

N° d'ordre: 3333

THESE

présentée

DEVANT L'UNIVERSITE DE RENNES 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITE DE RENNES 1

Mention : Informatique

PAR

Jaroslav KŘIVÁNEK

Equipe d'accueil : SIAMES-IRISA

Ecole doctorale : Informatique, Traitement du signal et Télécommunications

Composante universitaire : IFSIC

**Cache de luminance pour l'éclairage global en présence
de surfaces rugueuses**

Soutenue le 15 décembre 2005 devant la commission d'Examen

M.	Bruno Arnaldi	Président
MM.	Kadi Bouatouch	Directeurs
	Jiří Žára	
MM.	François Sillion	Rapporteurs
	Karol Myszkowski	
	Mathias Paulin	
M.	Sumanta Pattanaik	Examineurs
Mme.	Ivana Kolingerová	
M.	Jiří Sochor	

Czech Technical University in Prague
Faculty of Electrical Engineering



Radiance Caching for Global Illumination Computation on Glossy Surfaces

by

Jaroslav KŘIVÁNEK

A dissertation submitted to
the Faculty of Electrical Engineering, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

PhD program: Computer Science and Engineering

December 2005

Thesis supervisors:

Kadi Bouatouch
IRISA / INRIA Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex
France

and

Jiří Žára
Department of Computer Science and Engineering
Faculty of Electrical Engineering
Czech Technical University
Karlovo náměstí 13
121 35 Praha 2
Czech Republic

Abstract

Computation of global illumination is an important means in the pursuit of photorealism of computer-generated images. It is of particular importance in scenes with glossy surfaces, since it conditions correct perception of glossy material characteristics. This dissertation focuses on efficient computation of global illumination in presence of glossy surfaces with low-frequency reflectance functions. Computing global illumination on those surfaces is difficult with pure Monte Carlo sampling because of the sheer number of rays required to get a good estimate of indirect illumination. We propose radiance caching, a ray tracing-based algorithm for computing indirect illumination on glossy surfaces with low-frequency reflectance functions. The algorithm takes advantage of the shading smoothness on these surfaces by interpolating indirect illumination from a set of sparsely distributed, cached values. We discuss all the components necessary for an implementation of radiance caching: (1) an automatic choice of surfaces on which radiance caching is used; (2) compact representation for the cached illumination by spherical harmonics; (3) efficient spherical harmonics rotation that makes interpolation possible; (4) computation of translational gradient for smooth interpolation; and (5) an adaptive criterion for distributing the cached illumination values on surfaces. Radiance caching delivers high quality images in a shorter time compared to existing ray tracing-based methods. It can be used for final gathering in two-pass global illumination methods or as a stand-alone global illumination algorithm.

Keywords: Computer graphics, image synthesis, rendering, global illumination, Monte Carlo, spherical harmonics, radiance caching.

Résumé

L'illumination globale est un moyen permettant de produire des images de synthèse dites photoréalistes. Elle joue un rôle encore plus important dans le cas de scènes contenant des objets en parties spéculaires, c'est à dire non parfaitement lisses. Cette thèse traite principalement du problème du calcul de l'illumination globale dans le cas de ce type de scène où les objets sont caractérisés par des réflectances de basse fréquence. Le calcul de l'illumination globale s'avère très difficile lorsqu'on utilise un échantillonnage de type Monte Carlo car un grand nombre d'échantillons (rayons) est nécessaire pour obtenir une bonne estimation de l'éclairage indirect. C'est pourquoi nous proposons une méthode utilisant un cache de luminance, méthode basée sur le lancer de rayon et prenant en compte les surfaces spéculaires non parfaitement lisses (ayant une rugosité microscopique) dont les réflectances possèdent des caractéristiques de basse fréquence. L'algorithme proposé exploite la variation douce de l'éclairage sur une surface en interpolant l'éclairage indirect à partir de données éparses stockées dans le cache. Tous les outils et composants logiciels nécessaires à la mise en oeuvre de cet algorithme sont présentés dans cette thèse. En effet, il s'agit : du choix des surfaces sur lesquels la sauvegarde dans le cache doit être effectuée, de la représentation compacte des informations sauvegardées à l'aide d'harmoniques sphériques, de la rotation des harmoniques sphériques nécessaires lors de l'interpolation, du calcul de gradient translationnel aussi utilisé lors de l'interpolation, du critère adaptatif permettant de choisir les valeurs à stocker dans le cache. Notre méthode basée sur le cache de luminance permet de générer des images de grande qualité en un temps plus court que celui obtenu avec les méthodes existantes basées sur le lancer de rayon. Elle peut être aussi utilisée pour la collecte finale dans les méthodes à deux passes ou comme un algorithme d'illumination globale complet.

Mots clés: Informatique graphique, images de synthèse, rendu, éclairage global, Monte Carlo, harmoniques sphériques, cache de luminance.

Acknowledgements

I can hardly express the gratitude to my advisor in France, Kadi Bouatouch, for he has been a constant source of inspiration, moral support and encouragement throughout my Ph.D. studies. He took care of funding my research at IRISA and arranged my stay at the University of Central Florida, which turned out to be very fruitful for my work.

I would like to thank to my advisor in the Czech Republic, Jiří Žára for getting me interested in computer graphics, proving me the opportunity to start my Ph.D. at the computer science department in Prague, and especially for his assistance at the beginning of my studies.

Meeting, and having the opportunity to work with, Sumanta Pattanaik was a great honor to me. I was immediately infected with his enthusiasm and eagerness for solving various problems in graphics. I want to thank him for providing me with many ideas and for the numerous insightful discussions we have had. I cannot thank enough his wife, Laxmi, for introducing me to her exquisite Indian cuisine.

Pascal Gautron was an excellent colleague. I much enjoyed working with him for his knowledge, willingness to help, and his serenity.

I would like to thank the reviewers of my dissertation thesis, François Sillion, Karol Myszkowski, and Mathias Paulin for reviewing the manuscript and providing me with numerous ideas for its improvement. My special thanks are addressed to Vlastimil Havran for letting me use his long-developed rendering software GOLEM and for the numerous discussions we have had. Many thanks to all the other people who helped with my work: Jaakko Kontinnen, Mark Colbert, Zoran Nikoloski, Jiří Bittner, Marina Casiraghi, Eric Risser, and Gérard Choqueux.

I would like to thank to my parents for their constant support and encouragement. I spent most of my Ph.D. studies abroad. I want to thank all the people I met there for being so open and for introducing me to the culture of their countries.

During my stay at the University of Central Florida, my research was partially supported by ATI Research, Orlando.

Contents

Abstract	v
Résumé	vi
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	1
1.2 Our Approach to Indirect Illumination Computation	3
1.3 Contributions	4
1.3.1 Radiance Caching	4
1.3.2 Further Contributions	4
1.4 Thesis Overview	5
2 Background	7
2.1 Global Illumination	7
2.2 Indirect Illumination on Glossy Surfaces	9
2.2.1 Radiosity on Glossy Surfaces	9
2.2.2 Monte Carlo Ray Tracing on Glossy Surfaces	9
2.2.3 Our Approach to Indirect Illumination on Glossy Surfaces	11
2.3 Exploiting Illumination Smoothness	11
2.3.1 Previous Work	11
2.3.2 Irradiance Caching and Derived Algorithms	12
2.3.3 Our Choice of the Interpolation Scheme	15
2.4 Incoming Radiance Representation	15
2.4.1 Requirements	15
2.4.2 Previous Work	16
2.4.3 Our Choice of the Incoming Radiance Representation	16
2.4.4 Spherical and Hemispherical Harmonics	17
2.5 BRDF Representation	18
2.6 Summary	19

3	Radiance Caching	21
3.1	Radiance Caching in a Nutshell	21
3.2	Incoming Radiance Computation and Storage	22
3.3	Caching Scheme and Interpolation	23
3.4	Order of the BRDF and Radiance Representation	25
3.5	Rendering with Radiance Caching	28
3.6	Discussion	28
3.7	Results	31
3.8	Summary	35
4	Radiance Gradient Computation	39
4.1	Introduction	39
4.2	Solid Angle-Based Gradient	40
4.2.1	Gradient by Finite Differences	41
4.2.2	Gradient by Differentiation	41
4.2.3	Irradiance Gradient	44
4.2.4	Discussion	44
4.3	Stratification-Based Gradient	46
4.3.1	Radiance Gradient	46
4.3.2	Irradiance Gradient	49
4.4	Results	50
4.5	Summary	52
5	Spherical Harmonics Rotation	53
5.1	Introduction	53
5.2	Background	53
5.3	Fast Rotation Approximation	55
5.3.1	Rotation Approximation by Truncated Taylor Series	55
5.3.2	Error Analysis	57
5.3.3	Complexity	57
5.4	Application in Radiance Caching	58
5.4.1	Rotation Approximation Limiting Angle	60
5.4.2	Derivation of the Limiting Angle	60
5.4.3	Results	61
5.5	Application in Real-Time Shading	61
5.5.1	Preliminaries	61
5.5.2	Fast, Arbitrary BRDF Shading for Low-Frequency Lighting	63
5.5.3	Shading on Normal-Mapped Surfaces through a Per-Pixel Rotation	64
5.5.4	Results	64
5.6	Summary	65

6 Adaptive Radiance Caching	67
6.1 Introduction	67
6.2 Common Interpolation Formula	68
6.3 Adaptive Caching	69
6.3.1 Error Criterion	69
6.3.2 Full Convergence	70
6.4 Discernability Metric	73
6.5 Practical Issues	74
6.5.1 Adapting Record Density by Gradient Magnitude	74
6.5.2 Neighbor Clamping	75
6.5.3 Ray Leaking	76
6.6 Results and Discussion	78
6.7 Summary	81
7 Conclusion	89
7.1 Summary of Contributions	89
7.1.1 Radiance Caching	89
7.1.2 Gradient Computation	90
7.1.3 Spherical Harmonics Rotation	90
7.1.4 Adaptive Radiance Caching	91
7.2 Future Research	91
7.2.1 Direct Extensions of Radiance Caching	91
7.2.2 Wavelet Representation of Incoming Radiance	92
7.2.3 Frequency Analysis	93
Appendices	94
A Derivatives of Spherical and Hemispherical Harmonics	95
B Gradient Computation Code	97
C Spherical Harmonics Rotation Matrix Derivative	99
D Spherical Harmonics Rotation around z-Axis	101
Bibliography	102
Résumé en français	115

List of Figures

1.1	Comparison of a directly and indirectly illuminated glossy object.	2
2.1	Direct visualization of a photon map and an image obtained by final gathering.	8
2.2	Importance sampling for high- and low-frequency BRDFs.	11
2.3	Illustration of the irradiance caching scheme.	13
2.4	Pseudocode for outgoing radiance computation with irradiance caching.	14
3.1	Pseudocode for outgoing radiance computation with radiance caching.	23
3.2	Coordinate frame alignment in interpolation on curved surfaces.	24
3.3	Positions of radiance cache records in an example scene.	26
3.4	Adaptive BRDF representation.	27
3.5	Outline of rendering with radiance caching.	29
3.6	Information loss when radiance is interpolated on a curved surface.	30
3.7	Timing breakdown for the test scenes.	31
3.8	Two views of a Cornell box with a glossy back wall rendered with radiance caching and Monte Carlo importance sampling.	32
3.9	Frames from the Glossy Flamingo animation rendered with radiance caching and Monte Carlo importance sampling.	33
3.10	Time spent on radiance caching for the Glossy Flamingo animation.	34
3.11	Picture of Frank Gehry’s Walt Disney Hall in Los Angeles.	35
3.12	Rendering of a simple model of Walt Disney Hall in Los Angeles.	36
3.13	Another view of the Walt Disney Hall scene.	37
3.14	Comparison of a radiance caching rendering with a reference image.	38
4.1	Comparison of radiance interpolation without and with radiance gradients.	40
4.2	Gradient computation by displacing the hemisphere center.	42
4.3	Quantities in the computation of $\frac{\partial \cos \xi_k}{\partial q_x r_k^2}$	43
4.4	Comparison of irradiance gradient computation by our solid angle-based method and the method of Ward and Heckbert.	45
4.5	Uniform hemisphere subdivision for stratified radiance sampling.	47
4.6	Results of solid angle-based and stratification-based gradients.	49
4.7	Results of solid angle-based and stratification-based gradients.	50

4.8	Results of solid angle-based and stratification-based gradients.	51
5.1	Form of the spherical harmonics rotation matrix.	54
5.2	First derivative of the y rotation matrix at $\beta = 0$	56
5.3	Second derivative of the y rotation matrix at $\beta = 0$	57
5.4	C code for the y rotation using the “1.5th order” Taylor expansion.	58
5.5	Rotation approximation error $\varepsilon(\beta)$ as a function of angle β	59
5.6	Radiance caching renderings obtained with the approximate rotation.	62
5.7	Rendering times for the flamingo and sphere images in Figure 5.6.	63
5.8	Detail of a normal mapped vase rendered with our spherical harmonics rotation and with the simplified normal mapping.	65
5.9	More results of normal mapping with our spherical harmonics rotation compared to the simplified normal mapping.	66
6.1	Adaptive caching detects perceptible discontinuities and eliminates them.	69
6.2	Refinement when a discontinuity is detected.	70
6.3	Algorithm for attaining full convergence for a set of shading points.	72
6.4	The maximum possible difference of the two records’ distance from the step is given by their mutual distance.	76
6.5	Neighbor clamping helps to detect small, step-like geometry features that might otherwise be missed.	77
6.6	Inaccurate connection of polygons may result in ray leaking.	78
6.7	Displacement of the origin of hemisphere sampling from the polygon edge.	78
6.8	Image artifacts caused by ray leaking are reduced by using neighbor clamping.	79
6.9	Rendering statistics for adaptive radiance caching example scenes.	79
6.10	Scenes used to test adaptive irradiance caching.	81
6.11	Comparison of non-adaptive and adaptive caching for the Cornell box scene.	83
6.12	Comparison of non-adaptive and adaptive caching for the Teapot scene.	84
6.13	Comparison of non-adaptive (bottom half) and adaptive (top half) caching for the Walt Disney Hall scene.	85
6.14	Color coded approximation error and record positions for the Walt Disney Hall scene.	86
6.15	Automatic adaptation of record density to the sharpness of the BRDF.	87

Chapter 1

Introduction

The main focus of this thesis is developing an efficient algorithm for computing indirect illumination on glossy surfaces with low-frequency reflectance functions. In this chapter we present the motivation for our effort, give a short overview of our approach, and summarize our contributions.

1.1 Motivation

The goal of computer graphics is to create, or render, images according to the purpose of the user. *Photorealistic rendering* is used when images indistinguishable from real photographs are desirable. Applications of photorealism span fields such as: architecture, cinema, training, education, art and entertainment, to name a few.

One of the essential aspects of creating photorealistic images is that objects look as though made of real materials. Additionally, appearance of objects should match the scene lighting and be in accord with the scene configuration (*e.g.* reflections on specular objects). This match requires accurate, physically-based simulation of light transport in the scene, carried out by, so-called, global illumination algorithms. Algorithms based on Monte Carlo sampling [48, 22] are general enough to allow light transport simulation in arbitrary scenes with all possible material types. However, those algorithms are too slow. This is why simplifications of the simulated phenomena are often made to speed up rendering. The crudest and most efficient simplification is to reduce the light transport to *direct illumination*, taking into account only the light coming *directly* from the light sources. However, *indirect illumination*, *i.e.* the light arriving at a point after having interacted with other scene surfaces, significantly improves the realism of images, and hence is desirable.

No matter how accurate the light transport simulation is “the output is only as good as the inputs” [77]. For this reason, direct measurement of various scene properties, such as illumination, object shape, or surface reflectance properties (BRDF¹ measurement) is becoming prevalent. The BRDF measurements [73] reveal that the reflectance characteristics of many materials deviate from the commonly used idealizations: the ideal diffuse

¹bidirectional reflectance distribution function

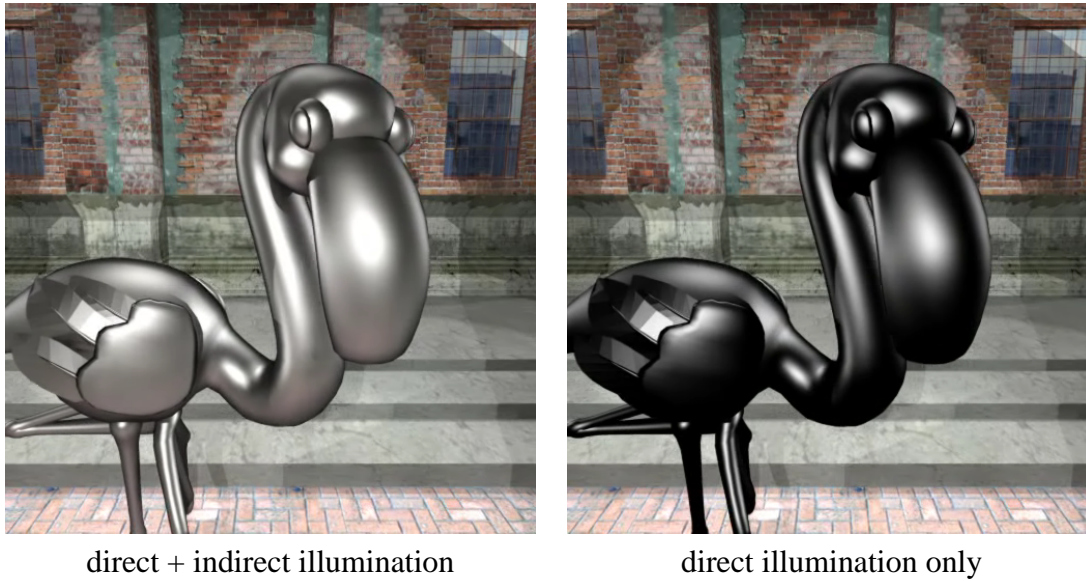


Figure 1.1: With indirect illumination (left), the flamingo looks as made of metal, whereas with direct illumination only (right) it looks as though made of a dark plastic.

and ideal specular reflection. Materials, such as metallic paints or various kinds of wood, often show *directional diffuse* reflectance, leading to the *glossy* material look. Therefore, it is important to handle the light transport on glossy surfaces in global illumination algorithms.

It is not known in general which lighting phenomena are essential for realism in rendered images, and which are less important [100]. Although the answer depends on the function, or purpose, of the image [24, 23], we argue that in realistic rendering *simulation of indirect illumination on glossy surfaces is indispensable* for the right image interpretation.

We observe that there is a significant difference between leaving out the indirect term on a diffuse surface and on a glossy one. Omitting the indirect illumination on a diffuse surface gives the feeling that “something is not right” but we still know that the object is diffuse. On a glossy surface, on the other hand, we might easily get the impression that the rendering is correct but that the object is made of something else than it is supposed to be. Figure 1.1 shows an example. The directly and indirectly illuminated flamingo on the left looks as though made of metal, whereas with direct illumination only, the flamingo on the right is plastic-looking.

This observation is in agreement with the study of Fleming *et al.* [25] which shows that recognizing various materials is strongly dependent on the characteristics of the incident illumination. People find it easier to judge material properties under rich, natural illumination as compared to point light illumination. Realizing that “natural illumination” means direct and indirect illumination, we come to a conclusion that indirect illumination on glossy surfaces is a must for correct perception of their material. Only by computing the indirect illumination can a rendering system make sure that an artist assigning a mea-

sured aluminum BRDF to an object will see the object made of aluminum in the resulting image.

Yet another reason for computing indirect illumination on glossy surfaces stems from the difficulty of replacing it by fill lights. (Fill lights are commonly used in digital lighting to add soft, natural illumination to the scene [8].) This difficulty is due to the fact that the appearance of glossy surfaces tells much more about the incident light field than the appearance of diffuse ones [77]. It is therefore difficult to fool the eye by using the fake fill lights to emulate the indirect illumination on a glossy surface.

For the reasons given above, we believe it is crucial to compute indirect illumination on glossy surfaces for obtaining realistic images. On surfaces showing sharp reflections, indirect illumination can be computed efficiently by means of Monte Carlo importance sampling. However, for glossy surfaces with low-frequency reflectance functions, showing blurry reflections, the performance of importance sampling significantly degrades. An efficient algorithm for computing the indirect illumination on glossy surfaces with low-frequency reflectance functions is the key contribution of this thesis.

1.2 Our Approach to Indirect Illumination Computation

We, first, observe that glossy surfaces with low-frequency reflectance functions show blurry reflections. In other words, the shading on those surfaces due to indirect illumination is smooth, *i.e.* it does not change rapidly.

In our algorithm for indirect illumination computation, we take advantage of the illumination smoothness by computing the indirect illumination only at a sparse set of surface points and interpolating elsewhere. The interpolation scheme is borrowed from the irradiance caching algorithm of Ward *et al.* [121], which was originally designed for computing the diffuse component of indirect illumination. We generalize irradiance caching by caching and interpolating the directional distribution of incident radiance, thereby allowing for computation of the view-dependent indirect illumination on glossy surfaces.

The incoming radiance is represented by a vector of coefficients with respect to spherical or hemispherical harmonics [27]. With this representation, the radiance interpolation is simply carried out by interpolating the coefficient vectors. Scene BRDFs are represented in the same basis as the incoming radiance. Due to the orthogonality of the basis functions, evaluating the integral of the product of incident radiance and the BRDF reduces to the dot product of the interpolated radiance coefficients and the BRDF representation coefficients, and hence is very efficient.

We improve the interpolation smoothness by the use of translational gradients in the spirit of [120]. To assure a truly artifact-free interpolation result, we use a perceptual error criterion to guide the density of locations at which the indirect radiance is computed and cached.

1.3 Contributions

1.3.1 Radiance Caching

The main contribution of this work is *radiance caching*, an efficient ray tracing-based algorithm for computing high quality indirect illumination on glossy surfaces [55]. Radiance caching can be used for final reconstruction (final gathering) in two-pass global illumination algorithms (*e.g.* in combination with photon mapping or radiosity). It can also serve as a stand-alone global illumination algorithm. Radiance caching can be used with arbitrary low-frequency BRDFs, including measured ones.

Compared to previous Monte Carlo ray tracing-based methods, radiance caching is faster and produces images of better quality. The efficiency is achieved by reusing the computed illumination through interpolation. The interpolation also suppresses the image noise inherent to Monte Carlo algorithms, yielding a better image quality.

Compared to radiosity methods, radiance caching has the advantages of a purely ray tracing-based algorithm: no restriction on the scene geometry, no problems with meshing, lower memory demands, robustness in presence of geometry modeling imperfections, and easy implementation.

Radiance caching is derived from irradiance caching of Ward *et al.* [121] and, therefore, it shares its advantages: Computation is concentrated only in visible parts of the scene, so time is not wasted on computing illumination for invisible scene areas. The result of the computation is view-independent, though, so it can be reused for different viewpoints in walk-through animation rendering. Finally, implementation and integration with an existing rendering system is easy.

1.3.2 Further Contributions

Solutions to a number of sub-problems arising in radiance caching are given in the thesis. The proposed methods are of general interest, though.

New methods for translational radiance gradient computation. The new methods are more general compared to previous ones. They allow radiance gradient computation for an arbitrary distribution of sample rays on the hemisphere and an arbitrary weighting function applied to the radiance samples.

Fast approximation to the spherical and hemispherical harmonics rotation. Our rotation procedure is several times faster than previous ones. Although approximate, it does not introduce visible errors in images. We utilize the fast rotation in radiance caching to align the local coordinate frame at different surface points. Furthermore, we demonstrate our rotation procedure in real-time lighting of bump-mapped objects illuminated by an environment map.

Adaptive refinement for radiance and irradiance caching. We propose an adaptive algorithm for guiding the spatial density of cached values in radiance and irradiance caching. The density is adapted to the rate of change of indirect illumination in

order to avoid visible interpolation artifacts and produce smooth interpolated illumination.

Solution of practical implementation problems. A number of heuristics are proposed to make an implementation of radiance and irradiance caching resistant to imperfections in scene geometry modeling and to caching behavior excesses.

1.4 Thesis Overview

The thesis is divided into seven chapters. This chapter presented the motivation for the main objective of the thesis—an algorithm for computing indirect illumination on glossy surfaces—and summarized our contributions. Chapter 2 gives background on global illumination and lists the related previous work. Chapter 3 describes the radiance caching algorithm, presents the results of its implementation, and compares them to the results of Monte Carlo importance sampling. In the subsequent chapters, solutions to particular sub-problems of radiance caching are discussed. Chapter 4 describes the translational gradient computation; Chapter 5 deals with the spherical and hemispherical harmonics rotation; Chapter 6 presents the adaptive refinement algorithm for guiding the density of cached values, and brings up some implementation issues in radiance and irradiance caching. Chapter 7 concludes the thesis with some ideas for future work.

Chapter 2

Background

In this section we give background information on global illumination, in particular on computing indirect illumination on glossy surfaces. We also give references to the related work and motivate various choices we have made in designing radiance caching.

2.1 Global Illumination

Global illumination is an extensively researched area of computer graphics. We assume a general familiarity of the reader with the topic. Dutré *et al.* [22] give a one-semester university course on global illumination. Pharr and Humphreys [75] describe a complete implementation of a physically-based lighting simulation system with the underlying theory. The books of Cohen and Wallace [16], and Sillion and Puech [91] are excellent references on radiosity. Glassner’s book [29] gives a more physics-oriented look at global illumination. The PhD theses of Lafortune [57] and Veach [106] treat the application of Monte Carlo methods in global illumination.

Global illumination algorithms are all designed to solve the rendering equation [48, 22] describing the global energy balance in a scene. The most common approaches to solving the rendering equation are: the finite elements methods, known as radiosity in computer graphics [16, 91]; and Monte Carlo sampling, referred to as stochastic ray tracing or Monte Carlo ray tracing [57, 22].

Final Gathering. Two-pass algorithms for solving the rendering equation are often used in realistic image synthesis due to their efficiency and quality of the resulting images. In the first pass, only a rough solution of the rendering equation is quickly computed for all surfaces in the scene. Radiosity [16, 91] or photon mapping [47] can be used for this purpose.

In the second pass, called *final gathering*, the image is rendered by accurately evaluating the outgoing radiance L^o at visible surface points \mathbf{p} in the direction ω_o of the camera:

$$L^o(\mathbf{p}, \omega_o) = \int_{\Omega_p} L^i(\mathbf{p}, \omega_i) f_r(\mathbf{p}, \omega_i, \omega_o) \cos \theta_i d\omega_i, \quad (2.1)$$

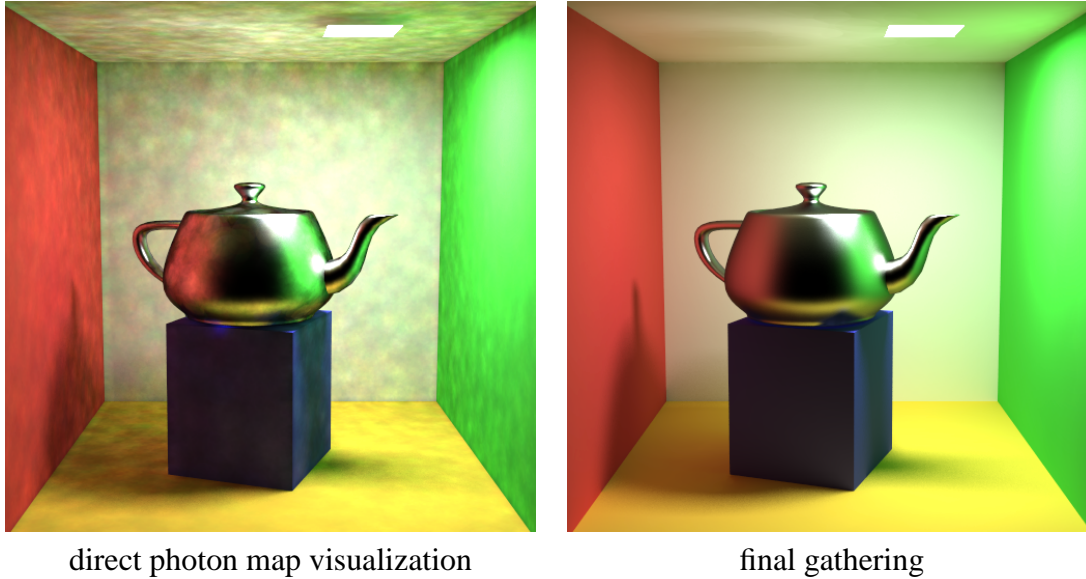


Figure 2.1: Rendering using the rough solution of the rendering equation stored in a photon map (left) leaves artifacts in the image. Final gathering (right) averages out the fluctuations of the rough solution and produces a high quality image.

Equation (2.1)—called the *illumination integral* in the rest of the thesis—describes light reflection on a surface point. Namely, it expresses the outgoing radiance leaving a point, $L^o(\mathbf{p}, \omega_o)$, in terms the incoming radiance L^i reflected off the surface at the point \mathbf{p} . The reflection characteristics of the surface are described by the bi-directional reflectance distribution function, or the BRDF, f_r . The integration domain $\Omega_{\mathbf{p}}$ is the hemisphere above \mathbf{p} and θ_i is the angle between the surface normal at \mathbf{p} and the incoming light direction ω_i . Radiance L is the quantity of interest in rendering, since it describes the amount of light reaching the eyes.

In a practical implementation, the illumination integral is split into two parts describing direct and indirect illumination. *Direct illumination* accounts for the light arriving at \mathbf{p} directly from the light source. This part is most often solved by numerical quadrature over the light source area [87, 1]. We do not deal with direct illumination computation in this thesis.

Indirect illumination is the light scattered by the scene surfaces at least once on its way from the light source to the point \mathbf{p} . In final gathering, the rough solution of the rendering equation computed in the first pass is used for evaluating the indirect term of the illumination integral: The value of the incoming radiance $L^i(\mathbf{p}, \omega_i)$ in the integrand is looked-up from the rough solution at the point \mathbf{q} , directly visible from \mathbf{p} in the direction of $-\omega_i$. Integrating the rough solution over the hemisphere effectively “averages out” the fluctuations of the rough solution and results in a high quality image (Figure 2.1). This technique is called final gathering, because the rough radiance estimate is “gathered” at the visible points to create the final image.

Rendering equation [48] can be solved even without the first pass by applying final gathering recursively: instead of reading the value of the indirect incoming radiance L^i

from the rough solution, it is computed by a recursive evaluation of the illumination integral. Without the first pass and without the recursive application, final gathering gives a first-bounce global illumination solution. Although this technique produces physically incorrect results, it has been reported to be sufficient for rendering plausible realistic images [103].

Summary. We have demonstrated that an efficient solution to the indirect part of the illumination integral (2.1) is an important constituent of a global illumination algorithm. It can be used for final gathering in a two-pass algorithm or, through recursive evaluation, as a stand-alone method for full global illumination computation. Radiance caching is used for the very purpose of evaluating the illumination integral on glossy surfaces.

2.2 Indirect Illumination on Glossy Surfaces

A number of solutions to computing indirect illumination on glossy surfaces have been proposed. We divide them into two groups: radiosity and Monte Carlo ray tracing.

2.2.1 Radiosity on Glossy Surfaces

The radiosity algorithm was originally designed to simulate the light transport in scenes with only diffusely reflecting surfaces [30, 38]. A number of algorithms have since been developed to generalize radiosity for glossy surfaces having view-dependent reflectance. Most methods handle the view-dependent light transport by storing a representation of directional radiance distribution at surface elements [42, 80, 89, 90, 74, 83, 64, 12, 99], similarly to radiance caching. Three-point method [5] and methods combining radiosity with particle tracing [31, 32] avoid storing the directional distributions while still handling light transport on glossy surfaces. Radiosity methods do not explicitly evaluate the illumination integral on the visible surfaces; they need a separate final gathering pass to produce high quality images [81, 82]. All the radiosity methods suffer from the following disadvantages: restrictions on the geometry representation (polygonal models only), high memory consumption due to fine polygon subdivision, laborious implementation, and sensitivity to inaccuracies in geometry modeling. For those reasons, ray tracing-based global illumination approaches are recently predominant in research and practice.

2.2.2 Monte Carlo Ray Tracing on Glossy Surfaces

Surprisingly little work in ray tracing has addressed the particular case of computing indirect illumination on glossy surfaces. In its original form, ray tracing [126] could only compute the indirect illumination on ideal specular or refractive surfaces by casting a single deterministic reflected or refracted ray. Cook *et al.* [17] were the first to show that by casting multiple reflected rays in randomized directions, indirect illumination can be computed on surfaces with arbitrary reflectance, including glossy ones. With the introduction of the rendering equation, Kajiya [48] has given a mathematical framework for

global illumination. He proposed to solve the integral involved in the rendering equation by means of Monte Carlo sampling.

Monte Carlo sampling produces disturbing image-noise whose level decreases with the square root of the computation time. It takes a great deal of time to produce noise-free images using uninformed Monte Carlo sampling. Consequently, a number of noise reduction techniques have been proposed (*e.g.* [59, 107, 58, 108, 109, 15]).

The main drawback of Monte Carlo sampling is that the computation starts from scratch at each point; there is no information reuse between points, despite the strong coherence between the illumination at surface points. The main advantage of Monte Carlo sampling is the unbiasedness [52] of the result, which allows its applications in predictive rendering [35].

BRDF Importance Sampling. A general technique used for noise reduction in Monte Carlo is *importance sampling*. In evaluating the illumination integral, importance sampling is effectuated by placing more samples, that is to say, sending more rays, along directions where the value of the integrand $L^i(\mathbf{p}, \omega_i) f_r(\mathbf{p}, \omega_i, \omega_o) \cos \theta_i$ is more significant. Since the incoming radiance $L^i(\mathbf{p}, \omega_i)$ is not known at the time of evaluating the integral, normalized product of the BRDF and the cosine term is used for guiding the sample ray placement. BRDF importance sampling requires the inversion of the BRDF, which can be found only for a few analytical models, such as Phong's [88], Ward's [118], and Lafortune's [60]. General methods for importance sampling of arbitrary BRDFs have been proposed recently [61, 14, 62].

BRDF importance sampling is effective for sharp, narrow-peaked BRDFs, since the whole integrand is characterized by the BRDF peak. Therefore, only a few sample rays are sufficient for an accurate evaluation of the illumination integral (see the left side of Figure 2.2).

For low-frequency BRDFs, on which we focus in this thesis, importance sampling does not help much in reducing the variance. The lobes of such BRDFs are wide, spanning a substantial part of the hemisphere, and the integrand of the illumination integral is predominantly characterized by the incoming radiance L^i . Hence, many sample rays are needed to get a reasonably accurate estimate of the illumination integral, which is slow (see the right side of Figure 2.2).

Virtual Light Sources. Algorithms have been proposed for representing the indirect illumination by a number of virtual light sources [51, 111]. With virtual light sources, any direct illumination algorithm (*e.g.* hardware rendering or lightcuts [114]) can be used to compute indirect illumination. The algorithms work for glossy surfaces, but a high number of virtual light sources may be required. Those algorithms have problems representing short-distance light transfers in corners.

Density Estimation Approaches. Indirect illumination on glossy surfaces can also be computed by means of density estimation techniques, among which the most popular is photon mapping [47]. Light particles, called photons, are distributed in the scene and

interpolation [121, 96, 129, 6, 92, 105, 98, 104, 103, 97, 3] with two different purposes: acceleration of high quality global illumination and fast previewing of a partially converged solution.

Only few of those interpolation techniques can handle indirect illumination on glossy surfaces and are aimed at fast, high-quality image rendering. Radiance interpolants of Bala *et al.* [6] are designed for the use in a deterministic ray tracer and are not directly applicable to Monte Carlo ray tracing. Smyk *et al.* [97] use reprojection of radiance samples in final gathering on glossy surfaces to avoid hemisphere sampling at each visible point. However, reprojection is costly because of the necessity to reevaluate the BRDF at the interpolation position. Durand *et al.* [21] show an adaptive image sampling algorithm for glossy surfaces. Using the results of their frequency analysis of light transport, they infer the necessary sampling rate for glossy objects based on the frequency cutoff of the BRDF. They “emphasize that this application is meant only as a proof of concept...” and indeed, the image rendered with their method looks quite different from the reference solution. Walter *et al.* [114] interpolate illumination on glossy surfaces in image space, but the interpolation scheme is tailored to their specific way of computing illumination.

Ward *et al.*’s irradiance caching [121] is of particular importance for us, since our radiance caching is a direct extension of this algorithm. Therefore, we describe the algorithm in more detail here.

2.3.2 Irradiance Caching and Derived Algorithms

Irradiance Caching. Irradiance caching [121, 120] is a Monte Carlo ray tracing-based method for computing the *diffuse* term of indirect illumination. Computing diffuse indirect illumination at a point with Monte Carlo sampling requires sending many sample rays (500-8000), because the diffuse BRDF lobe spans the whole hemisphere. Repeating this computation at each visible point is too costly. Since indirect illumination changes very slowly over diffuse surfaces, irradiance caching performs the hemisphere sampling only at a sparse set of locations and interpolates the indirect illumination elsewhere.

Because of the view-independence of the diffuse BRDF, irradiance, $E(\mathbf{p}) = \int_{\Omega_i} L^i(\mathbf{p}, \omega_i) \cos \theta_i d\omega_i$, is sufficient to represent the diffuse indirect illumination at a point \mathbf{p} . The diffuse outgoing radiance in an arbitrary direction ω_o can be computed from irradiance using the formula $L^o(\mathbf{p}, \omega_o) = E(\mathbf{p})\rho(\mathbf{p})/\pi$, where $\rho(\mathbf{p})$ is the diffuse reflectance (albedo) at \mathbf{p} .

Each new irradiance value, computed by the costly hemisphere sampling, is stored in a cache for later reuse. To avoid computing irradiance at each point, a simple caching scheme is used:

```

If one or more irradiance values is stored near this point
    Use stored value(s)
Else
    Compute and store new irradiance value at this point

```

The caching scheme is illustrated in Figure 2.3 and described algorithmically in Figure 2.4.

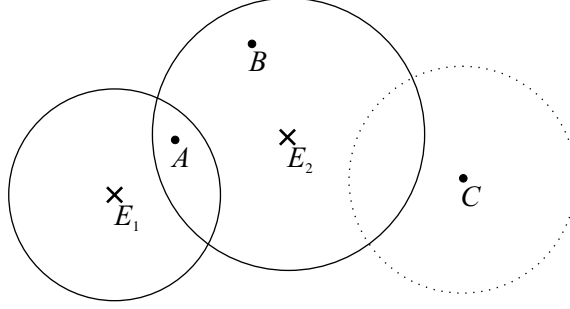


Figure 2.3: Irradiance values E_1 and E_2 were computed and stored in the cache previously. The weighted average of E_1 and E_2 is used for irradiance at the query point A . Point B uses E_2 . Point C results in computing a new irradiance value and storing it in the cache. (Figure adapted from [121].)

Irradiance at a point \mathbf{p} is interpolated from cached values by the weighted average:

$$E_{\mathcal{S}}(\mathbf{p}) = \frac{\sum_{i \in \mathcal{S}} (E_i + (\mathbf{n}_i \times \mathbf{n}) \cdot \nabla_r E_i + (\mathbf{p} - \mathbf{p}_i) \cdot \nabla_t E_i) w_i(\mathbf{p})}{\sum_{i \in \mathcal{S}} w_i(\mathbf{p})}, \quad (2.2)$$

where \mathbf{p}_i is the position of the i -th cached irradiance value E_i and \mathbf{n}_i is the surface normal at \mathbf{p}_i . The rotational and translational irradiance gradients, $\nabla_r E_i$ and $\nabla_t E_i$, are used for smoother interpolation. They are computed simultaneously with the irradiance value E_i and stored in the cache.

The weight of the i -th irradiance value for the weighted average at \mathbf{p} is given by the formula

$$w_i(\mathbf{p}) = \left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{R_i} + \sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i} \right)^{-1}, \quad (2.3)$$

where R_i is the harmonic mean of the distances to the objects visible from \mathbf{p}_i . The set \mathcal{S} of irradiance values that can be used for interpolation at \mathbf{p} is defined as

$$\mathcal{S} = \{i; w_i(\mathbf{p}) > 1/a\},$$

where a is a the user-defined desired approximation error. In other words, value i can be used only in the vicinity of its location, where the weight $w_i(\mathbf{p})$ is greater than the threshold $1/a$. The larger the value of a , the bigger the allowance for interpolation and the smaller the accuracy of the final image. If the set \mathcal{S} is empty at a given point, a new irradiance value is computed and stored in the cache.

The weight $w_i(\mathbf{p})$ is equal to the inverse of the upper bound on the irradiance gradient derived in [121]. As a consequence, density of the cached irradiance values is low in flat and open areas, where irradiance does not change rapidly, and high on curved surfaces or in the vicinity of other scene geometry, where more sudden irradiance changes are possible.

Algorithm: Irradiance caching scheme.

Input:

\mathbf{p} the point to compute the outgoing radiance at
 \mathbf{n} the surface normal at \mathbf{p}
 $\rho(\mathbf{p})$ diffuse reflectance at \mathbf{p}

Output:

$L^o(\mathbf{p}, \omega_o)$ outgoing radiance at \mathbf{p} due to the diffuse term of indirect illumination

```

1:  $\mathcal{S} := \text{look\_up\_cache}(\mathbf{p}, \mathbf{n})$ 
2: if  $\mathcal{S}$  is non-empty then
3:   // Interpolate irradiance using Equation (2.2).
4:    $E := E_{\mathcal{S}}(\mathbf{p})$ 
5: else
6:   // Sample hemisphere and store the new irradiance value in the cache.
7:    $[E, \nabla_r E, \nabla_t E, R] := \text{sample\_hemisphere}(\mathbf{p}, \mathbf{n})$ 
8:    $\text{store\_in\_cache}(\mathbf{p}, \mathbf{n}, E, \nabla_r E, \nabla_t E, R)$ 
9: end if
10: return  $E \cdot \rho(\mathbf{p}) / \pi$ 

```

Figure 2.4: Pseudocode for outgoing radiance computation with irradiance caching.

Algorithms Derived from Irradiance Caching. Slusallek *et al.* [96] and Kato [49] use sample reprojection with the purpose to better reuse the information from hemisphere sampling. Instead of caching and interpolating one aggregate value (irradiance E), they cache the incoming radiance samples from the hemisphere sampling and reproject them at the interpolation point. It is not clear, however, if the improvement of the interpolation quality cancels out the performance penalty caused by the reprojection. Tawara *et al.* [104] and Smyk *et al.* [97] also cache the incoming radiance samples used in hemisphere sampling. However, their primary motivation is not the reprojection, but the book-keeping needed to exploit the spatial coherence in animation rendering.

Smyk *et al.* [98] increase the density of the cached values based on the gradient magnitude to better accommodate to abrupt changes in indirect illumination. Tabellion and Lamorlette [103] integrate irradiance caching into their global illumination system for film rendering. They avoid an unduly high density of the cached values near corners by considering the projected pixel size, and storing at most one cached value for every other pixel.

Arikan *et al.* [3] increase the efficiency of irradiance caching by decomposing the indirect illumination into a near and a distant term. The distant term is computed at a very sparse set of positions in the scene using the costly hemisphere sampling. The near term is computed at a dense set of points near geometry features, but its computation is accelerated by ignoring occlusion for the near geometry. The near term is then used to add higher frequency, short-distance transport effect to the smoothly varying distant term.

2.3.3 Our Choice of the Interpolation Scheme

Motivation for Choosing Irradiance Caching. We chose irradiance caching as the starting point for developing radiance caching for the following reasons: First, irradiance caching has proven its practical usefulness in physically-based lighting simulation [119] and in film rendering [103]. Second, it is easy to implement and integrate into an existing ray tracer. Third, as a ray tracing-based algorithm, it does not impose any limitations on the scene geometry. Fourth, although computation is concentrated only in the visible parts of the scene, the cached irradiance is still view independent. Fifth, the view independence of the cached irradiance along with caching in object space¹ allows to reuse the cached irradiance values for different viewpoints in walk-through animation rendering. Sixth, since the interpolation is not image-based, it can be used at higher recursion levels in the ray tracing tree. Our radiance caching is designed to maintain those advantages and, additionally, to allow interpolation on glossy surfaces with low-frequency BRDFs.

Interpolated Quantity. Having chosen to build upon irradiance caching, we had to decide which physical quantity should be cached and interpolated on glossy surfaces. Irradiance E cannot be used due to the view-dependence of glossy surfaces. We have two possibilities: interpolate the outgoing radiance $L_{(\mathbf{p}, \omega_o)}^o$ for a fixed outgoing direction, or the directional distribution of incoming radiance over the hemisphere $L_{(\mathbf{p})}^i(\omega_i)$. If we choose to cache the outgoing radiance $L_{(\mathbf{p}, \omega_o)}^o$, as in [113, 117, 21], we will lose the view-independence of the cached information, one of the major strengths of irradiance caching. Therefore *we cache and interpolate the hemispherical incoming radiance*. This assures inheriting all the advantages of irradiance caching, including the view-independence of the cached information.

2.4 Incoming Radiance Representation

2.4.1 Requirements

Having chosen the hemispherical incoming radiance at a point as the most suitable candidate for caching and interpolation, we need to choose an appropriate representation meeting the following requirements:

- Allows to represent a function on the unit hemisphere (or the unit sphere).
- Consumes little memory.
- Allows efficient rotation.

The rotation is required to align the local coordinate frames at the interpolation point and at the point where the incoming radiance was computed.

¹as opposed to image space caching and interpolation

- Allows efficient evaluation of the illumination integral.
The illumination integral must be evaluated at every interpolation location to compute the outgoing radiance.

2.4.2 Previous Work

Piecewise constant representation [42, 9, 80, 12, 96, 49, 104, 97] represents a spherical function by storing its value at a discrete set of directions. This representation is simple but memory demanding and prone to aliasing.

Spherical harmonics [10, 89, 125, 90, 78, 77, 50, 94, 33, 93] are a generalization of Fourier series from the line to the unit sphere. They suppress aliasing and are efficient for representing low-frequency functions. However, representation of sharp functions require many coefficients and ringing may be a problem. Spherical harmonics have global support (they do not localize in directions), so changing the represented function at a small subset of the spherical domain generally causes all the representation coefficients to change. A function represented by spherical harmonics can be easily rotated by applying a linear transformation to the representation coefficients.

Hemispherical harmonics [27] are similar to spherical harmonics, but better suited for representing functions on a hemisphere. Other bases similar to spherical harmonics are Zernike polynomials [127, 53] and the hemispherical harmonics of Makhotkin [67]. Unlike for spherical and hemispherical harmonics, no rotation procedure is available for these basis functions.

Zonal harmonics [95] represent functions as a sum of rotationally symmetric lobes around arbitrary axes. The shape of each lobe is represented by a subset of spherical harmonics basis functions. Zonal harmonics have similar fitting properties as spherical harmonics, but require a non-linear optimization to fit a given function. The biggest advantage of zonal harmonics, compared to spherical or hemispherical ones, is the simplicity of rotation: it does not involve more than rotating the axes by a 3×3 Cartesian rotation matrix.

Representation by *wavelets* [74, 83, 84, 64, 99, 71, 72, 65, 115, 116, 13] has a number of advantages over spherical harmonics. Wavelets localize in directions, allow an adaptive representation of functions that are varying in a part of the domain and smooth elsewhere, and allow progressive evaluation. They can have problems with aliasing, though, unless high order wavelets are used. No rotation procedure exists for functions represented by wavelets.

2.4.3 Our Choice of the Incoming Radiance Representation

In radiance caching, we choose to represent the incoming radiance with *hemispherical and spherical harmonics* because they are *the only option for which an efficient rotation procedure is available*. Additionally, they have good anti-aliasing properties, low storage cost and are easy to use. We give more details on spherical and hemispherical harmonics in the following subsection.

Piecewise constant representation is not an option because it is too memory demanding and prone to aliasing. Zonal harmonics would have been an alternative if they did not require the slow non-linear optimization to fit a function. Wavelets are a suitable alternative to the use of spherical or hemispherical harmonics, especially for higher frequency BRDFs, but a rotation procedure must be first conceived.

2.4.4 Spherical and Hemispherical Harmonics

Our description of spherical harmonics is geared towards their practical use in radiance caching. A more general treatise can be found in physics literature [66, 43], or, within computer graphics, in [77, 33, 128].

Representation by Spherical Harmonics. Any function $L(\omega)$ defined on the unit hemisphere Ω can be approximated in terms of spherical harmonics as the sum

$$L(\omega) \approx \sum_{l=0}^{n-1} \sum_{m=-l}^l \lambda_l^m Y_l^m(\omega),$$

where ω is a direction in 3D and Y_l^m are the real spherical harmonics basis functions. Definition and evaluation algorithm for the spherical harmonics is given in [33]. Coefficients λ_l^m constitute the representation of $L(\omega)$ with respect to the spherical harmonics basis. n is the *order* of the approximation; there are n^2 coefficients in the approximation of order n .

Spherical harmonics of equal l index form a *band*, with one harmonic in the first band ($m = 0$), three in the second band ($m = -1, 0, 1$), five in the third band ($m = -2, -1, 0, 1, 2$), etc. Although the coefficients have two indices, l and m , they can be stored in a linear array $[\lambda_0^0, \lambda_1^{-1}, \lambda_1^0, \lambda_1^1, \dots]$, indexed by $i = l(l+1) + m$.

Projection. The representation coefficients λ_l^m are found by projecting the function L onto the spherical harmonics basis:

$$\lambda_l^m = \int_{\Omega} L(\omega) Y_l^m(\omega) d\omega.$$

This integral can be evaluated with Monte Carlo quadrature by generating N random, uniformly distributed directions ω_k over the unit hemisphere [22] and enumerating

$$\lambda_l^m = \frac{2\pi}{N} \sum_{k=1}^N L(\omega_k) Y_l^m(\omega_k). \quad (2.4)$$

Product integral. Integral of the product of two hemispherical functions represented by spherical harmonics is given by the dot product of the representation coefficients:

$$\int_{\Omega} L(\omega) F(\omega) d\omega = \sum_{l=0}^{n-1} \sum_{m=-l}^l \lambda_l^m f_l^m = \Lambda \cdot \Phi, \quad (2.5)$$

where $\Lambda = [\lambda_l^m]$ is the vector of coefficients for L and $\Phi = [f_l^m]$ is the vector of coefficients for F . This property is essential in radiance caching for an efficient evaluation of the illumination integral.

Hemispherical Harmonics. Hemispherical harmonics H_l^m have been proposed as an alternative to spherical harmonics for a more efficient representation of functions on the unit hemisphere [27]. They are obtained from spherical harmonics by shifting the definition domain from the full sphere to the hemisphere. Hemispherical harmonics provide a slightly better fit of a given function than spherical harmonics, which may reduce ringing when a too low number of coefficients is used.

All the properties discussed above for spherical harmonics are also valid for hemispherical harmonics. In the following text, we use H_l^m to denote both hemispherical and spherical harmonics, unless specified otherwise.

2.5 BRDF Representation

In radiance caching, we represent the BRDF in the same basis as the incoming radiance, *i.e.* spherical or hemispherical harmonics, in order to be able to use Equation (2.5) for an efficient evaluation of the illumination integral. In radiance caching, the BRDF representation is queried in the following way: given an outgoing direction ω_o , return the coefficients representing the BRDF lobe for that outgoing direction. Therefore, the BRDF representation suitable for radiance caching must allow for *fixing the outgoing direction*.

This requirement precludes the representation of the BRDF by the harmonics on the Cartesian product of two hemispheres [125, 63]. One can imagine this as a representation by the harmonics, where each coefficient is a hemispherical function represented, again, by the harmonics. Such a representation is not suitable for radiance caching, because it does not allow fixing the outgoing direction.

The BRDF representation of Sillion *et al.* [89] allows for fixing the outgoing direction, but is appropriate only for isotropic BRDFs. Since we want to handle general, anisotropic BRDFs, we have chosen the representation of Kautz *et al.* [50], which we describe in more detail.

BRDF Representation of Kautz *et al.* [50]. The BRDF is pre-multiplied by the cosine term, $\hat{f}_r(\mathbf{p}, \omega_i, \omega_o) = f_r(\mathbf{p}, \omega_i, \omega_o) \cos \theta_i$. For the sake of clarity, we omit the position \mathbf{p} in what follows.

The outgoing directions are discretized. For each discrete outgoing direction ω_o , the BRDF at a point is a hemispherical function of the incoming direction, $\hat{f}_{r(\omega_o)}(\omega_i)$, and as such, it can be directly represented by the harmonics:

$$\hat{f}_{r(\omega_o)}(\omega_i) \approx \sum_{l=0}^{n-1} \sum_{m=-l}^l c_{l(\omega_o)}^m H_l^m(\omega_i). \quad (2.6)$$

The representation coefficients $c_{l(\omega_o)}^m$ are computed by projecting $\hat{f}_{r(\omega_o)}(\omega_i)$ onto the harmonics basis:

$$c_{l(\omega_o)}^m = \int_{\Omega} \hat{f}_{r(\omega_o)}(\omega_i) H_l^m(\omega_i) d\omega_i. \quad (2.7)$$

This integral can be evaluated with Monte Carlo quadrature similar to Equation (2.4). We use $C_{(\omega_o)}$ to denote the whole coefficient vector representing the BRDF for the fixed outgoing direction ω_o .

The coefficient vector $C_{(\omega_o)}$ for each discrete outgoing direction is stored in a unit square texture. Each texel $[u, v]$ of the texture corresponds to one fixed outgoing direction. The outgoing directions are mapped onto the unit square $[0, 1] \times [0, 1]$ with the parabolic parametrization of Heidrich and Seidel [40]:

$$u = \frac{1}{2} \left(1 + \frac{x}{z+1} \right)$$

$$v = \frac{1}{2} \left(1 + \frac{y}{z+1} \right),$$

where x , y , and z are the Cartesian components of ω_o .

2.6 Summary

In this chapter we summarized the previous work related to radiance caching. We started by formulating final gathering as an evaluation of the illumination integral (2.1) at a point. We summarized previous radiosity-based global illumination algorithms suitable for scenes with glossy surfaces. We, then, described Monte Carlo importance sampling as a general ray tracing-based method for evaluating the illumination integral. We decided to exploit the smoothness of indirect illumination on surfaces with low-frequency BRDFs by interpolation, and described the related interpolation-based algorithms. We discussed various possible representations for directional distribution of incoming radiance, out of which we have chosen spherical and hemispherical harmonics. Finally, we described the BRDF representation of Kautz *et al.* [50], used in radiance caching.

Chapter 3

Radiance Caching

In this chapter we give a concise description of the radiance caching algorithm, describe its integration in a ray tracer and verify the algorithm on test scenes. Solutions to particular sub-problems of radiance caching are described in the later chapters of the thesis.

3.1 Radiance Caching in a Nutshell

The overall goal of radiance caching is high quality global illumination in scenes with glossy surfaces. In particular, radiance caching is a ray tracing-based algorithm designed for efficient computation of the indirect term of the illumination integral on glossy surfaces with low-frequency BRDFs. The only alternative method—Monte Carlo importance sampling—is slow and generates disturbing noise if used on such surfaces.

In its core, radiance caching is based on Monte Carlo sampling, too. However, it gains its efficiency by reusing the illumination computed at a point for locations in its vicinity. More specifically, directional distribution of incoming radiance L^i , computed at a point by the costly Monte Carlo hemisphere sampling, is stored in the *radiance cache* for later reuse through interpolation. The interpolation is possible since shading due to indirect illumination tends to change slowly as one moves over a surface with a low-frequency BRDF.

The caching and interpolation scheme used in radiance caching is borrowed from irradiance caching of Ward *et al.* [121], described in Section 2.3.2. The only difference is the cached quantity. While Ward *et al.* caches irradiance, we cache the directional distribution of incoming radiance at a point. This allows us to use the cached value to compute the outgoing radiance in an arbitrary outgoing direction on a glossy surface. We maintain the information about the directional distribution, since the BRDFs of glossy surfaces are view-dependent, *i.e.* for different outgoing directions, incoming light is reflected from different parts of the hemisphere of incoming directions.

3.2 Incoming Radiance Computation and Storage

To represent the directional distribution of incoming radiance at a point, which is a function defined on the unit hemisphere, we use the spherical or hemispherical harmonics. The properties of this representation are described in Section 2.4, where the justification of this choice among other alternatives is also given. Incoming radiance L^i is represented by a vector of coefficients $\Lambda = [\lambda_l^m]$ as

$$L^i(\omega) \approx \sum_{l=0}^{n-1} \sum_{m=-l}^l \lambda_l^m H_l^m(\omega),$$

where n is the representation order and H_l^m are the harmonics basis functions.

The coefficients λ_l^m are given by the projection of the incoming radiance L^i onto the harmonics basis:

$$\lambda_l^m = \int_{\Omega_p} L^i(\omega) H_l^m(\omega) d\omega.$$

Since our knowledge of L^i is based solely on hemisphere sampling (*i.e.* ray casting), we compute λ_l^m by Monte Carlo quadrature:

$$\lambda_l^m = \frac{2\pi}{N} \sum_{k=1}^N L^i(\omega_k) H_l^m(\omega_k), \quad (3.1)$$

where $L^i(\omega_k)$ is the incoming radiance from the sample direction ω_k , drawn from the uniform distribution over the hemisphere. Given two random numbers, ζ_1 and ζ_2 , uniformly distributed in $[0, 1)$, the spherical coordinates of a sample direction can be computed with $(\theta_k, \phi_k) = (\arccos \zeta_1, 2\pi\zeta_2)$. The number of sample directions, N , is specified by the user. We use a fixed N , observing that adaptive hemisphere sampling similar to [85, 86] is desirable. We will address adaptive sampling in future work.

Radiance Cache Record. The computed incoming radiance coefficients, along with other information, are stored as a record in the radiance cache, implemented as an octree for fast lookup [121]. Radiance cache record i contains the following items:

- position \mathbf{p}_i ,
- $[\mathbf{u}_i, \mathbf{v}_i, \mathbf{n}_i]$, axes of the local coordinate frame at \mathbf{p}_i ,
- coefficient vector $\Lambda_i = [\lambda_l^m]$ representing the incoming radiance,
- two derivative vectors $\frac{\partial \Lambda_i}{\partial x}$ and $\frac{\partial \Lambda_i}{\partial y}$ representing the x and y components of the translational radiance gradient, and
- R_i , the harmonic mean of distances to objects visible from \mathbf{p}_i .

Algorithm: Radiance caching scheme.

Input:

\mathbf{p} the point to compute the outgoing radiance at
 $[\mathbf{u}, \mathbf{v}, \mathbf{n}]$ axes of the local coordinate frame at \mathbf{p}
 $C_{(\mathbf{p}, \omega_o)}$ vector of coefficients representing the BRDF at \mathbf{p} for the outgoing direction ω_o

Output:

$L^o(\mathbf{p}, \omega_o)$ outgoing radiance at \mathbf{p} due to indirect illumination

```

1:  $\mathcal{S} := \text{look\_up\_cache}(\mathbf{p}, \mathbf{n})$ 
2: if  $\mathcal{S}$  is non-empty then
3:   // Interpolate incoming radiance coefficients using Equation (3.2).
4:    $\Lambda := \Lambda_{\mathcal{S}}(\mathbf{p})$ 
5: else
6:   // Sample hemisphere and store the new radiance record in the cache.
7:    $[\Lambda, \frac{\partial \Lambda}{\partial x}, \frac{\partial \Lambda}{\partial y}, R] := \text{sample\_hemisphere}(\mathbf{p}, [\mathbf{u}, \mathbf{v}, \mathbf{n}])$ 
8:    $\text{store\_in\_cache}(\mathbf{p}, [\mathbf{u}, \mathbf{v}, \mathbf{n}], \Lambda, \frac{\partial \Lambda}{\partial x}, \frac{\partial \Lambda}{\partial y}, R)$ 
9: end if
10: // Evaluate the illumination integral as the dot product of radiance and BRDF
    coefficients.
11: return  $\Lambda \cdot C_{(\mathbf{p}, \omega_o)}$ 

```

Figure 3.1: Pseudocode for outgoing radiance computation with radiance caching.

3.3 Caching Scheme and Interpolation

The caching scheme used in radiance caching to avoid computing the incoming radiance at each point, is borrowed from irradiance caching without any change:

```

If one or more radiance values is stored near this point
    Use stored value(s)
Else
    Compute and store new radiance value at this point

```

The scheme is illustrated in Figure 2.3. The formal description of this scheme, given in Figure 3.1, differs from that of irradiance caching (Figure 2.4) by the actions taken to interpolate the incoming radiance and compute the final result, the outgoing radiance L^o .

Radiance Interpolation. If a radiance cache query succeeds at a point \mathbf{p} (*i.e.* some radiance values are stored near \mathbf{p}), the incoming radiance coefficients are interpolated

with the weighted average:

$$\Lambda_{\mathcal{S}}(\mathbf{p}) = \frac{\sum_{i \in \mathcal{S}} \mathbf{R}_i \left(\Lambda_i + d_x \frac{\partial \Lambda_i}{\partial x} + d_y \frac{\partial \Lambda_i}{\partial y} \right) w_i(\mathbf{p})}{\sum_{i \in \mathcal{S}} w_i(\mathbf{p})}. \quad (3.2)$$

For each record i , contributing to the interpolation, the following operations are carried out. First, the radiance coefficient vector is adjusted based on the translational gradient, by adding the term $d_x \frac{\partial \Lambda_i}{\partial x} + d_y \frac{\partial \Lambda_i}{\partial y}$. Here, d_x and d_y are the displacements of $\mathbf{p} - \mathbf{p}_i$ along the x and y axes of the record's local coordinate frame, *i.e.* $d_x = (\mathbf{p} - \mathbf{p}_i) \cdot \mathbf{u}_i$ and $d_y = (\mathbf{p} - \mathbf{p}_i) \cdot \mathbf{v}_i$. The displacement along the local z axis is not taken into account, since it is typically very small. The two derivative vectors, $\frac{\partial \Lambda_i}{\partial x}$ and $\frac{\partial \Lambda_i}{\partial y}$, which constitute the gradient, are computed in the hemisphere sampling, when a new record is being created, as detailed in Chapter 4. They are stored with the new record in the cache. The use of gradients substantially improves the smoothness of the interpolated radiance.

After the gradient-based adjustment, the local coordinate frame at the record position \mathbf{p}_i is aligned with the frame at the interpolation point \mathbf{p} by rotating the cached incoming radiance (see Figure 3.2). This is done by multiplying the radiance coefficients by the rotation matrix \mathbf{R}_i as described in Chapter 5. The rotation has the side effect of replacing the rotational gradients used in irradiance caching.

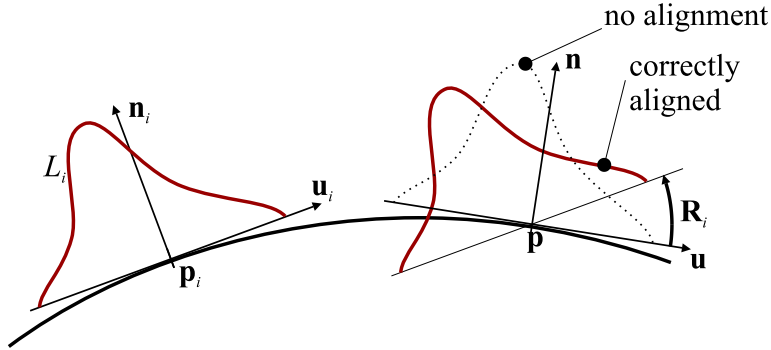


Figure 3.2: Rotation \mathbf{R}_i aligns the local coordinate frame at the record location \mathbf{p}_i with that at the interpolation point \mathbf{p} to make interpolation possible.

Finally, the rotated coefficients are multiplied by the weight $w_i(\mathbf{p})$ and added to the interpolation sum. With only a slight modification, we adopt the weighting function from irradiance caching:

$$w_i(\mathbf{p}) = \left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{\max\{R_-, \min\{R_+, R_i\}\}} + \sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i} \right)^{-1},$$

where \mathbf{n} is the surface normal at \mathbf{p} . Following the idea of Tabellion and Lamorlette [103], we let R_+ and R_- be respectively 50 and 2 times the square root of the projected pixel

area at the point \mathbf{p} . Limiting the minimum value of R_i by R_- avoids an excessively high record density in corners. On the other hand, R_i is not allowed to exceed R_+ in order to assure adequate record density in open exterior areas.

The definition of the set \mathcal{S} of records used for interpolation at \mathbf{p} is also taken from irradiance caching:

$$\mathcal{S} = \{i | w_i(\mathbf{p}) > 1/a\},$$

where a is a user defined allowed approximation error. The higher the value of a , the bigger the allowance for interpolation and the smaller the accuracy of the final image. The definition of the set \mathcal{S} means that the record i can be used for interpolation in the neighborhood of its location, where $w_i(\mathbf{p}) > 1/a$ holds. This neighborhood is larger in flat open areas, where the indirect illumination changes slowly over surfaces, and smaller on curved objects and near scene geometry, where the indirect illumination changes more quickly. This adapts the density of the cached radiance values to the expected rate of change of the indirect illumination, thereby guaranteeing smooth interpolation. The location of cached values in an example scene is shown in Figure 3.3.

The interpolation scheme was derived in [121] for diffuse surfaces. Even though the derivation does not hold for glossy surfaces, the interpolation gives surprisingly good results in radiance caching. However, there are some rare situations when it fails. This is why we have developed a perceptual error criterion to guide the record density, described in Chapter 6.

Outgoing Radiance Computation. To turn the interpolated directional distribution of incoming radiance into the outgoing radiance, the illumination integral (2.1) must be evaluated. Since hemispherical and spherical harmonics form an orthonormal basis, the integral evaluation reduces to the dot product:

$$L^o(\mathbf{p}, \omega_o) = \Lambda_{\mathcal{S}}(\mathbf{p}) \cdot C_{(\mathbf{p}, \omega_o)} = \sum_{l=0}^{n-1} \sum_{m=-l}^l \lambda_l^m c_{l(\mathbf{p}, \omega_o)}^m, \quad (3.3)$$

where $\Lambda_{\mathcal{S}}(\mathbf{p}) = [\lambda_l^m]$ is the interpolated incoming radiance coefficient vector and $C_{(\mathbf{p}, \omega_o)} = [c_{l(\mathbf{p}, \omega_o)}^m]$ is the coefficient vector representing the BRDF at \mathbf{p} for the outgoing direction ω_o .

3.4 Order of the BRDF and Radiance Representation

The BRDFs are represented with the method of Kautz *et al.* [50], described in Section 2.5. The representation for all scene BRDFs is computed before the rendering starts. In this section we describe how the representation order for the BRDF and, consequently, for the incoming radiance, is automatically selected. The described procedure is also used to automatically distinguish between low-frequency BRDFs, for which radiance caching will be used, and high-frequency ones, for which the indirect illumination will be computed with Monte Carlo importance sampling.

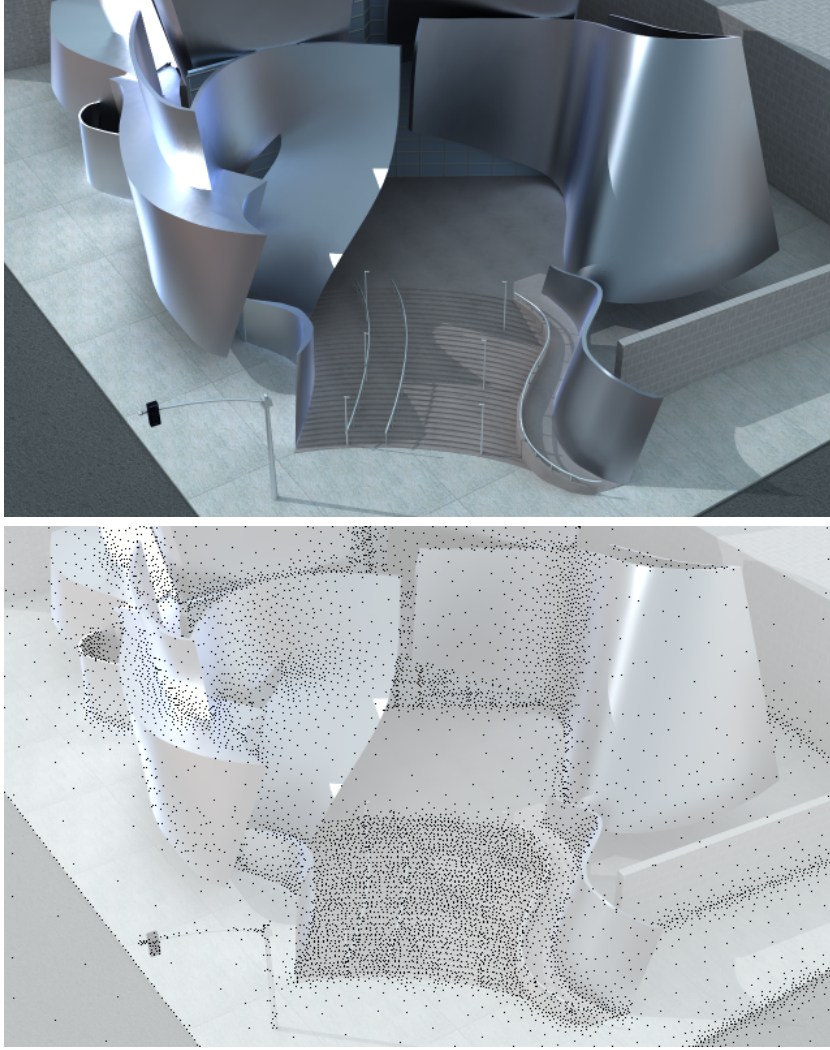


Figure 3.3: Positions of radiance cache records in an example scene. Hemisphere sampling was performed only at those positions during final gathering. The cached radiance was interpolated at all other positions. The top image shows the actual rendering, while the bottom image shows the record positions as black dots. Radiance caching was used for the metal walls and irradiance caching for all other surfaces.

Adaptive BRDF Representation. The user specifies the maximum allowed order n_{\max} and the maximum allowed BRDF representation error ϵ_{\max} . For each outgoing direction ω_o , the BRDF $f_r(\omega_o)$ is represented with the minimum order n sufficient to avoid exceeding ϵ_{\max} . The BRDF representation error is measured with the formula given by Ramamoorthi and Hanrahan [78]:

$$\epsilon = 1 - \frac{\sum_{l=0}^{n-1} \sum_{m=-l}^l |c_l^m(\omega_o)|^2}{\int_{\Omega} [f_r(\omega_o)(\omega)]^2 d\omega}. \quad (3.4)$$

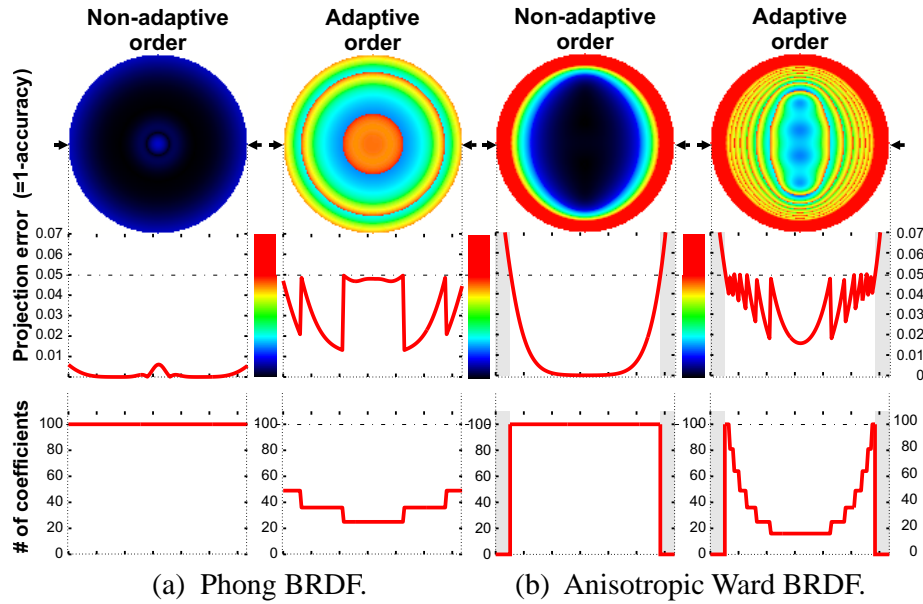


Figure 3.4: Adaptive BRDF representation for (a) Phong BRDF with exponent $h = 15$ and (b) anisotropic Ward BRDF [118] with $k_d = 0$, $k_s = 1$, $\alpha_x = 0.6$, $\alpha_y = 0.25$. The order of the hemispherical harmonics representation adapts to the BRDF without exceeding the specified maximum representation error (here 5%). The color disks show BRDF representation error for different outgoing directions ω_o . Directions are mapped to the disk with the parabolic parametrization. (One can imagine the disks as looking at the hemisphere from the top.) The graphs represent one scanline from the disk images (*i.e.* fixed y and varying x component of the outgoing direction). In the case of the Ward BRDF, radiance caching is not used for some directions, since the representation error would be too high.

The required order may significantly vary with the outgoing direction; higher order is usually needed for directions ω_o near the equator. If the BRDF representation is not accurate enough even with the maximum allowed order, we are dealing with a high-frequency BRDF, for which radiance caching will not be used. The BRDF representation is thus discarded. The adaptive BRDF representation is illustrated in Figure 3.4.

After this procedure, only low-frequency BRDFs are represented using the harmonics and radiance caching will be used for them. This constitutes the automatic criterion for discerning low- and high-frequency BRDFs in our rendering framework.

The maximum order n_{\max} should be selected such that a BRDF is classified as low-frequency if and only if radiance caching is more efficient than Monte Carlo importance sampling for that BRDF. While $n_{\max} = 10$ was a good value for our scenes, it does not necessarily have to be so in other ones. Higher n_{\max} allows using radiance caching for higher frequency BRDFs. The higher the BRDF frequency, the more rays required in hemisphere sampling, and the less reuse for a new record.

Incoming Radiance Representation Order. The order for the incoming radiance representation at the point \mathbf{p} is set to a value equal to the order of the BRDF representation at \mathbf{p} . By doing this, we discard the high frequencies of the incoming radiance distribution. However, the analysis of Ramamoorthi [77] shows that high frequencies of the incoming radiance do not affect the outgoing radiance, since a low-frequency BRDF acts as a low-pass filter upon the incoming radiance. This justifies our approach.

3.5 Rendering with Radiance Caching

Radiance caching is a part of a ray tracing approach to global illumination. At every ray-surface intersection, the outgoing radiance is evaluated with the illumination integral (2.1). The integral is split into parts, each of which is solved with a different technique:

- Direct illumination uses area sampling of light sources [87].
- Ideal specular reflections/refractions are solved by tracing a single deterministic ray.
- Ward’s irradiance caching computes the indirect diffuse term for *purely* diffuse surfaces.
- Two different techniques may be used for glossy surfaces.

Low-frequency BRDF. Our radiance caching computes the indirect *glossy and diffuse* terms.

High-frequency BRDF. Monte Carlo importance sampling computes the indirect glossy term and irradiance caching computes the indirect diffuse term.

Radiance caching is not used for high-frequency BRDFs, since for those BRDFs, importance sampling provides good accuracy with only a few sample rays, outperforming radiance caching. The steps of the rendering algorithm related to radiance caching are summarized in Figure 3.5.

3.6 Discussion

Rotation Loss. There is a loss of information when radiance is interpolated on a curved surface (Figure 3.6). Due to the rotation \mathbf{R}_i , a part of the radiance incident at \mathbf{p}_i should disappear under the surface (marked “a” in Figure 3.6) and should not contribute to the interpolated radiance at \mathbf{p} . A part of the radiance actually incident at \mathbf{p} is not represented by the radiance record at \mathbf{p}_i (marked “b” in Figure 3.6) and is missing in the interpolated radiance.

This problem is not due to the usage of a hemispherical basis for representing the incoming radiance, but stems from the fact that the incident radiance at a surface point *is* a hemispherical function. Using spherical harmonics instead of hemispherical harmonics

Algorithm: Rendering with radiance caching.

```

1: // preprocessing - BRDF representation
2: for (every BRDF in the scene) do
3:   if (BRDF suitable for radiance caching) then
4:     Compute and store the representation of the surface's BRDF.
5:   end if
6: end for

7: // rendering with radiance caching
8: for (every ray-surface intersection p) do
9:   Retrieve the representation of the BRDF at p.
10:  if (representation of the BRDF not available) then
11:    // high-frequency BRDF
12:    Monte Carlo importance sampling computes the indirect glossy term.
13:    Irradiance caching computes the indirect diffuse term.
14:  else
15:    // low-frequency BRDF; use radiance caching
16:    Compute outgoing radiance with Algorithm 3.1.
17:  end if
18: end for

```

Figure 3.5: Outline of rendering with radiance caching.

does not aid this problem. In practice the error introduced by this problem is very small because the difference between the normal at **p** and the normal at any record used for interpolation at **p** is small. We have not noticed any image artifacts due to this problem. Note also that Ward et al.'s irradiance caching suffers from the same problem.

Global versus Local Coordinates. Incoming radiance at a point can be represented either in the local coordinate frame at that point or in the global coordinate frame. This determines how the interpolation at **p** is performed:

- *Incoming radiance in the global frame*

```

for (each available record i) do
  Update the interpolation sums in Equation (3.2).
end for
Align the result with the local frame at p.
Compute the dot product.

```
- *Incoming radiance in the local frame*

```

for (each available record i) do
  Align the frame at pi with the frame at p.
  Update the interpolation sums in Equation (3.2).
end for
Compute the dot product.

```

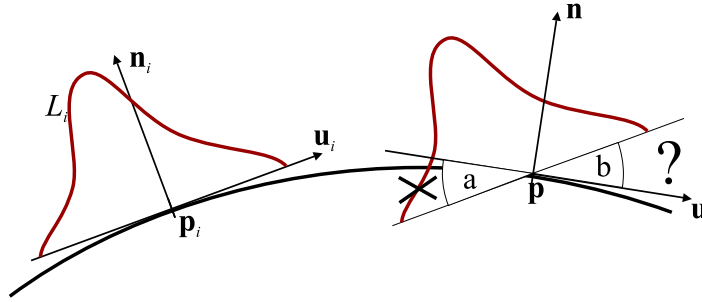


Figure 3.6: Information loss when radiance is interpolated on a curved surface. When interpolating at \mathbf{p} , a part of the incoming radiance cached at \mathbf{p}_i should disappear under the horizon. A part of radiance actually incident at \mathbf{p} is not represented by the radiance record at \mathbf{p}_i .

The final dot product (Equation 3.3) is always carried out in the local frame at \mathbf{p} . On curved surfaces, fewer rotations are needed if the incoming radiance is represented in the global frame. On the other hand, if the incoming radiance is represented in the local frame, no alignment (even with the BRDF) is needed on flat surfaces. One more reason for representing the incoming radiance in the local frame is that our fast rotation, described in Chapter 5, only works for the small rotations used for aligning the local frames of nearby points. However, it fails for the large rotations needed to align the local and the global frame. Note that hemispherical harmonics cannot be used to represent the incoming radiance in the global frame.

Suitability of Spherical and Hemispherical Harmonics. We use (hemi)spherical harmonics since a rotation procedure is available for them. Additionally, they are simple, computationally efficient (manipulations of vectors of floats), and they avoid aliasing. The essential disadvantage is the lack of directional localization. When we create a new radiance cache record, the full hemisphere must be sampled. The more directional the BRDF is, the more this approach becomes wasteful. With a basis that localizes in directions, only the required part of the hemisphere would need to be sampled. For this purpose one can use piecewise constant representation [96, 49, 104], but it would presumably introduce aliasing. The use of spherical wavelets [84] is probably a good choice, but a rotation procedure must first be conceived.

General limitations. The restriction to low-frequency BRDFs is the main limitation of radiance caching. The BRDF frequencies used in the example scenes of the Results section are near the limit of what our technique can handle. We believe that higher frequency BRDFs can be handled within the presented framework by using a localized basis such as wavelets. Additionally, radiance caching, as a gathering technique, cannot solve certain types of light transport phenomena, such as caustics. Those have to be solved by a more adapted technique.

3.7 Results

Figure 3.7 gives the breakdown of rendering times for three of the scenes we used to test radiance caching (Cornell Box, Glossy Flamingo, Walt Disney Hall). The timings were measured on a 2.2GHz Pentium 4, with 1 GB RAM, running Windows XP. The maximum harmonics order for radiance caching was set to $n = 10$, which corresponds to approximately 3.6 kB-sized radiance cache records.

We compared the images obtained with radiance caching to those obtained with Monte Carlo importance sampling in the same rendering time. Importance sampling uses the surface BRDF as the importance function. The two rendering methods exhibit artifacts of very different nature: high-frequency noise for importance sampling (‘specks’ in images) and low-frequency error in radiance caching (uneven illumination gradients). It is therefore difficult to compare rendering times needed to attain the same visual quality. Instead, we have chosen to fix the rendering time and compare the image quality delivered by the two methods.

	Cornell Box 1280 × 1280		Walt Disney Hall 1600 × 1000		Glossy Flamingo 1280 × 1280		
	Frame I	Frame II	Frame I	Frame II	Frame I	Frame II	Frame III
RC filling	28.7	9.2	251	380	63.5	53.2	63.9
RC interpolation	4.6	4.6	31.4	35.1	4.42	11.4	8.7
Total	82.9	68.7	538	659	108	101	104

Figure 3.7: Timing breakdown for the test scenes. ‘RC filling’ is the time spent on computing and adding radiance cache records. ‘RC interpolation’ is the time spent on looking up the existing radiance cache records and interpolating radiance. ‘Total’ is the total rendering time. The difference between the total time and the time spent on radiance caching consists of primary ray casting, direct lighting calculation, and irradiance caching (in Walt Disney Hall and Glossy Flamingo). All times given in seconds.

Cornell Box

Figure 3.8 shows renderings of a Cornell box with a glossy back wall (Phong BRDF, exponent 22), taken from two viewpoints at resolution 1280 × 1280. Except for the back wall, all objects are diffuse. Only direct lighting and first bounce indirect glossy lighting for the back wall were computed.

Images in the top row were computed with radiance caching with the allowed caching error set to $a = 0.15$ and the number of rays cast to sample each hemisphere set to $N = 6000$. The indirect glossy term took 33.3 seconds to compute for the left image; total rendering time was 82.9 sec (direct illumination uses 8 samples per pixel to sample the area source). The number of radiance cache records was 600. The time spent on the indirect glossy term computation in the right image was only 13.8 sec, since the records from the left rendering were retained and only 164 additional record were required.

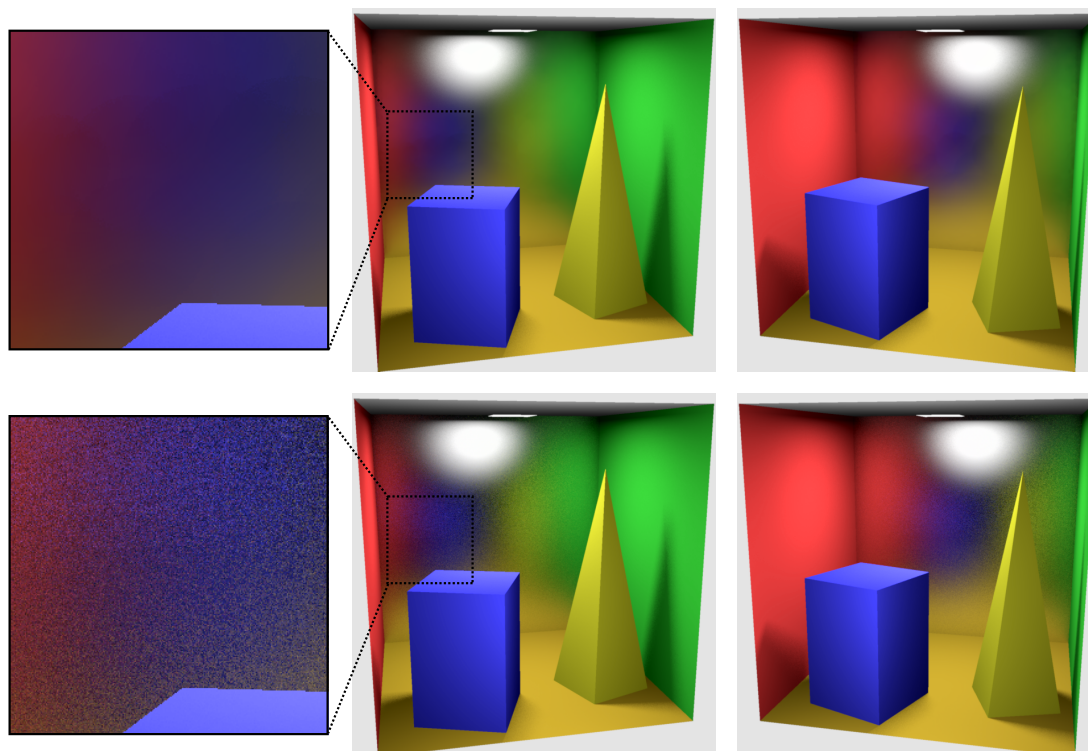


Figure 3.8: Two views of a Cornell box with a glossy back wall rendered with radiance caching (top) and Monte Carlo importance sampling (bottom).

The indirect glossy term for each of the two bottom images was computed in 35 seconds with Monte Carlo importance sampling using 12 reflected rays per point on a glossy surface. Those renderings exhibit a high noise level.

The average time spent on radiance caching for a 180 frames long animation with the camera moving between the position in the left and the right image was just 4.9 seconds per frame. Most of this time is spent on interpolation since only a few records are needed for additional frames. The average frame time with Monte Carlo importance sampling is 35 seconds since this method does not reuse any information from the previous frames. Even though this is 7 times more than for radiance caching, the quality is much lower.

Glossy Flamingo

Figure 3.9 shows three frames from the Glossy Flamingo animation. The flamingo was assigned the Phong BRDF (exponent 7) and all other surfaces are purely diffuse. The rendering resolution was 1280×1280 pixels. Full global illumination up to 4 bounces was computed. Irradiance caching was used to compute indirect lighting on diffuse surfaces.

First bounce indirect lighting on glossy surfaces for the images in the top row was computed using radiance caching; path tracing was used for deeper recursion levels. The allowed caching error was set to $a = 0.15$ and the number of rays for hemisphere sampling was $N = 1000$. Table 3.7 gives the rendering times for the three images when rendered

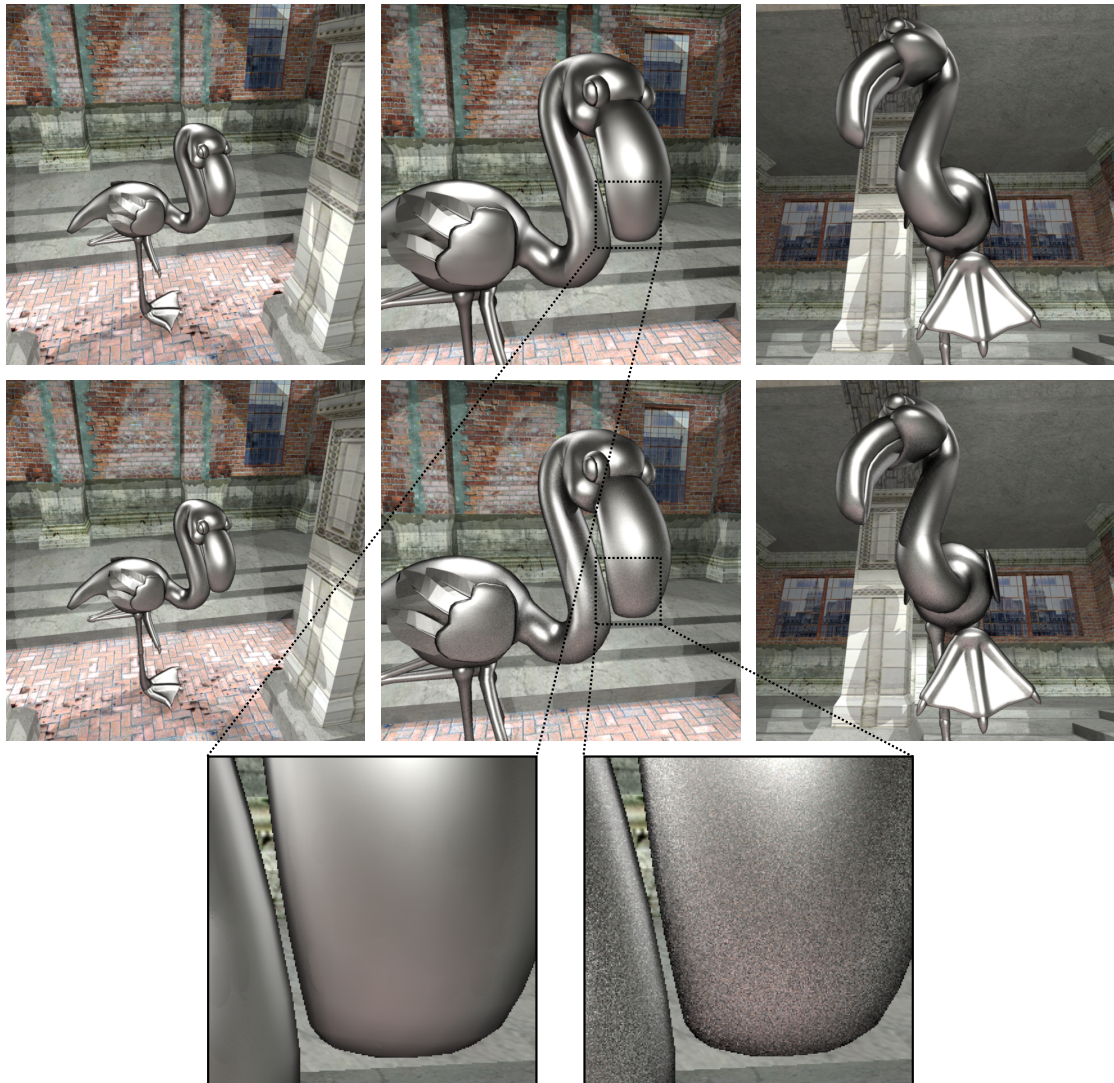


Figure 3.9: Frames from the Glossy Flamingo animation rendered with radiance caching (top row) and Monte Carlo importance sampling (second row).

independently, without retaining radiance cache records from previous renderings. Figure 3.10 shows the time spent on radiance caching in a 280 frames long animation with the camera moving between the shown images.

The images in the second row use Monte Carlo importance sampling instead of radiance caching. In order to keep the rendering time the same as for radiance caching, the number of reflected rays per point on a glossy surface was set to 12, 4, 6 respectively (from left to right).

The Glossy Flamingo scene is quite challenging for radiance caching because of the highly curved surfaces. On such surfaces, radiance cache records cannot be reused at as many pixels as on flat surfaces. Moreover, the costly rotation is required before each interpolation. In the image on the left, the flamingo occupies only a small part of the image and therefore we do not take advantage of the independence of radiance caching's com-

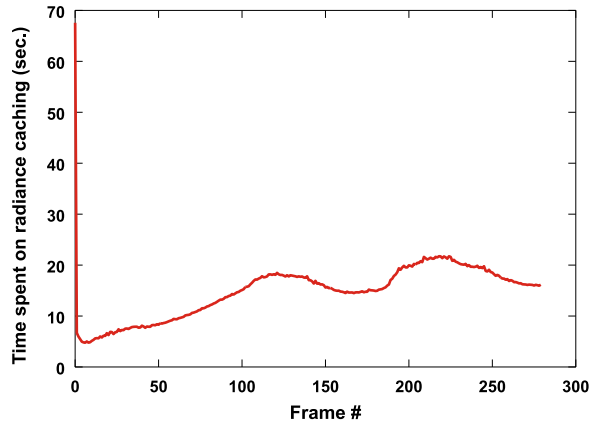


Figure 3.10: Time spent on radiance caching for the Glossy Flamingo animation. Cached records are shared among frames.

putation time on the image resolution. The quality advantage of radiance caching over Monte Carlo importance sampling can be seen only by a very close inspection. However, the images in the middle and on the right manifest the noise introduced by Monte Carlo sampling overtly. Notice also that for the animation rendering, the average time for radiance caching is 15 seconds per frame. Rendering the same animation using Monte Carlo importance sampling with only 2 reflected rays on a glossy pixel takes 27 seconds per frame spent on indirect glossy lighting computation. The quality obtained with radiance caching is considerably better.

Walt Disney Hall

Figures 3.12 and 3.13 show renderings of a simple model of the Walt Disney Hall in Los Angeles. The curved walls of the real building (see Figure 3.11) are covered with brushed metal tiles, whose BRDF we approximate with a three-lobe Lafortune model [60]. The illumination is due to the sun (modeled as a directional light), the sky (modeled as a constant blue light) and the surrounding urban environment (modeled as a constant brownish light). Similarly to the real building, walls reflect the sky or the surroundings depending on their normals and the viewpoint location (compare the Figures 3.12 and 3.13).

Full global illumination solution with one ray per pixel was computed at resolution 1600×1000 . Indirect illumination and the illumination coming from the sky and from the surroundings was computed in the same way: with irradiance caching on diffuse surfaces, with radiance caching on glossy surfaces in the top images, and with Monte Carlo importance sampling on glossy surfaces in the bottom images.

The allowed caching error was set to $a = 0.15$ leading to 10,200 records for Figure 3.12 and 13,500 records for Figure 3.13. We used 2000 rays for hemisphere sampling. The total rendering time was 538 s for Figure 3.12 and 659 s for Figure 3.13. Monte Carlo importance sampling in the bottom images used 30 reflected rays on each visible glossy pixel (first bounce) and one ray for higher order bounces. The rendering time was similar



Figure 3.11: Picture of Frank Gehry’s Walt Disney Hall in Los Angeles which was the inspiration for the Walt Disney Hall scene.

to that for radiance caching, but the quality of radiance caching results is substantially higher.

The BRDF used for the metallic walls [124] was fairly sharp; the Lafortune lobes had exponents of 16, 88 and 186. It is impossible to represent such a BRDF accurately with spherical or hemispherical harmonics of order 10—the average representation error was over 20%. Nevertheless, Figure 3.14 demonstrates that the radiance caching rendering does not show serious distortions of the material appearance compared to the reference solution¹. The images are noticeably different only in areas seen at very sharp grazing angles, for which the BRDF representation error is the highest (more than 50%).

3.8 Summary

In this chapter we described radiance caching, our interpolation-based algorithm for computing indirect illumination on glossy surfaces. The directional distribution of incoming radiance, computed at a point, is represented by spherical or hemispherical harmonics and stored in the radiance cache for later reuse through interpolation. Radiance interpolation consists of adjustment by the translational gradient, rotation, and interpolation of the projection coefficients. The outgoing radiance is computed from the interpolated incoming radiance using a simple dot product.

On the example scenes we have shown that radiance caching outperforms Monte Carlo importance sampling in terms of image quality obtained in the same rendering time. Radiance caching produces high quality images even for challenging scenes with highly curved surfaces and relatively sharp BRDFs.

¹The reference solution was computed with Monte Carlo importance sampling with 1,000 reflected rays.

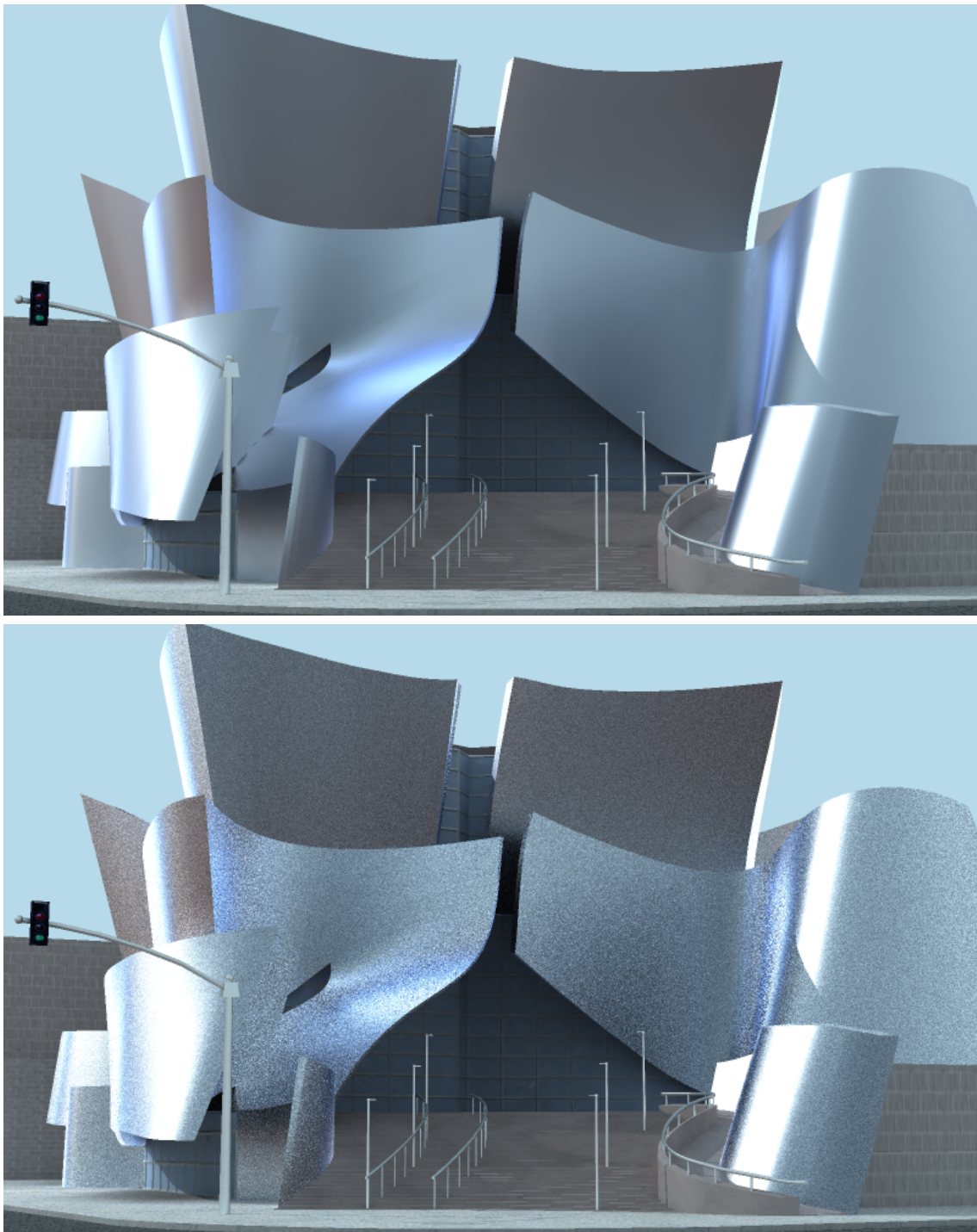


Figure 3.12: Rendering of a simple model of Walt Disney Hall in Los Angeles. The top image was rendered with radiance caching, the bottom one with Monte Carlo importance sampling. Both images were rendered in approximately the same time, 538 seconds.

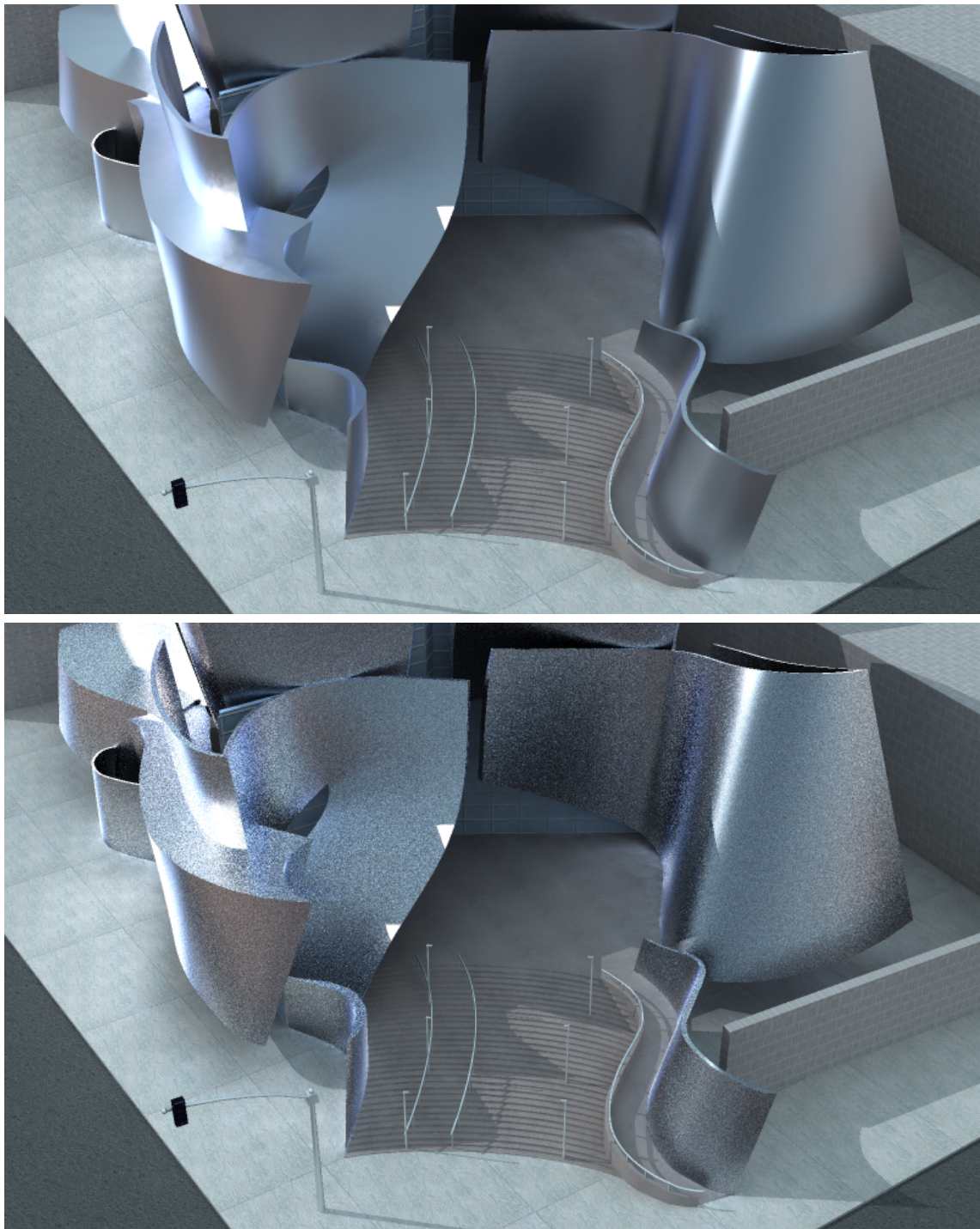


Figure 3.13: Another view of the Walt Disney Hall scene. The top image was rendered with radiance caching, the bottom one with Monte Carlo importance sampling. Both images were rendered in approximately the same time, 659 seconds.

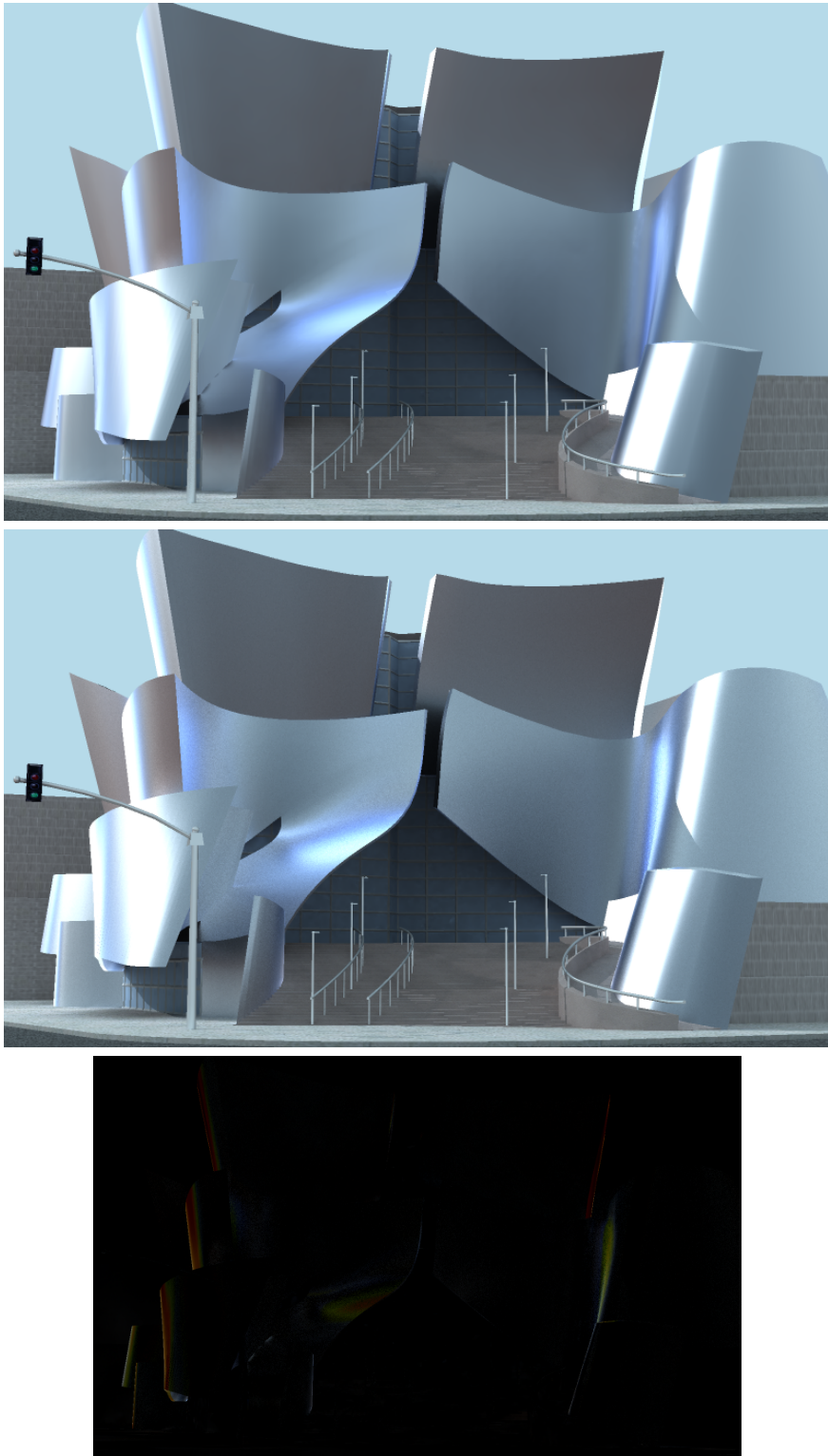


Figure 3.14: Comparison of a radiance caching rendering at the top with a reference image in the middle. The reference image was computed with Monte Carlo importance sampling using 1000 reflected rays per point. The difference image at the bottom shows that radiance caching causes error only in areas seen at sharp grazing angles, for which the BRDF representation has the highest error.

Chapter 4

Radiance Gradient Computation

Translational radiance gradients are used in radiance caching to improve the smoothness of the interpolated radiance. In this section we describe two methods for computing the gradients.

4.1 Introduction

Ward and Heckbert [120] found that in irradiance caching, the interpolation quality can be significantly improved by the use of translational and rotational irradiance gradients. The translational gradient characterizes the change of irradiance with a small displacement on a surface, while the rotational gradient describes the irradiance change with surface rotation. They also pointed out that the information contained in the hemisphere sampling, used in computing irradiance at a point, is sufficient to estimate the gradients.

We use *translational radiance gradients* in radiance caching to describe the change of the directional distribution of incoming radiance with a displacement over a surface. The gradients are used in the radiance interpolation formula (3.2) to effectively raise the interpolation order and produce smoothly varying indirect illumination. Improvement of the interpolation quality with the use of translational gradients is illustrated in Figure 4.1. Unlike Ward and Heckbert, we do not use rotational gradients, since the change of the directional distribution due to the rotation of the underlying surface is described by the rotation \mathbf{R}_i carried out in the interpolation (3.2).

Gradients are computed when a new radiance record is being created. A gradient is calculated from same incoming radiance samples $L^i(\omega_k)$ used in the quadrature formula (3.1) to compute the projection coefficients λ_l^m . For each coefficient λ_l^m , a gradient $\vec{\nabla}\lambda_l^m = \left[\frac{\partial\lambda_l^m}{\partial x}, \frac{\partial\lambda_l^m}{\partial y}, 0 \right]$ is computed. The derivatives $\frac{\partial\lambda_l^m}{\partial x}$ and $\frac{\partial\lambda_l^m}{\partial y}$ for all coefficients form two derivative vectors, $\frac{\partial\Lambda}{\partial x} = \left[\frac{\partial\lambda_l^m}{\partial x} \right]$ and $\frac{\partial\Lambda}{\partial y} = \left[\frac{\partial\lambda_l^m}{\partial y} \right]$, that are stored with the new record in radiance cache.

The translational gradients always lie in the tangent plane at the hemisphere center. In other words, we disregard the derivative with respect to the local z -axis. This is justified by the assumption of locally near flat surfaces, in which case the displacements along the z -axis are very small and therefore hardly influence the radiance change.

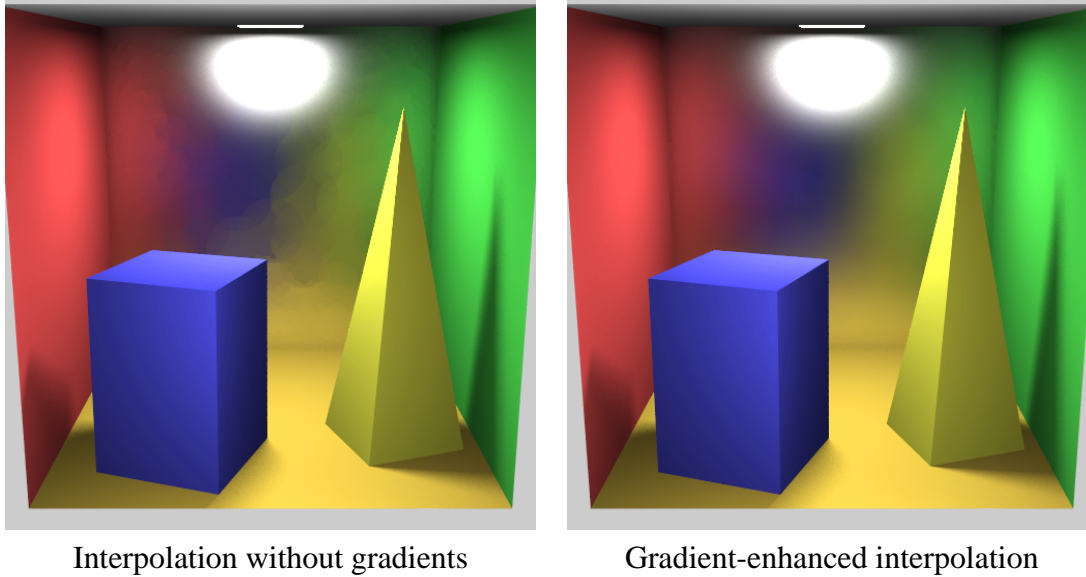


Figure 4.1: Comparison of radiance interpolation without (left) and with (right) translational radiance gradients. If gradients are not used, discontinuities are visible in the image (on the glossy back wall), while the use of gradients results in smooth interpolation with no image artifacts.

In this chapter we describe two different methods for computing the translational gradient. The first, solid angle-based method does not consider the changes in occlusion that can occur with translation—it can fail when the occlusion changes, indeed, take place. This is why we have developed another, stratification-based gradient computation method, which gives better results due to accounting for the changes in occlusion [54].

Previous work on illumination gradient computation includes the papers of Arvo [4] and Holzschuch and Sillion [41], who compute irradiance gradient due to polygonal emitters. In contrast, we compute radiance gradient due to arbitrary emitters, the knowledge about which is mediated solely through the hemisphere sampling. Computation of the solid angle-based gradient was independently developed by Annen *et al.* [2] for the use in precomputed radiance transfer.

4.2 Solid Angle-Based Gradient

The solid angle-based translational gradient can be computed in two equivalent ways. The first displaces the center of the hemisphere and approximates the gradient by *finite differences* of the coefficients λ_l^m . The second is based on *differentiating* the terms of the quadrature formula (3.1), used to compute the coefficients λ_l^m . In both cases, we compute the gradient $\vec{\nabla}\lambda_l^m = \left[\frac{\partial\lambda_l^m}{\partial x}, \frac{\partial\lambda_l^m}{\partial y}, 0 \right]$ by evaluating the partial derivatives $\partial\lambda_l^m/\partial x$ and $\partial\lambda_l^m/\partial y$.

4.2.1 Gradient by Finite Differences

We displace the hemisphere center \mathbf{p} , along the local x -axis, by Δx to \mathbf{p}' (Figure 4.2). For each Monte Carlo sample $L^i(\omega_k)$ we:

1. Compute the new direction ω'_k at \mathbf{p}' as $\omega'_k = \frac{\mathbf{q}_k - \mathbf{p}'}{r'_k}$. Here \mathbf{q}_k is the point hit by the ray from \mathbf{p} in direction ω_k and $r'_k = \|\mathbf{q}_k - \mathbf{p}'\|$. We also denote $r_k = \|\mathbf{q}_k - \mathbf{p}\|$. See Figure 4.2 for the terms used here.
2. Compute the solid angle Ω'_k associated with the new direction ω'_k . The solid angle Ω_k associated with each direction in Equation (3.1) is uniform and equal to $2\pi/N$. With the displacement of the point \mathbf{p} , the solid angles no longer remain uniform. The change in solid angle is due to the change in distance $r_k = \|\mathbf{q}_k - \mathbf{p}\|$ and orientation of the surface at \mathbf{q}_k , as seen from the hemisphere center \mathbf{p} or \mathbf{p}' . The solid angle before the displacement is

$$\Omega_k = \Delta A_k \frac{\cos \xi_k}{r_k^2} = \frac{2\pi}{N},$$

where ξ_k is the angle between the surface normal at \mathbf{q}_k and the vector from \mathbf{q}_k to \mathbf{p} . The area $\Delta A_k = \frac{2\pi}{N} \frac{r_k^2}{\cos \xi_k}$ is the part of the environment visible through Ω_k . It does not change with the displacement because we assume that the environment visible from \mathbf{p} and \mathbf{p}' is the same. After the displacement, the solid angle subtended by ΔA_k becomes

$$\Omega'_k = \Delta A_k \frac{\cos \xi'_k}{r_k'^2} = \frac{2\pi}{N} \frac{r_k^2}{r_k'^2} \frac{\cos \xi'_k}{\cos \xi_k}.$$

We, now, estimate the coefficient $\lambda_l^{m'}$ at \mathbf{p}' as:

$$\lambda_l^{m'} = \frac{2\pi}{N} \sum_{k=1}^N \frac{r_k^2}{r_k'^2} \frac{\cos \xi'_k}{\cos \xi_k} L^i(\omega_k) H_l^m(\omega'_k),$$

and, finally, we compute $\partial \lambda_l^m / \partial x = (\lambda_l^{m'} - \lambda_l^m) / \Delta x$.

The computation of $\partial \lambda_l^m / \partial y$ proceeds in a similar way. This completes the estimation of the translational gradient $\nabla \lambda_l^m$ at the point of interest through finite differences.

4.2.2 Gradient by Differentiation

In the second way of computing the solid angle-based gradient, the partial derivative $\partial \lambda_l^m / \partial x$ is calculated by differentiating the terms of Equation (3.1). We have seen in the previous section that $\Omega_k = \frac{2\pi}{N}$ does not remain constant with displacement of \mathbf{p} and therefore it has to be included in the sum and differentiated, too. By moving Ω_k inside the sum and using spherical coordinates, Equation (3.1) becomes

$$\lambda_l^m = \sum_{k=1}^N \Omega_k L^i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k), \quad (4.1)$$

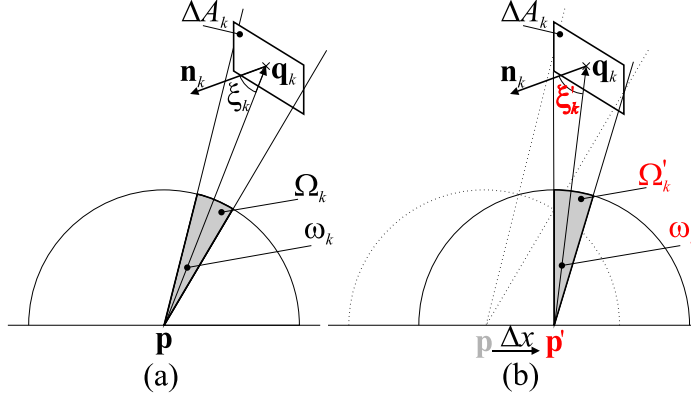


Figure 4.2: Gradient computation by displacing the hemisphere center from \mathbf{p} to \mathbf{p}' ((a) before and (b) after the displacement). The quantities changing with the displacement are (shown in red): sample ray direction ω_k , the solid angle Ω_k associated with this sample, and the angle ξ_k between the sample direction and the surface normal at the hit point \mathbf{q}_k . Neither the hit point \mathbf{q}_k nor the area ΔA_k visible through Ω_k change with the displacement.

with $\Omega_k = \frac{2\pi}{N}$ for uniform hemisphere sampling. We now differentiate (4.1):

$$\frac{\partial \lambda_l^m}{\partial x} = \sum_{k=1}^N \frac{\partial}{\partial x} \left(\Omega_k L^i(\theta_k, \phi_k) H_l^m(\theta_k, \phi_k) \right) \quad (4.2)$$

$$= \sum_{k=1}^N L^i(\theta_k, \phi_k) \left(\frac{\partial \Omega_k}{\partial x} H_l^m(\theta_k, \phi_k) + \Omega_k \frac{\partial H_l^m}{\partial x}(\theta_k, \phi_k) \right) \quad (4.3)$$

The derivative of the basis function is

$$\frac{\partial H_l^m}{\partial x}(\theta_k, \phi_k) = \frac{\partial \theta_k}{\partial x} \frac{\partial H_l^m}{\partial \theta_k}(\theta_k, \phi_k) + \frac{\partial \phi_k}{\partial x} \frac{\partial H_l^m}{\partial \phi_k}(\theta_k, \phi_k), \quad (4.4)$$

with [123]

$$\begin{aligned} \frac{\partial \theta_k}{\partial x} &= -\cos \theta_k \cos \phi_k / r_k, \\ \frac{\partial \phi_k}{\partial x} &= \sin \phi_k / (r_k \sin \theta_k). \end{aligned} \quad (4.5)$$

The derivatives of θ_k and ϕ_k with respect to y are

$$\begin{aligned} \frac{\partial \theta_k}{\partial y} &= -\cos \theta_k \sin \phi_k / r_k, \\ \frac{\partial \phi_k}{\partial y} &= -\cos \phi_k / (r_k \sin \theta_k). \end{aligned} \quad (4.6)$$

The derivatives $\partial H_l^m / \partial \theta_k$ and $\partial H_l^m / \partial \phi_k$ are given in Appendix A.

The derivative of the solid angle Ω_k is

$$\frac{\partial \Omega_k}{\partial x} = \frac{\partial}{\partial x} \Delta A_k \frac{\cos \xi_k}{r_k^2} = \Delta A_k \frac{\partial \cos \xi_k}{\partial x} \frac{1}{r_k^2}.$$

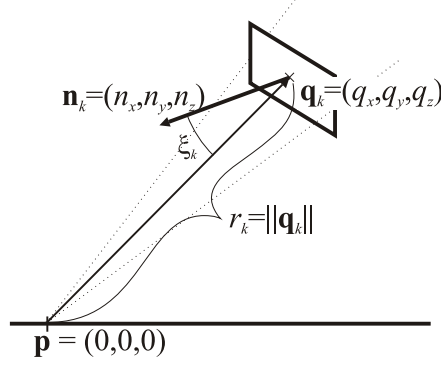


Figure 4.3: Quantities in the computation of $\frac{\partial}{\partial q_x} \frac{\cos \xi_k}{r_k^2}$.

The area $\Delta A_k = \frac{2\pi}{N} \frac{r_k^2}{\cos \xi_k}$ is the part of the environment visible through Ω_k . It does not change with the displacement. The change of $\cos \xi_k / r_k^2$ with the displacement of \mathbf{p} is opposite to its change with the displacement of $\mathbf{q}_k = (q_x, q_y, q_z)$, *i.e.*

$$\frac{\partial}{\partial x} \frac{\cos \xi_k}{r_k^2} = - \frac{\partial}{\partial q_x} \frac{\cos \xi_k}{r_k^2}.$$

The derivative $\frac{\partial}{\partial q_x} \frac{\cos \xi_k}{r_k^2}$ can be computed with the assumption that \mathbf{p} lies at the origin because only the relative position of \mathbf{p} and \mathbf{q}_k matters (see Figure 4.3). Since $\cos \xi_k = -\frac{\mathbf{n}_k \cdot \mathbf{q}_k}{r_k}$ and $r_k = \|\mathbf{q}_k\| = \sqrt{q_x^2 + q_y^2 + q_z^2}$, we have,

$$\begin{aligned} \frac{\partial}{\partial q_x} \frac{\cos \xi_k}{r_k^2} &= - \frac{\partial}{\partial q_x} \frac{n_x q_x + n_y q_y + n_z q_z}{(q_x^2 + q_y^2 + q_z^2)^{3/2}} \\ &= - \frac{r_k n_x + 3q_x \cos \xi_k}{r_k^4}. \end{aligned} \quad (4.7)$$

Here $\mathbf{n}_k = (n_x, n_y, n_z)$ is the surface normal at \mathbf{q}_k , expressed in the local coordinate frame at \mathbf{p} . Combining this result with $\Delta A_k = \frac{2\pi}{N} \frac{r_k^2}{\cos \xi_k}$, we get

$$\frac{\partial \Omega_k}{\partial x} = \Delta A_k \frac{\partial}{\partial q_x} \frac{\cos \xi_k}{r_k^2} = \frac{2\pi}{N} \frac{r_k n_x + 3q_x \cos \xi_k}{r_k^2 \cos \xi_k}. \quad (4.8)$$

Plugging Equations (4.8) and (4.4) into (4.3) we get the complete formula for $\partial \lambda_l^m / \partial x$. The derivation of $\partial \lambda_l^m / \partial y$ is similar—only Equation (4.5) must be replaced by Equation (4.6). A code fragment evaluating one term of the sum in Equation (4.3) is given in Appendix B.

The differentiation-based method was also independently developed by Annen *et al.* [2]. A similar gradient calculation was also proposed by Zaninetti *et al.* [129]. Their method disregards the change of Ω_k and hence does not provide good results.

4.2.3 Irradiance Gradient

Note that both ways of computing the solid angle-based gradient can still be used if H_l^m is replaced by any other hemispherical function. We also do not rely on uniform hemisphere sampling: any probability density $p(\theta, \phi)$ can be used for sampling. The solid angle then becomes $\Omega_k = \frac{1}{Np(\theta_k, \phi_k)}$ instead of $\Omega_k = \frac{2\pi}{N}$ used for the uniform sampling.

As an example, we compute the irradiance gradient $\vec{\nabla}E$ with a cosine-weighted hemisphere sampling. $H_l^m(\theta, \phi)$ is replaced by $\cos\theta$, the probability density of sampling in direction (θ, ϕ) is $p(\theta, \phi) = \frac{\cos\theta}{\pi}$ and therefore $\Omega_k = \frac{\pi}{N \cos\theta_k}$. The resulting irradiance gradient formula for the differentiation-based method is

$$\frac{\partial E}{\partial x} = \sum_{k=1}^N L^i(\theta_k, \phi_k) \left(\frac{\partial \Omega_k}{\partial x} \cos\theta_k - \frac{\pi \sin\theta_k}{N \cos\theta_k} \frac{\partial \theta_k}{\partial x} \right)$$

with

$$\frac{\partial \Omega_k}{\partial x} = \frac{\pi}{N \cos\theta_k} \frac{r_k n_x + 3q_x \cos\xi_k}{r_k^2 \cos\xi_k}.$$

We implemented this irradiance gradient computation method and that of Ward and Heckbert [120], and we compared them on an example scene (Figure 4.4). The results were similar for both methods. Ward and Heckbert's method gives better results when occlusion changes significantly with displacement. Otherwise our method gives slightly smoother results.

4.2.4 Discussion

The method based on finite differences and the differentiation-based method are equivalent and their results are indistinguishable. The former is easier to implement since we do not need to evaluate the basis function derivatives. The latter is numerically more stable near edges and corners and, also, slightly faster to evaluate.

For the derivation of the solid angle-based gradient we assumed:

- The radiance $L^i(\theta_k, \phi_k)$ from the point \mathbf{q}_k incident at \mathbf{p} does not change with the displacement of \mathbf{p} .
- Visibility of ΔA_k , the small area around \mathbf{q}_k , does not change with the displacement of \mathbf{p} .

Although none of these assumptions is necessarily valid in all scenes, they are reasonable for small displacements. The first assumption may break down when the surface at the hit point \mathbf{q}_k is highly specular. It is a good idea to exclude the rays hitting such surfaces from the gradient computation. The second assumption breaks down if a substantial occlusion change takes place with the displacement. As a consequence, the computed gradient is usually underestimated. To remedy this problem, we have developed a new gradient computation method, described in the next section, that incorporates the changes in occlusion into the computed gradient.

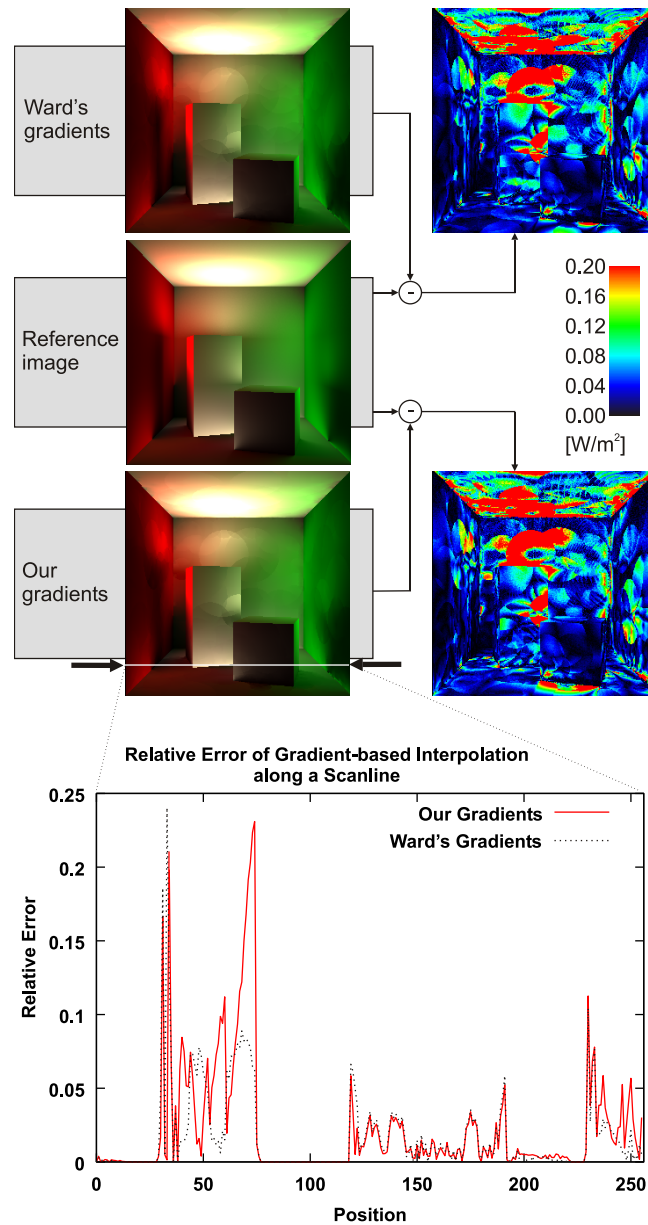


Figure 4.4: Comparison of irradiance gradient computation by our solid angle-based method and the method of Ward and Heckbert. The scene is a diffuse Cornell box; only first bounce indirect illumination is computed. The color-coded images show the difference between the gradient-enhanced interpolation and the reference solution (10,000 samples per hemisphere). RMS error of the images is 0.125 for Ward's method and 0.131 for our method. The graph shows relative error of the interpolation along a single scanline as compared to the reference solution. Ward's method gives better results when there are surfaces seen at very sharp grazing angles from the sampling point. Otherwise our method gives slightly lower error.

4.3 Stratification-Based Gradient

4.3.1 Radiance Gradient

For the computation of the stratification-based radiance gradient, we follow the idea of the irradiance gradient calculation from Ward and Heckbert [120], since it includes the occlusion changes into the gradient computation. However, we cannot use the irradiance gradient calculation directly, for two reasons. First, we distribute the samples uniformly over the hemisphere, whereas irradiance gradients are based on uniform *projected* solid angle sampling (*i.e.* the hemisphere is sampled more densely near the pole for irradiance gradient than for radiance gradient). Second, to compute the projection coefficients, we weight the incoming radiance samples by the basis functions H_l^m evaluated in the appropriate direction. No weighting is applied in the hemisphere sampling scheme used for irradiance gradients.

We reformulate Ward and Heckbert's irradiance gradient formula without assuming the uniform projected area sampling used in irradiance gradient computation. In our formulation, an arbitrary function can be used to weight the incoming radiance samples, which allows to project the incoming radiance onto an arbitrary hemispherical basis.

We assume stratified hemisphere sampling [22] for the gradient computation. With the stratification, the quadrature formula (3.1) for computing the projection coefficients of the incoming radiance becomes

$$\lambda_l^m = \frac{2\pi}{N \cdot M} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} L_{j,k}^i H_l^m(\theta_{j,k}, \phi_{j,k}), \quad (4.9)$$

where:

$L_{j,k}^i$ is the incoming radiance from the sample direction $(\theta_{j,k}, \phi_{j,k}) = \left(\arccos\left(1 - \frac{j+\zeta_j}{M}\right), 2\pi \frac{k+\zeta_k}{N} \right)$, and is computed by tracing a ray from the hemisphere center in the direction $(\theta_{j,k}, \phi_{j,k})$.

ζ_j, ζ_k are two uniformly distributed random numbers in $[0, 1)$, while $N \cdot M$ is the total number of sample directions and $N \approx 4M$.

The stratified sampling divides the hemisphere into cells of equal solid angle, or equal area on the unit hemisphere, given by

$$A_{j,k} = (\cos \theta_{j-} - \cos \theta_{j+})(\phi_{k+} - \phi_{k-}) = \frac{1}{M} \frac{2\pi}{N}. \quad (4.10)$$

The notation used in describing the geometry of the hemisphere division is illustrated in Figure 4.5, and summarized here:

(j, k) is the cell index.

$A_{j,k}$ is the cell area.

θ_{j-} is the elevation angle at the boundary between the current cell (j, k) and the previous cell $(j-1, k)$, $\theta_{j-} = \arccos\left(1 - \frac{j}{M}\right)$.

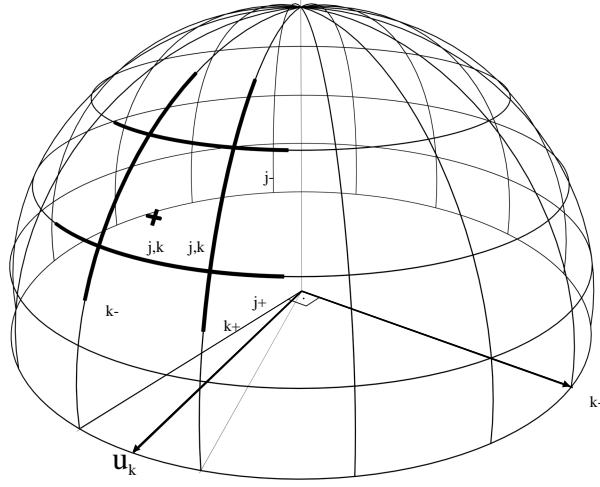


Figure 4.5: Uniform hemisphere subdivision for stratified radiance sampling. The hemisphere is subdivided into rectangular cells of the same area. One random direction is selected within each cell for tracing a secondary ray (depicted by the cross for the cell (j, k)).

θ_{j+} is the elevation angle at the boundary between the current cell (j, k) and the next cell $(j + 1, k)$, $\theta_{j+} = \arccos(1 - \frac{j+1}{M})$.

ϕ_{k-} is the azimuthal angle at the boundary between the current cell (j, k) and the previous cell $(j, k - 1)$, $\phi_{k-} = 2\pi \frac{k}{N}$.

ϕ_k is the azimuthal angle at the center of the current cell (j, k) , $\phi_k = 2\pi \frac{k+0.5}{N}$.

ϕ_{k+} is the azimuthal angle at the boundary between the current cell (j, k) and the next cell $(j, k + 1)$, $\phi_{k+} = 2\pi \frac{k+1}{N}$.

\hat{u}_k is the unit vector in direction $(\pi/2, \phi_k)$.

\hat{v}_{k-} is the unit vector in direction $(\pi/2, \phi_{k-} + \pi/2)$.

We estimate each cell's contribution to the gradient separately, and, then, sum the results up to get the total gradient estimate.

To estimate the gradient for the cell (j, k) , we observe how its area changes with respect to the displacement of the hemisphere center, along two near perpendicular vectors, \hat{u}_k and \hat{v}_{k-} , defined above. The displacement along \hat{u}_k causes a shift of the wall separating the considered cell (j, k) and its neighboring cell $(j - 1, k)$. The induced change of the cell area is

$$\begin{aligned} \nabla_{\hat{u}_k} A_{j,k} &= \nabla_{\hat{u}_k} \theta_{j-} \cdot \frac{\partial A_{j,k}}{\partial \theta_{j-}} \\ &= \frac{-\cos \theta_{j-}}{\min\{r_{j,k}, r_{j-1,k}\}} \cdot \sin \theta_{j-} (\phi_{k+} - \phi_{k-}) \\ &= \frac{2\pi \cos \theta_{j-} \sin \theta_{j-}}{N \min\{r_{j,k}, r_{j-1,k}\}}, \end{aligned}$$

where $\nabla_{\hat{u}_k}$ denotes the (scalar) directional derivative in the direction of \hat{u}_k , and $r_{j,k}$ is the distance from the hemisphere center to the closest surface in the sample direction $(\theta_{j,k}, \phi_{j,k})$. The derivative $\partial A_{j,k}/\partial \theta_{j_-} = \sin \theta_{j_-} (\phi_{k_+} - \phi_{k_-})$ follows directly from Equation (4.10).

Similarly, the displacement along \hat{v}_{k_-} causes a shift of the wall separating the considered cell (j, k) and its neighboring cell $(j, k-1)$. The induced change of the cell area is

$$\begin{aligned} \nabla_{\hat{v}_{k_-}} A_{j,k} &= \nabla_{\hat{v}_{k_-}} \phi_{k_-} \cdot \frac{\partial A_{j,k}}{\partial \phi_{k_-}} \\ &= \frac{-1}{\sin \theta_{j,k} \min\{r_{j,k}, r_{j,k-1}\}} \cdot (\cos \theta_{j_+} - \cos \theta_{j_-}) \\ &= \frac{1}{M \sin \theta_{j,k} \min\{r_{j,k}, r_{j,k-1}\}}. \end{aligned} \quad (4.11)$$

Here we used the equality $-(\cos \theta_{j_+} - \cos \theta_{j_-}) = 1/M$, which holds for uniform hemisphere sampling as a result of the way directions are generated in Equation (4.9).

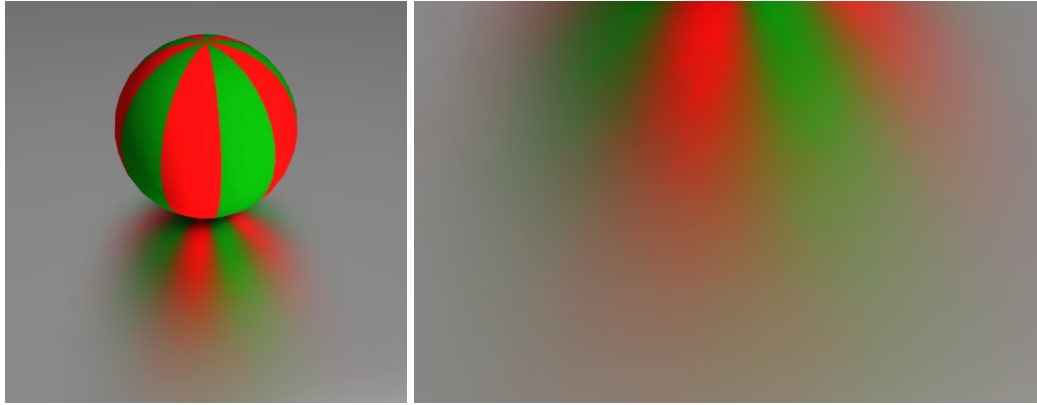
The change of incoming radiance arriving at the hemisphere center through the cell is given by interpolating the radiance from two neighboring cells with the area used as the blending factor:

$$\begin{aligned} \nabla_{\hat{u}_k} L_{j,k}^i &= \nabla_{\hat{u}_k} A_{j,k} (L_{j,k}^i - L_{j-1,k}^i), \\ \nabla_{\hat{v}_{k_-}} L_{j,k}^i &= \nabla_{\hat{v}_{k_-}} A_{j,k} (L_{j,k}^i - L_{j,k-1}^i). \end{aligned}$$

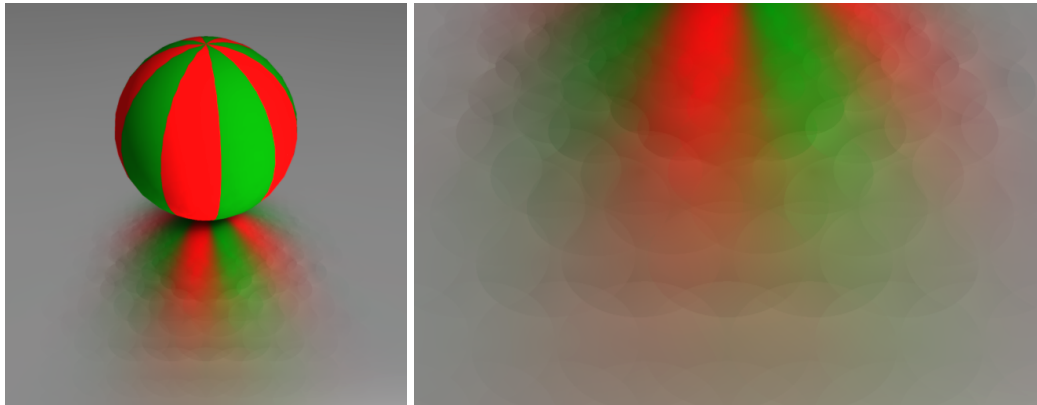
The final gradient formula for a coefficient λ_l^m is given by summing the marginal radiance changes over all hemisphere cells, weighted by the basis functions H_l^m :

$$\begin{aligned} \vec{\nabla} \lambda_l^m &= \sum_{k=0}^{N-1} \left[\hat{u}_k \frac{2\pi}{N} \sum_{j=1}^{M-1} \frac{\cos \theta_{j_-} \sin \theta_{j_-}}{\min\{r_{j,k}, r_{j-1,k}\}} (L_{j,k}^i - L_{j-1,k}^i) H_l^m(\theta_{j,k}, \phi_{j,k}) + \right. \\ &\quad \left. \hat{v}_{k_-} \frac{1}{M} \sum_{j=0}^{M-1} \frac{1}{\sin \theta_{j,k} \min\{r_{j,k}, r_{j,k-1}\}} (L_{j,k}^i - L_{j,k-1}^i) H_l^m(\theta_{j,k}, \phi_{j,k}) \right]. \end{aligned}$$

The change of occlusion is accounted for by taking the differences of the incoming radiance from the neighboring cells and using the $\min\{r_{j,k}, r_{j,k-1}\}$ and $\min\{r_{j,k}, r_{j-1,k}\}$ terms for estimating the relative wall movement between the current and the neighboring cell. The minimum of the two distances is important since “it is always the distance to the *closer* surface that determines rate of change in occlusion”, as pointed out by Ward and Heckbert [120]. On the other hand, the solid angle-based gradient treats each hemisphere cell independently from each other, based solely on the local geometrical information at the point where the cell’s sample ray hits another surface. Such an approach does not lend itself to estimating the changes in occlusion. To sum up, the stratification-based gradients are more accurate than the solid angle-based gradients and more general than the irradiance gradients of Ward and Heckbert [120].



Stratification-based gradients



Solid angle-based gradients

Figure 4.6: The solid angle-based gradients (bottom) do not properly handle significant change of occlusion in the sampled environment and leave visible interpolation artifacts. Taking the occlusion changes into account in the stratification-based gradients (top) results in a smoother indirect illumination interpolation on the glossy floor. The two images on the right are cut out from the two images on the left.

4.3.2 Irradiance Gradient

The radiance gradient formula above was derived for uniform hemisphere sampling with radiance samples weighted by the basis functions H_l^m . However, the same approach can be used to infer an *irradiance* gradient formula for the hemisphere sampling employed by Ward and Heckbert. Two changes have to be made. First, the weighting by $H_l^m(\theta_{j,k}, \phi_{j,k})$ is replaced by $\cos \theta_{j,k}$. Second, as a consequence of uniform *projected* solid angle sampling used in irradiance calculation, the equality $-(\cos \theta_{j_+} - \cos \theta_{j_-}) = 1/M$ does not hold anymore, since the definition of $\theta_{j,k}$, θ_{j_-} and θ_{j_+} is different from ours: $\theta_{j,k} = \arcsin \sqrt{\frac{j+\zeta_j}{M}}$, $\theta_{j_-} = \arcsin \sqrt{\frac{j}{M}}$, $\theta_{j_+} = \arcsin \sqrt{\frac{j+1}{M}}$. Instead of using the simplification $-(\cos \theta_{j_+} - \cos \theta_{j_-}) = 1/M$, the values $-(\cos \theta_{j_+} - \cos \theta_{j_-})$ have to be used

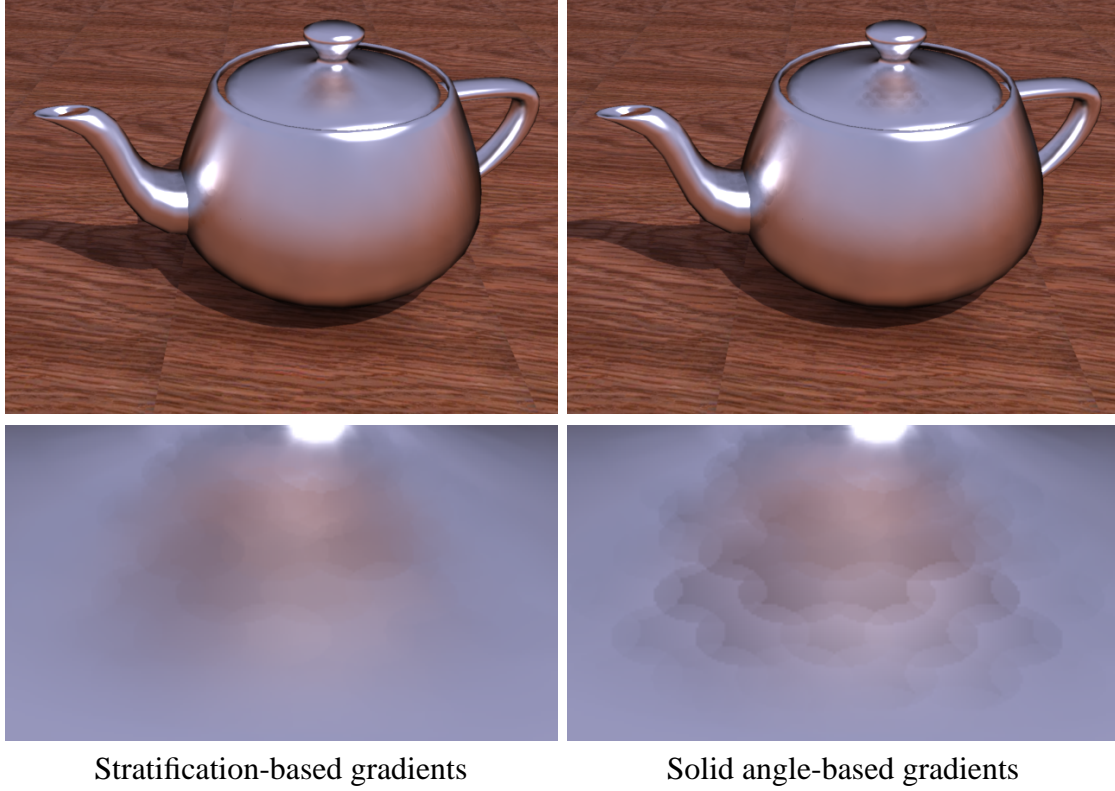


Figure 4.7: Indirect illumination on the teapot computed with radiance caching. The stratification-based gradients (on the left) provide smoother interpolation on the lid than the solid angle-based gradients (on the right).

directly in the final irradiance gradient formula:

$$\vec{\nabla}E = \sum_{k=0}^{N-1} \left[\hat{u}_k \frac{2\pi}{N} \sum_{j=1}^{M-1} \frac{\cos \theta_{j,k} \cos \theta_{j-} \sin \theta_{j-}}{\min\{r_{j,k}, r_{j-1,k}\}} (L_{j,k}^i - L_{j-1,k}^i) + \hat{v}_{k-} \sum_{j=0}^{M-1} \frac{\cos \theta_{j,k} (\cos \theta_{j-} - \cos \theta_{j+})}{\sin \theta_{j,k} \min\{r_{j,k}, r_{j,k-1}\}} (L_{j,k}^i - L_{j,k-1}^i) \right]$$

This formula yields numerically very similar results to the original formula of Ward and Heckbert [120] and the images generated with the two formulas are indistinguishable from each other. Performance of both methods is equal since all terms except from the hit distances $r_{j,k}$ and the incoming radiance samples $L_{j,k}^i$ can be precomputed.

4.4 Results

In this section we compare the results of the two gradient computation methods described above. The scene in Figure 4.6 is a pathological case for the solid angle-based gradient, but is handled correctly by the stratification-based gradient. The scene consists of a glossy

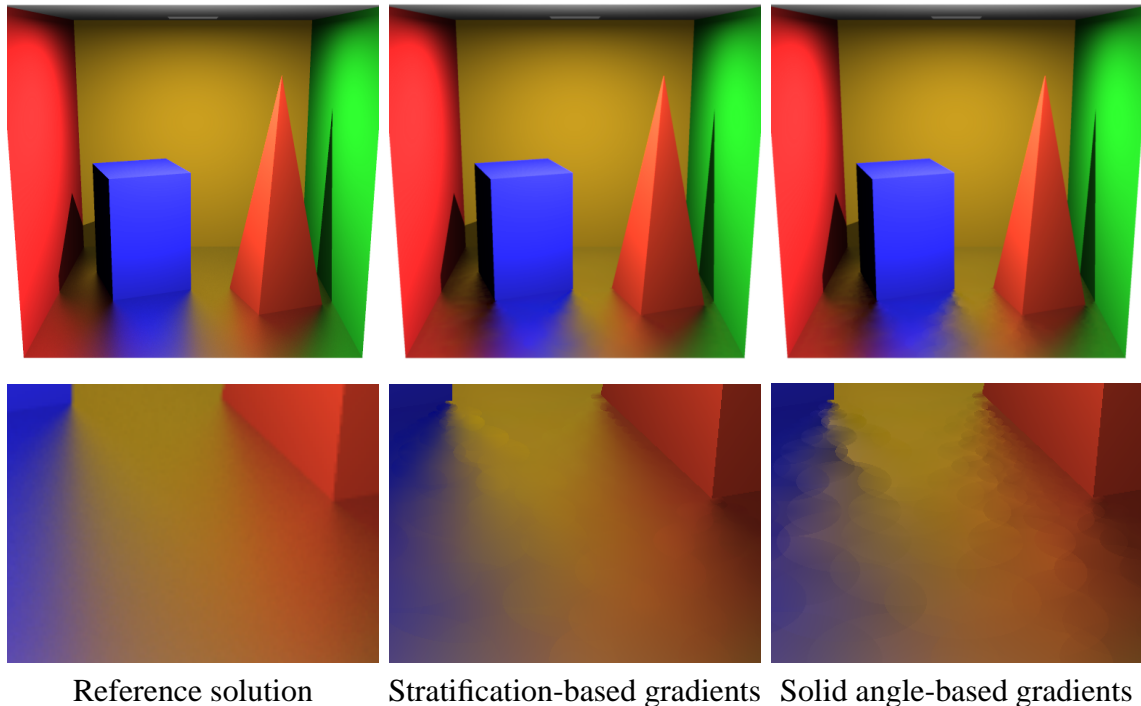


Figure 4.8: Indirect illumination on the glossy floor computed with path tracing (left) and with radiance caching using gradient-enhanced interpolation (middle and right). Compared to the solid angle-based gradients (on the right), the stratification-based gradients (in the middle) provide smoother interpolation near the occlusion changes caused by the box and the pyramid, and provide an image visually closer to the reference solution.

floor plane (isotropic Ward BRDF, $\rho_s = 0.9$, $\alpha = 0.15$, [118]) with a diffuse sphere on it, which is the only source of indirect illumination for the glossy floor. Consequently, there is a large change of occlusion with displacement on the part of the floor that reflects the silhouette of the sphere. Since the computation of the solid angle-based gradients assumes continuity of the reflected environment, this situation is not handled correctly and the interpolation shows discontinuities (Figure 4.6 at the bottom). On the other hand, the stratification-based gradients handle the occlusion change correctly and the sphere reflection on the floor is much smoother (Figure 4.6 at the top). Note that the interior of the sphere reflection is rendered near identically by both methods, since no occlusion changes prevent the solid angle-based gradients from performing well in that area.

Figures 4.7 and 4.8 illustrate, again, the same general observations. The solid angle-based gradients break down where changes of occlusion take place with displacement. In Figure 4.7, it is on the part of the lid that reflects the lid handle (shown in the blow-up); in Figure 4.8 it is on the floor reflecting the edges of the box and the pyramid (also shown in the blow-up). On the other hand, both methods perform similarly when no significant changes of occlusion take place. The teapot in Figure 4.7 is assigned the isotropic Ward BRDF [118] with $\rho_s = 0.9$, $\alpha = 0.15$, and the Cornell box floor in Figure 4.8 uses measured metallic BRDF fit with three Lafortune BRDF lobes [124, 60].

We can conclude from the example images, that the stratification-based gradient provides better image quality in scenes with important occlusion changes. The only advantage of the solid angle-based gradient is that no stratification of the hemisphere is required, which gives more freedom in choosing the hemisphere sampling scheme (*e.g.* using quasi Monte Carlo sampling [51]).

4.5 Summary

Two methods for computing the translational radiance gradient were described in this chapter. In both methods, the radiance gradient is estimated during hemisphere sampling from the incoming radiance samples, used for computing the projection coefficients of the incoming radiance. In the first method, the gradient is computed from the changes of mutual position and orientation of the surfaces contributing radiance to the hemisphere center, and the hemisphere center itself. The second method computes the gradient by summing up the changes of incoming radiance arriving at the hemisphere center through fixed cells on the hemisphere. On the example images rendered with both methods, we have shown that the latter method gives better results, since it takes occlusion changes into account. The computed gradients are stored in the radiance cache and used for smoother radiance interpolation.

Chapter 5

Spherical Harmonics Rotation

In this chapter we deal with rotation of functions represented by spherical harmonics. We propose a fast approximation for spherical harmonics rotation and we demonstrate how it can be applied in radiance caching. In addition, we show an application of the fast rotation in real-time rendering.

5.1 Introduction

In radiance caching, interpolating incoming radiance on curved surfaces involves aligning the local coordinate frames at the interpolation point and at the point where a cached radiance value was computed (see Figure 3.2). For this purpose, the cached incoming radiance has to be rotated—this was expressed in the radiance interpolation formula (3.2) by multiplying the coefficient vector Λ_i by the rotation matrix \mathbf{R}_i . This is a general rule: any function represented by spherical harmonics can be rotated by a linear transformation of the representation coefficients [33]. The existing methods for constructing the spherical harmonics rotation matrix [44, 45, 11, 50] are computationally intensive and make the interpolation in radiance caching slow.

To remedy this bottleneck, we propose an efficient approximation of spherical harmonics rotation, based on replacing a general spherical harmonics rotation matrix by its truncated Taylor expansion [56]. We show that our approximation is faster and has lower computational complexity in terms of spherical harmonics order than previous methods. We apply the proposed rotation in radiance caching and in real-time rendering with environment lighting. Unfortunately, our approximation is accurate only for small rotation angles. Yet, we show that this is not a serious restriction in our applications.

5.2 Background

This section provides background on spherical harmonics rotation and discusses previous rotation methods.

Problem Statement. Given a vector of spherical harmonics coefficients $\Lambda = [\lambda_l^m]$ representing a spherical function $L(\omega) = \sum_{l=0}^{n-1} \sum_{m=-l}^l \lambda_l^m Y_l^m(\omega)$, find a vector of coefficients $\Upsilon = [v_l^m]$ representing the rotated function $L(\mathcal{R}^{-1}(\omega)) = \sum_{l=0}^{n-1} \sum_{m=-l}^l v_l^m Y_l^m(\omega)$, where \mathcal{R} is the desired rotation.

Spherical harmonics are rotation invariant: rotation of any function represented by spherical harmonics of order n can be exactly represented by spherical harmonics of the same order. The rotation can be carried out as a linear transformation $\Upsilon = \mathbf{R}\Lambda$ with a block-sparse rotation matrix \mathbf{R} depicted in Figure 5.1. Note that coefficients between different spherical harmonics bands¹ do not interact. The problem is how to construct \mathbf{R} for a desired 3D rotation and order n . Different ways of solving this task are described in [33]. Our approach to spherical harmonics rotation, described in Section 5.3, avoids explicit construction of \mathbf{R} . We compare our approach with the methods of Ivanic and Ruedenberg [44, 45] and the $zxzxz$ -decomposition of Kautz *et al.* [50].

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 & 0 & 0 & 0 & \dots \\ \hline 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \dots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \dots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \dots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \dots \\ 0 & 0 & 0 & 0 & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \mathbf{X} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{R}^1 & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{R}^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Figure 5.1: Form of the spherical harmonics rotation matrix. (After [Green 2003]).

Ivanic and Ruedenberg [44, 45] construct \mathbf{R} recursively, starting from the sub-block \mathbf{R}^1 continuing over \mathbf{R}^2 up to \mathbf{R}^l for any given l . Elements of the sub-block \mathbf{R}^l are computed from elements of \mathbf{R}^{l-1} and \mathbf{R}^1 using rules summarized in [45, 33]. (See Figure 5.1 for the composition of matrix \mathbf{R} from the sub-blocks.)

A more efficient spherical harmonics rotation can be achieved with the method of Kautz *et al.* [50] that we call the $zxzxz$ -rotation. A general 3D rotation is first decomposed into zyz Euler angles (α, β, γ) . The rotation around y by angle β is then expressed as a rotation around x by $\pi/2$, a rotation around z by β and a rotation around x by $-\pi/2$. The angle of the two rotations around x is fixed; therefore, the rotation matrices for them can be pre-computed. The number of non-zero elements in the x rotation matrices is only a fourth of that of a general spherical harmonics rotation matrix. Additionally, a general rotation around z is very simple (see Appendix D), therefore the $zxzxz$ -rotation is more efficient than Ivanic and Ruedenberg's rotation.

Direct3D API [69] provides the `D3DXSHRotate()` call that rotates a function represented by spherical harmonics. The implementation is probably based on explicit for-

¹A *band* is formed by the harmonics of the same l -index. See Section 2.4.4.

mulas for the elements of the rotation matrix in terms of Euler angles [94], since it only works for orders up to $n = 6$. It is, furthermore, slower than the method of Ivanic and Ruedenberg [44, 45].

Choi *et al.*'s method [11] performs the rotation in complex domain and then converts the results back to real domain. (Our spherical harmonics and coefficient vectors are *real*.) According to [34], this procedure is slower than the method of Ivanic and Ruedenberg [44, 45].

None of the listed methods is fast enough to be performed each time interpolation is carried out in radiance caching. This is why we have developed a fast rotation approximation described in the next section.

5.3 Fast Rotation Approximation

This section describes the fast approximation of spherical harmonics rotation using a truncated Taylor expansion of the rotation matrix.

5.3.1 Rotation Approximation by Truncated Taylor Series

According to Euler's rotation theorem, any rotation in 3D space can be described by three angles. We decompose rotations using the zyz convention and express them as three subsequent rotations around z , y , and z axes by angles α , β , and γ , respectively, i.e. $\mathbf{R} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma)$.

Spherical harmonics rotation around z is simple and efficient (see Appendix D). It remains to find the rotation matrix $\mathbf{R}_y(\beta)$. We replace this matrix by its truncated Taylor expansion at $\beta = 0$:

$$\mathbf{R}_y(\beta) \approx \mathbf{I} + \beta \frac{d\mathbf{R}_y}{d\beta}(0) + \frac{\beta^2}{2} \frac{d^2\mathbf{R}_y}{d\beta^2}(0),$$

where \mathbf{I} is the identity matrix. Computation of the derivative matrices is described in Appendix C and they are depicted in Figures 5.2 and 5.3. The first derivative matrix $\frac{d\mathbf{R}_y}{d\beta}(0)$ has non-zero elements only on the superdiagonal and the subdiagonal. The elements of the second derivative matrix $\frac{d^2\mathbf{R}_y}{d\beta^2}(0)$ are non-zero only on the main diagonal and on the diagonal just below the subdiagonal and just above the superdiagonal. Therefore, the resulting rotation matrix approximation is very sparse. The rotation matrix $\mathbf{R}_y(\beta)$ does not have to be explicitly constructed at all because we know where the non-zero elements are.

In practice we use a "1.5th order" Taylor expansion, where any non-diagonal elements of the second derivative matrix are ignored. The "1.5th order" expansion is only slightly less accurate than the second order expansion, but incurs less computation. The C code in Figure 5.4 shows how simple the y rotation is using the "1.5th order" Taylor expansion. The arrays `dySubDiag` and `ddyDiag` contain the subdiagonal of $\frac{d\mathbf{R}_y}{d\beta}(0)$ and the diagonal of $\frac{d^2\mathbf{R}_y}{d\beta^2}(0)$, respectively. They are computed just once at the start-up of the application and remain constant throughout the run-time. The superdiagonal of $\frac{d\mathbf{R}_y}{d\beta}(0)$ does not have to

$$\frac{d\mathbf{R}_y}{d\beta}(0) = \begin{pmatrix} 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \frac{d\mathbf{R}_y^1}{d\beta}(0) & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \frac{d\mathbf{R}_y^2}{d\beta}(0) & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{d\mathbf{R}_y^3}{d\beta}(0) & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where

$$\frac{d\mathbf{R}_y^1}{d\beta}(0) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\frac{d\mathbf{R}_y^2}{d\beta}(0) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1.73 & 0 \\ 0 & 0 & 1.73 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\frac{d\mathbf{R}_y^3}{d\beta}(0) = \begin{pmatrix} 0 & 1.22 & 0 & 0 & 0 & 0 & 0 \\ -1.22 & 0 & 1.58 & 0 & 0 & 0 & 0 \\ 0 & -1.58 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2.45 & 0 & 0 \\ 0 & 0 & 0 & 2.45 & 0 & -1.58 & 0 \\ 0 & 0 & 0 & 0 & 1.58 & 0 & -1.22 \\ 0 & 0 & 0 & 0 & 0 & 1.22 & 0 \end{pmatrix}$$

Figure 5.2: First derivative of the y rotation matrix at $\beta = 0$. (Numbers are rounded to three significant digits.)

be stored, since the first derivative matrix is, like any other infinitesimal rotation matrix, antisymmetric [122].

All components for the full rotation $\mathbf{R} = \mathbf{R}_z(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma)$ are, now, available. The rotation proceeds as follows:

1. Decompose the desired rotation into the *zyz* Euler angles α , β and γ .
2. Rotate around *z* by α (see Appendix D).
3. Use `shRotYdiff15()` to rotate around *y* by β .
4. Rotate around *z* by γ .

It has to be emphasized that the described procedure only *approximates* spherical harmonics rotation and is usable only if the angle of rotation around *y* is small. An application using our approximation must make sure that this condition holds. The next section compares the approximation error for the first, the “1.5th”, and the second order Taylor expansions.

$$\frac{d^2 \mathbf{R}_y}{d\beta^2}(0) = \begin{pmatrix} 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \frac{d^2 \mathbf{R}_y^1}{d\beta^2}(0) & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \frac{d^2 \mathbf{R}_y^2}{d\beta^2}(0) & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{d^2 \mathbf{R}_y^3}{d\beta^2}(0) & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where

$$\frac{d^2 \mathbf{R}_y^1}{d\beta^2}(0) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$$\frac{d^2 \mathbf{R}_y^2}{d\beta^2}(0) = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -3 & 0 & 1.73 \\ 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 1.73 & 0 & -1 \end{pmatrix}$$

$$\frac{d^2 \mathbf{R}_y^3}{d\beta^2}(0) = \begin{pmatrix} -1.5 & 0 & 1.94 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 & 0 & 0 \\ 1.94 & 0 & -2.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 & 3.87 & 0 \\ 0 & 0 & 0 & 0 & -8.5 & 0 & 1.94 \\ 0 & 0 & 0 & 3.87 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & 1.94 & 0 & -1.5 \end{pmatrix}$$

Figure 5.3: Second derivative of the y rotation matrix at $\beta = 0$. (Numbers are rounded to three significant digits.)

5.3.2 Error Analysis

Let $\mathbf{R}_y(\beta)$ be the correct matrix for rotation around y by β and let $\mathbf{R}'_y(\beta)$ be our approximation. The error caused by rotating a coefficient vector Λ using our approximation is given by the following L_2 norm

$$\varepsilon(\beta) = \|\mathbf{R}_y(\beta)\Lambda - \mathbf{R}'_y(\beta)\Lambda\| = \|(\mathbf{R}_y(\beta) - \mathbf{R}'_y(\beta))\Lambda\| = \|\mathbf{D}(\beta)\Lambda\|$$

Maximum of the error $\varepsilon(\beta)$ over all unit length Λ is given by the L_2 norm of the matrix $\mathbf{D}(\beta)$, which is equal to the greatest singular value of $\mathbf{D}(\beta)$. Average error $\varepsilon(\beta)$ over all unit length Λ is given by the average singular value of $\mathbf{D}(\beta)$. Figure 5.5 shows the maximum and the average error $\varepsilon(\beta)$ and also the actual measured $\varepsilon(\beta)$ for a Phong lobe $\cos^7(\theta)$. Although the maximum error grows quite fast with β , the results for the Phong lobe show good accuracy up to $\beta = 25^\circ$.

5.3.3 Complexity

In this section we compare the complexity of Ivanic and Ruedenberg's rotation [44, 45] and the $zxzxz$ -rotation [50] with the complexity of our approximation. The complexities are expressed in terms of order n .

```

/** Rotate around y using the 1.5th order Taylor expansion
    @param beta      angle of rotation around y
 */
void shRotYdiff15(int order, float* dest, const float* src,
                  const float* dySubDiag, const float* ddyDiag, float beta)
{
    float bbeta = 0.5f*beta*beta;
    dest[0] = src[0];
    for(int i=1; i<order*order-1; i++) {
        dest[i] = src[i] * (1.0 + bbeta*ddyDiag[i]) +
            beta * (dySubDiag[i]*src[i-1] - dySubDiag[i+1]*src[i+1]);
    }
    dest[i] = src[i] * (1.0 + bbeta*ddyDiag[i]) +
        beta * dySubDiag[i] * src[i-1];
}

```

Figure 5.4: C code for the y rotation using the “1.5th order” Taylor expansion.

Ivanic and Ruedenberg’s method. The number of non-zero elements in a general spherical harmonics rotation matrix for order n is $N_{nz}(n) = \sum_{i=1}^n (2i-1)^2 = n(4n^2-1)/3$. Computation of each element of the matrix using Ivanic and Ruedenberg’s method [44, 45] is a constant-time operation, therefore the complexity of the spherical harmonics rotation matrix construction is $O(n^3)$. Complexity of transforming a spherical harmonics coefficient vector with the matrix is also $O(n^3)$.

zxzx-Rotation. One z -rotation involves $N_z(n) = 2n(n-1)$ multiplications; the cost of one x -rotation is $N_x(n) = \sum_{i=1}^n (i^2 - i + 1) = n(n^2 + 2)/3$. Rotation of one spherical harmonics vector with the $zxzx$ -rotation, thus, costs $3N_z(n) + 2N_x(n) = n(2n^2 + 18n - 14)/3 \in O(n^3)$ multiplications. This is only about a half of the number of multiplications needed for transforming a vector by a full spherical harmonics rotation matrix \mathbf{R} . Additionally, the rotation matrix itself does not have to be constructed in the $zxzx$ -rotation.

Our rotation approximation. There are $N_{dy}(n) = 5n^2$ multiplications in `rotYdiff15()`. The total cost of our rotation $2N_z(n) + N_{dy}(n) = 9n^2 - 4n \in O(n^2)$ is asymptotically lower than that of the previous methods. The advantage of our method in terms of speed becomes more pronounced as the order n increases, the downside being the lower accuracy for higher n .

5.4 Application in Radiance Caching

The proposed rotation approximation is used in radiance caching to align the local coordinate frames at an interpolation point \mathbf{p} and at a point \mathbf{p}_i where a cached radiance value was computed. Due to the interpolation criterion used in radiance caching, the normals

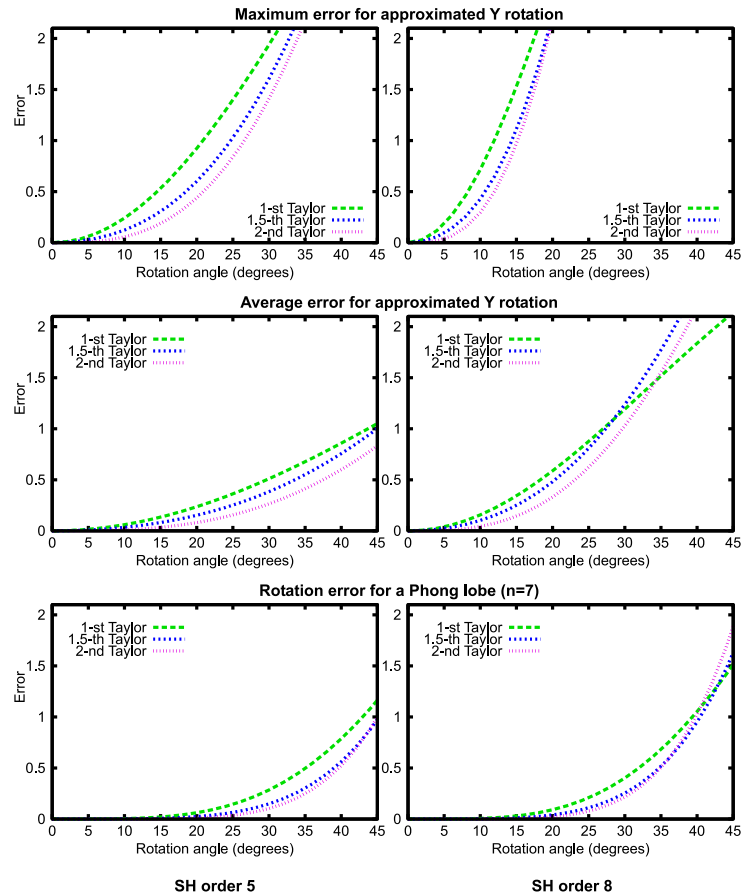


Figure 5.5: Rotation approximation error $\varepsilon(\beta)$ as a function of angle β for spherical harmonics of order 5 (left) and 8 (right). The plots in the first row show the maximum error for any unit length vector, the second row shows the average error over all unit length vectors and the third row shows the actual error for a Phong lobe $\cos^7(\theta)$. Error is expressed as the Euclidean distance between coefficient vectors. Each plot shows the error for the 1st, the “1.5th”, and the 2nd order Taylor expansion.

\mathbf{n}_i and \mathbf{n} at \mathbf{p}_i , respectively, and \mathbf{p} are always similar. (If they were not, the interpolation criterion would exclude the radiance value at \mathbf{p}_i from the interpolation at \mathbf{p} .) The angle between the normals \mathbf{n}_i and \mathbf{n} corresponds to the angle of rotation around the y -axis, β , in the zyz Euler decomposition. Hence, the rotation around y will always be small and our approximation can be safely used.

In radiance caching we also use the fast rotation approximation to rotate functions represented by *hemispherical harmonics*. The only change compared to spherical harmonics is the transformation of angle β ,

$$\beta' = \arccos(2 \cos \beta - 1).$$

Here β is the angle of rotation around y in the zyz decomposition of the desired rotation, and β' is the actual value used in our rotation procedure to multiply the derivative matrix. Reference [27] gives more details on hemispherical harmonics rotation.

Note that the approximate rotation cannot be used in radiance caching if the cached incoming radiance is represented in the global coordinate frame. In such a case, the rotation can be arbitrary, because of the necessity to align the global frame with the local frame at the interpolation point. Section 3.6 discusses the pros and cons of representing the incoming radiance in the global or the local frame.

5.4.1 Rotation Approximation Limiting Angle

To keep the error caused by the rotation approximation low, we use the approximation only for angles $\beta = \angle(\mathbf{n}, \mathbf{n}_i)$ smaller than a threshold β_{lim} . The accurate, but more costly, $zxzxz$ -rotation is used if $\beta > \beta_{\text{lim}}$. We set the limiting angle to $\beta_{\text{lim}} = 1.25a$, where a is the user specified maximum allowed interpolation error of radiance caching. This allows higher error in rotation if the user allows higher error in interpolation. Additionally, setting β_{lim} to a multiple of a has for consequence that the percentage of $zxzxz$ -rotations in a given scene is constant, no matter what value the user specifies for a .

If, on the other hand, we kept β_{lim} independent of the caching error a , increasing a would lead to a more frequent use of the $zxzxz$ -rotation. This would actually slow down the interpolation—an unexpected effect of increasing the allowed caching error.

Our rotation approximation and the $zxzxz$ -rotation may not meet in a visually continuous fashion at β_{lim} if a is high ($a > 0.3$ in our scenes). However, in such a case, the caching artifacts are more serious anyway and the rotation artifacts go unnoticed.

5.4.2 Derivation of the Limiting Angle

This section justifies setting the limiting angle β_{lim} to a multiple of the caching error a . We encourage the reader to review Section 3.3 on interpolation in radiance caching, as we will refer to the terms defined therein.

A radiance cache record i is included in interpolation at a point \mathbf{p} if $a > 1/w_i(\mathbf{p})$. Weight $w_i(\mathbf{p}) = \left(\frac{\|\mathbf{p} - \mathbf{p}_i\|}{R_i} + \sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i} \right)^{-1}$ is a function of the distance $\|\mathbf{p} - \mathbf{p}_i\|$ and the angle between normals $\beta = \angle(\mathbf{n}, \mathbf{n}_i)$. Consider a constant curvature surface with the osculating circle radius r . Then $\|\mathbf{p} - \mathbf{p}_i\| = 2r \sin \frac{\beta}{2}$ and therefore the weight is only a function of β , *i.e.*

$$1/w_i(\mathbf{p}) = f(\beta) = \frac{2r}{R_i} \sin \frac{\beta}{2} + \sqrt{1 - \cos \beta},$$

where R_i is the harmonic mean distance.

Our aim is to find for a given a and fixed r the value of β_{lim} such that for all accepted radiance cache records, the normal divergence is never more than β_{lim} . This can be done by inverting function f , which is unfortunately impossible to do in a closed form. Instead we take the first order Taylor expansion of f at $\beta = 0$, which is $f(\beta) \approx \beta(1/\sqrt{2} + \frac{r}{R_i})$, and we find $\beta_{\text{lim}} = a(1/\sqrt{2} + \frac{r}{R_i})^{-1}$. We choose $\frac{r}{R_i} = 0.1$ and get $\beta_{\text{lim}} = 1.25a$ (in degrees $\beta_{\text{lim}} = 70.1a$). The choice $\frac{r}{R_i} = 0.1$ means that all surfaces of curvature $10/R_i$ or smaller will be rendered without exceeding β_{lim} . To conclude, by setting $\beta_{\text{lim}} = 1.25a$, the slower

zxz -rotation will be used only on surfaces of curvature higher than $10/R_i$, regardless of the value of a .

5.4.3 Results

For the results in this section we used the “1.5th Taylor expansion” of the y rotation matrix (all non diagonal elements of the second derivative matrix are ignored). In Figure 5.6 we compare the results of radiance caching obtained by the correct and the approximated rotation. Instead of a side-by-side comparison, in which the results are visually indistinguishable, we show a color coded difference between the results of the two methods. Image areas exhibiting the maximum error are usually very curved. In those areas, visual artifacts, if any, are well-masked.

Figure 5.7 shows the rendering times for the flamingo and the sphere (Figure 5.6 left) with radiance caching. The rendering time only includes the interpolation from the cache. For spherical harmonics order $n = 6$, our method is 4 times faster than the zxz -rotation; for $n = 10$, it is 6 times faster.

5.5 Application in Real-Time Shading

This section shows how our fast rotation can be utilised in real-time rendering. We use the rotation for hardware-accelerated shading of normal-mapped objects illuminated by environment lighting.

5.5.1 Preliminaries

When using spherical harmonics or wavelets for real-time shading with environment lighting [50, 94, 63, 93, 115, 65, 72], one has to face the same alignment problem as in radiance caching: before evaluating the illumination integral, the environment lighting, represented in the global coordinate frame, must be rotated to match the local coordinate frame at each surface point.

In their real-time shading algorithm, Kautz *et al.* [50] perform the rotation of the environment lighting represented by spherical harmonics for each vertex, and report that the rotation is a bottleneck. Ng *et al.* [72] avoid the rotation by storing the BRDF multiple times, pre-rotated to the global frame for different surface normal directions. This approach wastes memory and prohibits the use of high frequency and/or anisotropic BRDFs. Pre-computed radiance transfer [94, 63, 93, 115, 65] avoids the rotation problem for smooth surfaces since the required rotation is included in the transfer matrix stored per vertex. However, if pre-computed radiance transfer is to be used on an object with the surface normal modulated by a texture, rotation would also be needed.

To demonstrate how our fast rotation approximation can be used in real-time shading, we extend the rendering method of Kautz *et al.* [50] to compute shading on objects with a per-pixel modulated normal. We start by giving a short overview of the original Kautz *et al.*'s method.

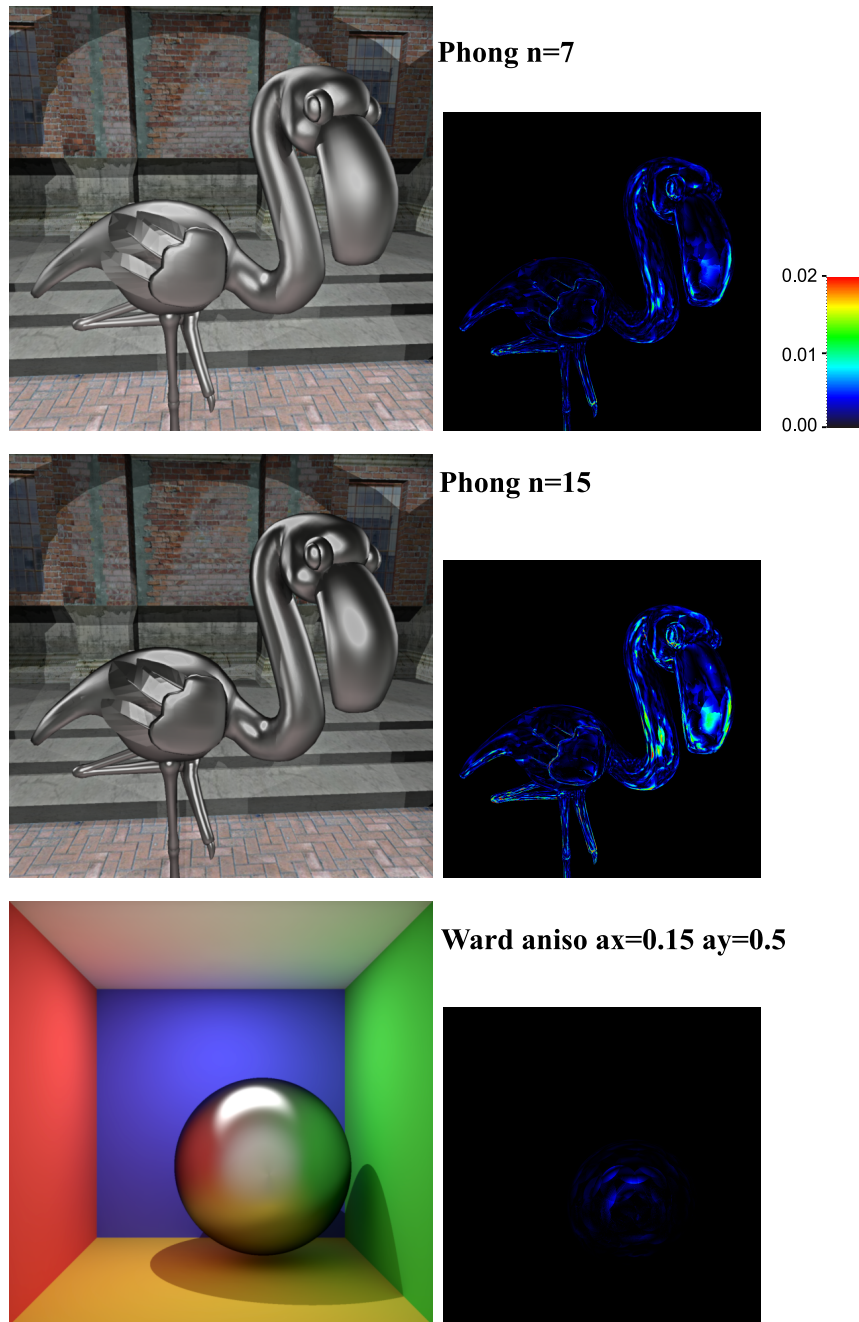


Figure 5.6: Left: Radiance caching renderings obtained with the approximate rotation. The flamingo is assigned a Phong BRDF with the exponent of 7 (top) or 15 (middle), and the sphere has an anisotropic Ward BRDF [118] with $\alpha_x = 0.15$, $\alpha_y = 0.5$. Spherical harmonics order $n = 10$ is used for all renderings. Right: Color coded difference between images with the approximate and the correct rotation, measured on a $[0, 1]$ RGB scale. The difference is below the visual threshold of 1% for most pixels.

Order	6		10	
	RT	TPR	RT	TPR
Flamingo				
Ignore	10.3 s	—	11.2 s	—
Our method	12.8 s	0.68 μs	16.9 s	1.54 μs
<i>zxzxz</i>	21.2 s	2.96 μ s	47.4 s	9.83 μ s
Ivanic	47.3 s	10.1 μ s	192 s	49.1 μ s
DirectX	76.4 s	17.9 μ s	—	—
Sphere				
Ignore	3.30 s	—	3.96 s	—
Our method	4.28 s	0.65 μs	6.13 s	1.44 μs
<i>zxzxz</i>	7.08 s	2.51 μ s	16.8 s	8.57 μ s
Ivanic	17.8 s	9.63 μ s	75.8 s	47.8 μ s
DirectX	30.3 s	17.9 μ s	—	—

Figure 5.7: Rendering times for the flamingo and sphere images in Figure 5.6 with radiance caching. The rendering time includes only interpolation from the cache. Various rotation methods are used for interpolation: Ignore (rotation is ignored), our method, *zxzxz*-rotation, the method of Ivanic and Ruedenberg and the DirectX rotation. ‘RT’ is the frame rendering time and ‘TPR’ is the time per rotation. There were $1,226,917 \times 3 = 3,680,751$ rotations for the flamingo and $501,420 \times 3 = 1,504,260$ rotations for the sphere.

5.5.2 Fast, Arbitrary BRDF Shading for Low-Frequency Lighting

Kautz *et al.* [50] uses graphics hardware to perform real-time shading of surfaces with arbitrary BRDFs illuminated by low-frequency environment lighting. By environment lighting we understand the light incident at an object from the whole sphere of directions. Shadowing is ignored and an assumption is made that the lighting is spatially invariant. Therefore, each point on an object surface receives the same illumination.

Spherical harmonics are used to represent both the environment lighting and BRDFs; the way BRDFs are stored is described in Section 2.5. The illumination integral can be evaluated by a dot product, similarly to the way it is done in radiance caching.

The rendering algorithm proceeds as follows:

1. [Per-vertex, CPU] Rotate the lighting coefficients to the local coordinate frame of vertex v_p . Send the rotated coefficient as vertex data to the GPU.
2. [Per-pixel, GPU] Look up the BRDF coefficients for the viewing direction transformed to the local coordinate frame.
3. [Per-pixel, GPU] Compute the dot product of the BRDF coefficients and the rotated lighting coefficients.

In this technique, the shading variance due to surface orientation is limited by the number of vertices in the mesh. A very fine mesh must be used for detailed shading. Our extension of the algorithm, presented in the next section removes this restriction.

5.5.3 Shading on Normal-Mapped Surfaces through a Per-Pixel Rotation

Normal maps modulate surface normals by a texture in order to represent small surface variations, for which an explicit geometry representation by a triangle mesh would be too bulky. Our extension of the shading algorithm of Kautz *et al.* allows to use normal mapping on an object illuminated by environment lighting, thereby decoupling the possible shading details from the number of mesh vertices.

We modify the original rendering algorithm in the following way (new steps are in italics):

1. [Per-vertex, CPU] Rotate the lighting coefficients to the local coordinate frame of vertex v_p . Send the rotated coefficient as vertex data to the GPU.
2. [Per-pixel, GPU] *Look up the normal map (normal map represents the modulation of the local coordinate frame at the pixel with respect to the frame given by the interpolated per-vertex normals).*
3. [Per-pixel, GPU] Look up the BRDF coefficients for the viewing direction transformed to the *modulated* local coordinate frame.
4. [Per-pixel, GPU] *Use our fast spherical harmonics rotation approximation to rotate the BRDF coefficients from the modulated local frame to the interpolated per-vertex local frame.*
5. [Per-pixel, GPU] Compute the dot product of local lighting and BRDF coefficients.

Modulation of surface normals by a normal map is usually limited to rather small angles; we can therefore safely use our rotation approximation. Thanks to the approximation simplicity, we were able to implement the per-pixel rotation in a pixel shader of the graphics hardware.

Our extension leads to a substantial improvement of visual quality as illustrated in Figures 5.8 and 5.9. It also allows using meshes with lower number of vertices than the original algorithm, which improves the overall rendering performance. Similar approach can also be used to augment visual richness of the spherical harmonics-based, pre-computed radiance transfer techniques [94, 50], by applying the per-pixel rotation on the transferred radiance.

To simplify normal mapping, one can ignore the per-pixel rotation (step 4) and use the normal map only to modulate the local frame for the BRDF lookup. Unlike our method, this simplified normal mapping generates flat looking surfaces and does not capture color variations on the surface bumps due to multicolored environment lighting.

5.5.4 Results

Figures 5.8 and 5.9 compare our results with the simplified normal mapping. We used spherical harmonics of order $n = 5$ (25 coefficients). The rotation approximation used the “1.5th order” Taylor expansion for bands $l = 1$ to $l = 3$ and the first order Taylor expansion

for band $l = 4$. Due to the limited pixel-shader instruction-count we had to use four passes to accommodate 25 coefficients. The frame rates for these images at resolution 800×600 were:

	Simplified	Ours
Vase (891 vertices)	58 fps	46 fps
Sphere (560 vertices)	65 fps	54 fps
Plane (25 vertices)	75 fps	56 fps

These figures were measured on a 2.2 GHz Pentium IV PC with ATI Radeon 9800 Pro GPU. The drop in the frame rate due to the rotation is more pronounced for low polygon count meshes, for which the rendering time is determined mostly by fragment processing.

Although those results are outdated by the power and capacities of more recent graphics hardware, the presented algorithm is useful as a demonstration of the possible uses of the fast rotation.

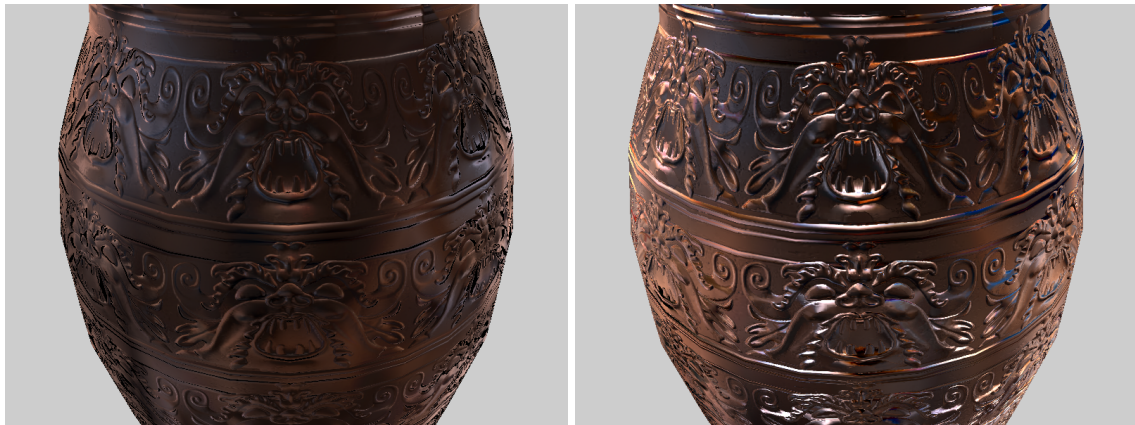


Figure 5.8: Detail of a normal mapped vase rendered with our spherical harmonics rotation (right) and with the simplified normal mapping (left). Normal mapping with our spherical harmonics rotation is more successful at conveying the shape approximated by the normal map. The vase is illuminated by the St Peter’s Basilica environment map [18]; the BRDF comes from a measurement of a brushed metal [124].

5.6 Summary

In this chapter we presented a fast algorithm for rotating functions represented by spherical harmonics. The main idea is to replace a general spherical harmonics rotation matrix by its truncated Taylor expansion. The algorithm has lower complexity and is faster than previous approaches. Although the rotation approximation is accurate only for small rotation angles, we have demonstrated its practical usefulness in radiance caching and real-time rendering. The rotation algorithm is simple enough to fit in the pixel shader of standard graphics hardware, which allows to apply the rotation on a per-pixel basis in real-time. We took advantage of this by shading normal mapped surfaces with arbitrary BRDFs illuminated by environment lighting.

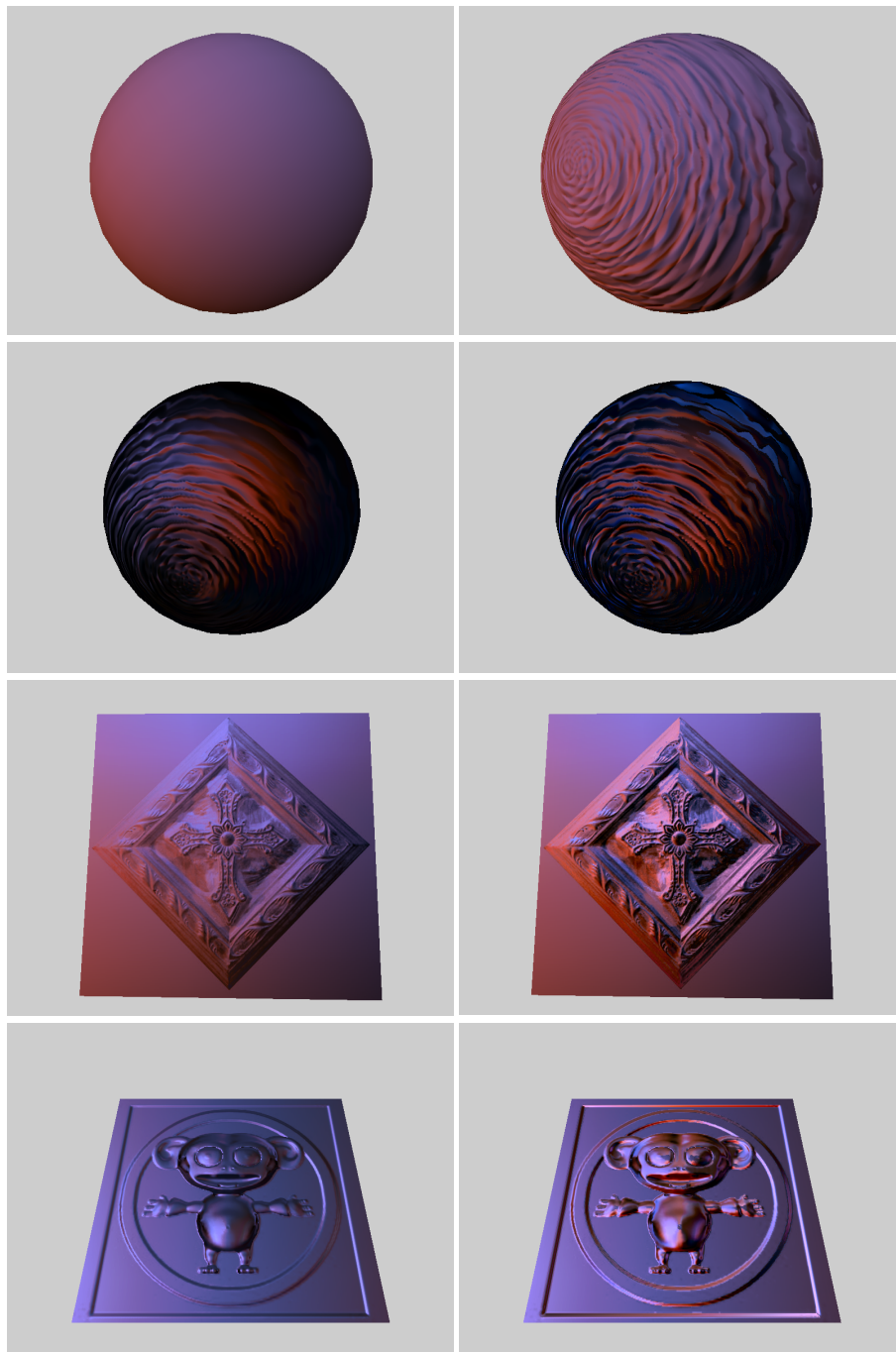


Figure 5.9: More results of normal mapping with our spherical harmonics rotation (right) compared to the simplified normal mapping (left). The BRDFs used were (from top to bottom) Lambertian, Phong, Ward isotropic, Ward anisotropic. The objects are illuminated by the Grace Cathedral environment map [18]. Note the color variations on the surface bumps captured by our method. For the Lambertian surface (top), the simplified normal mapping does not work since the BRDF is view independent. Our method is also more successful at revealing the effects of BRDF anisotropy (bottom).

Chapter 6

Adaptive Radiance Caching

In this chapter we describe an adaptive algorithm for guiding record density in radiance and irradiance caching. Record density is adapted to the rate of change of indirect illumination in order to avoid visible interpolation artifacts and produce smooth interpolated illumination. In addition, we discuss some implementation issues and propose heuristics for solving common problems of irradiance and radiance caching.

6.1 Introduction

To faithfully represent indirect illumination with only a sparse set of radiance or irradiance values, their density must be proportional to the rate of change of indirect illumination; otherwise, interpolation artifacts may appear. In irradiance caching, the upper bound on the illumination change is estimated based on the scene geometry. Even though the actual illumination conditions are disregarded in such an interpolation error metric, it generates a record density that gracefully follows the indirect illumination changes, and provides good image quality with a relatively low number of records.

In radiance caching, however, the task of estimating the rate of change of indirect illumination is more difficult. On glossy surfaces, not only the illumination characteristics, but also the sharpness of the surface BRDF and the viewing direction are important factors influencing the actual rate of change of indirect illumination over surfaces. It would be complicated to design an interpolation error criterion that takes all those factors into account, and the resulting formula is likely to be quite computationally involved. We believe that an adaptive algorithm that refines the record density based on a simple perceptual metric is more appropriate.

In this chapter we describe an adaptive algorithm for controlling record density in radiance and irradiance caching that we refer to as *adaptive caching*. It starts with an initial set of records and then refines their density where necessary to capture all the features of indirect illumination and to eliminate interpolation artifacts. The main source of artifacts in radiance and irradiance caching are visible discontinuities at a boundary of influence areas of two or more records. The error criterion guiding the adaptive refinement is designed to detect those discontinuities. The criterion is, thus, perceptual in the sense that it does not primarily care about the physical correctness of the image, but about the visi-

ble artifact elimination. This approach bears similarity with perceptually-based radiosity [28, 68], where a perceptual criterion is used to control adaptive patch refinement.

With adaptive caching, the record density is adapted to the actual illumination conditions better than with the original criterion used in radiance caching (see Section 3.3). If indirect illumination is simple, without sudden changes, adaptive caching generates fewer records and rendering is faster. Under complex indirect illumination, adaptive caching can generate more records than the original criterion would, but without the risk of compromising image quality by interpolation artifacts.

Adaptive caching is the most useful in rendering of glossy surfaces with localized indirect illumination features, where it locally increases the record density. With adaptive caching, the record density also automatically adapts to the BRDF sharpness. Additionally, record density is relaxed under strong direct illumination, where the indirect illumination quality does not matter much.

Besides the adaptive caching algorithm, we discuss a number of issues one must address in a practical implementation of radiance and irradiance caching. We propose a heuristics called *neighbor clamping* and show how it is useful for solving two problems of radiance and irradiance caching: missing small sources of indirect illumination and ray leaking.

6.2 Common Interpolation Formula

Since the adaptive algorithm described in this chapter applies both to radiance and irradiance caching, it is useful to use a common formulation for the two algorithms. To unify the interpolation formulas (2.2) and (3.2), we reformulate them by interpolating outgoing radiance contributions, instead of irradiance values E_i or coefficient vectors Λ_i :

$$L_{\mathcal{S}}^o(\mathbf{p}) = \frac{\sum_{i \in \mathcal{S}} L_i^o(\mathbf{p}) w_i(\mathbf{p})}{\sum_{i \in \mathcal{S}} w_i(\mathbf{p})}, \quad (6.1)$$

where $L_i^o(\mathbf{p})$ is the contribution of the i -th record to the outgoing radiance at the point \mathbf{p} . For irradiance caching at a point with diffuse reflectance $\rho(\mathbf{p})$, $L_i^o(\mathbf{p})$ is:

$$L_i^o(\mathbf{p}) = [E_i + (\mathbf{n}_i \times \mathbf{n}) \cdot \nabla_r E_i + (\mathbf{p} - \mathbf{p}_i) \cdot \nabla_t E_i] \rho(\mathbf{p}) / \pi.$$

For radiance caching, the outgoing radiance contribution is given by the dot product:

$$L_i^o(\mathbf{p}) = \left[\mathbf{R}_i \left(\Lambda_i + d_x \frac{\partial \Lambda_i}{\partial x} + d_y \frac{\partial \Lambda_i}{\partial y} \right) \right] \cdot C_{(\mathbf{p}, \omega_o)}.$$

In the rest of this chapter, we use the formulation (6.1), and the discussion applies to both irradiance and radiance caching, unless indicated otherwise.

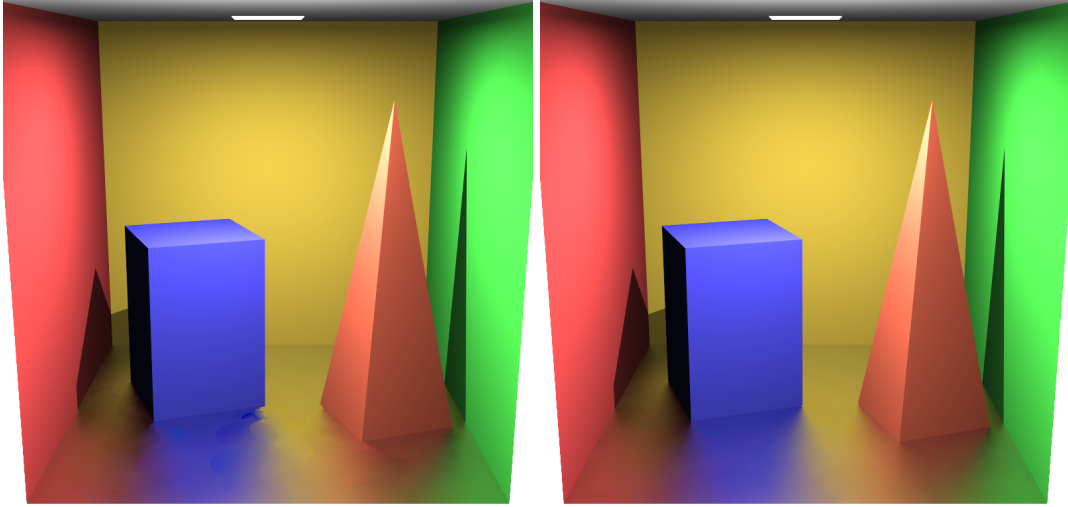


Figure 6.1: Left: Interpolation between two or more records can cause discontinuities (visible on the floor of the box). Right: Our algorithm detects the perceptible discontinuities and eliminates them.

6.3 Adaptive Caching

In this section we present the error criterion used to guide record density and an algorithm that uses this criterion to distribute records on object surfaces.

6.3.1 Error Criterion

Adaptive caching uses the original radiance caching interpolation scheme (Section 3.3) and an adaptive record density is achieved by *modulating the approximation error value, a , on a per-record basis*, based on our error criterion. So each record i now stores its own value of approximation error, a_i , instead of having one global value for the whole scene.

The goal of the adaptive record density control is to eliminate image artifacts caused by interpolation in places with high rate of change of indirect illumination. Those artifacts exhibit themselves as discontinuities at the boundary of influence areas of two or more records, see Figure 6.1. The influence area of a record is defined as $\{\mathbf{p}; w_i(\mathbf{p}) > 1/a\}$. Our error criterion reports a visible discontinuity if two records have a visibly different outgoing radiance contribution somewhere in the overlap of their influence areas. Whenever a visible discontinuity is detected, we decrease a record's value of a_i , effectively locally increasing the record density. In other words, we force the radiance contribution in the overlap areas to be indistinguishable from each other.

Consider a point \mathbf{p} at which the outgoing radiance is computed with the interpolation formula (6.1) from a set \mathcal{S} of at least two records. We say that the overlap of those records causes a discontinuity if a record i exists in \mathcal{S} , such that the outgoing radiance *without* this record's contribution, $L_{\mathcal{S}\setminus\{i\}}^o(\mathbf{p})$, is discernable from the outgoing radiance *with* this record's contribution, $L_{\mathcal{S}}^o(\mathbf{p})$.

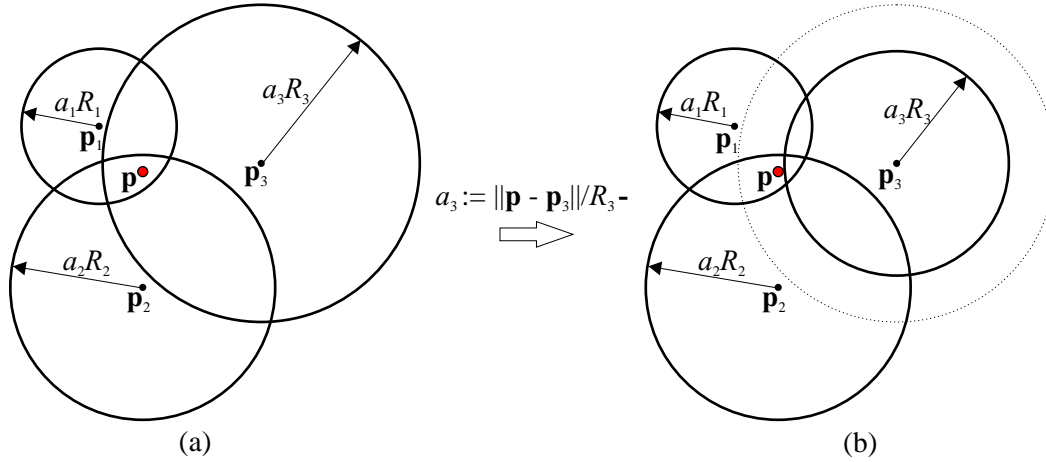


Figure 6.2: (a) If the outgoing radiance $L_{\mathcal{S}}^o(\mathbf{p})$, $\mathcal{S} = \{1, 2, 3\}$ at the point \mathbf{p} is visibly different from $L_{\mathcal{S} \setminus \{i\}}(\mathbf{p})$ for some $i \in \{1, 2, 3\}$, the overlap area is said to cause a discontinuity. (b) If a discontinuity is detected, the approximation error a_i of the record contributing with the minimum weight $w_i(\mathbf{p})$ is decreased (here record #3), so that point \mathbf{p} be outside the record's influence area.

If a discontinuity-causing record i is found, its contribution is excluded from the interpolated radiance at \mathbf{p} by decreasing its approximation error to $a_i := \frac{\|\mathbf{p} - \mathbf{p}_i\|}{R_i} - \varepsilon$, where ε is a very small number (see Figure 6.2). This assures that the condition $w_i(\mathbf{p}) > 1/a_i$ does not hold anymore and record i does not contribute to the interpolated radiance at \mathbf{p} . If there are more than one candidate for exclusion, we choose the one with the lowest weight $w_i(\mathbf{p})$, remove it from \mathcal{S} , and reiterate the process. Intuitively, the record with the lowest weight is relatively the farthest from point \mathbf{p} and should be excluded. The metric used to assess the discernibility of two radiance values is discussed in Section 6.4.

6.3.2 Full Convergence

The criterion described above can be used to compute indirect illumination at a point \mathbf{p} in a straightforward way:

$\mathcal{S} := \text{look_up_cache}(\mathbf{p}, \mathbf{n})$

If \mathcal{S} is empty, add new record.

If $|\mathcal{S}| = 1$, interpolate from the single record in \mathcal{S} .

If $|\mathcal{S}| > 1$, run the error criterion and interpolate from the records that remain in \mathcal{S} .

However, this approach does not work well, since a contribution from a record that will be shrunk later might have been mistakenly used earlier for interpolation. For a correct result, the record distribution must *fully converge* with respect to a set of shading points, before any record is used for interpolation at any of those points. We call *shading points* all the points with indirect illumination computed by means of radiance or irradiance caching.

Typically those are the surface points visible through image pixels either directly or after an ideal specular reflection or refraction.

Definition. Given a set of shading points \mathcal{P} , we say that the *record density has fully converged* for points in \mathcal{P} if:

1. each point in \mathcal{P} has at least one contribution from existing records, and
2. for all points having contributions from two or more records, the error criterion does not report a visible discontinuity.

Figure 6.3 describes an efficient procedure for attaining full convergence for a given set of shading points. The basic steps of the algorithm are motivated by the two following facts. First, after a new record is added to the cache, all the shading points in its influence area should be checked again by the error criterion. This is taken care of on line 11. Second, if the approximation error a_i of the record i is decreased, some shading points might find themselves without any valid contribution. New records should then be added to the cache. To this end, the algorithm works in passes (the outermost while loop of Algorithm 6.3). At the end of each pass (lines 16-18), we loop over all records whose a_i has changed in the current pass. For each of the changed records, all shading points within that record's influence area from the beginning of the pass are added to the queue to be checked by the error criterion. In order to find out what a record's influence area was at the beginning of the pass, each record stores its a_i value from the beginning of the pass. The whole algorithm finishes if no record has been changed in the last pass.

For the sake of efficiency, we store a list of record contributions with each shading point, so that the (ir)radiance cache does not have to be queried repetitively to find the contributing records for a shading point (line 7). Each contribution in the list stores a pointer to the contributing record, weight $w_i(\mathbf{p})$, outgoing radiance $L_i^o(\mathbf{p})$ (evaluated lazily as requested by the error criterion) and a change counter (described below).

Upon the reduction of a record's a_i value, some of the record's contributions may become invalid. It is thus necessary to check the validity of the entries in the contribution list (line 7) by re-evaluating the weight of a record that has changed since its contribution was last updated. To keep track of the contributions that require the validity check, we use *change counters*. Each record has a counter, incremented every time its a_i is reduced. The value of the change counter is copied to the contribution each time the validity of the contribution is verified on line 7. A contribution stays certainly valid as long as the record's counter and the contribution's counter are equal.

An alternative implementation of the convergence algorithm without the per-point contribution list uses a cache look-up on line 7 and leaves out line 10. It consumes less memory, is simpler to implement, but it is significantly slower.

Algorithm 6.3 ensures full convergence at its end. In our test, it always finished in at most eleven passes. The overhead due to this algorithm and due to the evaluation of the perceptual criterion rarely exceeds two seconds for a 800×800 image.

There are a few important details to make the algorithm efficient:

- pointQueue is implemented as a cyclic queue with a bit field to avoid multiple additions of one shading point.

Algorithm: Full convergence.

Input:

The set of shading points \mathcal{P} . Each shading point contains position, local coordinate frame, BRDF pointer, and a *list of record contributions*, empty at start.

Ensures:

All the shading points have at least one contribution from a cache record and for all points having contributions from two or more records, the error criterion does not report a visible discontinuity.

```

1: pointQueue.push( $\mathcal{P}$ )
2: while pointQueue not empty do
3:   // new pass starts here
4:   while pointQueue not empty do
5:     // for all points to be processed in this pass
6:      $p_s :=$  pointQueue.pop()
7:     remove invalid contributions from the shading point  $p_s$ 
8:     if zero contributions in  $p_s$  then
9:       add new record
10:      add a contribution to each shading point in the new record's influence
        area
11:      add each point in the new record's influence area to pointQueue
12:      else if more than one contribution in  $p_s$  then
13:        run error criterion on contributions in  $p_s$ 
14:      end if
15:    end while
16:    for all records changed in this pass do
17:      add to pointQueue all shading points within the record's influence area from
        the beginning of the pass
18:    end for
19:    // go to next pass
20: end while

```

Figure 6.3: Algorithm for attaining full convergence for a set of shading points.

- The initial value of a for a new record is copied from the nearest existing record in the cache and multiplied by 1.4. We found the multiplier 1.4 empirically. Lower values lead to unnecessarily many records after the convergence; higher values do not decrease the number of records but increase the overhead of the convergence algorithm.

- Shading points are stored in a k -D tree [47] to quickly locate the points in a record's influence area (lines 10 and 17).
- The contribution list for a shading point is a static array of length 6. We found that there are almost never more than six contributions for one shading point. Should this be the case, the excess contributions may safely be ignored (six is already enough). Using a dynamic linked list introduces a serious performance penalty because of the constant memory allocation and de-allocation.

This algorithm inverts the classical irradiance and radiance caching algorithm, where the basic operation is: ‘given a shading point, find all contributing records and interpolate’. On the other hand, the basic operation in our convergence algorithm is: ‘given a record, find all shading points it contributes to and add a contribution’. This is similar to radiance cache splatting [26] or reverse photon mapping [39].

6.4 Discernability Metric

This section describes the metric used in our error criterion to assess discernability of two radiance values. After some experimentation, we have opted for the simplest metric, the Weber law. Weber law says that the minimum perceptible change in a visual signal is given by the fixed fraction of the signal. Although not conservative under all circumstances, the threshold of 2% is widely used in computer graphics (*e.g.* [114]) and we used it in all our renderings.

The input of the perceptual metric are two radiance values L_1 and L_2 . L_1 is the interpolated outgoing radiance contribution from all overlapping records, plus direct illumination at \mathbf{p} ,

$$L_1 = L_{\mathcal{S}}^o(\mathbf{p}) + L^{o_{direct}}(\mathbf{p}).$$

L_2 does not contain the contribution from the tested record, i ,

$$L_2 = L_{\mathcal{S} \setminus \{i\}}^o(\mathbf{p}) + L^{o_{direct}}(\mathbf{p}).$$

The discernability metric based on the Weber law has the following form:

$$L_1 \text{ differs from } L_2 \quad \equiv \quad |Y(L_1) - Y(L_2)| > \Delta_{max},$$

where

$$\Delta_{max} = 0.02 Y(L_1) + \max\{\sigma(L_1), \sigma(L_2)\}.$$

$Y(\cdot)$ denotes the luminance channel of a tri-stimulus value. Estimates of the standard deviation $\sigma(L_1)$ and $\sigma(L_2)$ are added to compensate for the randomness of L_1 and L_2 , stemming from the fact that they are computed from quantities estimated by Monte Carlo hemisphere sampling. If the standard deviation was not added and only a few rays were used to sample a hemisphere when creating new cache records, it is likely that the irradiance or incoming radiance stored in neighboring records would be very different. In such a case, the criterion would constantly report visible discontinuities and lead to an excessive record density, which is undesirable.

To estimate the standard deviation of irradiance computed by Monte Carlo hemisphere sampling, an irradiance value E is computed from all the sample rays, and another value E' is computed from every second ray. Standard deviation of irradiance, estimated as $\sigma(E) = |E - E'|$ [19, 110], is then used to compute the standard deviation of the outgoing radiance. We use the standard deviation of irradiance even in radiance caching with good results.

We experimented with using the threshold elevation model of Ramasubramanian *et al.* [79] as the discernability metric. We decided to reject this approach since visible artifacts were produced near edges, where the model predicts a high elevation of the detectability threshold.

6.5 Practical Issues

This section describes useful heuristics and techniques that help making an implementation of radiance or irradiance caching practical. Although it contains rather engineering work than research results, we believe it is important to consider these issues in order to create a working implementation of either irradiance or radiance caching.

6.5.1 Adapting Record Density by Gradient Magnitude

The source code of Radiance lighting simulation system [119] contains a (never published) test that avoids a too low record density if the indirect illumination gradient is high. The test consists in comparing the upper bound of the translational irradiance gradient, derived from the “split sphere model” [121], with the actual irradiance gradient, estimated from the hemisphere sampling [120]. Should the magnitude of the actual irradiance gradient exceed the supposed upper bound, the actual gradient magnitude will be used in the error metric instead of the upper bound. Technically, this is done by clamping the value of R_i when a new record is being created:

$$\mathbf{if} \quad \frac{|\nabla_t E_i|}{E_i} > \frac{1}{R_i}, \quad \mathbf{then} \quad R_i := \left(\frac{|\nabla_t E_i|}{E_i} \right)^{-1}.$$

The use of this test substantially improves the image quality produced by irradiance caching.

We use a similar test in radiance caching:

$$\mathbf{if} \quad \frac{\sqrt{\|\frac{\partial \Lambda_i}{\partial x}\|^2 + \|\frac{\partial \Lambda_i}{\partial y}\|^2}}{\|\Lambda_i\|} > \frac{1}{R_i}, \quad \mathbf{then} \quad R_i := \left(\frac{\sqrt{\|\frac{\partial \Lambda_i}{\partial x}\|^2 + \|\frac{\partial \Lambda_i}{\partial y}\|^2}}{\|\Lambda_i\|} \right)^{-1}.$$

This test works aggregately on the whole coefficient vector which represents the result of the whole hemisphere sampling. Therefore, the test is neither view dependent nor it takes the BRDF sharpness into account. It helps to detect the most serious cases of high gradient change, but it leaves much space for improvement to our adaptive radiance caching.

6.5.2 Neighbor Clamping

In this section we propose *neighbor clamping*, a heuristics used to detect geometrically small sources of indirect illumination.

The radius of the i -th record's influence area $\{\mathbf{p}; w_i(\mathbf{p}) > 1/a_i\}$ on a flat surface is given by the product $a_i R_i$, which follows from the definition of $w_i(\mathbf{p})$. Here R_i is the harmonic mean of distances to visible surfaces from the record position, \mathbf{p}_i , computed from the ray lengths during hemisphere sampling. The closer the surrounding geometry, the smaller R_i , and the higher the record density. However, because the sample rays from which R_i is computed are only a random subset of all directions in the hemisphere, some geometry features are likely to be missed. The computed value of R_i is, then, too large and produces a too low record density. If the missed features are sources of strong indirect illumination, interpolation artifacts in the image result. Because of the randomness of ray directions, the features missed by one record might not be missed by another, which even amplifies the noticeability of the image artifacts.

Examples of features most commonly missed are steps of a staircase, or windowsills on a facade, which may be too small to keep the harmonic mean of the ray lengths low, yet too important in terms of indirect illumination to be missed. The left column of Figure 6.5 shows the artifacts due to the insufficient record density around a geometry feature (the steps).

Our goal is to make sure that no relevant geometry features are missed. Additionally, a geometry feature detected by the rays from one record should be detected by all nearby records, too.

Tabellion and Lamorlette [103] address this problem by computing the R -value of a record as the minimum of the ray lengths during the hemisphere sampling, instead of taking the harmonic mean. This increases the probability of detecting small geometry features and indeed, the step-like features are missed very rarely. However, after some experimentation we decided not to use the minimum ray length for the following reasons. Firstly, using the minimum ray length is overly sensitive to very small features that hardly have any importance for indirect illumination. Secondly, and more importantly, the minimum ray length on concave objects is very small, as the rays near the equator are usually extremely short. This produces an excessive record density on those objects. Tabellion solves this problem by excluding rays near the equator, hitting other surfaces at grazing angles, from the computation of the minimum distance [102]. But even with this modification, we found it difficult to obtain a decent distribution of records on concave surfaces of varying curvature by using the minimum ray length for computing R_i .

We have thus decided to compute R_i as the harmonic mean of the ray lengths, and to prevent random missing of geometry features by imposing an additional constraint on the difference of the R -values of neighboring records. The constraint stems from a basic observation on geometrical coherence in a scene. Consider a record j at position \mathbf{p}_j located on a floor at a distance d from a step (see Figure 6.4). Now consider another record at position \mathbf{p}_i . By triangle inequality, the maximum possible distance of \mathbf{p}_i from the step, in terms of d and the distance between the two records is $d + \|\mathbf{p}_i - \mathbf{p}_j\|$. Motivated

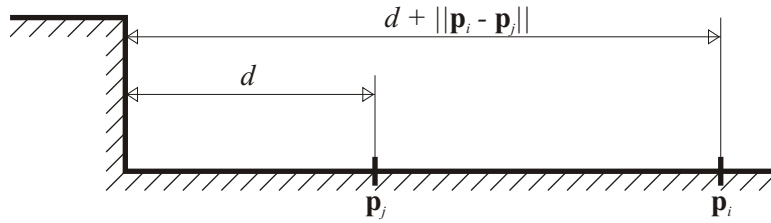


Figure 6.4: The maximum possible difference of the two records' distance from the step is given by their mutual distance.

by this observation, we never allow the R -values of two nearby records to differ by more than their mutual distance.

Technically, when a new record i is being added to the cache, we locate all nearby records j and clamp the new record's R_i value to $R_i := \min\{R_i, R_j + \|\mathbf{p}_i - \mathbf{p}_j\|\}$. After that we similarly clamp the nearby records' R -values by the new record's R_i -value. A consequence of this clamping is that a too large R -value of a record, caused by missing a geometrical feature, is clamped down by the R -value of some of the neighboring records that did not miss that feature.

This heuristic, which we call *neighbor clamping*, is fully justified when using the minimum ray length for computing R_i , but it gives very good results even for the harmonic mean. The features are almost never missed, and the overall distribution of records in the scene behaves well. Furthermore, we do not experience the problem with excessive record density on concave objects. Figure 6.5 demonstrates how neighbor clamping (right column) helps to detect small, step-like geometry features. Without neighbor clamping (left column), those features are often missed and artifacts appear in the image (see the detail of the stairs). Both images were rendered using the same number of records (7750). Without neighbor clamping, at least 20,000 records were required to get rid of the image artifacts on the stairs.

Neighbor clamping is useful especially for irradiance caching, where the chance of missing features is quite high because of the relatively low number of rays cast per hemisphere. We have not experienced problems with missing features in radiance caching, because the number of rays cast per hemisphere is usually higher than for irradiance caching. Therefore, we use the neighbor clamping heuristics in all test scenes for irradiance caching, but not for radiance caching. In the next section we show one more situation in which neighbor clamping proves to be very useful.

6.5.3 Ray Leaking

Irradiance and radiance caching are quite sensitive to imperfections in scene modeling; a typical example in which they break down is an inaccurate connection of adjacent edges of two polygons. This may be produced *e.g.* by an insufficient number of significant digits when a scene is exported to a text file.

Consider the situation in Figure 6.6, where there is a small gap between the floor polygon and the wall polygon. If a primary ray hits this gap, its intersection with the floor

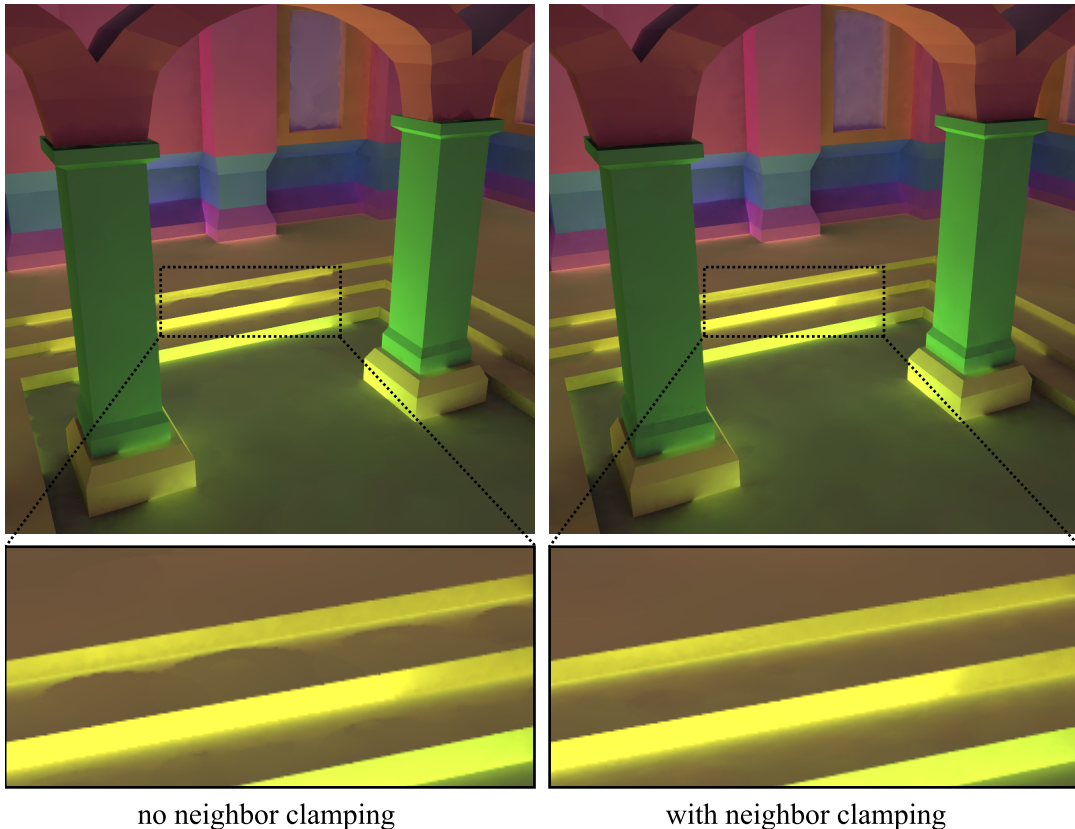


Figure 6.5: Without our neighbor clamping, small, step-like geometry features are often missed. Missing geometry features, which are strong sources of indirect illumination, causes image artifacts (see the detail of the steps in the left column). Neighbor clamping helps to detect the small geometry features and suppresses the image artifacts (see the right column). The images show only indirect illumination.

polygon can be found *behind* the wall polygon. As a consequence, rays that are supposed to hit the wall now *leak* either to the neighboring room or to infinity. The outcome of such an event is quite disastrous:

- The computed irradiance or incoming radiance is wrong.
- The harmonic mean of the ray lengths is much greater than it should be; therefore, the wrong (ir)radiance is extrapolated on a very large area.

The resulting image artifacts are shown in Figure 6.8 on the left. The following paragraphs discuss two possible solutions to this problem, among which the more successful is to use neighbor clamping.

The first solution is to place the origin of sampling rays away from polygon edges. Tabellion places the origin approximately to the centre of a micropolygon [103, 102]. Since this is not applicable in a generic ray tracer, our approach is to shift the origin perpendicularly from the nearest polygon edge to a fixed distance. We never shift the origin by more than half a projected pixel size. We also never shift the origin farther than to the

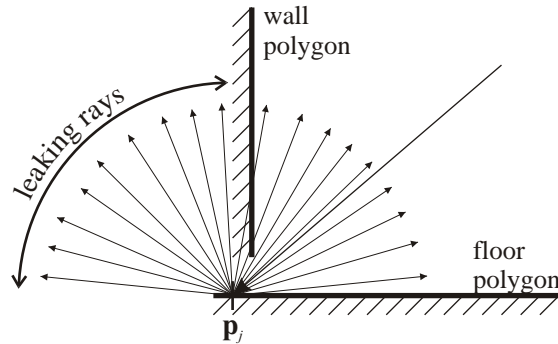


Figure 6.6: Inaccurate connection of polygons may result in ray leaking.

bisection line between the nearest edge and its neighboring edges (see Figure 6.7). The ray origin shifting suppresses ray leaking in some case, but is not completely dependable.

A better solution consists in using neighbor clamping. Records suffering from ray leaking have disproportionately bigger R -value than their neighbors not having this problem, thereby breaking the assumption of geometrical coherence used to derive the neighbor clamping heuristics. Therefore, ray leaking is reliably detected by the use of neighbor clamping and its consequences are alleviated by the reduction of the erroneous R -value. Figure 6.8 on the right shows how neighbor clamping detects and suppresses the effects of ray leaking. Besides small feature detection, ray leaking detection is another purpose that neighbor clamping serves well.

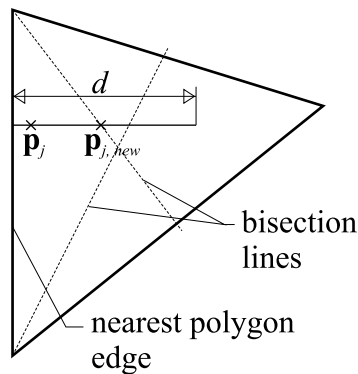


Figure 6.7: Origin of hemisphere sampling is offset from the nearest polygon edge to some fixed distance d , but never farther than to the bisection line between the nearest edge and its neighboring edges.

6.6 Results and Discussion

Adaptive Radiance Caching. Results of adaptive radiance caching are shown in Figures 6.11, 6.12, 6.13 and 6.14; the rendering statistics, measured on a PC with Intel Pentium M 1.5GHz and 512MB RAM, are given in Figure 6.9. Notice in the table that the

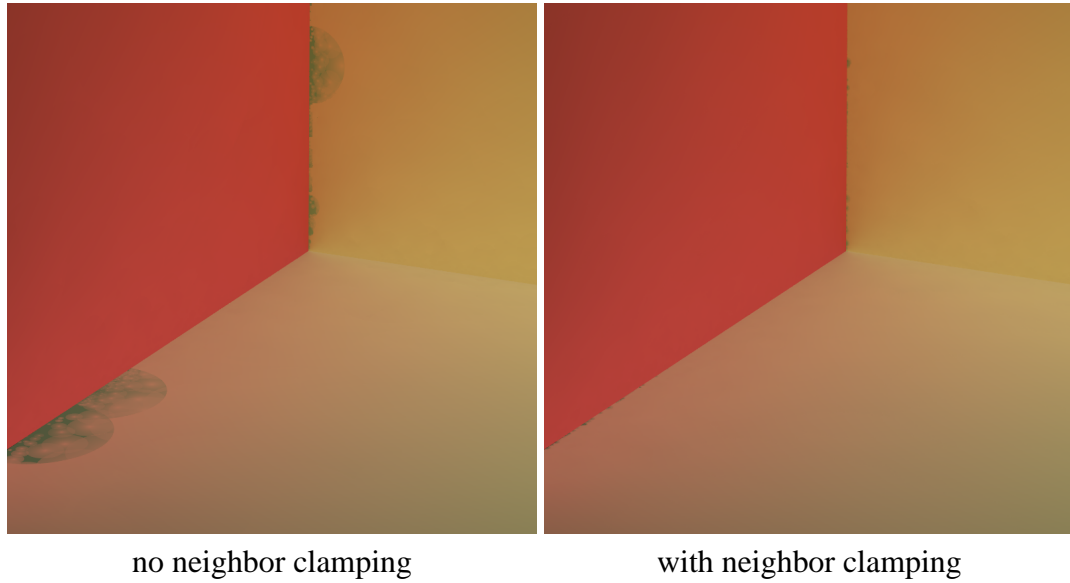


Figure 6.8: Serious image artifacts caused by ray leaking (left) are significantly reduced by using the proposed neighbor clamping heuristics (right).

	Cornell Box 800 × 800		Teapot 800 × 800		Walt Disney Hall 1280 × 800	
	adaptive	non-adaptive ($a = 0.23$)	adaptive	non-adaptive ($a = 0.15$)	adaptive	non-adaptive ($a = 0.18$)
# records	990	1010	1589	1695	4078	4144
Cache filling [s]	68.6	68.7	94.3	94.8	195	196
Overhead [s]	2.18	1.6	4.18	2.81	11.2	9.34

Figure 6.9: Rendering statistics for adaptive radiance caching example scenes. *Cache filling* is the time spent on creating radiance cache records (hemisphere sampling). *Overhead* is the time spent on interpolating from cache records and evaluating the perceptual error criterion. Notice that the extra overhead introduced by adaptive caching is very small.

extra overhead introduced by adaptive caching is very small. For each scene, we rendered one image with adaptive caching and another without adaptive caching using the same number of records. We compare the quality of the two renderings. Adaptation of record density by gradient magnitude, described in Section 6.5.1, is used for both irradiance and radiance caching and both adaptive and non-adaptive settings. Additionally, the neighbor clamping heuristics, described in Section 6.5.2, is used for irradiance caching. Apart from the rendered images, the figures show the map of the approximation error a and the record locations. Notice how the approximation error and the record density adapt to the rate of change of indirect illumination with our adaptive caching.

We do not provide a comparison with a reference solution because our error metric promotes smooth, artifact-free images, but does not ensure any error bound for the image. Our goal is to generate images that look good and are plausible, but not necessarily physically correct.

The floor of the box in Figure 6.11 uses the Lafortune BRDF [60] fitted to measured metallic BRDF data [124]. Only direct illumination and first bounce indirect illumination for the floor is used. Non-adaptive radiance caching has problems capturing the change of indirect illumination caused by the edges of the box and the pyramid. On the other hand, the record density behind the two objects is unnecessarily high. Adaptive caching produces a record density that reproduces all the details of the indirect illumination. To obtain an artifact-free image with non-adaptive caching, at least 3,300 records were required, which is 3 times more than with adaptive caching.

The scenes in Figures 6.12 and 6.13 use four indirect bounces and both irradiance and radiance caching. The teapot in Figure 6.12 has a Phong BRDF with the exponent of 16. Non-adaptive caching runs into troubles capturing the reflection of the spout on the teapot body, which adaptive caching renders faithfully by locally elevating record density. In this scene, adaptive caching also increases record density on the teapot rim, where there are no noticeable artifacts even with the non-adaptive algorithm. This suggests that a better perceptual metric, which possibly takes the projected record size into account, would probably allow even lower number of records without deteriorating image quality. The teapot scene required at least 2,800 records for an artifact-free image without adaptive caching, that is 1.7 times more than with adaptive caching.

In the Walt Disney Hall scene (Figures 6.13 and 6.14), the adaptive record density is useful not only for radiance caching (see cutouts 1 and 2), but also for irradiance caching. Adaptive irradiance caching nicely captures the caustics formed on the floor at the entrance to the building (cutout 3), and also delivers a better image quality on the staircase, where the non-adaptive caching leaves some smudges. Notice the visible discontinuity on the metal wall at the very right of the image. This is actually a discontinuity of first order, caused by the inversion of the indirect illumination gradient and by the change of the tint from brownish to bluish. Our current discernability metric does not capture those aspects of discontinuity perception. For an image quality similar to that produced by adaptive caching, the non-adaptive caching required as many as 29,000 records (7 times more than adaptive caching). 17,500 records were sufficient for non-adaptive irradiance caching, which is only 1.3 times more than with adaptive irradiance caching.

Figure 6.15 demonstrates how record density automatically adapts to the sharpness of a BRDF. The floor is assigned the isotropic Ward-Dür BRDF [20, 118] with $\rho_d = 0$, $\rho_s = 0.9$ and $\alpha = 0.60, 0.30$ and 0.15 (from left to right). With increasing BRDF sharpness, the record density automatically grows to reproduce the sharper features of the indirect illumination, without requiring any user intervention. The number of records is 335, 516, and 993, respectively.

Discussion of the Radiance Caching Results. Figures 6.11, 6.12 and 6.13 show that adaptive radiance caching mostly pays off if there is an indirect illumination feature that requires a locally high record density. In the non-adaptive caching, the only solution to render such a feature is to increase the record density in the whole image by manually altering the global value of the approximation error a . This is inefficient, since most of the image already has a high enough record density. We would like to point out that on

glossy surfaces, the localized features almost always appear; therefore, adaptive radiance caching is very profitable.

Adaptive Irradiance Caching Results and Discussion. To test our adaptive algorithm with irradiance caching, we rendered a number of scenes, of which two are shown in Figure 6.10. We have found that the number of records and the rendering times for both adaptive and non-adaptive irradiance caching are very similar. In our opinion, this is caused by the fact that localized indirect illumination features rarely appear on diffuse surfaces¹ and thus the adaptive record density does not bring as much profit to irradiance caching as it does to radiance caching.



Figure 6.10: Scenes used to test adaptive irradiance caching.

The irradiance caching results demonstrate that extending the combination of the original Ward’s error criterion [121] and the adaptation by the gradient magnitude (Section 6.5.1) with the newly proposed neighbor clamping heuristics (Section 6.5.2) results in a robust, dependable interpolation error criterion. Within the irradiance caching framework, it would probably be difficult to find an error criterion that computes the indirect illumination with a significantly lower number of records, without missing any important indirect illumination features.

6.7 Summary

In this chapter we described an adaptive algorithm for controlling record density in radiance and irradiance caching. The algorithm is based on detecting visible discontinuities caused by inadequate sampling of indirect illumination. If a discontinuity is detected, record density is locally increased in order to produce smooth, artifact free images. As a result, record density automatically adapts to the complexity of indirect illumination—fewer records are used for smoothly varying illumination and more records are generated

¹A notable exception are caustics, usually handled by other means than irradiance caching.

in areas of abrupt illumination changes. Moreover, the record density adapts to the BRDF sharpness and the viewing direction. The adaptation is automatic, without the need for user intervention. We demonstrated on a number of scenes that the image quality produced by adaptive caching is superior to that generated by the error criterion of radiance caching.

We also proposed the neighbor clamping heuristics, which significantly reduces the probability of missing important geometry features. Neighbor clamping also helps to detect, and alleviate the consequences of, ray leaking—a serious problem of irradiance and radiance caching.

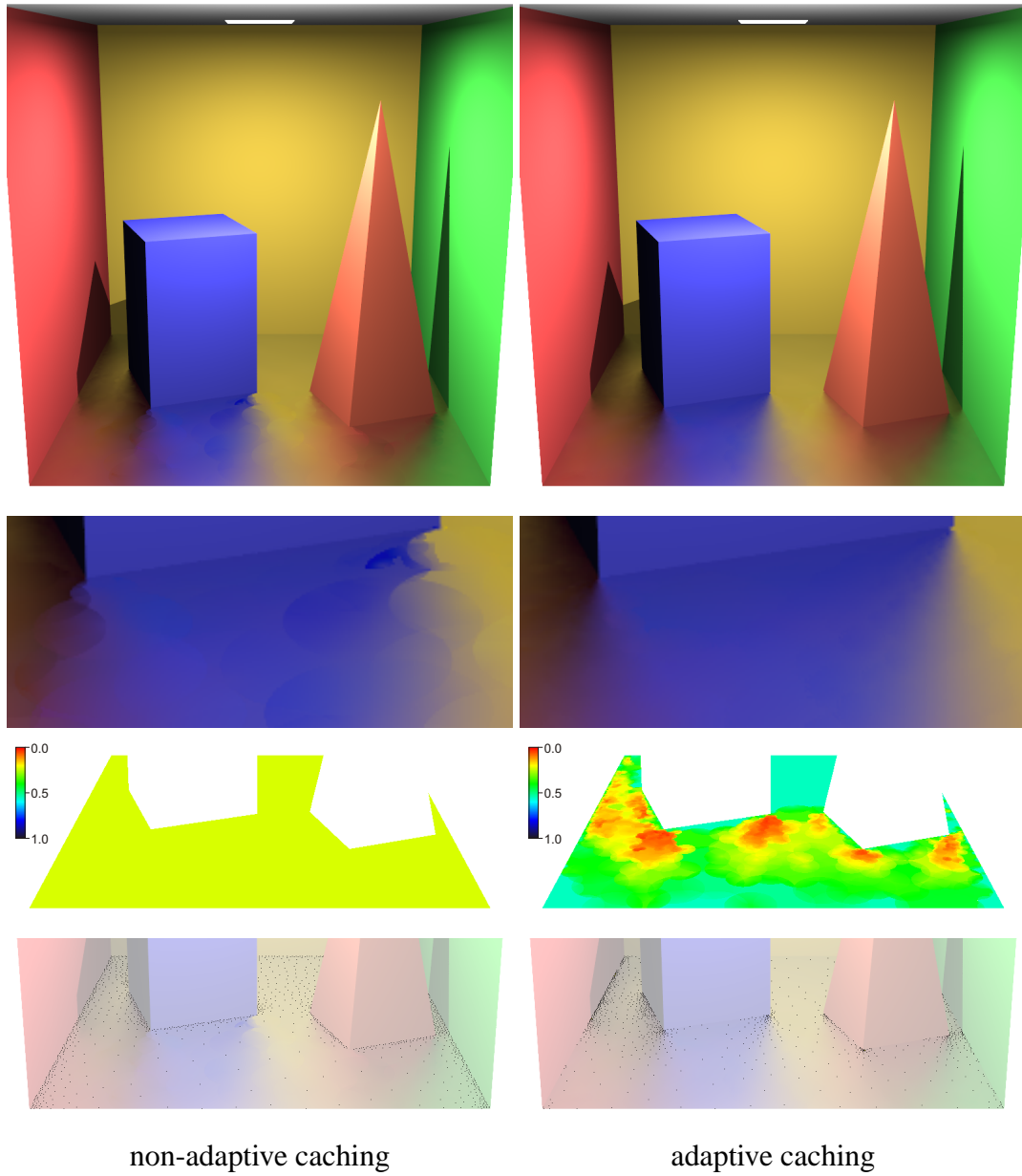


Figure 6.11: Comparison of non-adaptive (left) and adaptive (right) caching for the Cornell box scene. From top to bottom: the rendered images, cutouts from the rendered images, color coded values of the approximation error a and record positions.

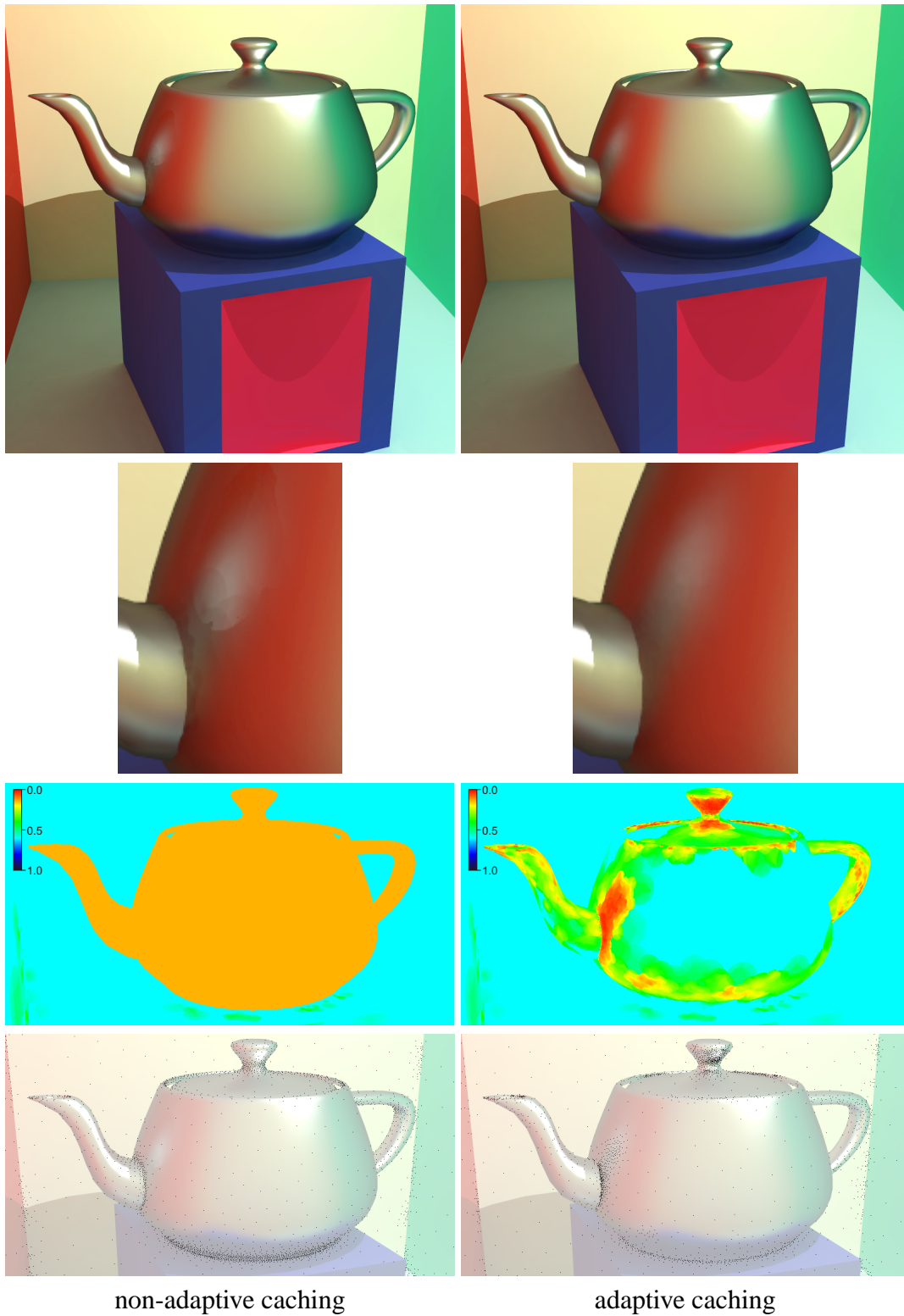


Figure 6.12: Comparison of non-adaptive (left) and adaptive (right) caching for the Teapot scene. From top to bottom: the rendered images, cutouts from the rendered images, color coded values of the approximation error α and record positions.

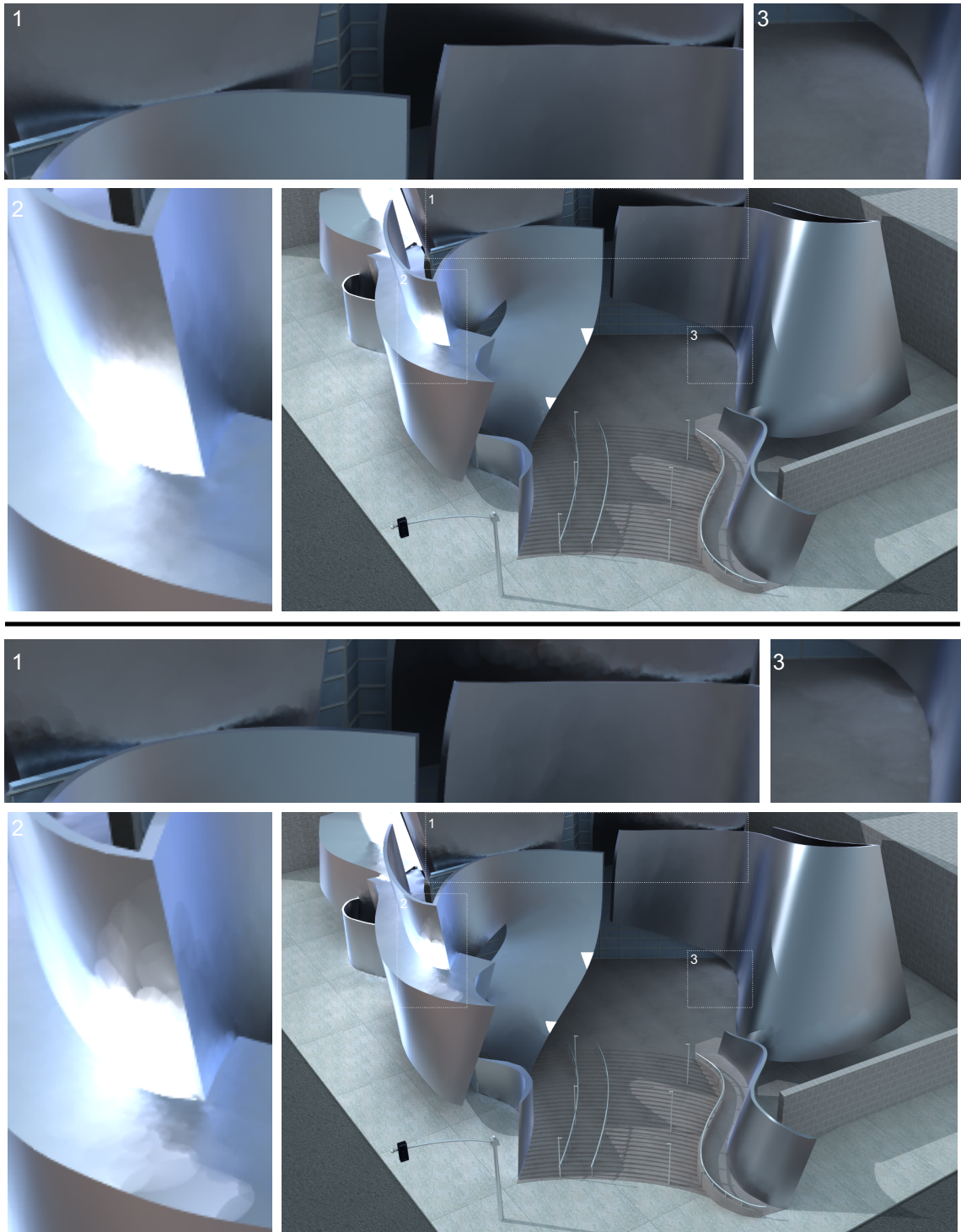


Figure 6.13: Comparison of non-adaptive (bottom half) and adaptive (top half) caching for the Walt Disney Hall scene. The cutouts show places where the most artifacts are left by the non-adaptive caching.

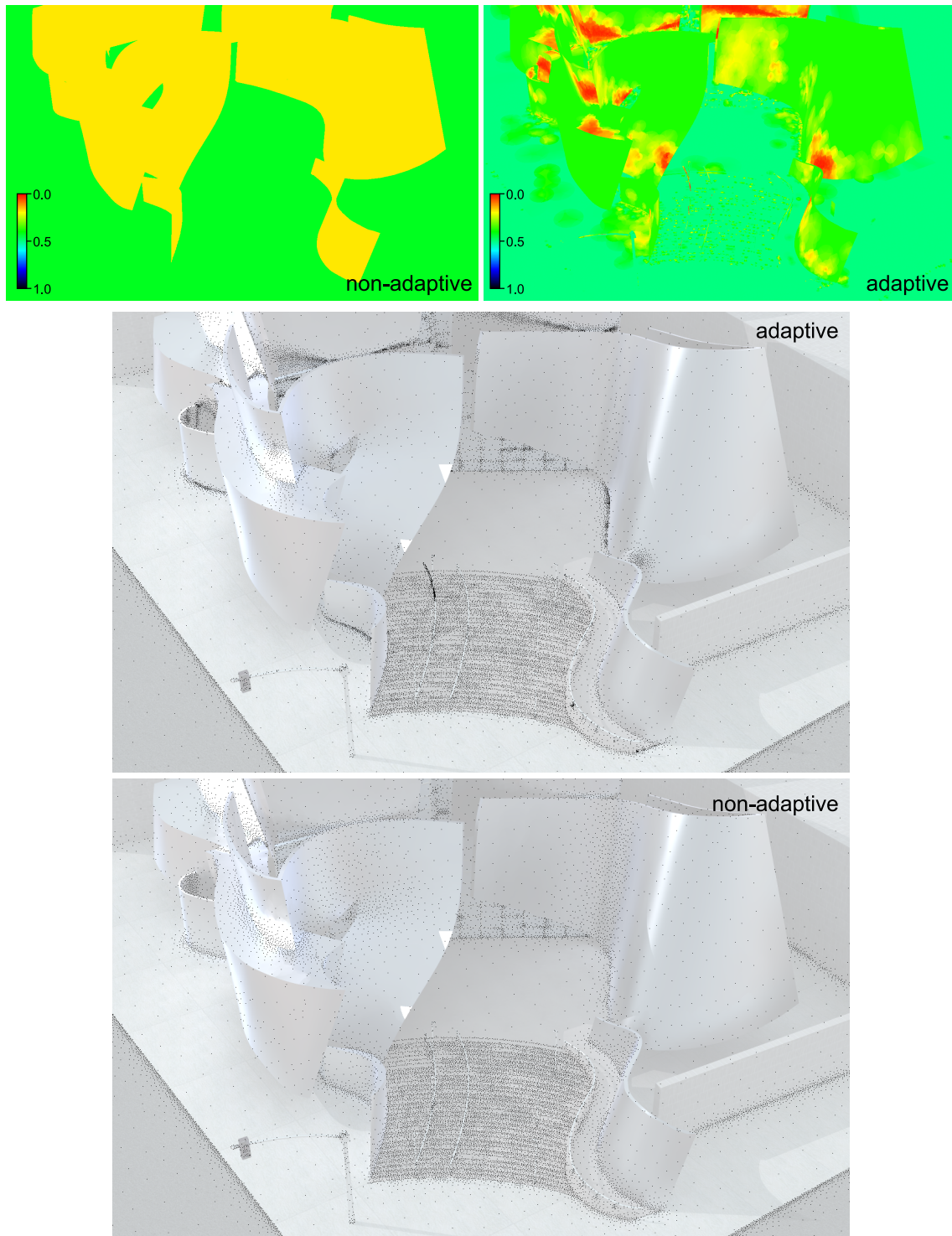


Figure 6.14: Color coded approximation error and record positions for the Walt Disney Hall scene.

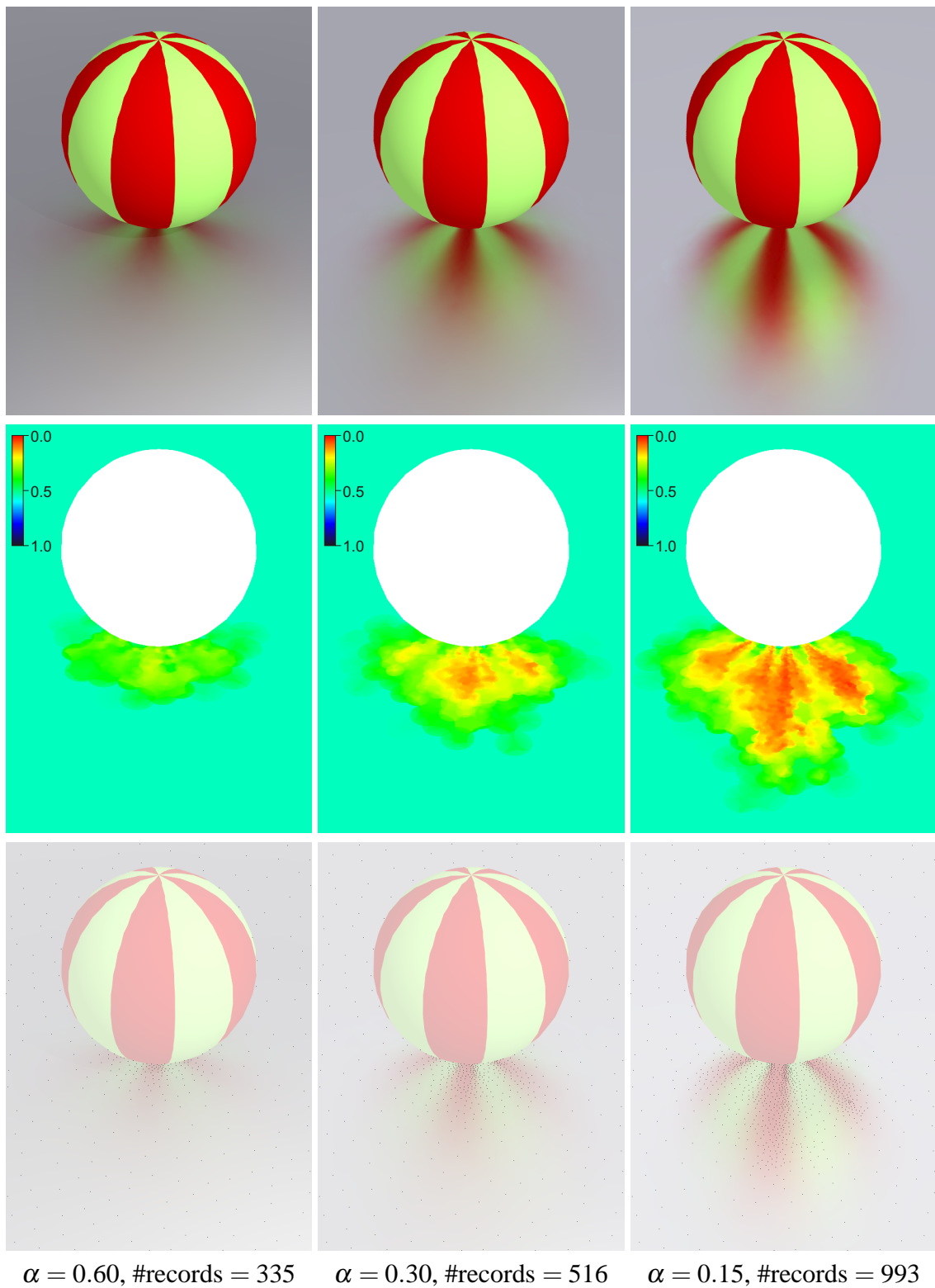


Figure 6.15: Automatic adaptation of record density to the sharpness of the BRDF. The floor is assigned the Ward-Dür BRDF with $\rho_d = 0$, $\rho_s = 0.9$ and (left to right) $\alpha = 0.60$, 0.30 , and 0.15 .

Chapter 7

Conclusion

In this chapter, we conclude the thesis by summarizing our contributions and suggesting ideas for future research.

7.1 Summary of Contributions

7.1.1 Radiance Caching

This thesis addresses the computation of global illumination in scenes with glossy surfaces. To this end, we propose, in Chapter 3, radiance caching, a ray tracing-based algorithm for computing indirect illumination on glossy surfaces with low-frequency BRDFs. Radiance caching efficiently evaluates the indirect term of the illumination integral (2.1) at a point. It can be employed for final gathering in two-pass methods for global illumination computation, or, through recursive evaluation, as a stand-alone global illumination algorithm.

The efficiency of radiance caching is achieved by interpolating indirect illumination from a sparse set of cached values, distributed on surfaces in the scene. The interpolation is possible thanks to the smoothness of shading due to indirect illumination. Radiance caching uses the interpolation scheme of Ward *et al.* [121], which was originally designed for caching and interpolating the view-independent, diffuse, term of indirect illumination. In radiance caching, this scheme is extended by caching the directional distribution of incoming radiance, which allows for interpolation on glossy surfaces with view-dependent BRDFs.

Each time the indirect illumination on a glossy surface needs to be evaluated, the radiance cache is queried. If one or more cached records are available near the query point, incoming radiance is interpolated from them. Otherwise, the directional distribution of incoming radiance is computed at the point and stored in the cache.

Monte Carlo sampling is used to estimate the directional distribution of incoming radiance at a point. It is, then, projected onto the spherical or hemispherical harmonics basis, and the resulting coefficient vector is stored as a new record in the radiance cache.

Incoming radiance is interpolated by a weighted average of the coefficient vectors. This involves several steps: (1) The cached coefficient vector is adjusted by the translational

gradient, stored in the cached radiance record, to make interpolation smoother. (2) The local coordinate frames at the interpolation point and at the point, where the cached incoming radiance was computed, are aligned by a rotation. (3) Finally, the coefficient vector of the rotated incoming radiance is multiplied by a weight and added to the weighted average.

BRDFs are represented in the same, orthonormal, basis as the incoming radiance. Therefore, evaluation of the illumination integral from the interpolated incoming radiance reduces to a dot product of the BRDF and incoming radiance coefficient vectors.

Radiance caching delivers high quality images in a shorter time compared to existing ray tracing-based methods. Compared to radiosity, radiance caching has the advantages of a purely ray tracing-based algorithm: no restriction on the scene geometry, no problems with meshing, lower memory demands.

The main disadvantage of radiance caching is the restriction to low-frequency BRDFs. Support for spatially varying BRDFs is also difficult. Moreover, as an interpolation-based method, radiance caching is biased.

7.1.2 Gradient Computation

Translational radiance gradients significantly improve the spatial smoothness of the interpolated incoming radiance in radiance caching. A translational gradient describes how the directional distribution of incoming radiance changes with a translation on a surface. Gradients are computed from the incoming radiance samples in Monte Carlo hemisphere sampling, when a new radiance record is being created.

In Chapter 4, we proposed two methods for computing the translational radiance gradient at a point. To obtain the total gradient estimate, both methods sum up the marginal contributions due to each sample ray in hemisphere sampling.

In the first, solid angle-based method, a solid angle is associated with each sample ray. The solid angle delimits a small area on a scene surface, contributing radiance to the hemisphere center. Gradient contribution due to one ray is given by the differential change of the solid angle and the area orientation, when the hemisphere center is displaced.

The second, stratification-based method divides the hemisphere into cells, each associated with one ray. A marginal contribution of a cell to the gradient is given by the change of its area due to the movement of walls separating it from its neighboring cells, when the hemisphere center is displaced. This method provides a better gradient estimate than the solid angle-based method, since it takes into account the changes of occlusion with a translation of the hemisphere center.

7.1.3 Spherical Harmonics Rotation

Spherical harmonics rotation is used in radiance caching for coordinate frame alignment in radiance interpolation. Existing rotation approaches cause a bottleneck in the interpolation procedure, since they are computationally involved.

To make the interpolation efficient, in Chapter 5, we proposed a fast, approximate, spherical harmonics rotation procedure. In our procedure, a general rotation is decom-

posed into zyz Euler angles. The rotation matrix around the y -axis is approximated by its truncated Taylor expansion, which is a very sparse matrix, thereby making the y -rotation efficient. Since the z -rotation is simple, too, the whole rotation is fast—at least four times faster than the $zxzxz$ -rotation [50], the most efficient existing rotation method.

Our rotation procedure has the computational complexity of $O(n^2)$ with respect to the spherical harmonics order n . This is less than the $O(n^3)$ complexity of previous methods.

The proposed rotation approximation is accurate only for small rotation angles. Yet, we show that this is not a serious restriction in our applications, which, apart from radiance caching, include real-time shading of bump-mapped surfaces under environment lighting.

7.1.4 Adaptive Radiance Caching

To faithfully represent indirect illumination with only a sparse set of radiance or irradiance values, their density must be proportional to the rate of change of indirect illumination. To this end, radiance caching uses the interpolation error criterion from [121]. Since this criterion was derived for diffuse surfaces, it may, in some cases, underestimate the rate of change of indirect illumination on glossy surfaces, and produce visible interpolation artifacts.

To deal with this problem, we proposed, in Chapter 6, an adaptive algorithm for guiding the record density in radiance caching. The algorithm is based on detecting visible discontinuities in the overlap of the influence areas of two records. If a discontinuity is detected, the influence area of one of the records is reduced in size, and, as a consequence, local record density is increased.

With adaptive caching, the record density is adapted to the actual illumination conditions better than with the original criterion used in radiance caching, especially on surfaces with localized indirect illumination features. Additionally, adaptive caching automatically accommodates the record density to the BRDF sharpness.

In Chapter 6, we also proposed a heuristics called neighbor clamping. Based on observations about geometry coherence in scenes, the heuristics is designed to limit the maximum difference between the radii of influence areas of nearby records. The radius of two records is not allowed to differ by more than their mutual distance. Neighbor clamping significantly reduces the probability of missing small geometry features. It also helps to detect, and alleviate the consequences of, ray leaking—a common problem of irradiance and radiance caching.

7.2 Future Research

7.2.1 Direct Extensions of Radiance Caching

Staying within the framework of radiance caching, as proposed in this thesis, the following issues should be addressed in future research.

Number of Rays for Hemisphere Sampling. The number of rays, required for good results, varied from 1,000 to 8,000 in our test scenes. The number depends on the variance of illumination in the scene and the presence of small localized sources of indirect illumination. In addition, it depends on the perceptibility of fluctuations of the computed indirect illumination, which is higher on flat surfaces. An automatic selection of the number of rays is desirable.

Adaptive Hemisphere Sampling. In hemisphere sampling, some directions exhibit higher variance of incoming radiance than others; some directions are more important for the computed result, because they contribute more energy. Such directions should be sampled with a higher number of rays. To this end, one can use hierarchical quadrature [48, 86, 85] or importance sampling with the incoming radiance as the importance function. A rough estimate of the directional distribution of incoming radiance for the purpose of importance sampling can be obtained from a photon map [46]. We suggest using the cached directional distribution from a nearby radiance record as the importance function.

Strong Secondary Light Sources. Radiance caching, as a light gathering technique, has difficulties detecting small, localized secondary light sources. Those often appear on glossy surfaces in the form of highlights. Caustics, produced by those secondary light sources, can be more efficiently reconstructed by photon tracing, *i.e.* by following light particles from the primary light sources towards the receiving surfaces [47]. But how does one divide the light paths into those that generate caustics and those that generate smooth indirect illumination, in the scenes with general reflectance functions? The frequency analysis of light transport [21] could, possibly, give an answer.

BRDF Separation. Traditionally, BRDFs are separated into a view-independent, diffuse part, and a view-dependent, specular part. However, for the purposes of indirect illumination computation, it might be useful to separate the BRDF into a general low-frequency part, and a high-frequency residual. Indirect illumination for the low-frequency part would, then, be computed by radiance caching, or a similar interpolation-based technique. For the high-frequency part, indirect illumination could be computed by Monte Carlo importance sampling with only a few sample rays.

7.2.2 Wavelet Representation of Incoming Radiance

The restriction to low-frequency BRDFs is the main limitation of radiance caching. Efficient radiance caching on surfaces with high-frequency BRDF would be enabled by sampling only the part of the hemisphere, where the BRDF has important values. Caching radiance from a partially sampled hemisphere requires a basis that allows localization in directions, such as spherical wavelets [84]. By its hierarchical nature, a wavelet representation can also be used for adaptive hemisphere sampling, and for the separation of high- and low-frequency parts of BRDFs, suggested above. The use of spherical wavelets in

radiance caching poses two main challenges: rotation and translational gradient computation.

Wavelet Rotation. A rotation procedure must be conceived for wavelet representation to make interpolation on curved surfaces possible. We believe that the basic idea of our fast spherical harmonics rotation, which consists in replacing a general rotation matrix by its truncated Taylor expansion, is also applicable to wavelet rotation.

Another, less flexible, option is to represent the cached incoming radiance in the global coordinate frame, which avoids rotation in radiance interpolation. The alignment of the BRDF with the interpolated incoming radiance can, then, be effectuated by storing several versions of the BRDF representation, pre-rotated to the global frame for different orientations of the local coordinate frame, similarly to [72].

Translational Gradient. Another challenge is to devise a translational gradient formula suitable for wavelet bases, for which our gradient computation methods is not directly applicable. We believe that the derivation, used in Section 4.3 for the stratification-based gradient, can be employed for this purpose.

7.2.3 Frequency Analysis

The frequency analysis of light transport of Durand *et al.* [21] is related to the interpolation-based global illumination algorithms through the Nyquist theorem. Results of such an analysis can be used in radiance caching to devise a more complete interpolation error criterion than that borrowed from irradiance caching. It would also be interesting to investigate using those results in unbiased, mutation-based global illumination algorithms [109, 15] to design optimal mutation strategies for various terms of light transport, depending on their frequency content. In particular, optimal mutation strategies for glossy surfaces should be designed.

Appendix A

Derivatives of Spherical and Hemispherical Harmonics

Partial derivatives for spherical harmonics are:

$$\frac{\partial Y_l^m}{\partial \theta}(\theta, \phi) = \begin{cases} -\sqrt{2}K_l^m \cos(m\phi) \sin(\theta) \frac{dP_l^m}{dx}(\cos \theta) & \text{if } m > 0 \\ -\sqrt{2}K_l^m \sin(-m\phi) \sin(\theta) \frac{dP_l^{-m}}{dx}(\cos \theta) & \text{if } m < 0 \\ -K_l^0 \sin(\theta) \frac{dP_l^0}{dx}(\cos \theta) & \text{if } m = 0, \end{cases}$$

$$\frac{\partial Y_l^m}{\partial \phi}(\theta, \phi) = \begin{cases} 0 & \text{if } m = 0 \\ -mY_l^{-m}(\theta, \phi) & \text{otherwise,} \end{cases}$$

where K_l^m is the spherical harmonics normalization constant:

$$K_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}}.$$

P_l^m are the associated Legendre polynomials, the derivative of which can be evaluated with the recurrence formula:

$$\frac{dP_l^m}{dx}(x) = \begin{cases} \frac{1}{x^2-1} \left(x l P_l^m(x) - (m+l) P_{l-1}^m(x) \right) & \text{if } m < l \\ -(-1)^m x (2m-1)!! (1-x^2)^{\frac{m}{2}-1} & \text{if } m = l, \end{cases}$$

where $x!!$ is the double factorial (product of all odd integers less than or equal to x).

The partial derivatives for hemispherical harmonics are:

$$\frac{\partial H_l^m}{\partial \theta}(\theta, \phi) = \begin{cases} -2\sqrt{2}\tilde{K}_l^m \cos(m\phi) \sin(\theta) \frac{dP_l^m}{dx}(2\cos\theta - 1) & \text{if } m > 0 \\ -2\sqrt{2}\tilde{K}_l^m \sin(-m\phi) \sin(\theta) \frac{dP_l^{-m}}{dx}(2\cos\theta - 1) & \text{if } m < 0 \\ -2\tilde{K}_l^0 \sin(\theta) \frac{dP_l^0}{dx}(2\cos\theta - 1) & \text{if } m = 0, \end{cases}$$

$$\frac{\partial H_l^m}{\partial \phi}(\theta, \phi) = \begin{cases} 0 & \text{if } m = 0 \\ -mH_l^{-m}(\theta, \phi) & \text{otherwise,} \end{cases}$$

with the following normalization constant:

$$\tilde{K}_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{2\pi(l+|m|)!}} = \sqrt{2}K_l^m.$$

Appendix B

Gradient Computation Code

The following C++ code fragment evaluates one term of the sum in Equation (4.3).

```
// -----  
// This code snippet projects the incoming radiance on hemi(spherical) harmonics  
// and computes the translational gradient of the coefficient vector.  
//  
// The code is invoked once for every ray used in hemisphere sampling when  
// a new radiance cache record is being created.  
//  
// Identifiers are  
// float *H..... evaluated basis functions  
// float *dH_dt  
// float *dH_dp..... derivatives of the basis functions w.r.t theta and phi  
// CVector3D raydir_loc... ray direction in the local coordinate frame  
// float cos_theta, cos_phi, sin_theta, sin_phi  
// ..... spherical coordinates of the ray direction  
// _sphBasis->EvalDerivCartVInt()  
// ..... evaluates the basis functions (SH or HSH) and their  
// ..... derivatives up to 'desiredOrder' in a given direction.  
// float r ..... distance to the hit point  
// float ir ..... 1/r  
// CVector3D U, V, N .... axes of the local coordinate frame  
// float cos_xi ..... angle between the ray direction and the surface  
// ..... normal at the hit point  
// float *dx_gray, *dy_gray .. auxiliary arrays  
//  
// Results of the computation are:  
// result..... 3 vectors of projection coefficients, one vector  
// ..... for each color channel  
// dx, dy ..... 3 vectors of derivatives of the projection  
// ..... coefficients, one vector for each color channel  
//  
// After processing all rays, the vectors 'result', 'dx' and 'dy'  
// have to be multiplied by  $2\pi/N$ , where N is the number of cast rays.  
// -----
```

```

// evaluate the (H)SH basis functions + their derivatives
_sphBasis->EvalDeriv(desiredOrder,raydir_loc, &H,&dH_dt,&dH_dp);

// transform the normal at the hit point to the local frame
CVector3D nl = ToLocalFrame(outgoingHitInfo.GetNormal(),U,V,N);

// derivatives of Omega w.r.t x and y
float dOmega_dx = ( nl.x + 3.0f*raydir_loc.x*cos_xi ) / (r*cos_xi);
float dOmega_dy = ( nl.y + 3.0f*raydir_loc.y*cos_xi ) / (r*cos_xi);

// derivatives needed to evaluate dH/dx
// using dH/dx = dH/dtheta*dtheta/dx + dH/dphi*dphi/dx
float dTheta_dx = -cos_theta * cos_phi * ir;
float dTheta_dy = -cos_theta * sin_phi * ir;
float dPhi_dx   =  sin_phi / (r * sin_theta);
float dPhi_dy   = -cos_phi / (r * sin_theta);

// L_i - independent computation...
for(int i=0; i<sqrDesiredOrder; i++)
{
    dx_gray[i] = H[i]      * dOmega_dx +
                dH_dt[i] * dTheta_dx +
                dH_dp[i] * dPhi_dx;
    dy_gray[i] = H[i]      * dOmega_dy +
                dH_dt[i] * dTheta_dy +
                dH_dp[i] * dPhi_dy;
}

// Li is the incoming radiance along the rays
for(ch=0; ch < channels; ch++) {
    if(Li[ch]<=0) { continue; }
    // multiply vector 'H' by scalar Li[ch] and add to 'result'
    shvecf::madvsip(sqrDesiredOrder, result.GetChannel(ch), H, Li[ch]);
    // multiply + add
    shvecf::madvsip(sqrDesiredOrder, dx->GetChannel(ch), dx_gray, Li[ch]);
    shvecf::madvsip(sqrDesiredOrder, dy->GetChannel(ch), dy_gray, Li[ch]);
}

```

Appendix C

Spherical Harmonics Rotation Matrix Derivative

This appendix describes an algorithm for calculating the k -th derivative of the spherical harmonics rotation matrix around y , $\frac{d^k \mathbf{R}_y}{d\beta^k}$. The algorithm is based on Ivanic and Ruedenberg's rotation matrix construction [44, 45] and retains its structure. Elements of the derivative matrix block $\frac{d^k \mathbf{R}_y^l}{d\beta^k}$ are indexed by m_1 and m_2 and we denote them $R_y^{(k)}(l, m_1, m_2)$.

We start with bands $l = 0$ and $l = 1$:

$$\begin{array}{ll}
 R_y^{(k)}(0, 0, 0) = 1^{(k)} & R_y^{(k)}(1, 0, 0) = \cos^{(k)}(\beta) \\
 R_y^{(k)}(1, -1, -1) = 1^{(k)} & R_y^{(k)}(1, 0, 1) = -\sin^{(k)}(\beta) \\
 R_y^{(k)}(1, -1, 0) = 0 & R_y^{(k)}(1, 1, -1) = 0 \\
 R_y^{(k)}(1, -1, 1) = 0 & R_y^{(k)}(1, 1, 0) = \sin^{(k)}(\beta) \\
 R_y^{(k)}(1, 0, -1) = 0 & R_y^{(k)}(1, 1, 1) = \cos^{(k)}(\beta)
 \end{array}$$

where $\beta = 0$ and $1^{(k)}$ is the derivative of one ($1^{(k)} = 1$ if $k = 0$ and $1^{(k)} = 0$ if $k > 0$). For higher bands, we compute simultaneously the zero-th, first, second, ... *maxderiv*-th derivative:

```

for  $l = 2 \dots n - 1$  do
  for  $k = 0 \dots \text{maxderiv}$  do
    for  $m_1 = -l \dots l$  do
      for  $m_2 = -l \dots l$  do
         $R^{(k)}(l, m_1, m_2) := u_{m_1 m_2}^l \cdot dU^{(k)}(l, m_1, m_2) +$ 
           $v_{m_1 m_2}^l \cdot dV^{(k)}(l, m_1, m_2) +$ 
           $w_{m_1 m_2}^l \cdot dW^{(k)}(l, m_1, m_2)$ 
      end for
    end for
  end for
end for

```

Numerical coefficients $u_{m_1 m_2}^l$, $v_{m_1 m_2}^l$ and $w_{m_1 m_2}^l$ are the same as in the original paper [45]. Functions dU , dV and dW are defined as:

$$dU^{(k)}(l, m_1, m_2) = dP^{(k)}(l, m_1, m_2, 0)$$

$$dV^{(k)}(l, m_1, m_2) = \begin{cases} \frac{dP^{(k)}(l, m_1 - 1, m_2, 1) - dP^{(k)}(l, -m_1 + 1, m_2, -1)}{\sqrt{2}dP^{(k)}(l, 0, m_2, 1)} & \text{if } m_1 > 1 \\ \frac{dP^{(k)}(l, 1, m_2, 1) + dP^{(k)}(l, -1, m_2, -1)}{\sqrt{2}dP^{(k)}(l, 0, m_2, -1)} & \text{if } m_1 = 0 \\ \frac{dP^{(k)}(l, -m_1 - 1, m_2, -1) + dP^{(k)}(l, m_1 + 1, m_2, 1)}{dP^{(k)}(l, m_1 + 1, m_2, 1)} & \text{if } m_1 < -1 \end{cases}$$

$$dW^{(k)}(l, m_1, m_2) = \begin{cases} \frac{dP^{(k)}(l, m_1 + 1, m_2, 1) + dP^{(k)}(l, -m_1 - 1, m_2, -1)}{dP^{(k)}(l, m_1 - 1, m_2, 1) - dP^{(k)}(l, -m_1 + 1, m_2, -1)} & \text{if } m_1 > 0 \\ \text{otherwise} & \end{cases}$$

with

$$dP^{(k)}(l, m_1, m_2, i) = \begin{cases} \frac{dT^{(k)}(1, i, 0, l - 1, m_1, m_2)}{dT^{(k)}(1, i, 1, l - 1, m_1, l - 1) - dT^{(k)}(1, i, -1, l - 1, m_1, -l + 1)} & \text{if } |m_2| < l \\ \frac{dT^{(k)}(1, i, 1, l - 1, m_1, -l + 1) + dT^{(k)}(1, i, -1, l - 1, m_1, l - 1)}{dT^{(k)}(1, i, -1, l - 1, m_1, l - 1)} & \text{if } m_2 = -l \end{cases}$$

and

$$dT^{(k)}(l, m_1, m_2, l', m'_1, m'_2) = \sum_{i=0}^k \binom{k}{i} R_y^{(i)}(l, m_1, m_2) \cdot R_y^{(k-i)}(l', m'_1, m'_2).$$

Function dT implements the product derivative rule $(fg)^{(k)} = \sum_{i=0}^k \binom{k}{i} f^{(i)} g^{(k-i)}$, where $f^{(k)}$ denotes the k -th derivative.

Appendix D

Spherical Harmonics Rotation around z-Axis

The spherical harmonics around the z -axis can be computed efficiently without constructing the rotation matrix $\mathbf{R}_z(\alpha)$ using the following procedure:

```
for  $l = 0 \dots n - 1$  do  
   $v_l^0 := \lambda_l^0$   
  for  $m = 1 \dots l$  do  
     $v_l^{-m} := \lambda_l^{-m} \cos(m\alpha) - \lambda_l^m \sin(m\alpha)$   
     $v_l^m := \lambda_l^{-m} \sin(m\alpha) + \lambda_l^m \cos(m\alpha)$   
  end for  
end for
```

The sines and cosines of multiple angles can be computed with the recurrence formula:

$$\begin{aligned}\sin(m\alpha) &= 2\sin((m-1)\alpha)\cos(\alpha) - \sin((m-2)\alpha) \\ \cos(m\alpha) &= 2\cos((m-1)\alpha)\cos(\alpha) - \cos((m-2)\alpha)\end{aligned}$$

The number of multiplications in the rotation procedure is $N_z(n) = 2n(n-1)$ thus the complexity is $O(n^2)$.

Bibliography

- [1] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2003)*, 22(3):605–612, 2003.
- [2] Thomas Annen, Jan Kautz, Frédo Durand, and Hans-Peter Seidel. Spherical harmonic gradients for mid-range illumination. In *Proceedings of the Eurographics Symposium on Rendering 2004*, pages 331–336. Eurographics Association, 2004.
- [3] Okan Arikan, David A. Forsyth, and James F. O’Brien. Fast and detailed approximate global illumination by irradiance decomposition. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2005)*, 24(3):1108–1114, 2005.
- [4] James Arvo. The irradiance Jacobian for partially occluded polyhedral sources. In *SIGGRAPH ’94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 343–350, New York, NY, USA, 1994. ACM Press.
- [5] Larry Aupperle and Pat Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *SIGGRAPH ’93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 155–162, New York, NY, USA, 1993. ACM Press.
- [6] Kavita Bala, Julie Dorsey, and Seth Teller. Radiance interpolants for accelerated bounded-error ray tracing. *ACM Trans. Graph.*, 18(3):213–256, 1999.
- [7] Kavita Bala, Bruce Walter, and Donald Greenberg. Combining edges and points for interactive high-quality rendering. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2003)*, 22(3), 2003.
- [8] Jeremy Birn. *Digital Lighting & Rendering*. New Riders Press, 2000.
- [9] Chris Buchalew and Donald Fussell. Illumination networks: fast realistic rendering with general reflectance functions. In *SIGGRAPH ’89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 89–98, New York, NY, USA, 1989. ACM Press.
- [10] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. In *SIGGRAPH ’87: Proceedings of the 14th*

- annual conference on Computer graphics and interactive techniques*, pages 273–281, New York, NY, USA, 1987. ACM Press.
- [11] Cheol Ho Choi, Joseph Ivanic, Mark S. Gordon, and Klaus Ruedenberg. Rapid and stable determination of rotation matrices between spherical harmonics by direct recursion. *J. Chem. Phys.*, 111(19):8825–8831, 1999.
- [12] Per H. Christensen, Dani Lischinski, Eric J. Stollnitz, and David H. Salesin. Clustering for glossy global illumination. *ACM Trans. Graph.*, 16(1):3–33, 1997.
- [13] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2005)*, 24(3):1166–1175, 2005.
- [14] Luc Claustres, Mathias Paulin, and Yannick Boucher. BRDF Measurement Modelling using Wavelets for Efficient Path Tracing. *Computer Graphics Forum*, 22(4):701–716, December 2003.
- [15] David Cline, Justin Talbot, and Parris Egbert. Energy redistribution path tracing. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2005)*, 24(3):1186–1195, 2005.
- [16] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1993.
- [17] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 137–145, 1984.
- [18] Paul Debevec. Light probe gallery. <http://www.debevec.org/Probes/>, 2005.
- [19] Kirill Dmitriev and Hans-Peter Seidel. Progressive path tracing with lightweight local error estimation. In *Vision, modeling, and visualization 2004 (VMV-04)*, pages 249–254, Stanford, USA, 2004. Akademische Verlagsgesellschaft Aka.
- [20] Arne Dür. On the ward model for global illumination. Unpublished material, 2005.
- [21] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. A frequency analysis of light transport. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2005)*, 24(3):1115–1126, 2005.
- [22] Philip Dutré, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. A. K. Peters Ltd., Natick, Massachusetts, 2003.
- [23] James A. Ferwerda. Three varieties of realism in computer graphics. In *Proceedings of SPIE Human Vision and Electronic Imaging '03*, pages 290–297, 2003.
- [24] James A. Ferwerda and Fabio Pellacini. Functional difference predictors (FDPs): measuring meaningful image differences. In *Asilomar Conference on Signals, Systems, and Computers*, 2003.

- [25] Ronald W. Fleming, Ron O. Dror, and Edward H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, 3:347–368, July 2003.
- [26] Pascal Gautron, Jaroslav Křivánek, Kadi Bouatouch, and Sumanta N. Pattanaik. Radiance cache splatting: A GPU-friendly global illumination algorithm. In *Rendering Techniques 2005, Eurographics Symposium on Rendering*, June 2005.
- [27] Pascal Gautron, Jaroslav Křivánek, Sumanta N. Pattanaik, and Kadi Bouatouch. A novel hemispherical basis for accurate and efficient rendering. In *Rendering Techniques 2004, Eurographics Symposium on Rendering*, pages 321–330. Eurographics Association, June 2004.
- [28] Simon Gibson and Roger Hubbard. Perceptually-driven radiosity. *Computer Graphics Forum*, 16(2):129–140, June 1997.
- [29] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1995.
- [30] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 213–222, New York, NY, USA, 1984. ACM Press.
- [31] Xavier Granier and George Drettakis. Incremental updates for rapid glossy global illumination. In *Proceedings of Eurographics 2001*, 2001.
- [32] Xavier Granier and George Drettakis. A final reconstruction framework for an unified global illumination algorithm. *ACM Transaction on Graphics*, 23(2), 2004.
- [33] Robin Green. Spherical harmonic lighting: The gritty details. In *Game Developers' Conference*, 2003.
- [34] Robin Green. E-mail communication, 2004.
- [35] Donald P. Greenberg, Kenneth E. Torrance, Peter Shirley, James Arvo, Eric Lafortune, James A. Ferwerda, Bruce Walter, Ben Trumbore, Sumanta Pattanaik, and Sing-Choong Foo. A framework for realistic image synthesis. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 477–494, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [36] Eduard Gröller. *Coherence in Computer Graphics*. PhD thesis, Technische universität Wien, 1992.
- [37] Baining Guo. Progressive radiance evaluation using directional coherence maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 255–266, New York, NY, USA, 1998. ACM Press.

- [38] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 197–206, New York, NY, USA, 1991. ACM Press.
- [39] Vlastimil Havran, Robert Herzog, and Hans-Peter Seidel. Fast final gathering via reverse photon mapping. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2005)*, 24, 2005.
- [40] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, hardware-accelerated shading and lighting. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 171–178, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [41] Nicolas Holzschuch and François Sillion. Accurate computation of the radiosity gradient with constant and linear emitters. In *Sixth Eurographics Workshop on Rendering*, June 1995.
- [42] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 133–142, New York, NY, USA, 1986. ACM Press.
- [43] Tetsuro Inui, Yukito Tanabe, and Y. Onodera. *Group theory and its applications in physics*. Springer-Verlag Telos, 2nd corrected edition, 1996.
- [44] Joseph Ivanic and Klaus Ruedenberg. Rotation matrices for real spherical harmonics. direct determination by recursion. *J. Phys. Chem.*, 100(15):6342–6347, 1996.
- [45] Joseph Ivanic and Klaus Ruedenberg. Additions and corrections : Rotation matrices for real spherical harmonics. *J. Phys. Chem. A*, 102(45):9099–9100, 1998.
- [46] Henrik Wann Jensen. Importance driven path tracing using the photon map. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 326–335. Springer Verlag, 1995.
- [47] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters Ltd., Natick, Massachusetts, 2001.
- [48] James T. Kajiya. The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, New York, NY, USA, 1986. ACM Press.
- [49] Toshi Kato. Photon mapping in Kilauea. In *SIGGRAPH 2002, Course Notes No. 43*, pages 122–191, 2002.
- [50] Jan Kautz, Peter-Pike Sloan, and John Snyder. Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proceedings of the 13th*

- Eurographics workshop on Rendering*, pages 291–296. Eurographics Association, 2002.
- [51] Alexander Keller. Instant radiosity. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 49–56, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [52] David Kirk and James Arvo. Unbiased sampling techniques for image synthesis. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 153–156, New York, NY, USA, 1991. ACM Press.
- [53] Jan J. Koenderink, Andrea J. van Doorn, and Marigo Stavridi. Bidirectional reflection distribution function expressed in terms of surface scattering modes. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume II*, pages 28–39, London, UK, 1996. Springer-Verlag.
- [54] Jaroslav Křivánek, Pascal Gautron, Kadi Bouatouch, and Sumanta Pattanaik. Improved radiance gradient computation. In *SCCG '05: Proceedings of the 21th spring conference on Computer graphics*. ACM Press, 2005.
- [55] Jaroslav Křivánek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics*, 11(5), September/October 2005.
- [56] Jaroslav Křivánek, Jaakko Konttinen, Sumanta Pattanaik, and Kadi Bouatouch. Fast approximation to spherical harmonic rotation. Technical Report 1728, IRISA, Rennes, France, July 2005.
- [57] Eric Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Katholieke Universiteit Leuven, February 1996.
- [58] Eric Lafortune and Yves Willems. A 5D tree to reduce the variance of monte carlo ray tracing. In *Proceedings of the Eurographics Workshop on Rendering*, pages 11–20, 1995.
- [59] Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In *Computer Graphics Proceedings*, pages 145–153, December 1993.
- [60] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [61] P. Lalonde and A. Fournier. Generating reflected distribution from BRDF data. *Computer Graphics Forum (Proceedings of EUROGRAPHICS '97)*, 16(3):C293–C300, 1997.

- [62] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient BRDF importance sampling using a factored representation. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2004)*, 23(3):496–505, 2004.
- [63] Jaakko Lehtinen and Jan Kautz. Matrix radiance transfer. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 59–64. ACM Press, 2003.
- [64] Robert R. Lewis and Alain Fournier. Light-driven global illumination with a wavelet representation. In *Proceedings of the Seventh Eurographics Workshop on Rendering*, pages 11–20, 1996.
- [65] Xinguo Liu, Peter-Pike Sloan, Heung-Yeung Shum, and John Snyder. All-frequency precomputed radiance transfer for glossy objects. In *Proceedings of the Eurographics Symposium on Rendering*, pages 337–344, 2004.
- [66] T. M. MacRobert and I. N. Sneddon. *Spherical harmonics: an elementary treatise on harmonic functions, with applications*. Pergamon Press, Oxford, England, 3rd rev. edition, 1967.
- [67] Oleg A. Makhotkin. Analysis of radiative transfer between surfaces by hemispherical harmonics. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 56(6):869–879, 1996.
- [68] Ignacio Martin, Xavier Pueyo, and Dani Tost. An image-space refinement criterion for linear hierarchical radiosity. In *Graphics Interface '97*, pages 26–36. Canadian Human-Computer Communications Society, 1997.
- [69] Microsoft. DirectX 9.0b SDK, October 2004.
- [70] Don P. Mitchell. Generating antialiased images at low sampling densities. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 65–72, New York, NY, USA, 1987. ACM Press.
- [71] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2003)*, 22(3):376–381, 2003.
- [72] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2004)*, 23(3):477–487, 2004.
- [73] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of BRDF models. In *Proceedings of Eurographics Symposium on Rendering*. The Eurographics Association, 2005.
- [74] Sumanta N. Pattanaik and Kadi Bouatouch. Haar wavelet: A solution to global illumination with general surface properties. In *Proceedings of Fifth Eurographics Workshop on Rendering*, 1995.

- [75] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, San Francisco, CA, 2004.
- [76] Frederic P. Pighin, Dani Lischinski, and David Salesin. Progressive previewing of ray-traced images using image-plane discontinuity meshing. In *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 115–125, New York, NY, 1997. Springer Wien.
- [77] Ravi Ramamoorthi. *A Signal-Processing Framework for Forward and Inverse Rendering*. PhD thesis, Stanford University, 2002.
- [78] Ravi Ramamoorthi and Pat Hanrahan. Frequency space environment map rendering. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 517–526, New York, NY, USA, 2002. ACM Press.
- [79] Mahesh Ramasubramanian, Sumanta N. Pattanaik, and Donald P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 73–82, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [80] Bernard Le Saëc and Christophe Schlick. A progressive ray-tracing based radiosity with general reflectance functions. In *Proceedings of the Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, June 1990.
- [81] Annette Scheel, Marc Stamminger, and Hans-Peter Seidel. Thrifty final gather for radiosity. In *Rendering Techniques 2001 (Proc. of Eurographics Workshop on Rendering 2001)*. Eurographics, Spinger-Verlag, June 2001. held in London, UK.
- [82] Annette Scheel, Marc Stamminger, and Hans-Peter Seidel. Grid based final gather for radiosity on complex clustered scenes. In *Computer Graphics Forum, Proceedings of Eurographics 2002*, volume 21. Blackwell, September 2002.
- [83] Peter Schröder and Patrick Hanrahan. Wavelet methods for radiance computations. In *Proceedings Fifth Eurographics Workshop on Rendering*, pages 310–326, 1995.
- [84] Peter Schröder and Wim Sweldens. Spherical wavelets: efficiently representing functions on the sphere. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 161–172, New York, NY, USA, 1995. ACM Press.
- [85] Xavier Serpaggi and Bernard Péroche. An adaptive method for indirect illumination using light vectors. In *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2001)*, volume 20(3), pages 278–287. Blackwell Publishing, 2001.
- [86] Peter Shirley and Kenneth Chiu. Notes on adaptive quadrature on the hemisphere. Technical Report TR-411, Indiana University, July 1994.

- [87] Peter Shirley and Changyaw Wang. Direct lighting calculation by Monte Carlo integration. In *Proceedings of the Second Eurographics Workshop on Rendering*, pages 54–59, New York, NY, 1994. Springer-Verlag.
- [88] Peter S. Shirley. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, University of Illinois at Urbana-Champaign, 1991.
- [89] François Sillion, James R. Arvo, Spehne H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 187–196, New York, NY, USA, 1991. ACM Press.
- [90] François Sillion, George Drettakis, and Cyril Soler. A clustering algorithm for radiance calculation in general environments. In *Rendering Techniques 1995 (Proceedings of the Eurographics Workshop on Rendering '95)*, June 1995.
- [91] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, 1994.
- [92] Maryann Simmons and Carlo H. Séquin. Tapestry: A dynamic mesh-based display representation for interactive rendering. In *Proceedings of the 11th Eurographics Workshop on Rendering*, pages 329–340, June 2000.
- [93] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2003)*, 22(3):382–391, 2003.
- [94] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02)*, pages 527–536. ACM Press, 2002.
- [95] Peter-Pike Sloan, Ben Luna, and John Snyder. Local, deformable precomputed radiance transfer. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2005)*, 24(3):1216–1224, 2005.
- [96] Philipp Slusallek, Wolfgang Heidrich, and Hans-Peter Seidel. Radiance maps: An image-based approach to global illumination. *SIGGRAPH '98, Technical Sketch*, 1998.
- [97] Miłosław Smyk, Shin ichi Kinuwaki, Roman Durikovic, and Karol Myszkowski. Temporally coherent irradiance caching for high quality animation rendering. *Computer Graphics Forum (Proceedings of EUROGRAPHICS '05)*, 24(3), 2005.
- [98] Miłosław Smyk and Karol Myszkowski. Quality improvements for indirect illumination interpolation. In *Proceedings of the International Conference on Computer Vision and Graphics*, 2002.

- [99] Marc Stamminger, Annette Scheel, Xavier Granier, Frederic Perez-Cazorla, George Drettakis, and François Sillion. Efficient glossy global illumination with interactive viewing. *Computer Graphics Forum*, 19(1), 2000.
- [100] William A. Stokes, James A. Ferwerda, Bruce Walter, and Donald P. Greenberg. Perceptual illumination components: a new approach to efficient, high quality global illumination rendering. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2004)*, 23(3):742–749, 2004.
- [101] Gabor Szegö. *Orthogonal polynomials*. American Mathematical Society, Providence, Rhode Island, 4 edition, 1975.
- [102] Eric Tabellion. E-mail communication, 2005.
- [103] Eric Tabellion and Arnauld Lamorlette. An approximate global illumination system for computer generated films. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2004)*, 23(3):469–476, 2004.
- [104] Takehiro Tawara, Karol Myszkowski, and Hans-Peter Seidel. Exploiting temporal coherence in final gathering for dynamic scenes. In *Proceedings of Computer Graphics International (CGI '04)*, 2004.
- [105] Parag Tole, Fabio Pellacini, Bruce Walter, and Donald P. Greenberg. Interactive global illumination in dynamic scenes. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2002)*, 21(3):537–546, July 2002.
- [106] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, December 1997.
- [107] Eric Veach and Leonidas J. Guibas. Bidirectional estimators for light transport. In *Proceedings of Eurographics Rendering Workshop*, pages 147–162, June 1994.
- [108] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, New York, NY, USA, 1995. ACM Press.
- [109] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [110] Valdimir Volevich, Karol Myszkowski, Andrei Khodulev, and Edward A. Kopylov. Perceptually-informed progressive global illumination solution. Technical Report 99-1-002, The University of Aizu, 1994.
- [111] Bruce Walter, Gün Alppay, Eric Lafortune, Sebastian Fernandez, and Donald P. Greenberg. Fitting virtual lights for non-diffuse walkthroughs. In *SIGGRAPH '97:*

- Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 45–48, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [112] Bruce Walter, George Drettakis, Donald P. Greenberg, and Oliver Deussen. Enhancing and optimizing the render cache. In *Proceedings of the 10th Eurographics Workshop on Rendering*, June 2002.
- [113] Bruce Walter, George Drettakis, and Steven Parker. Interactive rendering using render cache. In *Proceedings of the 13th Eurographics Workshop on Rendering*, pages 19–30, 1999.
- [114] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. Lightcuts: a scalable approach to illumination. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2005)*, 24(3):1098–1107, 2005.
- [115] Rui Wang, John Tran, and David Luebke. All-frequency relighting of non-diffuse objects using separable BRDF approximation. In *Proceedings of the Eurographics Symposium on Rendering*, pages 345–354, 2004.
- [116] Rui Wang, John Tran, and David Luebke. All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph. (Proceedings of SIGGRAPH 2005)*, 24(3):1208–1215, 2005.
- [117] Gregory Ward and Maryann Simmons. The holodeck ray cache: an interactive rendering system for global illumination in nondiffuse environments. *ACM Trans. Graph.*, 18(4):361–368, 1999.
- [118] Gregory J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 265–272, New York, NY, USA, 1992. ACM Press.
- [119] Gregory J. Ward. The radiance lighting simulation and rendering system. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 459–472, New York, NY, USA, 1994. ACM Press.
- [120] Gregory J. Ward and Paul S. Heckbert. Irradiance gradients. In *Proceedings of the Third Eurographics Workshop on Rendering*, pages 85–98, 1992.
- [121] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 85–92, New York, NY, USA, 1988. ACM Press.
- [122] Eric W. Weisstein. Infinitesimal rotation. From *MathWorld – A Wolfram Web Resource*. <http://mathworld.wolfram.com/InfinitesimalRotation.html>, 2004.
- [123] Eric W. Weisstein. Spherical coordinates. From *MathWorld – A Wolfram Web Resource*. <http://mathworld.wolfram.com/SphericalCoordinates.html>, 2004.

- [124] Stephen H. Westin. Lafortune BRDF for RenderMan. <http://www.graphics.cornell.edu/westin/lafortune/lafortune.html>, 2005.
- [125] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 255–264, New York, NY, USA, 1992. ACM Press.
- [126] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, 1980.
- [127] James C. Wyant and Katherine Creath. Basic wavefront aberration theory for optical metrology. In *Applied optics and Optical Engineering, Vol XI*, pages 27–39. Academic Press, Inc., 1992.
- [128] Christopher R. Wyman. *Fast Local Approximation to Global Illumination*. PhD thesis, University of Utah, 2004.
- [129] Jacques Zaninetti, Xavier Serpaggi, and Bernard Péroche. A vector approach for global illumination in ray tracing. In *Proceedings of EUROGRAPHICS*, 1998.

Résumé en français

Introduction

L'illumination globale est un moyen permettant de produire des images de synthèse dites photoréalistes. Elle joue un rôle encore plus important dans le cas de scènes contenant des objets en partie spéculaires, c'est à dire non parfaitement lisses. Cette thèse traite principalement du problème du calcul de l'illumination globale dans le cas de ce type de scène où les objets sont caractérisés par des réflectances de basse fréquence.

Le calcul de l'illumination globale s'avère très difficile lorsqu'on utilise un échantillonnage de type Monte Carlo car un grand nombre d'échantillons (rayons) est nécessaire pour obtenir une bonne estimation de l'éclairage indirect. C'est pourquoi nous proposons une méthode utilisant un cache de luminance, méthode basée sur le lancer de rayon et prenant en compte les surfaces spéculaires non parfaitement lisses (ayant une rugosité microscopique) dont les réflectances possèdent des caractéristiques de basse fréquence. L'algorithme proposé exploite la variation douce de l'éclairage sur une surface en interpolant l'éclairage indirect à partir de données éparses stockées dans le cache. Tous les outils et composants logiciels nécessaires à la mise en oeuvre de cet algorithme sont présentés dans cette thèse. En effet, il s'agit : du choix des surfaces sur lesquelles la sauvegarde dans le cache doit être effectuée, de la représentation compacte des informations sauvegardées à l'aide d'harmoniques sphériques, de la rotation des harmoniques sphériques nécessaires lors de l'interpolation, du calcul de gradient translationnel aussi utilisé lors de l'interpolation, du critère adaptatif permettant de choisir les valeurs à stocker dans le cache.

Notre méthode basée sur le cache de luminance permet de générer des images de grande qualité en un temps plus court que celui obtenu avec les méthodes existantes basées sur le lancer de rayon. Elle peut être aussi utilisée pour le regroupement final dans les méthodes à deux passes ou comme un algorithme d'illumination globale complet.

Dans la section suivante nous résumerons les contributions apportées par cette thèse puis nous conclurons et proposerons de nouvelles perspectives.

Résumé des Contributions

Les deux premiers chapitres de cette thèse sont introductifs. Le premier présente les raisons qui nous ont motivés pour effectuer un calcul d'illumination globale sur des surfaces spéculaires rugueuses. Quant au second, il présente les notions de base de

l'illumination globale, en particulier ceux concernant l'éclairage indirect sur des surfaces spéculaires rugueuses. Nos contributions sont décrites dans les chapitres 3 à 6 de cette thèse. Dans cette section, nous résumons le contenu de ces 4 chapitres.

Cache de Luminance

Cette thèse traite du problème de calcul de l'illumination globale dans des scènes contenant des surfaces spéculaires rugueuses (*glossy surfaces* en anglais). C'est pourquoi nous proposons dans le chapitre 3 le cache de luminance (*radiance cache* en anglais), un algorithme basé sur le lancer de rayon et permettant de calculer l'éclairage indirect sur des surfaces spéculaires rugueuses caractérisées par des réflectances de basse fréquence. La méthode s'appuyant sur un cache de luminance sera appelée dorénavant *CL*. Elle évalue de manière efficace (faible coût de calcul) le terme d'éclairage indirect exprimé en un point par une intégrale d'illumination (*Illumination Integral* en anglais). L'intégrale d'illumination exprime la luminance réfléchiée par une surface, sous forme d'une intégrale sur l'hémisphère du produit entre la luminance incidente et la BRDF¹ de la surface. La méthode *CL* peut être aussi utilisée pour le regroupement final (*final gathering* en anglais) dans les méthodes à deux passes ou comme un algorithme récursif d'illumination globale complet.

L'efficacité de la méthode *CL* est due au fait que l'éclairage indirect peut être calculé par interpolation de valeurs d'éclairage indirect déjà calculées en certains points positionnés de manière éparse sur les surfaces et stockées dans le cache de luminance. Cette interpolation est possible en raison de la faible variation de l'éclairage indirect sur une surface. *CL* utilise le schéma d'interpolation de Ward *et al.* [121], qui a été conçu à l'origine pour stocker (dans le cache) et interpoler le terme diffus de l'éclairage indirect, c'est à dire la composante diffuse indirecte. Notre méthode *CL* étend ce schéma au stockage dans le cache des luminances incidentes qui varient avec la direction incidente, ce qui permet d'effectuer l'interpolation de l'éclairage indirect sur des surfaces spéculaires rugueuses caractérisées par des BRDF dépendant du point de vue.

A chaque fois qu'on nécessite le calcul de l'éclairage indirect en un point P sur une surface spéculaire rugueuse, le cache de luminance est sollicité. Si un enregistrement (ou plus) se trouve dans le cache et correspond à un (ou plusieurs) point(s) proche(s) du point P , alors la luminance incidente en P est interpolée à partir de ce (ou ces) point(s). Sinon, la luminance incidente en P est évaluée et stockée dans le cache.

L'échantillonnage par Monte Carlo est utilisé pour estimer la luminance incidente en un point. Cette luminance est ensuite projetée dans une base de fonctions d'harmoniques sphériques ou hémisphériques [66, 43, 77, 27], le vecteur de coefficients résultant de cette projection est stocké dans le cache en tant que nouvel enregistrement.

La luminance incidente est interpolée comme une moyenne pondérée de vecteurs de coefficients de projection. Ceci s'effectue en plusieurs étapes : (1) chaque vecteur de coefficients stocké dans la cache est ajusté par l'ajout d'un gradient translationnel, se trouvant dans un enregistrement stocké dans le cache, afin d'assurer une interpolation douce, (2) les repères locaux en un point P (pour lequel l'interpolation est en cours de

¹Fonction de Distribution de Réflectance Bidirectionnelle

calcul) et en un point dans le cache sont alignés à l'aide d'une rotation, (3) enfin, le vecteur de coefficients de la luminance incidente, ayant subi la rotation, est multiplié par un poids et ajouté à la moyenne pondérée.

Les BRDF sont représentées de la même manière que les luminances incidentes, c'est à dire qu'elles sont projetées dans une base de fonctions d'harmoniques sphériques ou hémisphériques [50]. Par conséquent, l'évaluation de l'intégrale d'illumination, à partir de la luminance incidente interpolée, se réduit à un produit scalaire des vecteurs de coefficients de projection de la BRDF et de la luminance incidente [101].

Les surfaces, sur lesquelles s'applique notre méthode *CL*, sont sélectionnées de manière automatique. Si la BRDF d'une surface peut être représentée par des harmoniques sphériques d'un certain ordre avec une faible erreur [78], alors la BRDF est un signal de basse fréquence et le cache de luminance peut être utilisé pour une éventuelle interpolation. Sinon, il s'agit d'une BRDF présentant de hautes fréquences, ce qui rend le cache inutilisable. Cette démarche est effectuée pour toutes les BRDF de la scène avant que le rendu ne commence.

Comparée aux méthodes basées sur le lancer de rayon, notre méthode *CL* produit des images de haute qualité en un temps plus court. Par rapport à la méthode de radiosité, elle présente l'avantage d'un algorithme de lancer de rayon pur, c'est à dire aucune restriction sur la géométrie de la scène, pas de problème de maillage, peu de ressources mémoire.

Le seul inconvénient de notre méthode *CL* est la contrainte basse fréquence imposée aux BRDF. En outre, étant donné que notre méthode *CL* recourt à une technique d'interpolation, elle fournit par conséquent une estimation biaisée de l'illumination indirecte, le biais étant une petite erreur systématique.

Calcul de Gradient de Luminance

Les gradients translationnels de luminance améliorent la douce variation spatiale de la luminance incidente interpolée dans le cas d'une méthode d'illumination utilisant un cache de luminance. Un gradient translationnel décrit la manière dont la distribution directionnelle de la luminance incidente varie avec toute translation sur une surface. Les gradients sont calculés à partir des enregistrements de luminance incidente, dans le cas d'un échantillonnage de type Monte Carlo, lors du calcul d'un nouvel enregistrement.

Dans le chapitre 4, nous proposons deux méthodes de calcul de gradient translationnel en un point. Pour obtenir le gradient total estimé, les deux méthodes cumulent les contributions marginales de chaque rayon échantillon lancé à travers l'hémisphère placé autour de ce point.

Dans la première méthode, basée sur les angles solides, à chaque rayon est associé un angle solide. Ce dernier délimite une petite zone, sur une surface de la scène, contribuant à la luminance du centre de l'hémisphère. La contribution, en terme de gradient, due à un rayon est donnée par la variation différentielle de l'angle solide et de l'orientation de la zone après déplacement du centre de l'hémisphère.

La seconde méthode, basée sur une stratification, divise l'hémisphère en cellules, chacune étant associée à un rayon. Une contribution marginale d'une cellule au gradient est donnée par la variation de son aire due au mouvement des murs qui la séparent de

ses voisines, après déplacement du centre de l'hémisphère. Cette méthode fournit une meilleure estimation de gradient que celle obtenue avec la première méthode puisqu'elle prend en compte les changements d'occlusion après translation du centre de l'hémisphère.

Rotation d'Harmoniques Sphériques

La rotation d'harmoniques sphériques est utilisée pour l'alignement de repères lors d'interpolation de luminance. Les méthodes existantes de calcul de rotation [44, 45, 11, 50] rendent très coûteuses (en terme de temps de calcul) la procédure d'interpolation.

Afin de rendre la procédure d'interpolation efficace, nous proposons dans le chapitre 5 une procédure de rotation d'harmoniques sphériques approximative, plus rapide que les méthodes précédentes.. Dans cette procédure, une rotation est décomposée en angles d'Euler zyz . La matrice de rotation autour de l'axe y est approchée par son développement limité, elle est alors creuse, ce qui rend la rotation autour de y efficace. Comme la rotation autour de l'axe z est simple, la rotation complète est alors à son tour rapide. Elle est 4 fois plus rapide qu'une rotation $zxzxz$ [50], méthode de rotation la plus rapide parmi celles existantes.

La complexité de notre procédure de rotation est de $O(n^2)$, n étant l'ordre des harmoniques sphériques. Elle est bien inférieure à celle des méthodes précédentes qui est de l'ordre de $O(n^3)$.

Notre méthode d'approximation de la rotation n'est précise que pour de petits angles de rotation. Nous montrons que ce n'est pas une restriction forte dans le cas de nos applications qui, en plus du rendu basé sur le cache de luminance, incluent l'éclairage temps réel de surfaces bosselées (*bumped surfaces* en anglais) par des environnements éclairés (*environment lighting* en anglais).

Cache de Luminance Adaptatif

Afin de représenter l'éclairage indirect de manière fiable, la densité des enregistrements d'éclairement ou de luminance incidente doit être proportionnelle au taux de variation de cet éclairage. Notre algorithme *CL* utilise un critère d'erreur introduite lors de l'interpolation de [121]. Toutefois, ce critère est introduit dans le cas du calcul d'éclairement diffus indirect. Lors de l'application de notre algorithme à des surfaces rugueuses, ce critère peut mener dans de rares cas à une sous-estimation du changement de luminance incidente. Cette sous-estimation peut alors avoir pour conséquence des défauts d'interpolation visibles dans l'image finale.

Afin de résoudre ce problème, nous proposons dans le Chapitre 6 un algorithme adaptatif améliorant la répartition des enregistrements de luminance incidente. Cet algorithme est basé sur la détection des discontinuités visibles dans les zones couvertes par plusieurs enregistrements. Si une discontinuité est détectée, les zones d'influence des enregistrements correspondants sont réduites, entraînant une augmentation de la densité locale d'enregistrements.

Cette méthode adaptative permet d'adapter la densité d'enregistrements aux conditions d'éclairage de manière plus précise que celle proposée dans [121]. L'augmentation de

qualité est particulièrement visible sur des surfaces comportant des détails d'éclairage indirect fins. De plus, notre méthode permet également d'adapter la densité d'enregistrements à la forme de la BRDF.

Dans le Chapitre 6, nous proposons également une heuristique appelée seuillage des voisins (ou *neighbor clamping* en anglais). En se basant sur la cohérence de la géométrie dans les scènes, cette heuristique est conçue pour limiter la différence maximale de rayon d'influence entre deux enregistrements voisins : la différence des rayons d'influence ne doit pas être supérieure à la distance séparant ces deux enregistrements. Le seuillage des voisins permet d'augmenter significativement la prise en compte des détails géométriques dans l'éclairage indirect. De plus, cette méthode aide à la détection et à la réduction d'un problème commun aux algorithmes de cache d'éclairement et de luminance: fuite de rayons (*ray leaking* en anglais), i.e. lorsqu'une partie des rayons émis pour le calcul d'un enregistrement passe au travers de la jonction entre deux murs de la scène.

Conclusion et Perspectives

La contribution principale de cette thèse est l'algorithme de cache de luminance, permettant de calculer efficacement une solution d'illumination globale dans des scènes comportant des surfaces spéculaires rugueuses. A l'aide de notre méthode, le coûteux calcul d'éclairage indirect n'est pas réalisé pour chaque point visible, mais les valeurs d'éclairage sont interpolées à partir d'un ensemble d'enregistrements. De plus, plusieurs sous-problèmes de l'algorithme de cache de luminance ont été résolus. Ces contributions incluent: de nouvelles méthodes de calcul de gradient translationnel de luminance, une approximation rapide de la rotation de fonctions projetées dans une base d'harmoniques sphériques ou hémisphériques, une méthode de raffinement adaptatif pour les caches d'éclairement et de luminance, et une solution à plusieurs problèmes pratiques d'implémentation.

Il existe de nombreux axes de recherche pour améliorer l'algorithme de cache de luminance. Parmi celles-ci, nous pouvons tout d'abord mentionner la définition d'une méthode automatique pour le choix du nombre de rayons à envoyer lors de l'échantillonnage de l'hémisphère. Par ailleurs, l'hémisphère devrait être échantillonné de manière adaptative, i.e. en envoyant plus de rayons dans les zones de forte variance [48, 86, 85] ou importance [46]. De plus, un phénomène manquant dans l'algorithme de cache de luminance est la prise en charge des caustiques projetées sur des surfaces rugueuses. Pour cela, un critère automatique permettant de décider quels sont les chemins lumineux à suivre est nécessaire. Enfin, lors du calcul d'éclairage indirect, il pourrait être utile de séparer la BRDF en deux composantes : basse fréquence et haute fréquence. L'éclairage indirect de la composante basse fréquence serait calculé à l'aide de l'algorithme de cache de luminance. La composante résiduelle, haute fréquence, serait estimée à l'aide d'un très faible nombre de rayons par la méthode de Monte Carlo en utilisant un échantillonnage d'importance.

La limitation principale de notre algorithme est la restriction des BRDF aux basses fréquences. Une prise en charge efficace des BRDF haute fréquence dans le cache

de luminance pourrait être réalisée à l'aide d'ondelettes sphériques [84]. Toutefois, l'utilisation des ondelettes sphériques dans notre algorithme présente deux problèmes ouverts: la rotation et le calcul de gradients de translation.

VU :

Le Directeur de Thèse

Jiří ŽÁRA

Kadi BOUATOUCH

VU :

Le Responsable de l'Ecole Doctorale

VU pour autorisation de soutenance

Rennes, le

Le Président de l'Université de Rennes 1

Bertrand FORTIN

VU après soutenance pour autorisation de publication :

Le Président de Jury,

Bruno ARNALDI

Abstract: Computation of global illumination is an important means in the pursuit of photorealism of computer-generated images. It is of particular importance in scenes with glossy surfaces, since it conditions correct perception of glossy material characteristics. This dissertation focuses on efficient computation of global illumination in presence of glossy surfaces with low-frequency reflectance functions. Computing global illumination on those surfaces is difficult with pure Monte Carlo sampling because of the sheer number of rays required to get a good estimate of indirect illumination. We propose radiance caching, a ray tracing-based algorithm for computing indirect illumination on glossy surfaces with low-frequency reflectance functions. The algorithm takes advantage of the shading smoothness on these surfaces by interpolating indirect illumination from a set of sparsely distributed, cached values. We discuss all the components necessary for an implementation of radiance caching: (1) an automatic choice of surfaces on which radiance caching is used; (2) compact representation for the cached illumination by spherical harmonics; (3) efficient spherical harmonics rotation that makes interpolation possible; (4) computation of translational gradient for smooth interpolation; and (5) an adaptive criterion for distributing the cached illumination values on surfaces. Radiance caching delivers high quality images in a shorter time compared to existing ray tracing-based methods. It can be used for final gathering in two-pass global illumination methods or as a stand-alone global illumination algorithm.

Keywords: Computer graphics, image synthesis, rendering, global illumination, Monte Carlo, spherical harmonics, radiance caching.

Résumé: L'illumination globale est un moyen permettant de produire des images de synthèse dites photoréalistes. Elle joue un rôle encore plus important dans le cas de scènes contenant des objets en parties spéculaires, c'est à dire non parfaitement lisses. Cette thèse traite principalement du problème du calcul de l'illumination globale dans le cas de ce type de scène où les objets sont caractérisés par des réflectances de basse fréquence. Le calcul de l'illumination globale s'avère très difficile lorsqu'on utilise un échantillonnage de type Monte Carlo car un grand nombre d'échantillons (rayons) est nécessaire pour obtenir une bonne estimation de l'éclairage indirect. C'est pourquoi nous proposons une méthode utilisant un cache de luminance, méthode basée sur le lancer de rayon et prenant en compte les surfaces spéculaires non parfaitement lisses (ayant une rugosité microscopique) dont les réflectances possèdent des caractéristiques de basse fréquence. L'algorithme proposé exploite la variation douce de l'éclairage sur une surface en interpolant l'éclairage indirect à partir de données éparses stockées dans le cache. Tous les outils et composants logiciels nécessaires à la mise en oeuvre de cet algorithme sont présentés dans cette thèse. En effet, il s'agit : du choix des surfaces sur lesquels la sauvegarde dans le cache doit être effectuée, de la représentation compacte des informations sauvegardées à l'aide d'harmoniques sphériques, de la rotation des harmoniques sphériques nécessaires lors de l'interpolation, du calcul de gradient translationnel aussi utilisé lors de l'interpolation, du critère adaptatif permettant de choisir les valeurs à stocker dans le cache. Notre méthode basée sur le cache de luminance permet de générer des images de grande qualité en un temps plus court que celui obtenu avec les méthodes existantes basées sur le lancer de rayon. Elle peut être aussi utilisée pour la collecte finale dans les méthodes à deux passes ou comme un algorithme d'illumination globale complet.

Mots clés: Informatique graphique, images de synthèse, rendu, éclairage global, Monte Carlo, harmoniques sphériques, cache de luminance.