  $uv$  image axis to  $xy$  axis and matching the center of the

image to point  $(0, 0, 1)$  of the sphere.



**Figure 2.3:** *Coordinate Conversion from an image space to a spherical space. (a) represents mapping between spherical coordinates and  $uv$  image coordinates. (b) describes mapping between cartesian coordinates and spherical coordinates.*

### 2.3.5 Monte-Carlo Integration

We precompute the SH coefficients via Monte-Carlo integration. The Monte-Carlo integration is the numerical integration method for random sampling. Let  $f(x)$  a function defined over  $A$  and  $p(x)$  a probability density function defined over  $A$ . The Monte-Carlo integration estimates the integration without bias follows:

$$\begin{aligned}
 E\left[\frac{1}{N} \sum_{i=0}^{n-1} \frac{f(x_i)}{p(x_i)}\right] &= \int_A \cdots \int_A \frac{1}{N} \sum_{i=0}^{n-1} \frac{f(x_i)}{p(x_i)} p(x_i) dx_0 \cdots dx_n \\
 &= \frac{1}{N} \sum_{i=0}^{n-1} \int_A \frac{f(x_i)}{p(x_i)} p(x_i) dx_i = \frac{1}{N} \sum_{i=0}^{i=n-1} I = I,
 \end{aligned} \tag{2.18}$$

where  $x_i$  is a random variable;  $I$  is the integration  $\int_A f(x) dx$ ;  $N$  is the number of coefficients.

## 2.4 Material Appearance

We can distinguish objects because objects have different material appearances. A paper reflects incoming light all over a hemisphere with same radiance and a mirror reflects light only in the bounced direction. In computer graphics, the several analytic appearance models are proposed to describe the material appearance accurately. In this section, we will introduce the diffuse reflection and the several analytic appearance models.

### 2.4.1 Diffuse Reflectance Model

Most objects look the same color during changing a view angle, known as diffuse reflection. The reason for it is that the unit surface of the object spreads light energy over a hemisphere in cosine manner. For example, as the viewing angle  $\theta$  from the surface normal increases, the surface density increases as  $1/\cos\theta$ . These two terms cancel out each other. Therefore, the radiance for diffuse materials is not physically changed even the view point is changed.

### 2.4.2 Phong Reflectance Model

The phong reflectance model [6] is an empherical model that represents the diffuse reflection and the specular reflection of surfaces:

$$f_r(x, w_i, w_o) = K_a + K_d(w_i \cdot N_x) + K_s(B(w_i) \cdot w_o)^n, \quad (2.19)$$

where  $K_a$  is the ambient coefficient;  $K_d$  is the diffuse coefficient;  $K_s$  is the specular coefficient;  $w_i$  is the incident direction;  $w_o$  is the viewing direction;  $n$  is the shiness factor;  $B(w_i)$  is the bounded direction of the incident direction  $w_i$ . In our thesis, we apply the Phong reflectance model without the ambient term. We compute the specular reflection as convolution of the isotropic specular lobe in SH domain.

### 2.4.3 Ward Reflectance Model

The Ward reflectance model [7] uses an exponential function for the specular lobe instead while the Phong reflectance model uses the cosine term. It also parameterizes the reflectance ratio for incoming radiance with the halfway angle  $\theta_h$ , an angle between the halfway vector and the surface normal where the halfway vector is the sum of an incoming vector and an outgoing vector. The equation is as below:

$$f_r(w_i \rightarrow w_o) = \rho_d/\pi + \rho_s \frac{1}{\cos\theta_i \cos\theta_r} \frac{e^{-\tan^2\theta_h/\alpha^2}}{4\pi\alpha^2}, \quad (2.20)$$

where  $\rho_d$  is a parameter for diffuse reflection;  $\rho_s$  is a parameter for specular reflection;  $\theta_i$  is an incoming angle;  $\theta_o$  is an outgoing angle;  $\theta_h$  is an angle between the halfway vector and the surface normal;  $\alpha$  is a parameter for the shininess.

### 2.4.4 Cook-Torrance Reflectance Model

The Cook-Torrance reflectance model [8] is the geometric optics-based reflectance model. It has three terms to represent the reflectance: the Frensel function, the microfacet distribution function, and the geometric masking and shadow function. The Frensel function describes the fraction of reflected light in

terms of the halfway angle. The microfacet distribution function represents the distribution of orientation of microfacets. The geometric masking and shadow function describes the occlusion of microfacets. With these three terms, we can compute the reflected light, which is not occluded by microfacets, in the viewing direction. It predicts that the max value of the reflectance is located near the surface, not in the bounded direction exactly. The grazing peaks, called as off-specular peaks, is caused by shadowing and masking. The equation for Cook-Torrance reflectance model is below:

$$f_r(w_i \rightarrow w_r) = \frac{F(\theta_h)}{\pi} \frac{D_{Beckmann}(\theta_h)G(\theta_i, \theta_r)}{\cos\theta_i \cos\theta_r}, \quad (2.21)$$

where  $F$  is the Fresnel function;  $D_{Beckmann}$  is the microfacet distribution model which is the slope distribution in terms of the slop parameter  $alpha$ , assuming that the fluctuation in height of the surface is Gaussian;  $G$  is the shadowing and masking function.

## Chapter 3. Previous Work

The precomputed radiance transfer techniques are improved by changing the basis or compressing the computation after Sloan et al. [1] propose the framework of the precomputed radiance transfer at first. This section introduces the precomputed radiance transfer using spherical harmonic in detail and overviews the development of other precomputed radiance transfer techniques over a decade.

### 3.1 Spherical Harmonic-based PRTs

The spherical harmonic-based precomputed radiance transfer can render the diffuse lighting with only 9 coefficients within 1% error [9]. However the high-frequency shadowing and specular representation need a large set of of basis functions.

#### 3.1.1 Spherical Harmonic Prefiltering

Ramamoothi and Hanrahan [9] propose the precomputed radiance transfer for Lambertian reflection using spherical harmonics. They use three orders of spherical harmonics and take no consideration of visibility and interreflection. Irradiance  $E$  per vertex is computed as the convolution (equation (2.3)) of the environment light map and the cosine term in SH domain:

$$E_{l,m} = \sqrt{\frac{4\pi}{2l+1}} A_l L_{l,m}, \quad (3.1)$$

where  $E_{l,m}$  is the SH coefficients for the irradiance;  $A_l$  is the SH coefficients for the cosine term;  $L_{l,m}$  is the SH coefficients for the environment light map.

They first decompose the environment light map and the cosine term into the SH vectors  $L$  and  $A$ . For rendering, they compute the irradiance  $E(\theta, \phi)$  which is a function of the surface normal  $n = (x, y, z, 1)^t$  via a vector-matrix multiplication as follows:

$$E(n) = n^t M n = n^t \begin{pmatrix} c_1 L_{2,2} & c_1 L_{2,-2} & c_1 L_{2,1} & c_2 L_{1,1} \\ c_1 L_{2,-2} & -c_1 L_{2,2} & c_1 L_{2,-1} & c_2 L_{1,-1} \\ c_1 L_{2,1} & c_1 L_{2,-1} & c_3 L_{2,0} & c_2 L_{1,0} \\ c_2 L_{1,1} & c_2 L_{1,-1} & c_2 L_{1,0} & c_4 L_{0,0} - c_5 L_{2,0} \end{pmatrix} n, \quad (3.2)$$

where  $c_1$  is 0.429043;  $c_2$  is 0.511664;  $c_3$  is 0.743125;  $c_4$  is 0.886227;  $c_5$  is 0.247708.

### 3.1.2 The Precomputed Radiance Transfer

Sloan et al. [1] presents the seminal paper about the precomputed radiance transfer. They render the specular reflection and the diffuse reflection with the Phong reflection model, taking shadows and interreflection into account. This section explains the method for direct lighting with shadow.

For diffuse lighting, the cosine weighted visibility  $O$  at each vertex is stored as the SH vector form. The diffuse color at a vertex requires the dot product of the cosine weighted visibility  $O$  and the environment map  $L$ :

$$c_d = \rho_x \sum_{i=0}^{n^2-1} L_i O_i, \quad (3.3)$$

where  $\rho_x$  is the albedo at the vertex;  $n$  is the order of spherical harmonics;  $L_i$  is the  $i$ th SH coefficients for the environment light map.

The specular lighting is computed as the convolution of the specular lobe and the incoming radiance at a vertex. To obtain the SH vector of the incoming radiance, Sloan et al. conducts the SH product projection of the environment light map  $L$  and the visibility  $V$ . Then, they convolve the incoming radiance with a  $z$ -aligned specular lobe and finally estimate the value of the convolution at a bounded view direction:

$$c_s(w_v) = \sum_{i=0}^{n^2-1} \sqrt{\frac{2\pi}{4l+1}} G_i \sum_{j=0}^{n^2-1} M_{ij} L_j y_i(B(w_v)), \quad (3.4)$$

where  $G_i$  is the SH coefficients for the  $z$ -aligned specular lobe;  $M_{ij}$  is the product projection matrix for visibility  $V$ ;  $w_v$  is the viewing direction;  $B(w_v)$  is the bounded direction for the viewing direction. Sloan et al. [10] applied the principal component analysis (PCA) clustering method to accelerate their work [1].

### 3.1.3 Arbitrary BRDF Shading using Spherical Harmonics

Kautz et al. [11] render the arbitrary BRDF via the precomputed radiance technique. They decompose the BRDF into the SH coefficients for each viewing direction, storing it as  $n^2$  texture images  $F_i$ , where the texture coordinate represents the viewing direction:

$$F_i(w_o) = \int_S f(w, w_o) y_i(w) dw, \quad (3.5)$$

where  $F_i(w_o)$  is the  $i$ th SH coefficients for the BRDF in terms of the direction  $w_o$ ;  $w_o$  is the viewing direction;  $y_i$  is the  $i$ th SH basis function;  $w$  is the incoming direction. The exit radiance at vertex  $x$  is the dot product of the SH vectors for the viewing direction  $w_o$  and for the environment light map  $L$ :

$$L(x \rightarrow w_o) = \int_S f(x, w, w_o) L(x \leftarrow w) dw = \sum_{i=0}^{n^2-1} F_i(w_o) L_i. \quad (3.6)$$

In the rendering stage, they compute the exit direction  $w_o$  and pluck the SH coefficient  $F_i(w_o)$  from the stack of texture images.

## 3.2 Wavelet-based PRTs

Compared to the spherical harmonic-based precomputed radiance transfer, the wavelet-based precomputed radiance transfer can approximate the high-frequency detail with less coefficients.

### 3.2.1 Non-linear Wavelet Lighting Approximation

Ng et al. [4] 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 diffuse scenes at arbitrary viewing directions.

They divide the rendering equation into two terms: light and a transport function. The transport function is the multiplication of visibility, the BRDF, and the cosine term. It is not related to the outgoing direction in both rendering cases. The rendering equation can be considered as below:

$$\begin{aligned} L_o(x) &= \int_{\Omega} L_{env}(w_i \rightarrow x) f_r(w_i, w_o(x)) (w_i \cdot N_x) dw_i \\ &= \int_{\Omega} L_{env}(w_i) T(x, w_i) dw_i, \end{aligned} \quad (3.7)$$

where  $L_o$  is an outgoing radiance to the viewer;  $L_{env}$  is an environment light map;  $f_r$  is a BRDF;  $N_x$  is the surface normal at point  $x$ ;  $w_i$  is an incoming direction. Then they decompose the each rendering factors into the wavelet coefficients. The rendering equation can be arranged as the matrix multiplication:

$$\begin{aligned} L_o(x) &= \int_{\Omega} L_{env}(w_i) T(w_i) dw_i = \int_{\Omega} \left( \sum_j L_j W_j(w_i) \right) \left( \sum_k T_j W_k(w_i) \right) dw_i \\ &= \sum_j \sum_k L_j T_k \int_{\Omega} W_j(w_i) W_k(w_i) dw_i = \sum_j \sum_k L_j T_k, \end{aligned} \quad (3.8)$$

where  $W_i$  is a wavelet basis function;  $L_j$  is the wavelet coefficient for light;  $T_k$  is the wavelet coefficient for transport;  $L_o(x)$  is the outgoing radiance at point  $x$ . They extended their work [4] to triple products of wavelet integrals for rendering all-frequency direct illumination, allowing to change the viewpoint and the light direction.

## 3.3 All-Frequency PRTs

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. [12] 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. Liu et al. [13] and Wang et al. [14] 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 [13]. Kautz et al. [15] proposed a hemispherical rasterization method that recomputes visibility for self-shadowing of dynamic objects. Inger et al. [16] recently proposed a locally adaptive product approach using discrete cosine transformation to achieve a real-time frame rate with high vertex counts.

### 3.4 Other Basis PRTs

Sloan et al. [3] 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 [17] 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 [18] recently introduced a sparse ZH factorization method for rotating harmonic coefficients efficiently.

In this thesis, 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.

## Chapter 4. Rendering Implementation using SH

In this section, we briefly introduce the SH rendering algorithm. SH rendering can be divided into two major steps: precomputation and SH lighting. In the precomputation step, light-related information such as a light environment map, ambient occlusion and a BRDF is decomposed into SH vectors. In the rendering step, instead of casting and tracing rays, linear calculation of the SH vectors is required to get color of each vertex.

### 4.1 Precomputation

We decompose three light components such as an environment light map, an ambient occlusion, and a BRDF into the spherical harmonic coefficients. In the rendering stage, we compute the rendering equation as the linear operation among the SH vectors. We can change the light and the BRDF lobe on the fly as we separate the rendering equation into three lighting factors. Section 4.1.1 describes the precomputation for the light environment map. Section 4.1.2 explains the integration for the ambient occlusion. We introduces the BRDF SH projection in Section 4.1.3.

#### 4.1.1 Light Environment Map Integration

We use the environment radiance map (see Section 2.3.4) as the incoming light. We calculate the SH coefficients as the numerical integration of the products of a environment light map and spherical basis functions over an image:

$$L_l^m = \int_S L_i(s) y_l^m(s) ds \simeq \sum_{(i,j) \in I} y(s(i,j)) L_i(s(i,j)) \Delta s(i,j),$$

where  $(i,j)$  is a pixel;  $s(i,j)$  is the corresponding cartesian coordinate (figure 2);  $\Delta s(i,j)$  is delta for pixel  $(i,j)$  which can be computed as  $(2\pi/W)^2 (\sin\theta/\theta)$ . To this end, for a given pixel (i,j), the incoming direction  $(x, y, z)$  can be computed refer to Section 2.3.4.

### 4.1.2 Ambient Occlusion SH Projection

In order to compute the incoming radiance at each vertex, we prebuilt ambient occlusion as SH coefficients per vertex via Monte Carlo integration:

$$O_l^m = \int_{\Omega} O(w)y_l^m(w)dw \simeq \frac{4\pi}{N} \sum_{w \in \Omega} O(w)y_l^m(w) \quad (4.1)$$

where  $O$  is ambient occlusion. We cast 5000 rays over a hemisphere for each vertex to obtain visibility information. Rays are casted in the stratified jittered pattern. To accelerate visibility test, the voxel grid structure is used. We write the ambient occlusion SH coefficients at every vertex into a file for one scene. The file is loaded on the memory before the rendering stage.

### 4.1.3 BRDF SH Projection

We model specular reflection as convolution of zonal harmonic specular lobes over incident light. We perform SH projection for an isotropic Phong lobe  $\rho_s(w_i, w_o) = (B(\vec{w}_i) \cdot \vec{w}_o)^s$ . We precompute the finite number of Phong lobes for all specularities in the scene.

## 4.2 SH Lighting

In SH lighting framework, we compute the rendering equation as the linear matrix computation among the SH coefficients. This method can compute the rendering equation faster than the Monte-Carlo integration. We can also change the level of lighting frequency. Due to the physical differences between diffuse reflection and specular reflection, we separate reflectance into diffuse (Equation (4.2)) and specular reflection terms (Equation (4.3)) [19].

### 4.2.1 Diffuse SH Lighting

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

$$L_d(x, \vec{w}_o) = \int_{\Omega} L(w_i)O(w_i)\rho_d dw_i = \rho_d L \cdot O, \quad (4.2)$$

where  $L$  is a rotated environment light map;  $O$  is an ambient occlusion which is the product of visibility and the cosine term;  $\rho_d$  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.

### 4.2.2 Specular SH Lighting

For specular lighting, we compute isotropic specular reflection  $L_s$  by convolving the specular reflectance function along the incoming radiance [1]:

$$L_s(\vec{N}, \vec{w}_o) = \int_{\Omega} L_i(\vec{w}_i) O(\vec{w}_i) \rho_s(\vec{w}_i, \vec{w}_o) dw_i, \quad (4.3)$$

where we employ the isotropic Phong model, assuming the specular lobe  $\rho_s$  as a single symmetric lobe with a specular parameter  $k$ :  $\rho_s(w_i, w_o) = (B(\vec{w}_i) \cdot \vec{w}_o)^k$  [20], 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 Equation (4.3) as below [1]:

$$\begin{aligned} L_s(x, \vec{w}_o) &= \int_{\Omega} R(B(\vec{w}_o) - \vec{w}_i) \rho_z(\vec{w}_i) dw_i \\ &= R * \rho_z(B(\vec{w}_o)), \end{aligned} \quad (4.4)$$

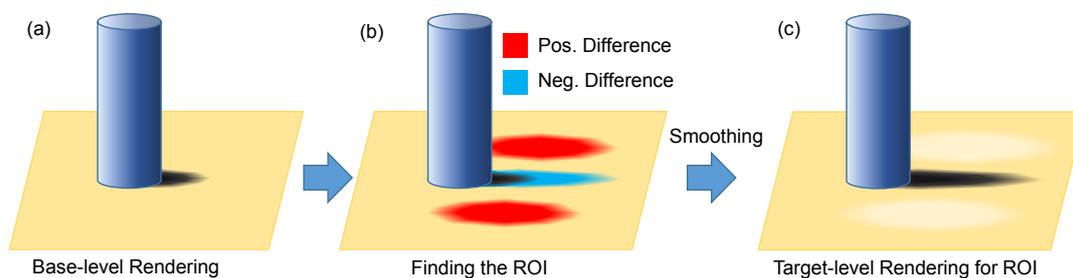
where  $R$  is an incoming radiance on  $\rho_z$ ;  $B(\vec{w}_o)$  is a reflection of  $\vec{w}_o$ ;  $\rho_z$  is a  $z$ -aligned specular lobe.

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  $\rho_z$  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 thesis, we focus on reducing the computational costs in specular lighting within the genuine spherical harmonic framework.

## Chapter 5. 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. To this end, we first compute the base-level lighting and find the regions of interest (ROIs) through multi-level filtering as Figure 5.1 shows. After refining the ROIs, we compute the high-frequency rendering only for the ROIs.

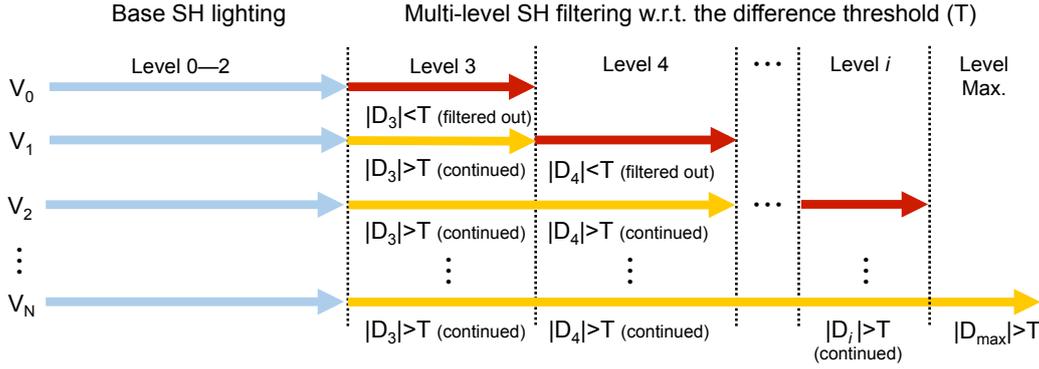


**Figure 5.1:** Schematic overview of our multi-level SH lighting framework. (a) We compute the reference SH lighting initially and (b) compute the difference between the current and the reference level in SH lighting. We define the narrow ROIs for the next level computation and (c) compute the next level.

### 5.1 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 region of interests (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  $T$ ). Figure 5.2 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. We conduct the multi-filtering for each vertex. For vertex  $x$ , we first call the  $COMPUTESHLIGHT(l, x)$ , which compute the lighting at level  $l$ , from zero to  $BASE$  in order to compute the base-level lighting at lines 3-4. Then we compute the further lighting level by level and accumulate the lighting in variable  $diff[x]$ . While accumulating the lighting difference, we check whether the accumulated difference is significant or not with the threshold  $T$  at line 9. If it is significant, we break the loop and continue to process another vertex.



**Figure 5.2:** 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.

---

**Algorithm 1** Multi-Level SH Filtering

---

```

1: procedure MULTILEVELFILTERING
2:   for vertex  $x$  in scene do
3:     for level  $i$  from 0 to  $BASE$  do
4:        $base[x] += COMPUTESHLIGHT(i, x)$ 
5:     end for
6:      $diff[x] = 0$ 
7:     for level  $l$  from  $BASE + 1$  to  $TARGET$  do
8:        $diff[x] += COMPUTESHLIGHT(l, x)$ 
9:       if  $diff[x] < T$  then
10:         $REMOVEFROMROI(x)$ 
11:      end if
12:    end for
13:  end for
14:   $REFINEROI()$ 
15: end procedure

```

---

### 5.1.1 Diffuse ROI

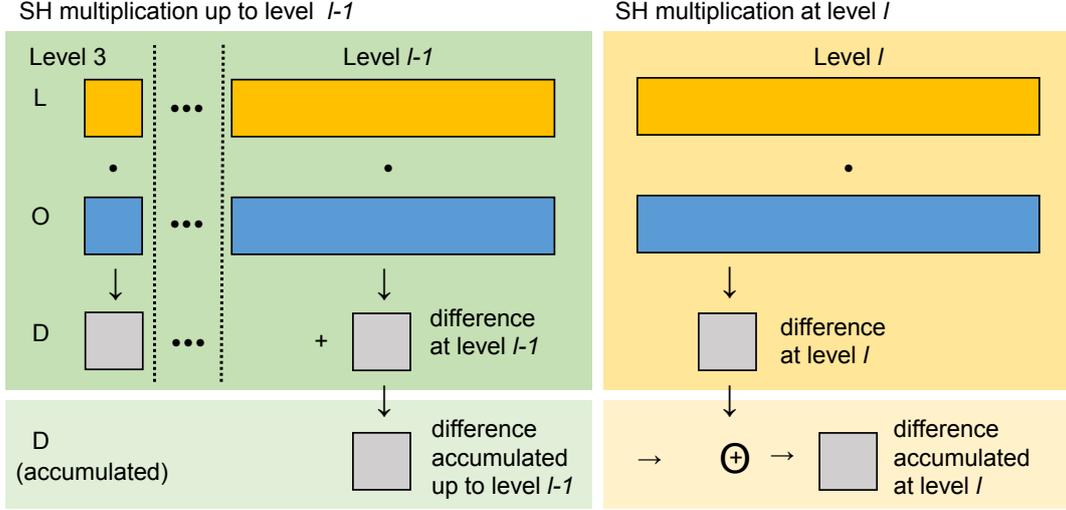
Our method calculates the specular and the diffuse lighting separately adopting the Phong reflectance model. Our method first determines the reference base-level diffuse lighting for every vertex. The reference base level for diffuse reflection is limited up to the second level (the first top three layers) as the three SH layers are fundamentally important to render diffuse reflection [21]. 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 Figure 5.3):

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

where  $L_{diffuse}^l$  is a diffuse reflection at a level  $l$  for a vertex  $x$ ;  $L_i$  is an  $i$ -th SH coefficient for an environment

light map;  $O_i$  is an  $i$ -th SH coefficient for the ambient occlusion at a vertex  $x$ .



**Figure 5.3:** Computation of SH diffuse lighting. We determine the diffuse ROI of the vertices at level  $l$ .

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 Figure 6.2. 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. We call  $COMPUTEDIFFUSE(l, x)$  instead of  $COMPUTESHLIGHT(l, x)$  in Algorithm 1 to compute the diffuse lighting at level  $l$  on vertex  $x$ .

---

**Algorithm 2** Compute Diffuse Lighting at Level  $l$

---

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

### 5.1.2 Specular ROI

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 Equation (4.4) 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 Figure 5.4(b)):

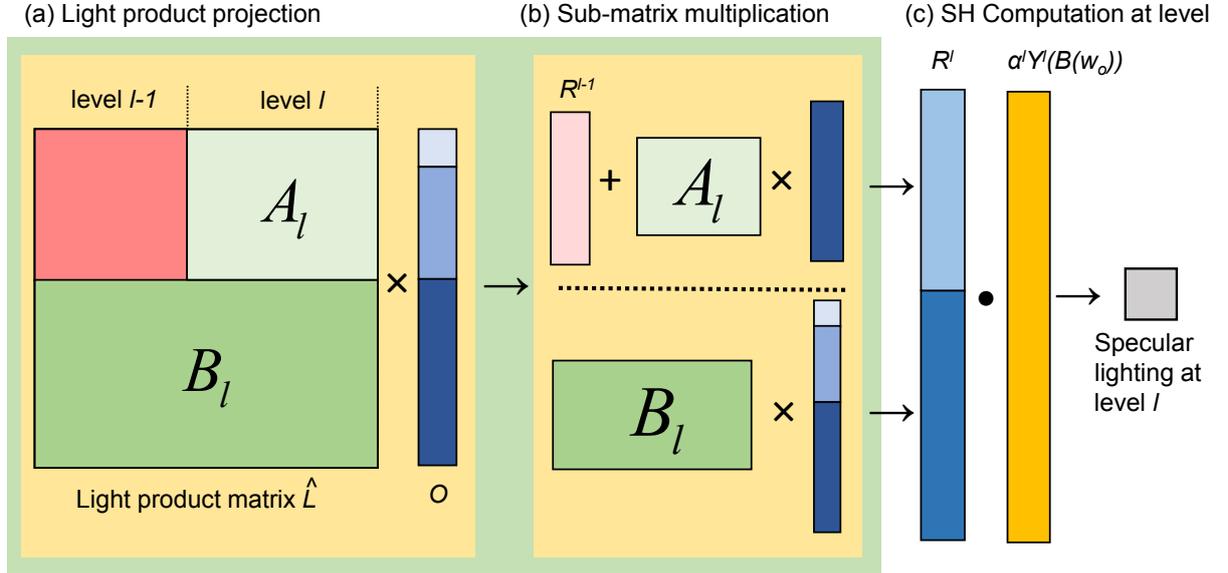
$$R_i^l = \begin{cases} \sum_{j=0}^{(l+1)^2-1} \hat{L}_{ij} O_j, & i \geq l^2 \\ R_i^{l-1} + \sum_{j=0}^{l^2-1} \hat{L}_{ij} O_j, & i < l^2 \end{cases},$$

where  $R_i^l$  is the  $i$ -th SH coefficient for the product of the ambient occlusion  $O$  and the light  $L$  at the level  $l$ ; where  $\hat{L}_{ij}$  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 Equation (2.2)). 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 Figure 5.4(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  $l$ ;  $\alpha_i$  is the  $i$ -th convolution coefficient (Equation (2.3));  $y_i(B(w_o))$  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  $T$ .



**Figure 5.4:** 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  $\alpha Y(B(w_o))$ .

Algorithm 3 presents the pseudo code of the computation of specular lighting at each level. Here,  $COMPUTESPECULAR(l, x, r)$  computes the specular lighting at level  $l$  for vertex  $x$ . We first compute the SH coefficients for the incoming radiance  $R$ . At line 4-7, we conduct the matrix multiplication of the

sub-matrix  $B_l$  and the ambient occlusion  $O$  to compute the SH coefficient of the incoming radiance at level  $l$ . The matrix  $B_l$  is the submatrix of the light product projection matrix  $\hat{Y}$ , obtained by selecting the rows from  $l^2$  to  $(l+1)^2 - 1$  and the columns from 0 to  $(l+1)^2 - 1$ . For the SH coefficients from zero level to  $l-1$  level, the ambient occlusion  $O$  is multiplied with the sub-matrix  $A_l$  at lines 10-13. We obtain the submatrix  $A_l$  of the light product projection matrix by selecting the rows from 0 to  $l^2 - 1$  and the columns from  $l^2$  to  $(l+1)^2 - 1$ . Line 8 and line 14 compute the convolution and the evaluation at the bounded direction as the dot products.

---

**Algorithm 3** Compute Specular Lighting at Level  $l$

---

```

1: procedure COMPUTESPECULAR(L,X,R)
2:    $levelSpecular = 0$ 
3:   for  $i$  from  $l^2$  to  $(l+1)^2 - 1$  do
4:      $R(i) = 0$ 
5:     for  $j$  from 0 to  $(l+1)^2 - 1$  do
6:        $R(i)+ = \hat{Y}(i, j)O(x, j)$ 
7:     end for
8:      $levelSpecular+ = R(i)\alpha^i Y^i(B(w_o))$ 
9:   end for
10:  for  $i$  from 0 to  $l^2 - 1$  do
11:    for  $j$  from  $l^2$  to  $(l+1)^2 - 1$  do
12:       $R(i)+ = \hat{Y}(i, j)O(x, j)$ 
13:    end for
14:     $levelSpecular+ = R(i)\alpha^i Y^i(B(w_o))$ 
15:  end for
16: end procedure

```

---

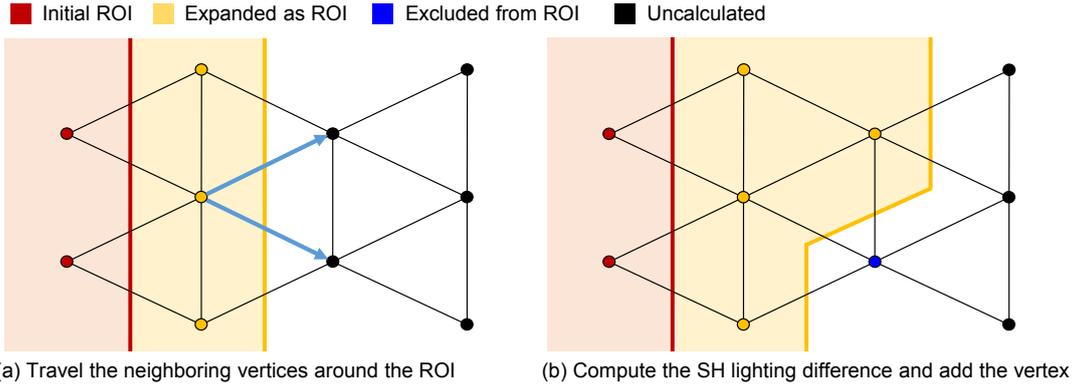
## 5.2 Refinements

### 5.2.1 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 Figure 5.5).

### 5.2.2 Smoothing the Boundaries

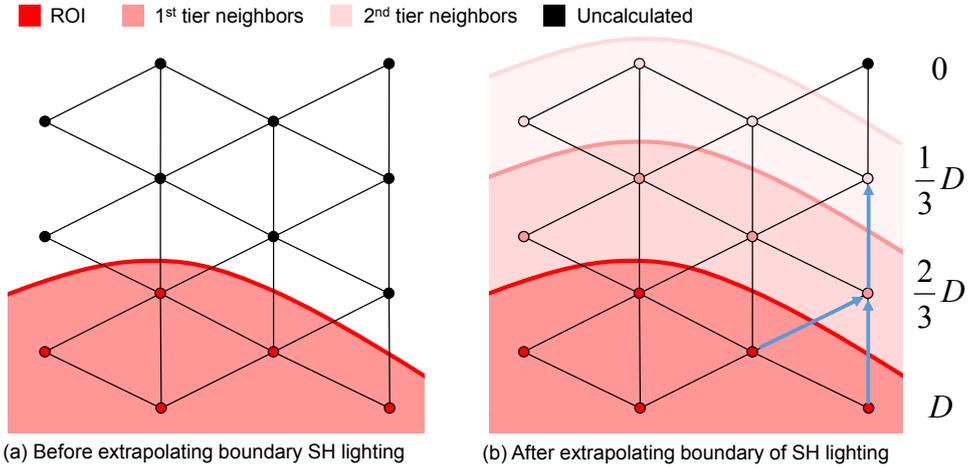
We render high-order SH lighting exclusively within the ROIs, yielding sharp-edge artifacts often around the ROIs. This edge artifacts occur as the computational trade-off between the speed and the accuracy



**Figure 5.5:** *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.*

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  $T = 0.02$ ) 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 Figure 5.6). Note that we take all the neighboring SH lightings of the connected vertices into account by averaging them.



**Figure 5.6:** *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.*

## Chapter 6. Results

We compare our work with the naive spherical harmonic lighting method to demonstrate our results. Compared to the naive SH lighting, we achieve about two times speed up with high peak signal-to-noise ratio(PSNR). The following subsections describe the implementation, experiment measurements, and results in detail.

### 6.1 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 Section 5.1). 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.

### 6.2 Experiment Measurements

We measure the frame rates to demonstrate our method’s performance. First, we pick the one view point for the scene. We turn the scene orientation with one rotation about z-axis in counter-clock wise for 10 seconds. Rotating the scene orientation, we measure frame rates ten times for each second and get the average of them. The filtering-out rates for diffuse reflection and specular reflection are also counted while measuring frame rates. Note that we measure the performance of direct illumination.

In order to demonstrate our visual quality, we measure the peak signal noise ratio. We capture the two images, one for our method and one for the full calculation, at the same view point and store it in PNG file format. Then, we calculate the mean square error of two images and compute the peak signal

noise ratio as below:

$$PSNR = 20 * \log_{10}\left(\frac{255}{\sqrt{MSE}}\right), \quad (6.1)$$

where 255 is the max signals of PNG; MSE is the mean square error of two images. For video, we measure the peak signal noise ratio of the start images.

### 6.3 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 6.1 compares the performance of our method with three objects and various environment maps. For diffuse reflection, our method filters out about 15% of visible vertices, while it filters about 80% of the visible vertices out for specular reflection. We improve the frame rate with about 2.5 speedup compared to the naive SH lighting.

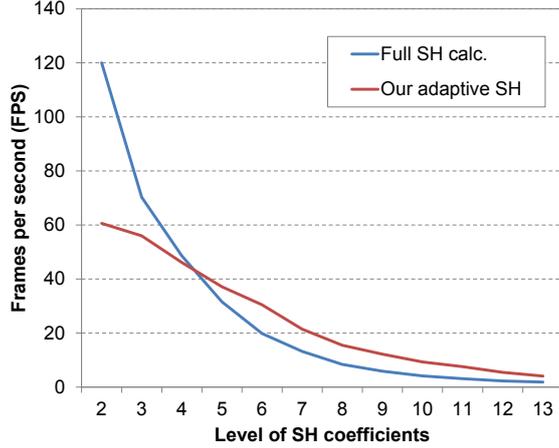
Environment map	Model	# of vertices	SH level	# of SH coeffs.	FPS (ours)	FPS (full cal.)	Filtering ratio (diffuse)	Filtering ratio (specular)	Speed gain
Peter’s basilica	Armadillo	52k	10	121	13.7	5.4	13%	80%	2.53
Euclayptus Grove	Armadillo	52k	10	121	12.1	5.5	18%	73%	2.20
Galileo tomb	Armadillo	52k	10	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 6.1:** *Quantitative measurements of our method with four objects with various environmental maps.*

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 6.1 and Table 6.2 compare the impact of the level of SH coefficients on the rendering speed.

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 6.2:** *Frame-rate comparison between the full SH coefficients computation and our adaptive methods.*

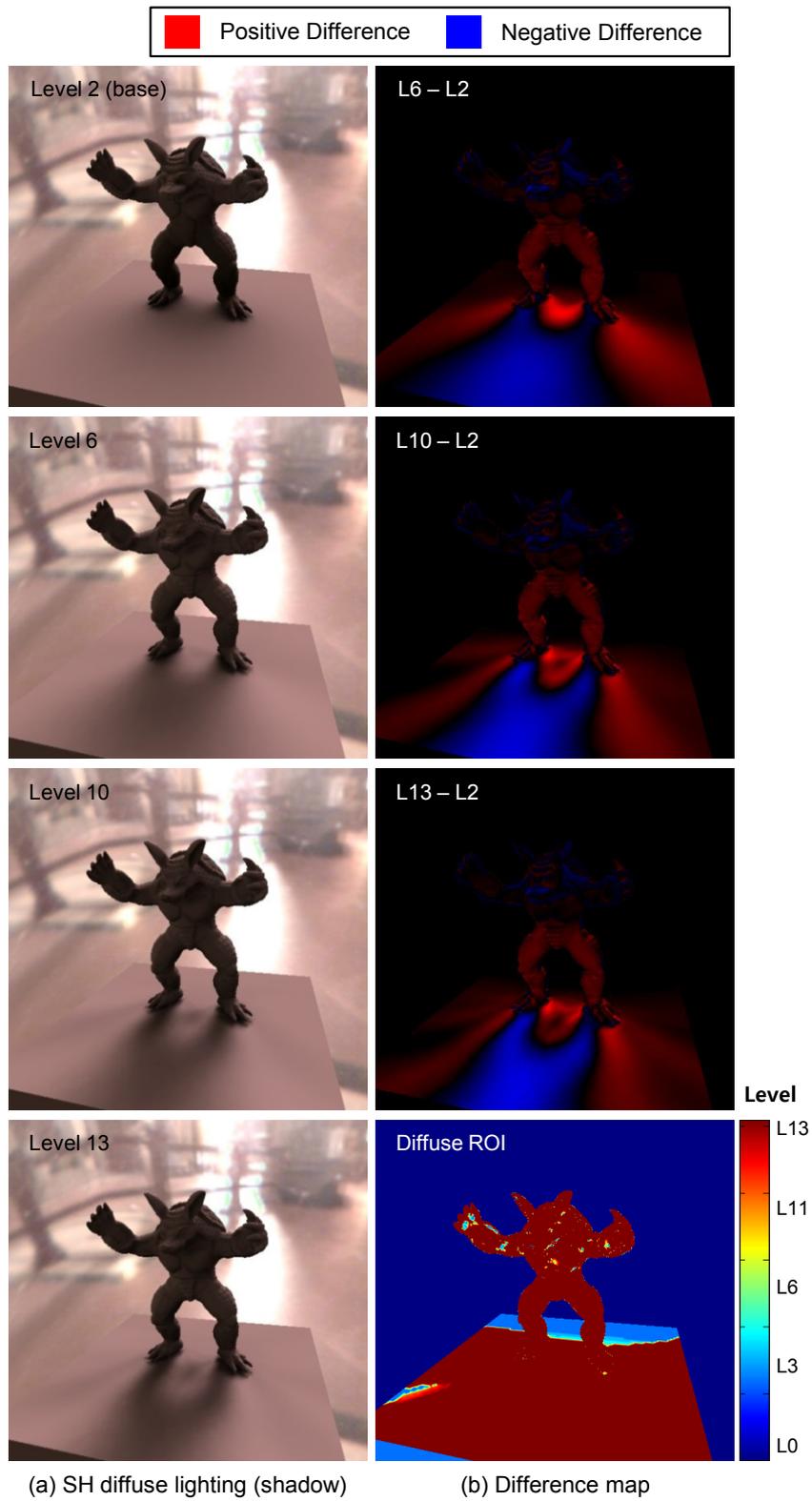


**Figure 6.1:** *The level of SH coefficients impacts on rendering speed of our method (model: Happy Buddha).*

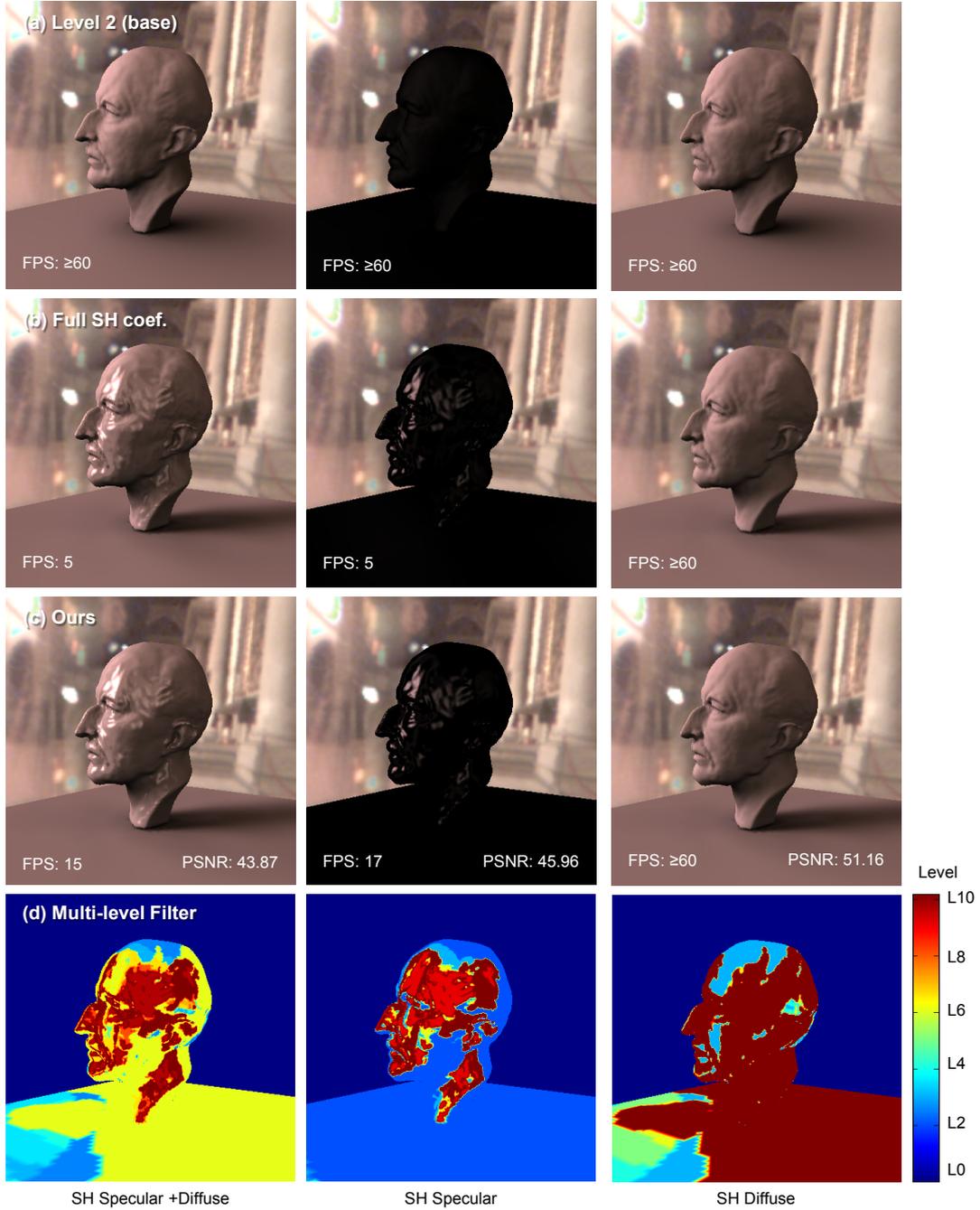
Figure 6.2 compares the SH diffuse lighting, the accumulated difference map and the initial ROI. Note that the diffuse shadows become sharper as the SH level increases. This shadow frequency change can be regarded as the accumulated SH light differences. Our initial ROI are filtered level-by-level in respect to the accumulated SH lighting differences.

Figure 6.3 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 vertices only within the ROI (the red region of Figure 6.3(d)). Our specular and diffuse lighting speeds up the rendering time by a factor of 2.6 with a high PSNR as 39.39, 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.

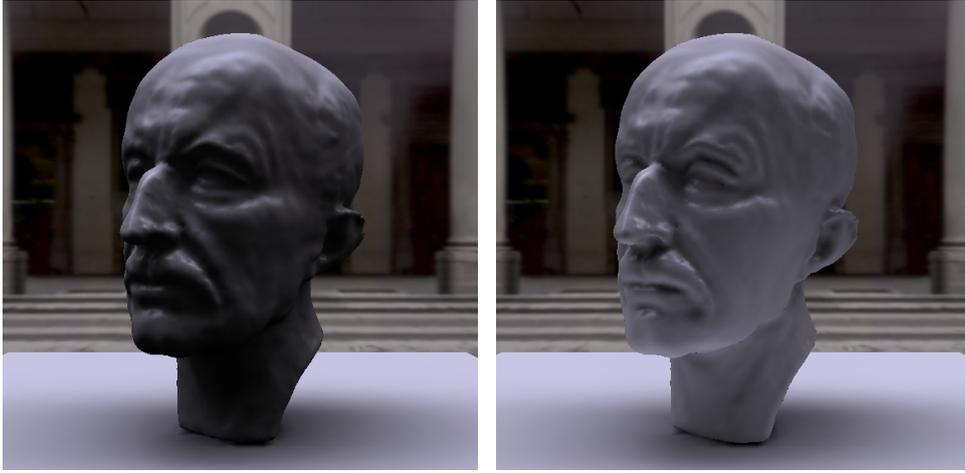
Figure 6.4 presents global illumination of our method. Our method can compute indirect lighting by decomposing the second-light transports into the SH coefficients in the precomputation step. Note that we only consider indirect diffuse lighting because the second-bounced specular reflection may drop the overall performance.



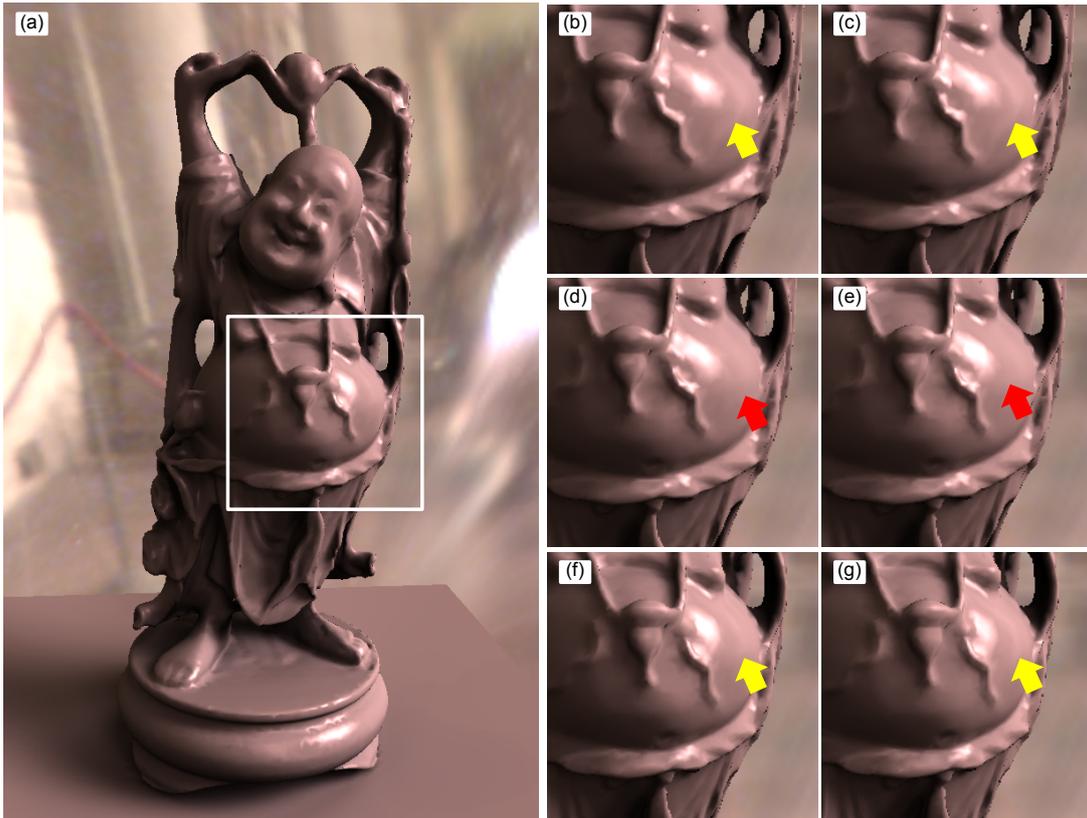
**Figure 6.2:** Level-by-level difference of our SH diffuse shadow lighting. (a) shows the SH diffuse lighting at each level and (b) shows the difference map of the levels against the reference base level.



**Figure 6.3:** 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 6.4:** *Global illumination. The left figure shows direct illumination of the “Max Planck’s head” and the right figure shows global illumination of it. Our algorithm can render indirect illumination by tracking the second-bounced rays for diffuse reflection and decomposing the light transports into the SH coefficients.*



**Figure 6.5:** *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)–(d) show smoother transition from the specular highlight to the diffuse color. However, the frame (e) bluntly shows the sudden changes of the ROI, lacking middle-level specularity while the object spins. (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.*

## Chapter 7. Discussion and Future Work

As shown in Table 6.1, 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 to 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.

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.

Our method is based on the threshold mechanism. When the threshold is too high, the high-frequency details are missed. In other words, our method sacrifices our frame rates to keep the significant visual details when the threshold is too low. The careful selection of threshold is needed to a produce plausible rendering in terms of both frame rates and visual quality.

We apply the multi-level filtering method frame by frame exclusively. It causes the flickering artifact where some portions of the scene may belong in the ROI off and on. Figure 6.5 presents the visual differences between frames in the video. Note that the flickering effect occurs even though we refine the ROIs by traveling the neighbor of ROIs. We could trace back that the ringing artifact of spherical harmonic lighting affects the traveling algorithm of our method(described in Section 5), yielding such visual artifacts between frames. We may resolve it by proposing frame-dependent filtering method or decreasing the threshold.

## Chapter 8. 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.

## 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] A. R. Edmonds, *Angular momentum in quantum mechanics*. Princeton University Press, 1996.
- [3] 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.
- [4] 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.
- [5] P. E. Debevec, “Rendering synthetic objects into real scenes,” in *SIGGraph-98*, 1998, pp. 189–198.
- [6] B.-T. Phong, “Illumination for computer generated pictures,” *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, Jun. 1975.
- [7] G. J. Ward, “Measuring and modeling anisotropic reflection,” *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 265–272, 1992.
- [8] R. L. Cook and K. E. Torrance, “A reflectance model for computer graphics,” *ACM Transactions on Graphics (TOG)*, vol. 1, no. 1, pp. 7–24, 1982.
- [9] 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.
- [10] 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.
- [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, X. Liu, H.-Y. Shum, and J. Snyder, “Bi-scale radiance transfer,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 370–375, Jul. 2003.
- [13] 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.
- [14] 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.

- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] J. Kautz and M. D. McCool, “Interactive rendering with arbitrary brdfs using separable approximations,” in *Rendering Techniques 99*. Springer, 1999, pp. 247–260.
- [20] B. T. Phong, “Illumination for computer generated pictures,” *Commun. ACM*, vol. 18, no. 6, pp. 311–317, Jun. 1975.
- [21] R. Ramamoorthi, “Precomputation-based rendering,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 4, pp. 281–369, 2009.

# Summary

## Locally Adaptive Products for Genuine Spherical Harmonic Lighting

광이동 전처리 기법은 난반사와 정반사 물체에 복잡한 조명을 처리하는 기술로 널리 쓰이고 있다. 파동함수는 조명의 모든 주파수를 표현하는데 있어 효과적이거나 구면조화함수는 회전불변 성질을 갖고 있기에 조명과 시점을 바꾸는데 있어 용이하다. 하지만 구면조화 함수의 경우 실시간 조명을 갱신하기 위해서 함수의 개수가 제한되며 이로 인해 높은 주파수를 표현하는 구면조화 함수들은 제외되게 된다. 그러므로 구면조화 조명 기법은 오직 열린 난반사 물체를 조명하는데 사용된다. 본 논문에서 우리는 순수 구면조화 조명 기법에서 고차 구면조화 조명 함수를 국부 적응적으로 계산한 간단하지만 실용적인 필터링 해법을 제시한다. 우리의 접근 방식은 크게 두 가지다. 첫째로, 우리는 고주파수 구면조화함수 조명이 필요한 관심영역을 구하기 위하여 정점들에 대해 다중레벨 필터링을 하였다. 우리는 누락된 관심영역을 미리 구한 관심영역의 이웃 정점을 순회하며 채웠다. 제안된 기법은 그래픽스 하드웨어에 의존치 않고도 난반사와 정반사 물체를 위한 구면조화 조명에서의 고차 계산을 실시간에 가능케 하였다.

## 감 사 의 글

이 논문을 완성하기까지 많은 사람들의 도움이 있었습니다. 먼저 논문지도에 가장 많은 도움을 주신 김민혁 지도 교수님, 심사 위원으로써 아낌없는 조언을 해주신 박진아 교수님, 윤성의 교수님께 감사드립니다.

컴퓨터를 하겠다는 저의 꿈을 믿어주시고 지원해주신 어머니, 아버지께 감사 드립니다. 또한 저의 활력소가 되어준 동생에게 고맙다고 전하고 싶습니다.

그리고 저와 석사생활을 함께한 랩 동기 길주형, 논문 제출까지 버팀목이 되준 인창이형, 언제나 응원해 준 승환이, 긍정적 사고의 해봄이형, 막내 석준이, 그리고 묵묵히 일하는 영범이형께 감사하다는 말을 전하고 싶습니다.

마지막으로 석사생활 동안 저와 동고동락했던 모든분들께 감사합니다.

# 이 력 서

이 름 : 이 주 호

생 년 월 일 : 1990년 9월 6일

출 생 지 : 서울시 강서구 등촌동 691-1 동성아파트 103동 803호

본 적 지 : 서울시 강서구 등촌동 691-1 동성아파트 103동 803호

주 소 : 대전 유성구 구성동 카이스트 전산학과 E3빌딩 3422호

E-mail 주 소 : jhlee@vclab.kaist.ac.kr

## 학 력

2006. 3. - 2008. 2. 한성과학고등학교 (2년 수료)

2008. 3. - 2012. 6. 한국과학기술원 전산학과 (B.S.)

## 경 력

2012. 9. - 2013. 6. 한국과학기술원 전산학과 일반조교

## 학 회 활 동

1. Joo H. Lee, Min H. Kim, *Locally Adaptive Products for Genuine Spherical Harmonic Lighting*, 22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, June., 2014.

## 연 구 업 적

1. Joo H. Lee, Min H. Kim, *Locally Adaptive Products for Genuine Spherical Harmonic Lighting*, proceedings of WSCG, June., 2014.