© Copyright 1995
Per Henrik Christensen

Hierarchical Techniques for Glossy Global Illumination

by

Per Henrik Christensen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Washington

1995

Approved by:

(Co-chairperson of Supervisory Committee)

(Co-chairperson of Supervisory Committee)

Program authorized to offer degree:

(date)

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 1490 Eisenhower Place, P.O. Box 975, Ann Arbor, MI, 48106, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature: ______________________________

Date: ______________________________

Hierarchical Techniques for Glossy Global Illumination

by *Per Henrik Christensen*

Chairpersons of supervisory committee:
Professors *David H. Salesin* and *Anthony D. DeRose*
Department of Computer Science and Engineering
University of Washington

Abstract

This dissertation concerns efficient computation of realistic images. To compute realistic synthetic images, the effect of global illumination is essential. Ray tracing algorithms solve the global illumination problem for specular interreflections, and radiosity algorithms solve it for diffuse interreflections. But computing a solution is more complicated when the surfaces are glossy. This dissertation describes hierarchical techniques for efficient solution of the glossy global illumination problem. Two types of hierarchy are utilized: wavelets to accurately represent radiance distributions on surface patches, and clusters to approximately represent radiant intensity from groups of surface patches. Without hierarchical techniques, the solution time would be quadratic in the number of patches and $O(n_b^{1.5})$ in the number of basis functions n_b. The hierarchical techniques make solution time linear in both the number of patches and the number of basis functions. This reduction is significant since the numbers of patches and basis functions are large for accurate solutions in realistic environments. Furthermore, directional importance is used to focus refinement of the solution on parts that contribute significantly to a particular view of the scene. Our method is the first finite-element method capable of handling complex glossy scenes.

Table of Contents

List of Figures **vi**

Notation **vii**

Chapter 1: Introduction **1**

1.1 Global Illumination 2
1.2 Glossy Reflection 4
1.3 Solution Methods 5
1.4 Hierarchical Techniques 11
1.5 Contributions 15
1.6 Dissertation Overview 16

Chapter 2: Global Illumination **19**

2.1 Simplifications 20
2.2 Radiance 21
2.3 Propagation 23
2.4 Reflection 26
2.5 Equilibrium Equations 29
2.6 Further Simplifications 32
2.7 Choice of Representation 34

Chapter 3: Discretization **37**
3.1 Preliminaries 39
3.2 Bases 40
3.3 Dual Bases 42
3.4 The Galerkin Method 44
3.5 Simple Algorithm 47
3.6 Complexity of the Simple Algorithm 50

Chapter 4: Importance **52**
4.1 Motivation 53
4.2 Background and Overview 54
4.3 Adjoints 55
4.4 Two-point Importance 57
4.5 Directional Importance 60
4.6 Diffuse Importance 63

Chapter 5: Wavelet Representation **66**
5.1 Hierarchical Bases 67
5.2 Multiresolution Analysis 69
5.3 Multidimensional Wavelet Bases 75
5.4 The Hierarchical Galerkin Method 84
5.5 Wavelet Algorithm 90
5.6 Refinement 92

Chapter 6: Clustering **96**
6.1 Background 98
6.2 Light Transfer between Clusters 100
6.3 Importance 105
6.4 Bounds 106
6.5 Discretization 111

6.6 Clustering Algorithm 115
6.7 Creation of Clusters 116
6.8 Refinement 118

Chapter 7: Implementation **120**
7.1 Surface Geometry 121
7.2 Light Sources 121
7.3 Reflection Models and Texture Maps 122
7.4 Efficient Computation of Transport Coefficients 123
7.5 Adaptive Numerical Integration 124
7.6 Visibility 125
7.7 Data Structure for Basis Function Coefficients 128
7.8 Data Structure for Transport Coefficients 130
7.9 Final Gather 132

Chapter 8: Results **136**
8.1 Three Patches 137
8.2 Scene of Medium Complexity 141
8.3 Asymptotic Behavior 143
8.4 Accuracy of Cluster Approximation 145
8.5 Results for a Highly Glossy Scene 147
8.6 Comparison with a Monte Carlo Method 148
8.7 A Complex Interior 149

Chapter 9: Conclusion **153**
9.1 Contributions 153
9.2 Future Directions 155

Bibliography **158**

List of Figures

1.1 Diffuse, glossy, and specular reflection 4
1.2 A typical ray traced image 6
1.3 A typical radiosity image 9

2.1 Radiance 22
2.2 Radiance along a ray is invariant 23
2.3 Geometry 24
2.4 Flux 24
2.5 Light propagation 26
2.6 Differential irradiance 27
2.7 Helmholz reciprocity 28
2.8 Light reflection 29
2.9 The radiance equation 30
2.10 The “rendering equation” 31

3.1 Example of a basis 41
3.2 Example of a dual basis 43
3.3 Normalized box functions 44
3.4 Computation of transport coefficient 46

4.1 The direct contribution of radiance distribution L 59

5.1 Haar scaling functions ϕ_i^j 70

5.2 Haar wavelets ψ_i^j 71
5.3 Linear B-splines and splinelets 73
5.4 Coiflets 73
5.5 Multiwavelets 74
5.6 Flatlets 74
5.7 Spatial projection 75
5.8 Angular projection 77
5.9 Radiance distribution before and after angular mapping 77
5.10 A continuous wavelet basis with just one scaling function 79
5.11 Standard and nonstandard basis construction 82
5.12 Two-dimensional Haar functions 82
5.13 Flatland geometry 85
5.14 Matrix for parallel lines 86
5.15 Standard Haar matrix 87
5.16 Error vs. number of transports 88
5.17 Sending and receiving basis functions 94

6.1 Light transport between two clusters 97
6.2 Cluster of patches and exitant point approximation 103
6.3 Cluster of patches and incident point approximation 104
6.4 A cluster with children 117

7.1 Ward's reflection model 123
7.2 Tree of basis function coefficients on a patch 129
7.3 Example of links between basis functions 131
7.4 New links to be considered 131

8.1 Three patches 137
8.2 Refinement of radiance distribution on patch 2 138
8.3 Refinement of radiance distribution on patch 3 139

8.4 Convergence of the radiance distribution 140
8.5 Solutions for scene of medium complexity 142
8.6 Time spent in various stages of the new method 144
8.7 Time spent by different clustering strategies 145
8.8 Glossy sphereflake 146
8.9 Sphereflake computed using our method 147
8.10 Comparison of sphereflake images 149
8.11 Interior with both glossy and diffuse surfaces 150

Notation

Throughout this dissertation, scalars, points and functions are set in slanted typeface, vectors and matrices are set in boldface, and operators are set in calligraphic typeface,

The following symbols and notation are used:

Symbol	Meaning
A_x	Area around point x [m^2]
B	Radiosity [W/m^2]
b	Basis function
$\overline{b}$	Dual basis function
$\mathbf{B}$	Basis for a space
$\widetilde{\mathbf{B}}$	Finite basis for subspace
$\overline{\mathbf{B}}$	Dual basis for a space
E	Irradiance [W · sr/m^2]
f_r	Bidirectional reflectance-distribution function, BRDF [1/sr]
G	Geometric term [sr/m^2]
$\mathcal{G}$	Propagation operator
I	Radiant intensity [W/sr]
	(continued on next page)

(continued from previous page)

I	Identity operator
L	Radiance [W/m^2/sr]
L_e	Emitted radiance
L_r	Reflected radiance
$\mathbf{L}$	Radiance matrix
$\widetilde{\mathbf{L}}$	Finite radiance matrix
L^*	Two-point transport intensity [W/m^4]
$\mathcal{L}^2$	The space of square-integrable functions
n_b	Number of basis functions in the global illumination solution
n_p	Number of patches in the scene
r	Receiving patch
R	Receiving cluster
$\mathcal{R}$	Reflectance operator
s	Sending patch
S	Sending cluster
$\mathcal{T}$	Transport operator
$\mathbf{T}$	Transport matrix
u, v	Parameters of a parametric surface
V^j	Function space
W^j	Orthogonal complement space
x, y, z	Points
$\mathit{\Gamma}$	Directional importance (dimensionless)
$\boldsymbol{\Gamma}$	Importance matrix
$\widetilde{\boldsymbol{\Gamma}}$	Finite importance matrix
$\mathit{\Gamma}^*$	Two-point importance [sr/m^2]

(continued on next page)

	(continued from previous page)
Υ	Importance intensity [m^2]
ϕ_k^j	Scaling function in space V^j with translation k
ψ_k^j	Wavelet in space W^j with translation k
Ψ	Smits' type of diffuse importance
$\vec{\omega}$	Direction
$\vec{\omega}_{xy}$	The direction from point x to point y
$\vec{\omega}_{SR}$	The direction from cluster S to cluster R
Ω	The sphere
$\Omega+$	The hemisphere above a point

Notation	Meaning
$\equiv$	"is defined as"
$\langle f \mid g \rangle$	Inner product of functions f and g
$f \perp g$	Orthogonal functions, $\langle f \mid g \rangle = 0$
O^*	Adjoint operator
f^*	Adjoint function
$O \circ \mathcal{P}$	Operator composition, $(O \circ \mathcal{P})f(x) \equiv O(\mathcal{P}(f(x)))$
$\lceil f(x,y) \rceil_{A,B}$	Upper bound of function f over domain $A \times B$

Acknowledgments

First of all, I would like to thank my two advisors, Professors David Salesin and Tony DeRose, for their inspiration and guidance. David's drive and enthusiasm for computer graphics has affected me deeply and is a great source of motivation. His stream of innovative ideas seems endless. Tony has an overwhelming knowledge of mathematics and physics. He never ceases to impress me with his ability to explain complicated mathematics in clear and simple terms — a rare talent. It was his seminar on wavelets that got me interested in using wavelets for global illumination.

Eric Stollnitz, a graduate student at the Department of Applied Mathematics, joined our research on glossy global illumination at an early stage. He has contributed significantly to the theory and implementation described in this dissertation. Our many discussions of these subjects have been a great help. Also, he has patiently answered many of my questions about terminology.

Dr. Dani Lischinski started as a post-doc in the summer of 1994 and turned out to be an invaluable source of ideas, advice, and programs. His sense of what is practical and feasible was essential to complete the work presented here. Also, his understanding and support of my desire to graduate was extremely helpful.

I would also like to thank Dr. Peter Schröder and Dr. Steven "Shlomo" Gortler for many discussions of wavelet methods by e-mail and in person. They have been very helpful in explaining their research.

Professor Linda Shapiro was my advisor in two years of research in the area of physics-based computer vision, an area I find immensely interesting. Computer vision can be considered "reverse computer graphics": where computer graphics computes images of 3D shapes, computer vision computes 3D shapes from images.

Professor Jørgen Staunstrup is the reason I came to study at the University of Washington. While completing my Master's project at the Technical University of Denmark under his supervision, he urged me to apply to the Department of Computer Science at University of Washington.

Niels-Jørgen Christensen taught my first computer graphics class, and was later advisor of my first computer graphics project, a ray tracer for CSG models. He was the first to suggest to me the need for a clustering method for glossy global illumination. Dr. Per Skafte Hansen's immense knowledge of linear algebra has proved helpful on more than one occasion.

Also, I would like to thank my apartment-mate Henrik for putting up with me through our five years of living together. Finally, I'd like to thank my friends in Seattle for showing me that there is a life outside the University, and my friends in Denmark for staying in touch.

The scene in figure 8.5 was created by Eric Stollnitz. The interior in figure 8.11 was originally created by Matt Hyatt at Cornell University's Program of Computer Graphics, and revised by Eric Stollnitz. Taweewan Siwadune created the palm tree in the same figure.

To my parents, Eva and Evald, and girlfriend My.

Chapter 1

Introduction

One of the fundamental goals of computer graphics is to create artificial images that are indistinguishable from photographs. Given a specification of geometry, emission, and reflection properties in a three-dimensional scene illuminated by some light sources, we want to compute the resulting distribution of light in the scene and create a two-dimensional image of it. Realistic image synthesis is useful for visualizing non-existent environments in fields such as architectural design, interior design, illumination engineering, industrial design, virtual reality, and special effects for movies.

To generate realistic images, it is necessary to accurately simulate the propagation and reflection of light in the three-dimensional scene. By imitating the physics we can approximate the visual impression of the scene. Among the physical effects that need to be modeled are the dependencies of reflected light on the direction, color, and intensity of the incident light; the orientation, material, and texture of the reflecting surface; and the direction of reflection.

"Local illumination" is illumination directly from the light sources. Since there are usually few light sources in a scene, the local illumination at a point in a scene can be computed fairly easily. However, the light emitted from the light sources is reflected in the scene, causing indirect illumination. This indirect illumination means that the light at a point depends not only on light from the light sources, but on light from all points visible to it. To complicate matters even more, the light at these points in turn depends on light at other points, and so on. All in all, the light at a point indirectly depends on light in the entire scene — even from parts of the scene not directly visible to that point. This dependence motivates the term "global illumination".

1.1 Global Illumination

To create realistic images, we need to take global illumination into account. Without global illumination, the objects in the scene would only be illuminated directly from the light sources. Several subtle, but visually important, effects would then be missing from the computed images. "Color bleeding" refers to the effect where one object picks up the tint of another nearby object. For example, a white wall next to a red carpet will appear slightly pink. Without global illumination, there would be no color bleeding. A real shadow of an object is not completely black since other objects in the environment reflect light into the shadow area. Without global illumination, the shadows of objects would be pitch black. When two glossy objects are near each other, the bright highlight from one object reflects so strongly to the other object that a secondary highlight appears. Without global illumination, there would be no secondary highlights. These effects, and other global illumination effects, are essential to convey the illusion of a real scene. Introductions to the global illumination problem can be found in several textbooks [24, 32, 36, 86].

As already described, global illumination problems are challenging because we need a solution at all points in all directions, and because the light at all points depend on each other. The most direct simulation of light transport would trace photons from the light sources through the scene to the eye. However, the number of photons in a non-trivial scene is overwhelming. For example, a 100-Watt light bulb emits on the order of 10^{27} photons in the visible spectrum each second. Simulating such huge numbers is far beyond the capability of even the most powerful computers. Therefore careful approximations have to be made. The approximations should be chosen so that the resulting solution is close to the real solution, while enabling much faster computation. One solution method selects a subset of these photons to simulate. Unfortunately, millions of photons have to be simulated to get images without noticeable noise, especially since only a fraction of the photons are going to reach the eye. Another solution method traces the photon paths backwards from the eye. In this way, one is sure to only simulate photons that actually reach the eye. However, if all types of reflection are simulated, even this solution method requires the tracing of millions of photon paths to create images without noticeable noise. Yet another solution method approximates the infinitely many points and directions as a finite set of small areas and directions and computes an approximate solution for these areas and directions. This results in a very large number of interdependencies.

These solution methods are conceptually simple, but they are not efficient. Considerable effort has been put into developing efficient methods for solving the global illumination problem, and this dissertation is a part of that effort. On a wider scale, the methods that are useful for solving the global illumination problem are also useful for general transport theory and physics simulation. Therefore, new techniques developed for global illumination may turn out to be useful in other fields as well.

1.2 Glossy Reflection

Reflection of light from a surface can be divided into three types: diffuse, glossy, and specular. A diffuse reflection scatters all reflected light evenly in all directions on the hemisphere above the surface point. Specular reflection reflects only in the mirror direction, where the angle of incidence equals the angle of reflectance. Glossy reflection is directional but not restricted to a single direction. These three types of reflection are illustrated in figure 1.1. In computer graphics, glossy reflection is typically modeled using microfacet theory. There is a plethora of glossy reflection models [12, 15, 25, 47, 53, 74, 94, 98]. Any of these models can be used with the techniques developed here, as could any other realistic model.

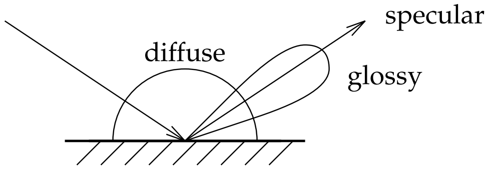


Figure 1.1

Diffuse, glossy, and specular reflection.

Real materials are not perfectly diffuse or specular; therefore, the simulation of glossy reflection is essential for realistic image synthesis. For purely diffuse reflection, one can reduce the dimensionality of the global illumination problem since there is no directional variation. Algorithms for this special case are often referred to as radiosity algorithms. For purely specular reflection, one can exploit the property that light is only reflected in a single direction using ray tracing algorithms. But for glossy reflections, neither advantage can be exploited. So glossy global illumination is, in some sense, the hardest problem of these three. However, as we shall see, many of the same solution techniques that have been developed for diffuse global illumination can also be used for glossy global illumination. The main difference is that the directionality of the reflections increases the dimensionality of the problem.

1.3 Solution Methods

The global illumination problem has been attacked with two fundamentally different solution methods: *Monte Carlo* and *finite elements*. The well-known *ray tracing* and *radiosity* methods are simple versions of these methods, but simulate only special cases of global illumination. Ray tracing is a simple version of the Monte Carlo method. It is efficient, but only simulates purely specular reflections within the scene. The term “radiosity” is used to describe a low-dimensional version of the finite-element method. There are now reasonably efficient radiosity methods, but they only simulate purely diffuse reflection.

1.3.1 Monte Carlo Methods

Monte Carlo methods simulate the transport of photons, either forwards from the light sources or backwards from the eye, and generate the solution by counting how many photons reach each pixel in the image.

The first Monte Carlo method used for global illumination was ray tracing [34, 101]. It traces the photon paths (rays) backwards from the eye. All three types of reflection (diffuse, glossy, and specular), are simulated in the direct illumination from the light sources, but photon paths are only traced backwards in the mirror direction. Therefore, only specular global illumination is simulated. Other types of global illumination are approximated by a constant — the so-called “ambient term”. Thus, all diffuse and glossy interreflections are ignored, meaning that effects such as color bleeding (which is diffuse interreflection) are missing. As a result, ray traced images look too shiny and mirror-like. For an example, see figure 1.2.

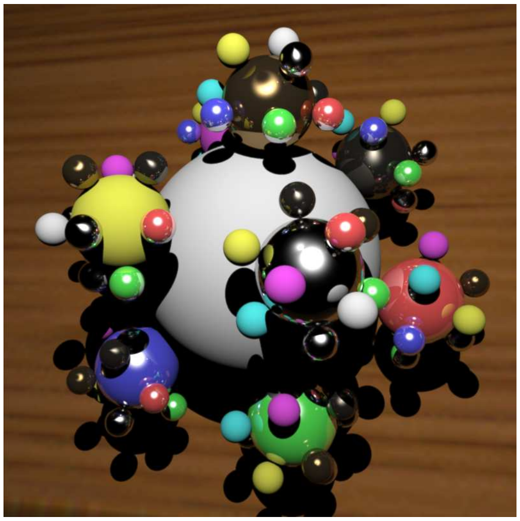


Figure 1.2

A typical ray traced image

To take glossy and diffuse interreflections into account, many more rays have to be traced through the scene. This added complexity is caused by the fact that a photon reflected in some direction from a glossy surface could have come from any direction within a cone before the reflection. Even worse, a photon reflected from a diffuse surface could have come from any direction on the entire hemisphere before the reflection. For specular reflection, in contrast, the photon could only have come from the mirror direction. There are two techniques to simulate general photon transport: *distribution ray tracing* and *path tracing*. Cook [26] introduced distribution ray tracing, where each time a ray hits a surface, a new set of rays is traced. The new rays are cast in the directions where the photon most likely came from. If the contribution along a ray is sufficiently small, no further rays are cast. Nevertheless, this technique gives a combinatorial explosion of rays to trace. Kajiya [54] used path tracing. He observed that if only a single photon path is followed for each intersection, the combinatorial explosion can be avoided entirely. The direction of the new ray is determined statistically from the reflection model, that is, where the photon most likely came from. Ward's

"Radiance" system [99, 100] reuses values from previous rays when a ray hits a surface near a ray that has already been traced. This gives a substantial savings, especially for diffuse interreflections.

Pattanaik [68, 69], Veach and Guibas [96], and others use techniques that simultaneously trace light both from the light sources and backwards from the eye, and compute reflections by combinations of these paths.

In general, Monte Carlo methods that account for all three types of reflection give noisy artifacts. These methods also have slow convergence; in order to reduce variance of the noise by half, four times as many photons have to be simulated.

1.3.2 Finite-Element Methods

In finite-element methods, the unknown light distribution in the scene is represented in a finite basis, and an approximation to the solution is found in that basis. Recent advances such as hierarchical representation [9, 41, 46, 79, 88], importance-driven refinement [17, 89], error control [8, 59], discontinuity meshing [49, 60], etc. have made finite-element methods a promising approach to global illumination. Most of these advances have been for diffuse global illumination, but, as we shall see, they can also be generalized to glossy global illumination.

Goral *et al.* [39] introduced the finite-element solution method for global illumination. They considered the simplified problem of diffuse global illumination, a problem often informally called "radiosity" since radiosity is a characterization of diffuse light. When all emission and reflection is diffuse, light leaving a surface is uniform in all directions, and only spatial variation has to be computed. The finite-element solution method has long been used to simulate thermal radiation [82, 91, 92], a problem very similar to global illumination. Both are transport problems where equilibrium is expressed as an integral equation, and in both cases the solution is radiation from surfaces of objects. Solving is a matter of computing coefficients of the basis functions: The influence of each basis function on every other basis function is computed and arranged in a so-called "form factor" matrix. The form factor matrix defines a finite linear system of equations. Solving this system of equations gives the unknown basis function coefficients and thereby an approximation of the radiosity in the scene. This method for approximate solution of an integral equation is known as the Galerkin method.

This method was improved in two major ways. The first improvement was faster solution of the linear system using Southwell iteration (also known as "progressive refinement" or "shooting" [22, 40]). The second improvement was the use of basis functions that ensure better convergence: orthogonal polynomials (Troutman and Max [95] and Zatz [105]) and piecewise-linear functions with discontinuities along the dominant discontinuities of the solution (Heckbert [49] and Lischinski *et al.* [60]).

Simulating only diffuse reflections is a rough approximation to the general global illumination problem. Images computed with only diffuse reflection contain no highlights, and look matte and dull. An example of an image of a scene with purely diffuse reflection is shown in figure 1.3.

Wallace *et al.* [97] extended the finite-element method to also handle a single specular reflection between diffuse reflections. This was done by taking a single bounce of specular reflection into account when computing form factors between diffuse surfaces. In this way, only diffuse light distributions have to be represented, while some specular reflections are simulated. A second pass using distribution ray tracing computes the chains of specular and glossy reflections that reach the eye. However, glossy-to-diffuse transfers and multiple specular reflections followed by diffuse reflection were not accounted for. Sillion and Puech [84] extended this method to handle diffuse and multiple specular reflections (but still no glossy interreflections). This combination of diffuse

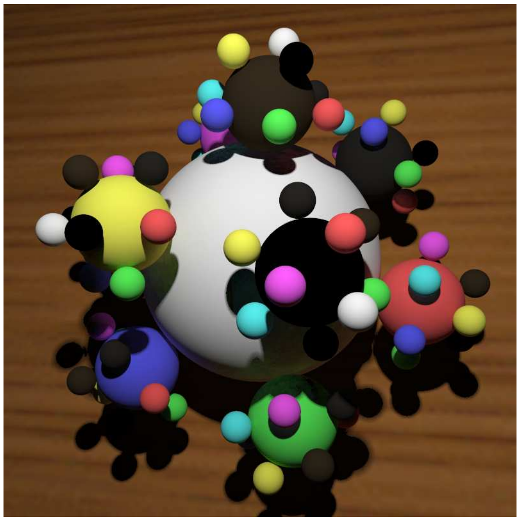


Figure 1.3

A typical radiosity image

and specular reflection was done by taking multiple specular reflections into account when computing the influence of diffuse basis functions on each other (the form factors). Specular surfaces are treated as additional paths along which light can be transported between diffuse surfaces. A final ray tracing pass computes the chains of specular reflections that reach the eye. In this way, only diffuse light distributions have to be represented, while multiple specular reflections are taken into account.

Immel *et al.* [51] were the first to use the finite-element solution method for glossy global illumination. They used piecewise-constant basis functions for both spatial and directional variation. Diffuse reflection is a simple special case of glossy reflection and can easily be handled. After the finite-element solution is computed, it is rendered using interpolation between discrete directions. Since the dimensionality of glossy global illumination is higher than that of diffuse global illumination, more basis functions and light transports are required. However, the transports are sparse; not all basis functions need to interact since a basis function with a given directional support only transports light to basis functions on a fraction of the patches in the scene. Despite this, the method uses a lot of time and space, so only very limited environments could be handled. Shao *et al.* [81] computed a directional solution similar to Immel *et al.*, but followed it by a second pass of distribution ray tracing to improve the image quality by avoiding some discretization artifacts.

Later, Sillion *et al.* [85] extended their method for diffuse and specular reflections to include glossy reflection, resulting in a finite-element solution method capable of handling all three types of reflection. They used spherical harmonics to represent directional variation and piecewise-constant functions to represent spatial variation of light. As mentioned above, diffuse reflection is a simple special case of glossy reflection and can easily be handled by finite-element solution methods for glossy global illumination. Specular reflection is slightly more difficult. Even though glossy reflection is specular in the limit, the finite-element solution method is not suitable for specular reflections since spiky light distributions are difficult to represent in a basis. However, specular reflection can be incorporated by extending the method used for combining diffuse and specular global illumination: multiple specular reflections are simulated when computing the influence of glossy light distributions on each other, and a ray tracing pass computes the specular reflections reaching the eye directly. In this way, the specular reflections are taken fully into account, even though the light distribution of the solution does not contain specular spikes.

In our research, we use the finite-element solution method since it promises better convergence than the Monte Carlo method. We focus on glossy reflections since diffuse reflection is a simple special case and specular reflections can be incorporated along the lines of Sillion *et al.* [84, 85]. If we can solve the glossy global illumination problem more efficiently, the general global illumination problem can also be solved more efficiently.

1.4 Hierarchical Techniques

This dissertation describes efficient hierarchical techniques to improve the finite-element solution method for the glossy global illumination problem. Two hierarchical representations of light are used: *wavelets* for detailed representation of light at surface patches, and a *clusters* for approximate representation of light from collections of surfaces. Furthermore, *importance* is used to refine the hierarchy such that light transports are computed with an accuracy proportional to their contribution to the image.

With nonhierarchical representations of light with spatial and directional variation, light is transported from each basis function to all basis functions on surface patches visible within its directional support. This requirement leads to algorithms where the number of transports is quadratic in the number of patches in the scene and $O(n_b^{1.5})$, where n_b is the number of basis functions in the solution. (The order is $O(n_b^{1.5})$, and not $O(n_b^2)$ as one might expect at first glance, since each basis function only transports light to other basis functions within its directional support.) The asymptotic complexity can be improved using hierarchical techniques: the number of transports becomes linear both in the number of patches and in the number of basis functions. The reduction to a linear number of transports is significant since the numbers of patches and basis functions are large for accurate solutions in realistic environments.

1.4.1 Wavelets

As mentioned above, hierarchical bases reduce the complexity from $O(n_b^{1.5})$ to linear in the number of basis functions n_b. Multiresolution analysis provides firm theoretical ground for describing hierarchical bases. Wavelets are a set of orthogonal basis functions; in a wavelet basis, a function is represented as a coarse overall shape with detail at increasing resolutions. Alpert [3] discovered that in a wavelet representation, a smooth integral operator can be represented as a sparse matrix without introducing significant error. This approximation can be utilized for global illumination, since light transport can be represented as an operator. Wavelets have also been used in as diverse fields as signal analysis, approximation theory, image compression, mathematics, and numerical analysis.

Hierarchical bases were first used for diffuse global illumination by Hanrahan *et al.* [45, 46] and for glossy global illumination by Aupperle and Hanrahan [9]. Wavelets were first used for diffuse global illumination by Gortler *et al.* [41, 78] and for glossy global illumination by Christensen *et al.* [19], Schröder and Hanrahan [79], and Pattanaik and Bouatouch [70]. The application of wavelets to global illumination introduces special requirements in the hierarchical basis, and the trade-offs come out differently than in other fields.

In this dissertation, the criteria for selecting wavelet bases for glossy global illumination are described. A four-dimensional wavelet basis, as required to represent directional light distributions, is constructed from a univariate wavelet basis. The possible choices of operator decomposition are also described. The operator decomposition determines which basis functions can interact. The reasons for choosing a standard operator decomposition are discussed.

1.4.2 Clustering

Realistic environments contain many surface patches, and light must be transported between all these patches to simulate a physical light transport. However, there is no need to transport light *directly* between the surface patches. Significant performance gains can be obtained by clustering groups of patches together and transport light between these clusters. Without a method to cluster surface patches, light has to be transported directly between all patches, resulting in a computational complexity that is quadratic in the number of patches. We use a clustering technique to reduce to linear complexity. The use of error bounds ensures that the cluster approximation is only used if the error introduced by the approximation is acceptable. Initially, light is transported between large clusters. Later, the transports are refined where higher accuracy is needed, so that transports are between smaller clusters or patches.

Clustering techniques were first developed for applications in astrophysics. Appel [5], Barnes and Hut [11], and Greengaard [43] considered efficient solutions to the n-body problem; that is, how n bodies influence each other through, for example, gravitational or electrostatic forces. When the number of bodies is in the thousands or millions, as occurs for simulating development of galaxies, a quadratic complexity is prohibitive. They therefore devised techniques to group bodies into a hierarchy of clusters and to compute the relatively weak forces between distant clusters instead of between all pairs of bodies. In this way, the computational complexity is reduced to linear in the number of bodies.

Similar clustering algorithms for diffuse global illumination have recently been proposed [55, 76, 83, 88, 104]. These techniques extend the hierarchy of basis functions upward to a hierarchy of clusters of surface patches. In many respects, the most successful work to date is that of Smits *et al.* [88]. Their method creates the clusters automatically, uses error bounds to guide the solution process, and has $O(n_p \log n_p)$ complexity in the number of patches n_p. They distinguish between two types of transports: α-links and β-links. To compute a bound for an α-link transport between two clusters, a bound is first computed on the flux density incident on the receiving cluster due to the sending cluster. This is done by bounding the contribution of each source patch (taking into account its area, orientation, and radiance, but ignoring occlusion within the cluster), and summing these contributions. Next, by considering the projected area and maximum reflection of each receiving patch, a bound on the transfer can be computed. The time to compute the bound is linear in the sum of the number of patches in the two clusters, giving a total link cost of $O(n_p \log n_p)$. An asymptotically lower total link cost can be achieved using coarser bounds called β-links. These bounds ignore the orientations of the patches contained in each cluster. For a given transport between clusters, it is assumed that each source patch is directly facing and visible to all the receivers, each receiving patch is highly reflective, and all the patches are as close as possible (within each cluster) to the other cluster. This bound can be computed in constant time, resulting in a total link cost linear in the number of patches. Because this bound is so crude, β-links are only used to represent the most negligible transports between clusters.

In this dissertation, an alternative clustering method that also works for glossy global illumination is described. The geometry of each cluster of surfaces is represented as a point that emits and reflects light according to some directional distribution. Such an approximation is less accurate when considering transports between clusters that are near each other, and is more accurate for distant clusters. The point approximation simplifies the calculation of light transfers between clusters since the geometry inside the cluster is abstracted away. Light is not transported between two clusters when the point approximation is too coarse; instead, light is transported between their subclusters or individual patches. When light has been transported between clusters, the light incident on each cluster is reflected off the patches inside the cluster, and light on the patches is added up to compute the distribution of light leaving the cluster.

1.4.3 Importance

Use of importance-driven refinement reduces the number of transports drastically by reducing the asymptotic constants. With importance, a transport is only refined if it is at a position and direction that contributes significantly to the image and has significant error.

Importance has been used in nuclear engineering since the 1940's [57]. It was introduced to the computer graphics field by Smits *et al.* [89]. They considered a view-dependent solution of the diffuse global illumination problem. By knowing the viewpoint of the image of the scene, the computations can be focused on the light transports that have a high contribution to the image. This does not mean that light not contributing directly to the image is ignored; it is just computed to an accuracy proportional to its contribution. Smits *et al.* defined importance as an adjoint function of radiosity, and developed a transport equation for this type of diffuse importance.

The use of importance has been extended to glossy global illumination. Aupperle and Hanrahan [9] and Pattanaik [68, 69] defined directional importance to be an adjoint of radiance. This type of importance directly gives the contribution of a given radiance distribution. In Christensen *et al.* [17] we found it advantageous to define directional importance in a different way. We define it so that importance is transported exactly like radiance, simplifying the algorithm. Later, Schröder and Hanrahan [79] and Veach and Guibas [96] used the same type of importance for a spatial parameterization of radiance and for a Monte Carlo solution method, respectively.

1.5 Contributions

The global illumination problem is challenging because all light in a scene is interdependent. For scenes with glossy reflections, the problem is even more challenging because of the high dimensionality of the unknown light distributions. Whereas diffuse global illumination is concerned with how light reflecting from every surface point affects light reflecting from all other points, in glossy global illumination one must consider how light reflecting *in every direction* from every surface point affects light reflecting *in all directions* from all other surface points. This higher dimensionality makes solution of global illumination problems in glossy environments much more difficult than in diffuse environments.

The contributions of this dissertation are a variety of techniques to improve the efficiency of the finite-element solution method for glossy global illumination. The techniques are generalizations and extensions of techniques known from the simpler problem of diffuse global illumination. The main contributions are:

- Directional importance: The most obvious type of importance for glossy global illumination is an adjoint of radiance. Instead, a type of importance that is algorithmically more convenient than an adjoint of radiance is introduced.
- Wavelet representation: Wavelet representation of the radiance at each surface reduces the number of transports from $O(n_b^{1.5})$ in the number of basis functions n_b to linear.
- Clustering: Clustering groups of surface patches reduces the number of transports from quadratic in the number of surface patches to linear.

A somewhat smaller contribution concerns adjoints. The radiance equation [24] and Kajiya's rendering equation [54] are alternative descriptions of light transport. They are shown to be adjoint equations, and radiance and two-point transport intensity are shown to be adjoint functions.

1.6 Dissertation Overview

Chapters 2 and 3 contain the background for the contributions of this dissertation: detailed descriptions of the glossy global illumination problem and a simple finite-element solution method. In chapter 2, the propagation and reflection of light is described and the integral equation that governs light transport is presented. There are two alternative formulations of this integral equation:

the radiance equation [24] and Kajiya's rendering equation [54]. Even though these two formulations are equivalent, the radiance equation is preferred since it has a parameterization local to each patch. The radiance equation is continuous; it describes the light at a point (in some direction) as the weighted integral of light from all other points. In chapter 3, a finite-element solution method for this equation is described. If a sufficiently fine basis is chosen, this solution method will iteratively compute an accurate solution, but the number of transports is quadratic in the number of patches and $O(n_b^{1.5})$ in the number of basis functions n_b.

The speed of the finite-element solution method is improved by adaptive hierarchical techniques presented in chapters 4, 5, and 6. In chapter 4, importance and its use for refinement is described. Importance gives the contribution of light to the image, and based on this contribution and an estimate of the error in each light transport, it is decided which transports to refine. The theory of importance is motivated by its use in hierarchical bases, but is derived directly from light transport theory. Chapter 5 covers the adaptive representation of light from a surface patch in a wavelet basis, as well as light transports between wavelets. Several wavelet bases are described, and their advantages and disadvantages for use in the glossy global illumination problem are discussed. Also, the construction of four-dimensional wavelet bases for light (and importance) is described, followed by a description of several operator decompositions for sparse and efficient transports. In chapter 6 the grouping of surface patches into clusters is described. Each cluster is a directional point approximation of the light and importance leaving the surface patches contained in the cluster. Efficient directional bounds are given on light transfers, and these bounds are used for refinement. Together, these three techniques — importance, wavelets, and clustering — enable an initially very coarse solution to be refined only where the improvement is significant to the final image. The use of wavelet representation of radiance makes the number of transports linear in the number of basis functions, and clustering of surface patches makes the number of transports linear in the number of patches.

In chapter 7, an efficient algorithm, using the previously discussed improvements, is presented. Various practical issues such as data structures, visibility computation, and rendering using a final gather pass are also discussed. Results from the implementation are presented in chapter 8. The method is tested on a simple scene to show convergence, and on more complex scenes to demonstrate the ability to handle complicated scenes. The most complex scene has nearly 8,000 surface patches. Finally, conclusions from the research presented here, as well as suggestions for future research, are given in chapter 9.

“LUX SIT” (Let there be light)
Genesis

Chapter 2

Global Illumination

Light is electromagnetic radiation at wavelengths visible to the human eye. In general, light is a function of position, direction, wavelength, phase, polarization, and time. Here we will make several simplifying assumptions, to emphasize the global illumination.

This chapter is a formal description of the aspects of light transport that are of interest when solving the global illumination problem. First, radiance — a measure of light — is defined. Then the propagation and reflection of light is described. Equilibrium distribution of light can be formulated as two equivalent equations, and we choose the equation that is advantageous for our purposes. The simplifications from assuming glossy reflection (no ideal specular reflections) that we will utilize are discussed, and further assumptions and simplifications utilized in the extensive research in diffuse global illumination are mentioned. Radiance can be parameterized directionally or spatially. The directional parameterization gives a smaller number of initial light transports.

2.1 Simplifications

To simplify the simulation of light transport and reflection, we eliminate many of the variables that light depends on. These simplifications are made to emphasize the global and glossy aspects of light transport.

- The scene is assumed to consist of objects in a clear medium such as dry air or vacuum. The medium does not participate in the light transport; all emission, reflection and absorption takes place at surfaces. This gives a reduction of the dimension of the position from three to two since a point on a surface can be specified by two parameters. By contrast, with participating media, such as fog and smoke, light is absorbed and reflected in the medium. Therefore, a three-dimensional (volume) description of the light is necessary. Global illumination in scenes with participating media is considered, for example, by Rushmeier *et al.* [77].
- The band of visible wavelengths is approximated with three discrete wavelengths: red, green, and blue. We also assume that there are no phosphorescent or fluorescent materials in the scene so that we can ignore interaction between wavelengths. These assumptions enable us to consider the three color bands independently, in effect solving three instances of the same (monochrome) problem simultaneously. Peercy [71] used a description of reflection for all visible wavelengths. Glassner [35] and Peercy *et al.* [72] describe the effects of phosphorescent and fluorescent reflection.
- Light propagation is approximated with ray optics. This approximation eliminates effects of wave optics such as diffraction and interference, which are due to phase. Gondek *et al.* [38] modeled interference effects caused by the phase of light.

- Only uniformly polarized light is considered. Polarization can be described as a linear combination of polarization in just two directions, and therefore is not too difficult to model. Each light ray has two parameters, and the reflection model has to take polarization into account. Wolff and Kurlander [103] considered polarization effects in computer graphics.

- A static scene with an equilibrium of light is assumed. Caused by the extremely high velocity of light and the relatively small scenes we consider, light equilibrium occurs very fast after a change in the scene or illumination. With this assumption we can ignore time-dependence in all equations.

Furthermore, transmission of light through surfaces is not modeled. However, this extension is straightforward since transmission does not increase the dimensionality of the problem. It merely doubles the amount of work at each transport since there is both reflection and transmission each time light hits a surface. Wallace *et al.* [97] showed how transmission effects could be incorporated into global illumination computations.

2.2 Radiance

Let x and y be points on surfaces, and let $\vec{\omega}$ and $\vec{\omega}_{xy}$ be directions. The direction from x to y is denoted $\vec{\omega}_{xy}$, thus $\vec{\omega}_{xy} = -\vec{\omega}_{yx}$.

Definition *Radiance:*
Radiance $L(y,\vec{\omega})$ is the power emanating from y per unit solid angle in the direction $\vec{\omega}$ per unit projected area perpendicular to that direction,

$$L(y,\vec{\omega}) \equiv \frac{d\Phi}{d\vec{\omega} \cdot d\vec{A}_y} = \frac{d\Phi}{d\vec{\omega}\, dA_y \cos\theta}\,.$$

For an illustration of these terms, see figure 2.1.

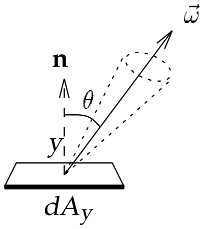


Figure 2.1

Radiance L at point y in direction ω. The surface normal at y is denoted **n**, and the angle between the normal and direction ω is θ.

Radiance is measured in watts per square meter per steradian [W/m^{2}sr]. Radiance is a four-dimensional quantity since a point on a surface can be described by two parameters, and a direction also can be described by two parameters.

The radiance L from a point y in direction $\vec{\omega}$ is the sum of two terms: *emitted radiance* L_e and *reflected radiance* L_r,

$$L(y,\vec{\omega}) = L_e(y,\vec{\omega}) + L_r(y,\vec{\omega})\,.$$

In the global illumination problem, the emitted radiance is known, while the reflected radiance is the unknown we have to determine.

The radiance in the entire scene (or in parts of the scene) can be represented by a *radiance distribution* L. We call this function a "distribution" since it is non-negative everywhere.

Radiance is defined in terms of power emanating from a surface. There is a similar definition for power incident on a surface.

Definition *Incident radiance*[1]*:*
Incident radiance $L^{in}(y,\vec{\omega})$ *is the power incident on* y *per unit solid angle in direction* $\vec{\omega}$ *per unit projected area perpendicular to that direction.*

---

[1] This quantity is also called "field radiance" by some authors [8].

Radiance along a ray is invariant, so if points x and y are mutually visible, we have

$$L^{\text{in}}(y,\vec{\omega}_{xy}) = L(x,\vec{\omega}_{xy})\,, \tag{2.1}$$

as illustrated in figure 2.2.

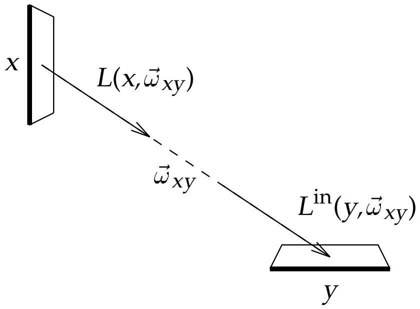


Figure 2.2

Radiance along a ray is invariant.

We shall now see how light is transported in the scene. The transport consists of two parts, propagation from one point to another point, and reflection at the receiving point.

2.3 Propagation

Consider two points x and y that are facing each other and unoccluded. With each point is associated a differential area, dA_x and dA_y respectively. The angles between the line segment xy and the respective normals of dA_x and dA_y are θ_x and θ_y. These terms are illustrated in figure 2.3.

Seen from point x, the solid angle $d\vec{\omega}$ subtended by the differential area dA_y is $\cos\theta_y\, dA_y \big/ \|x-y\|^2$. In the direction towards y, the differential area dA_x has projected area $\cos\theta_x\, dA_x$. Now consider the flux flowing in the beam from dA_x to dA_y; see figure 2.4. This flux is the product of radiance $L(x,\vec{\omega}_{xy})$, the solid an-

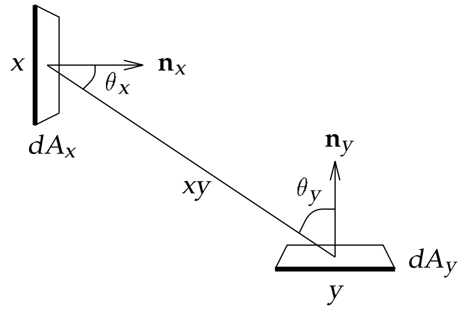


Figure 2.3

Geometry at points x and y. The surface normals of differential areas dA_x and dA_y are $\mathbf{n}_x$ and $\mathbf{n}_y$, respectively. The angles between the line segment xy and the normals of dA_x and dA_y are θ_x and θ_y.

gle subtended by dA_y, and the projected area of dA_x,

$$d\Phi = L(x, \vec{\omega}_{xy}) \frac{\cos\theta_y \, dA_y}{\|x - y\|^2} \cos\theta_x \, dA_x .$$

The flux is zero if the points are occluded or facing away from each other.

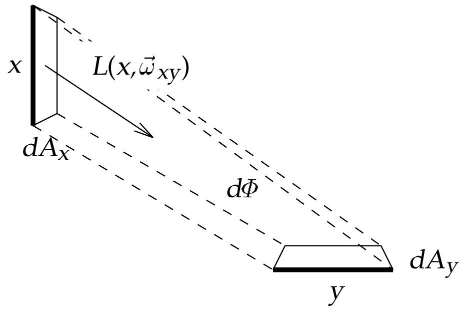


Figure 2.4

Flux from differential area dA_x to differential area dA_y.

To simplify notation, we will use the following definition.

Definition *Geometric term:*

The geometric term $G(x, y)$ is defined as

$$G(x, y) \equiv V(x, y) \cdot \frac{\cos\theta_x \cos\theta_y}{\|x - y\|^2}, \tag{2.2}$$

where $V(x,y)$ is 1 if x and y are mutually visible (that is, the line segment xy is unoccluded and x and y are on surfaces that are facing each other) and 0 otherwise.

The geometric term has units steradian per square meter [sr/m^2], and is symmetric in its arguments:

$$G(x,y) = G(y,x)\,.$$

Now we can write the flux flowing in the beam from dA_x to dA_y as

$$d\Phi = G(x,y)\, L(x, \vec{\omega}_{xy})\, dA_x\, dA_y\,. \tag{2.3}$$

We will now define a quantity that is the product of radiance and the geometric term.

Definition *Two-point transport intensity*[2]*:*
Two-point transport intensity $L^*(x,y)$ from point x to point y is flux per unit differential area at the sender and receiver,

$$L^*(x,y) \equiv \frac{d\Phi}{dA_x\, dA_y} = G(x,y)\, L(x, \vec{\omega}_{xy})\,. \tag{2.4}$$

Two-point transport intensity has units [W/m^4]. It is a function of four variables since each of its endpoints can be described by two parameters. The propagation of light is illustrated in figure 2.5.

If we let $\mathcal{G}$ denote the operator that multiplies a function by the geometric term G, we can write equation (2.4) in operator notation,

$$L^* = \mathcal{G} L\,. \tag{2.5}$$

We call the operator $\mathcal{G}$ the "propagation operator".

---

[2] Kajiya [54] introduced the term "two-point transport intensity". He used the symbol I for this quantity, but I is the standard symbol for radiant intensity. Instead, we use the symbol L^* since two-point transport intensity will turn out to be an adjoint function of radiance L.

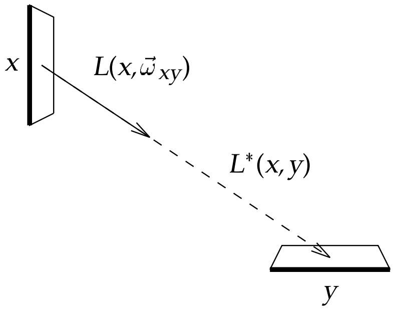


Figure 2.5

Light propagation from point x to point y.

Similar to equations (2.4) and (2.5), we can write the *emitted two-point transport intensity* as $L^*_e(x,y) = G(x,y)\,L_e(x,\vec{\omega}_{xy})$, or simply $L^*_e = \mathcal{G}L_e$.

2.4 Reflection

To describe reflection from a surface, we first have to define a few terms: irradiance, differential irradiance, and bidirectional reflectance distribution function.

Definition *Irradiance:*

Irradiance E is the energy per unit area received from other surfaces,

$$E(y) \equiv \int_{\Omega+} L^{\text{in}}(y,\vec{\omega}) \cos\theta_y \, d\vec{\omega}\,, \tag{2.6}$$

where the integration is over the hemisphere $\Omega+$ *above point y.*

The units of irradiance are watts per square meter [$\mathrm{W/m^2}$].

Differential irradiance is the contribution to irradiance from a differential solid angle. The differential irradiance at a point y from a differential solid angle $d\vec{\omega}$ in direction $\vec{\omega}$ is

$$dE(y,d\vec{\omega}) \equiv L^{\text{in}}(y,\vec{\omega}) \cos\theta_y \, d\vec{\omega}\,,$$

see figure 2.6a. We will now rewrite this expression in terms of areas instead of solid angles. At point y, the differential solid angle subtended by a differential area dA_x is

$$d\vec{\omega}_{xy} = \frac{\cos\theta_x\, dA_x}{\|x-y\|^2}, \tag{2.7}$$

as illustrated in figure 2.6b.

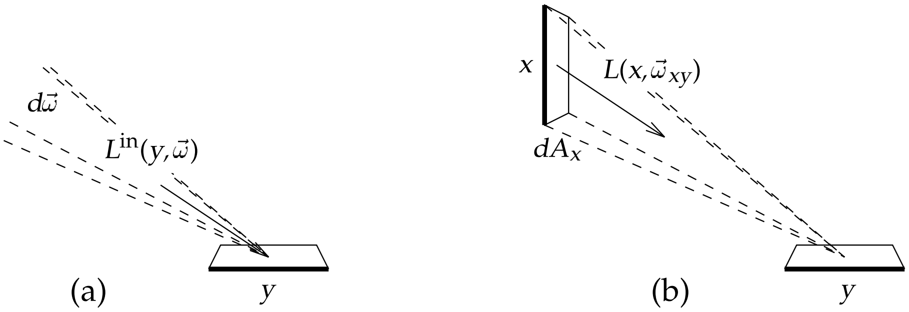


Figure 2.6

Differential irradiance to a point y. (a) From a differential solid angle, (b) from a differential area.

Using equations (2.1), (2.2), (2.4) and (2.7), we can rewrite differential irradiance as

$$\begin{aligned} dE(y, d\vec{\omega}_{xy}) &\equiv L^{\text{in}}(y,\vec{\omega}_{xy}) \cos\theta_y\, d\vec{\omega}_{xy} \\ &= L(x,\vec{\omega}_{xy}) \cos\theta_y \frac{\cos\theta_x\, dA_x}{\|x-y\|^2} \\ &= L(x,\vec{\omega}_{xy})\, G(x,y)\, dA_x \\ &= L^*(x,y)\, dA_x. \end{aligned} \tag{2.8}$$

Now that we know what differential irradiance is, we can define the bidirectional reflectance distribution function.

Definition *Bidirectional reflectance distribution function, BRDF:*
The BRDF is the ratio of reflected radiance (in some direction $\vec{\omega}_r$) to differential irradiance from some solid angle $d\vec{\omega}_i$ of incident directions,

$$f_r(\vec{\omega}_i, y, \vec{\omega}_r) \equiv \frac{dL_r(y,\vec{\omega}_r)}{dE(y, d\vec{\omega}_i)}. \tag{2.9}$$

The BRDF has units [sr^{-1}]. It is called “bidirectional” because it depends on two directions, and “distribution function” because it is non-negative everywhere. As a consequence of Helmholtz reciprocity [64], the BRDF stays unchanged if the incident and exitant directions are swapped[3]:

$$f_r(\vec{\omega}, x, \vec{\omega}') = f_r(-\vec{\omega}', x, -\vec{\omega})\,;$$

see figure 2.7. The Helmholz principle is equivalent to saying that if a photon moves along a path, it will follow the same path if its direction is reversed.

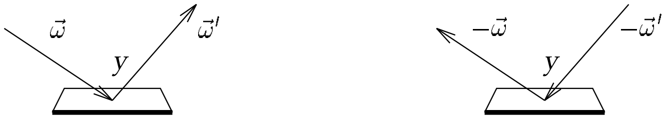


Figure 2.7

Helmholz reciprocity: the BRDF for incident direction ω and exitant direction ω' is the same as for incident direction $-\omega'$ and exitant direction $-\omega$.

The reflected radiance in direction $\vec{\omega}$ is the integral of reflection of differential irradiance from the entire hemisphere above y,

$$L_r(y, \vec{\omega}) = \int_{\vec{\omega}_i \in \Omega_+} f_r(\vec{\omega}_i, y, \vec{\omega})\, dE(y, d\vec{\omega}_i)\,. \tag{2.10}$$

Using the expression for irradiance in equation (2.8), we can rewrite equation (2.10) as

$$L_r(y, \vec{\omega}) = \int_x f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, L^*(x, y)\, dA_x\,. \tag{2.11}$$

The integration is over all points on all surfaces in the scene. This version of reflection is illustrated in figure 2.8.

---

[3] The BRDF is most often expressed with both incident and exitant directions specified by directions pointing away from the reflecting point. This, we feel, is counter-intuitive since the BRDF is defined as a ratio of an incident and an exitant quantity. Here we use the convention that the incident direction points towards the reflecting point, and the reflected direction points away.

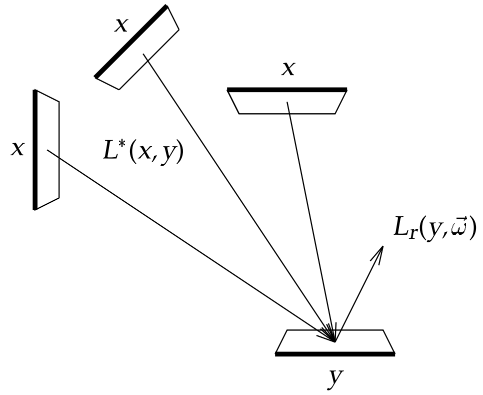


Figure 2.8

Light reflection at point y.

If we let $\mathcal{R}$ denote the operator that multiplies a function by the BRDF and integrates over all surfaces in the scene, we can write equation (2.11) in a simpler form using operator notation as $L_r = \mathcal{R}L^*$. Since radiance is the sum of emitted and reflected terms, $L = L_e + L_r$, we get

$$L = L_e + \mathcal{R}L^* . \tag{2.12}$$

2.5 Equilibrium Equations

We will now describe the equations that have to be solved for the global illumination problem. The equations describe two things at the same time. First, each equation describes the light at a given point, given light everywhere else. Second, each equation describes an equality that holds everywhere at equilibrium. We will describe two equivalent equations: a formulation using radiance giving the radiance equation, and a formulation using two-point transport intensity giving the "rendering equation".

Combining the previously introduced relations, equations (2.5) and (2.12), between radiance and two-point transport intensity,

$$L^* = \mathcal{G}L \qquad \text{and} \qquad L = L_e + \mathcal{R}L^* ,$$

we can write two equivalent equilibrium equations. The first is the *radiance equation*

$$L = L_e + \mathcal{R}\mathcal{G}L, \tag{2.13}$$

or, equivalently,

$$L = L_e + \mathcal{T}L, \tag{2.14}$$

where $\mathcal{T} \equiv \mathcal{R} \circ \mathcal{G}$ is the operator for combined propagation and reflection. We call the operator $\mathcal{T}$ the "transport operator" since it transports light emanating at one surface to light emanating at another surface. Writing the operators in equation (2.13) out in full yields

$$L(y,\vec{\omega}) = L_e(y,\vec{\omega}) + \int_x f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, G(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x.$$

These terms are illustrated in figure 2.9. The radiance equation was used by Immel *et al.* [51], Sillion *et al.* [85], Aupperle and Hanrahan [9] and others.

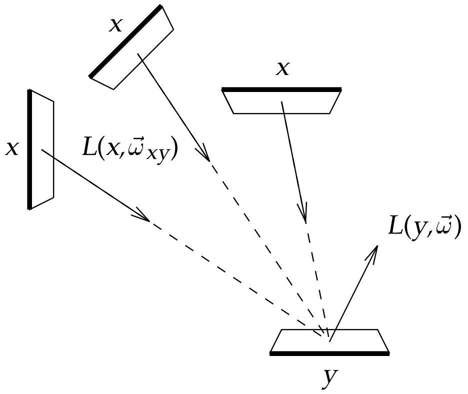


Figure 2.9

The radiance equation: radiance from point y in direction ω depends on radiance from all points x towards y.

The second equilibrium equation that can be created from equations (2.5) and (2.12) was introduced by Kajiya [54], who named it *"the rendering equation"*:

$$L^* = \mathcal{G}L_e + \mathcal{G}\mathcal{R}L^*, \tag{2.15}$$

or, equivalently,

$$L^* = L_e^* + \mathcal{T}^* L^* , \tag{2.16}$$

where we define $\mathcal{T}^* \equiv \mathcal{G} \circ \mathcal{R}$. (It will later turn out that this operator is the adjoint of $\mathcal{T}$, and for this reason we use the symbol $\mathcal{T}^*$.) Writing the operators in this equation out in full yields

$$L^*(y,z) \;=\; L_e^*(y,z) + G(y,z) \int_X f_r(\vec{\omega}_{xy}, y, \vec{\omega}_{yz})\, L^*(x,y)\, dA_x .$$

These terms are illustrated in figure 2.10.

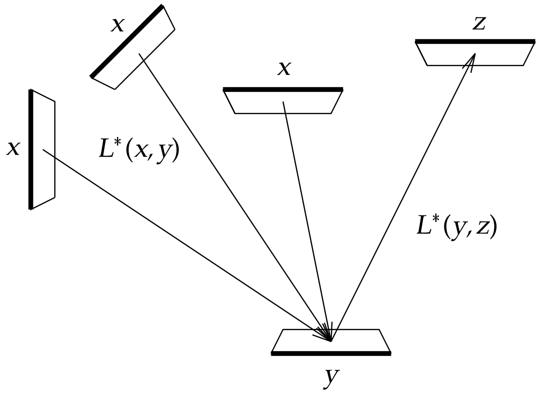


Figure 2.10

The "rendering equation": two-point transport intensity from point y to point z depends on two-point transport intensity from all points x to y.

Both equilibrium equations are multidimensional *Fredholm integral equations of the second kind* [28, 48]. Such equations are in general written

$$f(\mathbf{y}) = g(\mathbf{y}) + \int_D K(\mathbf{x},\mathbf{y})\, f(\mathbf{x})\, d\mathbf{x} ,$$

where the function f is to be determined, and functions g (the "boundary condition") and K (the "kernel") are known. They are integral equations because the unknown function f appears inside the integral. They are Fredholm equations since the domain of integration is constant. And they are of the second kind because the unknown function f is also outside the integral. (A Fredholm integral equation of the first kind is $g(\mathbf{y}) = \int_X K(\mathbf{x},\mathbf{y})\, f(\mathbf{x})\, d\mathbf{x}$.)

Informally, we call a single application of the operator $\mathcal{T}$ or $\mathcal{T}^*$ a "bounce" since it corresponds to one propagation and reflection of light. For example, if L is a radiance distribution, then $\mathcal{T}L$ is that distribution after exactly one bounce.

2.6 Further Simplifications

As it was described in the introduction, reflection can be divided into diffuse, glossy, and specular components. Here we will first describe glossy global illumination, the topic of the remainder of this dissertation. Afterwards, we will discuss the further simplification to purely diffuse reflection since that problem has lower dimensionality, but the same solution method. A lot of previous work has been done for the diffuse case, work that served as inspiration for the solution methods described here.

2.6.1 Glossy Global Illumination

In this dissertation we simplify the general global illumination problem by not simulating specular reflections. As already mentioned, specular reflection can easily be incorporated using a method similar to Sillion *et al.* [85], where the basis functions only store glossy radiance distributions, but specular reflections are incorporated in the transports by other means.

There are many models of glossy reflection, ranging from the very simple Phong model [15] to more accurate models such as the Beckmann-Spizzichino [12], Torrance-Sparrow [94], and other models [25, 47, 53, 74, 98]. Any of these reflection models can be used for our glossy global illumination algorithm. In fact, any model that simulates realistic reflection can be used. We use Ward's reflection model [98] since it is a good compromise between physical accuracy and speed of computation.

Some of these glossy reflection models (including Ward [98]) incorporate *anisotropic* reflection. The reflection from a surface is anisotropic if a rotation (about the normal) of the material changes the reflection. Examples of materials with anisotropic reflection are brushed metal, scales on a butterfly, feathers, and the semiprecious stone "tiger-eye".

It is interesting to note that even realistic models of reflection from real, matte surfaces are not perfectly diffuse; that is, they have some angular variation [66, 102].

2.6.2 Diffuse Global Illumination

Diffuse global illumination is a special case of glossy global illumination and is easily handled by our method. If both the reflection and the emitted radiance are independent of direction, then the radiance everywhere in the scene will be independent of direction, and $f_r(\vec{\omega}, y, \vec{\omega}')$ and $L_e(y, \vec{\omega})$ can be written as $f_r(y)$ and $L_e(y)$, respectively. This simplification means that radiance is two-dimensional instead of four-dimensional since it only depends on position on a surface. As a consequence, we have the simpler *diffuse radiance equation*,

$$L(y) = L_e(y) + \int_x f_r(y)\, G(x,y)\, L(x)\, dA_x\,. \tag{2.17}$$

In this case, the BRDF f_r is independent of x and can be moved outside the integration:

$$L(y) = L_e(y) + f_r(y) \int_x G(x,y)\, L(x)\, dA_x\,. \tag{2.18}$$

This equation might not look much simpler than the general radiance equation (2.13), but since the unknown L is two-dimensional instead of four-dimensional, it is much faster to solve. There has been a significant amount of research on this simplified problem [22, 23, 24, 39, 41, 46, 48, 49, 50, 59, 60, 61, 75, 78, 81, 86, 88, 89, 95, 104, 105].

Traditionally, this equation is often formulated in terms of radiosity B, and we show that formulation here for completeness and comparison. The radiosity of a point is the integral of radiance leaving the point in all directions. For a diffuse surface, radiosity is proportional to radiance,

$$B(x) \equiv \int_{\Omega+} L(x) \cos\theta_x \, d\vec{\omega} = \int_{\Omega+} \cos\theta_x \, d\vec{\omega} \, L(x) = \pi L(x) \, .$$

Substituting radiosity for radiance in equation (2.18), and letting π on both sides cancel, yields

$$B(y) = B_e(y) + f_r(y) \int_x G(x,y) \, B(x) \, dA_x \, .$$

Instead of the BRDF, we can use the reflectance ρ, the ratio of all reflected radiance to irradiance. It can be shown that for diffuse reflection, $\rho(y)$ equals $\pi f_r(y)$ (see Cohen and Wallace [24]). Substituting diffuse reflectance for the BRDF, we get the well-known *radiosity equation*

$$B(y) = B_e(y) + \frac{\rho(y)}{\pi} \int_x G(x,y) \, B(x) \, dA_x \, .$$

The fact that the first finite-element methods in global illumination computed radiosity in diffuse scenes has caused a use of the radiometric term “radiosity” to also mean “a finite-element method for diffuse global illumination”.

2.7 Choice of Representation

Recall from section 2.5 that there are two equivalent formulations of equilibrium: the radiance equation and the rendering equation. In this section, we decide which formulation to use and how to parameterize the unknown distribution. The choice is based on efficiency and the minimum number of transports.

Kajiya [54] introduced the use of rigorous equilibrium equations in global illumination. He expressed his equilibrium equation (the “rendering equation”) using two-point transport intensity. Since then, radiance has been preferred by most researchers [9, 19, 51, 70, 79, 85] because it is a standard term in the radiometry field.

Two fundamentally different parameterizations of radiance have been used for glossy global illumination. Immel *et al.* [51] and Sillion *et al.* [85] represent the light in the scene as directional radiance distributions, a function of two spatial and two angular variables on each surface patch. The radiance distribution in a scene can be divided into radiance distributions on patches; each such radiance distribution is a function of only local variables. By contrast, Aupperle and Hanrahan [9], Pattanaik and Bouatouch [70], and Schröder and Hanrahan [79] represent the radiance from one point towards another. This gives a parameterization with four spatial variables. The relation between directional and spatial parameterization of radiance is simply $L(x,y) = V(x,y)\,L(x,\vec{\omega}_{xy})$.

A clustering approach (like the one presented in chapter 6) reduces the complexity to $O(n_p)$, independent of the parameterization. However, here we will argue that the minimum number of light transports is lower for a directional parameterization than for a spatial parameterization. Even with a hierarchy of clusters, there is a significant minimum number of light transports. The reason is that light should only be transported between clusters that are some minimum distance apart (at the very least, they should not overlap), for the approximation to be valid. With a directional parameterization, the light transports are between pairs of clusters, while with spatial parameterization, the transports are between triples of clusters. For a given scene, the minimum number of non-overlapping pairs of clusters is significantly lower than the minimum number of non-overlapping triples of clusters. In short, the minimum number of transports with clustering is $O(n_p)$ for both parameterizations, but the asymptotic constant is lower for directional parameterization than with spatial. We therefore choose a directional parameterization of radiance, like Immel *et al.* and Sillion *et al.*

Without clustering, the difference between spatial and directional parameterization is even more dramatic. Assume that the scene is initially split into n_p patches. The coarsest possible directional parameterization of radiance has no directionality; it is constant over the entire patch and in all directions on the hemisphere, and requires only one basis function per patch. Therefore, the initial, very rough solution of the radiance transport equation requires $O(n_p^2)$ transports between basis functions. For a spatial parameterization, the coarsest representation at each patch has one basis function in the direction of every other patch, that is, n_p basis functions per patch. At the coarsest level, all n_p basis functions leading to a patch transport light to all n_p basis functions leaving that patch. Since there are n_p patches, $O(n_p^3)$ light transports are required. So with a spatial parameterization, $O(n_p^3)$ transports are necessary, while for a directional parameterization, $O(n_p^2)$ transports are sufficient.

"What is it else? A madness most discreet,
a choking gall and a preserving sweet."
"Romeo and Juliet" by Shakespeare

Chapter 3

Discretization

In the formulation of the glossy global illumination problem described in the previous chapter, the emitted radiance, as well as geometry and reflectance of the scene, are known. The unknown that needs to be solved for is the radiance in the entire scene (at all surface points in all directions).

The solution to the radiance equation $L = L_e + \mathcal{T}L$ is formally $L = (I - \mathcal{T})^{-1}L_e$. Furthermore, reflections are energy dissipating so the operator norm of $\mathcal{T}$ is less than 1, and therefore we know that the inverse, $(I - \mathcal{T})^{-1}$, exists. So one might think that computing the radiance L is simple. However, the complex geometry of nontrivial environments makes the transport operator $\mathcal{T}$ very complicated, so analytical solution is hopeless.

Instead, we have to solve the radiance equation numerically. There are two widespread methods for numerical solution of integral equations: Monte Carlo and finite elements. Monte Carlo methods [34, 67, 96, 99, 100] sample the solution by simulating the transport of selected photons (forwards from the light sources, backwards from the eye, or both). Such sampling methods generally give noisy artifacts and have slow convergence, unless only specular reflections are considered. Recent advances in finite-element techniques (such as hierarchical representation [9, 19, 41, 46, 79, 88, 89], importance-driven refinement [17, 89], error control [8, 59], discontinuity meshing [49, 60], etc.) have made it a promising approach.

We therefore use a finite-element solution method: we approximate the unknown radiance in a finite basis and compute an approximate solution using the Galerkin method. This method replaces the continuous operator $\mathcal{T}$ with a finite matrix $\widetilde{\mathbf{T}}$, whose entries represent the influence of each basis function on every other basis function. So instead of transporting light between infinitely many points, we transport light between a finite number of basis functions. With this finite matrix formulation, the solution can be found by iteratively solving a linear system of equations.

First, we introduce some concepts necessary to describe the discrete solution method. Then we describe how radiance can be represented in a basis. The Galerkin solution method for the radiance equation is described, both for exact and approximate solution. Lastly, a simple iterative solution method with fixed resolution is described.

The basis can be either nonhierarchical [39, 51, 85, 95, 105] or hierarchical [9, 19, 22, 41, 46, 70, 79, 88, 89]. Hierarchical bases enable faster, adaptive solution. In this chapter, we only consider nonhierarchical bases to keep the initial presentation simple. In chapters 5 and 6, more efficient hierarchical bases will be described.

3.1 Preliminaries

In the following, we will describe general concepts such as inner products, orthogonality, and function spaces.The concepts apply to all functions, but we will concentrate on radiance distributions since these are the functions we are interested in here.

We use the following standard definition of an inner product of two functions, although many other inner products are possible.

Definition *Inner product:*
The (standard) inner product of two functions f and g (both defined on domain D) is

$$\langle f \,|\, g \rangle \equiv \int_D f(\mathbf{x})\, g(\mathbf{x})\, d\mathbf{x}\,. \tag{3.1}$$

Example The inner product of two radiance distributions, L_1 and L_2, is

$$\langle L_1 \,|\, L_2 \rangle = \int_{\Omega+} \int_S L_1(x,\vec{\omega})\, L_2(x,\vec{\omega})\, dA_x\, d\vec{\omega}\,,$$

where S is all surfaces in the environment, and $\Omega+$ is the hemisphere above each point x.

To describe hierarchical bases, the concept of orthogonality is necessary.

Definition *Orthogonal functions:*
Two functions f and g are orthogonal if their inner product is zero,

$$\langle f \,|\, g \rangle = 0\,.$$

If f and g are orthogonal, we write $f \perp g$.

Example The functions $f(x) = \sin x$ and $g(x) = \cos x$ are orthogonal on the domain $[0,2\pi]$ since $\langle \sin \,|\, \cos \rangle = \int_0^{2\pi} \sin x \cos x\, dx = 0$.

In order to formalize the concept of hierarchical bases, we need to know to which function space radiance distributions belong.

Definition *Square-integrable functions:*
The space of square-integrable functions is the set of functions f that fulfill

$$\int |f(x)|^2\, dx < \infty .$$

The space of square-integrable functions is often denoted $\mathcal{L}^2$.

Radiance distributions belong to the function space of square-integrable functions since real radiance distributions fulfill the criterion

$$\int_{\Omega+} \int_S |L(x,\vec{\omega})|^2\, dA_x\, d\vec{\omega} < \infty .$$

We also use the terminology that radiance is a square-integrable function and that radiance is an $\mathcal{L}^2$ function.

A linear space with a norm is called a *Banach space*. A *Hilbert space* is a Banach space where the norm comes from an inner product. The norm for the Hilbert space of square-integrable functions $\mathcal{L}^2$ is $\sqrt{\langle f\,|\,f\rangle}$.

Example The norm of a radiance distribution L is

$$\|L\| = \sqrt{\langle L\,|\,L\rangle} = \left[\int_{\Omega+} \int_S L(x,\vec{\omega})^2\, dA_x\, d\vec{\omega} \right]^{1/2} .$$

3.2 Bases

A radiance distribution L can be represented in the basis $\mathbf{B} = [b_1(x,\vec{\omega})\; b_2(x,\vec{\omega})\; \cdots]$ by writing L as a series expansion,

$$L(x,\vec{\omega}) = \sum_{i=1}^{\infty} \ell_i\, b_i(x,\vec{\omega}) , \tag{3.2}$$

where ℓ_i are the coefficients of the representation. Each basis function b is defined on a four-dimensional domain. It has two spatial parameters to describe a point on a surface and two angular parameters to describe a direction. For a given space V, there are many possible sets of orthogonal basis functions that span V. But once the basis functions are chosen, the representation of L is uniquely determined (that is, the coefficients ℓ are unique).

Equation (3.2) can be written in matrix form as $L(x,\vec{\omega}) = \mathbf{B}(x,\vec{\omega})\mathbf{L}$, where $\mathbf{L}$ is an infinite column matrix whose i-th entry is ℓ_i. When no confusion can arise, we suppress the arguments and simply write

$$L = \mathbf{BL}.$$

Examples Let's look at a very simple, finite-dimensional space: the space of linear functions on the interval $[0,1]$. A basis for this space consists of the linear B-splines

$$b_1(x) = 1 - x \qquad \text{and} \qquad b_2(x) = x,$$

see figure 3.1. The linear function $f(x) = 2 + 3x$ on the interval can be represented as $f(x) = 2b_1(x) + 5b_2(x)$.

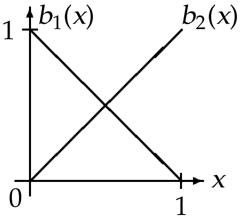


Figure 3.1

Example of a basis $\mathbf{B} = [b_1, b_2]$.

In the original formulation of diffuse global illumination, by Goral *et al.* [39], piecewise-constant functions were used as a basis for spatial variation of radiosity. In subsequent work, Heckbert [48], Zatz [105], and Troutman and Max [95] used orthogonal polynomials to improve the ability of the basis to represent radiosity distributions.

In the more general context of radiance, Immel *et al.* [51] used piecewise-constant basis functions for both spatial and angular variation. Later, Sillion *et al.* [85] used spherical harmonics for the angular variation and piecewise-constant basis functions for the spatial variation. Spherical harmonics are continuous, and therefore give angularly continuous solutions.

Given a radiance distribution L, how do we compute the coefficients ℓ? This is the topic of the following section.

3.3 Dual Bases

One method for computing the coefficients of the representation of a radiance distribution requires a dual basis. To describe dual bases, we first extend the definition of an inner product (3.1) to entire matrices of inner products.

Definition *Matrix of inner products:*
Let $[\langle \mathbf{F} \,|\, \mathbf{G} \rangle]$ *be the matrix whose entries are the inner products of elements of* $\mathbf{F}$ *and* $\mathbf{G}$*. Specifically, if* $\mathbf{F} = [f_1 \; f_2 \; \cdots\,]$ *and* $\mathbf{G} = [g_1 \; g_2 \; \cdots\,]$ *are two row matrices of functions, then* $[\langle \mathbf{F} \,|\, \mathbf{G} \rangle]$ *is the matrix whose ij-th entry is* $\langle f_i \,|\, g_j \rangle$*. Likewise, the inner product of a row matrix of functions with a single function,* $[\langle \mathbf{F} \,|\, g \rangle]$*, is the row matrix consisting of elements* $\langle f_1 \,|\, g \rangle$, $\langle f_2 \,|\, g \rangle$, $\ldots$.

Definition *Dual basis:*
Let $\overline{b}_i$ *be a linear combination of b's such that* $\langle \overline{b}_i \,|\, b_j \rangle = \delta_{ij}$*, where* δ_{ij} *is Kronecker's delta. The function* $\overline{b}_i$ *is the dual function of* b_i*. The dual basis associated with basis* $\mathbf{B}$ *is* $\overline{\mathbf{B}} = [\overline{b}_1(x,\vec{\omega}) \; \overline{b}_2(x,\vec{\omega}) \; \cdots\,]$*. In matrix form, the dual basis is characterized by the relation* $[\langle \overline{\mathbf{B}} \,|\, \mathbf{B} \rangle] = \mathbf{I}$*, where* $\mathbf{I}$ *is the identity matrix.*

The dual basis is unique when it is required to be formed by linear combination of primals, as above.

Example The dual basis corresponding to the previous example consists of the functions

$$\overline{b}_1(x) = 4b_1(x) - 2b_2(x) \qquad \text{and} \qquad \overline{b}_2(x) = -2b_1(x) + 4b_2(x)$$

since with these duals, $\langle \overline{b}_1 \,|\, b_1 \rangle$ and $\langle \overline{b}_2 \,|\, b_2 \rangle$ equal 1, and $\langle \overline{b}_1 \,|\, b_2 \rangle$ and $\langle \overline{b}_2 \,|\, b_1 \rangle$ equal 0. These functions are shown in figure 3.2.

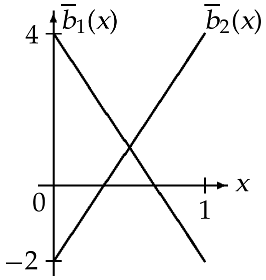


Figure 3.2

Example of a dual basis $\overline{\mathbf{B}} = [\overline{b}_1, \overline{b}_2]$.

Definition *Self-dual bases:*

For orthonormal bases, $\langle b_i \,|\, b_j \rangle = \delta_{ij}$*, so* $\overline{b}_i = b_i$*. Therefore, orthonormal bases are self-dual, meaning that the dual basis is identical to the basis itself,* $\overline{\mathbf{B}} = \mathbf{B}$.

Example A basis for piecewise-constant functions on the interval $[0,1]$ is the n functions evenly dividing that interval:

$$b_i(x) = \begin{cases} \sqrt{n}, & \text{for } \frac{i-1}{n} \le x < \frac{i}{n} \\ 0, & \text{elsewhere} \end{cases}$$

An example of such a basis (with n equal to 4) is shown in figure 3.3. These functions are known as the "normalized box functions". They are orthonormal and therefore self-dual.

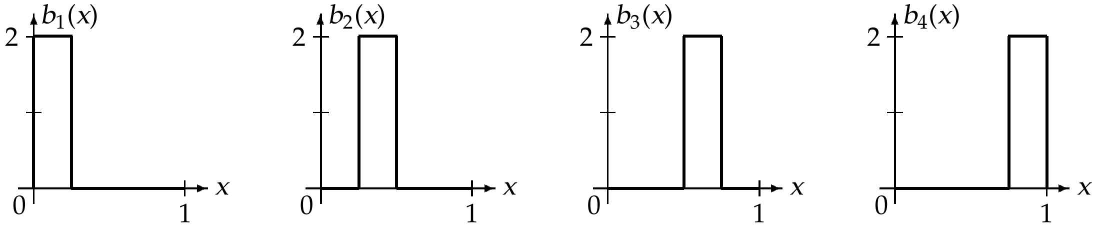


Figure 3.3

Example of a self-dual basis consisting of the normalized box functions $b_1, \ldots, b_4$.

We can use dual functions to compute the coefficients of a representation of a function, for example a radiance distribution. Let b be a set of basis functions for function space $\mathcal{L}^2$, and $\overline{b}$ their duals. The coefficients ℓ_i of the linear combination $L = \sum_{i=1}^{\infty} \ell_i b_i$ are computed as inner products with dual functions,

$$\ell_i = \langle \overline{b}_i \,|\, L \rangle \,,$$

since

$$\langle \overline{b}_i \,|\, L \rangle = \langle \overline{b}_i \,|\, \textstyle\sum_j \ell_j b_j \rangle = \textstyle\sum_j \ell_j \langle \overline{b}_i \,|\, b_j \rangle = \ell_i \,.$$

We will use this type of computation in the following description of the Galerkin method.

3.4 The Galerkin Method

An integral equation can be approximately solved using the *Galerkin method*. We will now describe this method, and show how it can be used to solve the discrete radiance equation exactly or approximately.

3.4.1 Exact Solution

Let $\mathbf{B}$ be a basis for radiance L, and let $\mathbf{BL}$ and $\mathbf{BL}_e$ be the representations of radiance and emitted radiance in this basis. The coefficients $\mathbf{L}_e$ are known since

L_e and $\mathbf{B}$ are known and $\mathbf{L}_e = [\langle \overline{\mathbf{B}} \,|\, L_e \rangle]$. The coefficients $\mathbf{L}$ are the unknowns we want to solve for in order to determine the radiance L. We can obtain a system of equations for the unknown entries of $\mathbf{L}$ by substituting $L = \mathbf{BL}$ and $L_e = \mathbf{BL}_e$ into the radiance equation (2.13), and using linearity of the operator $\mathcal{T}$ to yield

$$\mathbf{BL} = \mathbf{BL}_e + \mathcal{T}(\mathbf{BL}) = \mathbf{BL}_e + (\mathcal{T}\mathbf{B})\mathbf{L}\,.$$

By applying the linear operator $[\langle \overline{\mathbf{B}} \,|\, \cdot \rangle]$ to both sides of this equation, we get

$$[\langle \overline{\mathbf{B}} \,|\, \mathbf{BL} \rangle] = [\langle \overline{\mathbf{B}} \,|\, \mathbf{BL}_e \rangle] + [\langle \overline{\mathbf{B}} \,|\, (\mathcal{T}\mathbf{B})\mathbf{L} \rangle]\,.$$

Using linearity and the duality relation $[\langle \overline{\mathbf{B}} \,|\, \mathbf{B} \rangle] = \mathbf{I}$, we arrive at the *discrete radiance equation*,

$$\mathbf{L} = \mathbf{L}_e + \mathbf{TL}\,. \tag{3.3}$$

In this infinite system of linear equations, $\mathbf{T} \equiv [\langle \overline{\mathbf{B}} \,|\, \mathcal{T}\mathbf{B} \rangle]$ is an infinite matrix representing the transport operator $\mathcal{T}$. We call this matrix the *transport matrix*.

3.4.2 Transport Coefficients

The *rs*-th entry of $\mathbf{T}$ is a *transport coefficient*, representing the influence of the coefficient of b_s on the coefficient of b_r. It can be written explicitly as

$$\begin{aligned} T_{r \leftarrow s} &= \langle \overline{b}_r \,|\, \mathcal{T} b_s \rangle \\ &= \int_{\vec{\omega}} \int_y \overline{b}_r(y, \vec{\omega})\, (\mathcal{T} b_s)(y, \vec{\omega})\, dA_y\, d\vec{\omega} \\ &= \int_{\vec{\omega}} \int_y \overline{b}_r(y, \vec{\omega}) \int_x f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, G(x, y)\, b_s(x, \vec{\omega}_{xy})\, dA_x\, dA_y\, d\vec{\omega}\,, \end{aligned} \tag{3.4}$$

where the notation "$r \leftarrow s$" is to emphasize that $T_{r \leftarrow s}$ represents the influence of the *sender* s on the *receiver* r. In this integral, the domain of x is the spatial support of the sending basis function b_s; the domain of y is the spatial support of the dual receiving basis function $\overline{b}_r$; and the domain of $\vec{\omega}$ is the angular support of $\overline{b}_r$ (directions on a hemisphere above y). Figure 3.4 shows an example of the meaning of a transport coefficient. With the directional parameterization, most transport coefficients are known *a priori* to be zero since the directional support of the sending basis function only coincides with the spatial support of a small fraction of the other basis functions. Therefore, the transport matrix is sparse. In section 3.6 we argue that $O(n_b^{1.5})$ out of the $O(n_b^2)$ elements are nonzero.

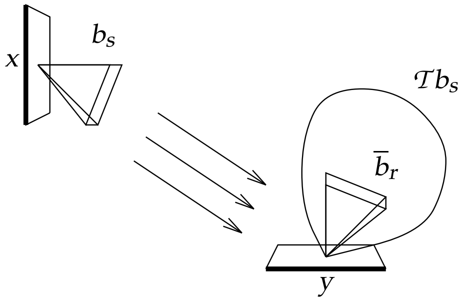


Figure 3.4

Computation of transport coefficient between two basis functions: the inner product of $\mathcal{T}b_s$ and $\overline{b}_r$.

3.4.3 Approximate Solution

Above, we used an exact representation **LB** of L in the infinite basis **B**. In such an infinite basis, the discrete radiance equation (3.3) is an exact solution to the radiance equation (2.13). Now we will use a finite basis to get an approximate solution of the radiance equation. We approximate L in a finite basis $\widetilde{\mathbf{B}} = [b_1(x,\vec{\omega})\cdots b_n(x,\vec{\omega})]$ as $L \approx \widetilde{\mathbf{B}}\widetilde{\mathbf{L}}$.

The *truncated radiance equation* is

$$\widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}_e + \widetilde{\mathbf{T}}\widetilde{\mathbf{L}}. \tag{3.5}$$

This approximate solution method for the original integral equation (2.13) is known as the Galerkin method. The Galerkin method seeks a solution that is exact once $\mathcal{T}$ and L_e have been approximated in the finite-dimensional space spanned by the finite basis $\widetilde{\mathbf{B}}$.

The accuracy of a solution can be improved either by introducing more basis functions with smaller support, by using higher-order basis functions with the same support (Heckbert [48], Zatz [105], and others), or by choosing basis functions that more closely match the solution (such as discontinuity meshing [49, 60]).

3.4.4 Related Methods

There are a number of related methods to solve discretized equations approximately. They differ in the requirement to the solution, more specifically to the *residual*

$$r \equiv \widetilde{\mathbf{B}}\widetilde{\mathbf{L}}_e + \mathcal{T}\widetilde{\mathbf{B}}\widetilde{\mathbf{L}} - \widetilde{\mathbf{B}}\widetilde{\mathbf{L}}\,.$$

- The Galerkin method insures that the residual is orthogonal to the basis functions b_i.
- The *least squares* method ensures that the residual is orthogonal to the images of the basis functions, $\mathcal{T}b_i$.
- The *point collocation* method ensures that the residual is zero at a finite number of points and directions, $r(x_i, \vec{\omega}_i) = 0$.

The term *finite elements* is a common name for all these methods. More on the theory behind the Galerkin method and other finite-element methods can be found in Delves and Mohamed [28] and the excellent course notes by Arvo [6].

3.5 Simple Algorithm

Here we will describe a simple, iterative, nonhierarchical solution method for the truncated radiance equation. Later, when we improve the solution method with hierarchical bases and adaptive refinement, this simple algorithm will be the core of the improved algorithms. The simple solution method has four components:

1. Decide on the finite basis $\widetilde{\mathbf{B}}$ for radiance in the scene.
2. Compute the influence of each basis function on every other basis function by forming the matrix $\widetilde{\mathbf{T}} = [\langle \overline{\widetilde{\mathbf{B}}} \,|\, \mathcal{T}\widetilde{\mathbf{B}} \rangle]$.
3. Solve the truncated radiance equation $\widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}_e + \widetilde{\mathbf{T}}\widetilde{\mathbf{L}}$.
4. Render an image of the radiance distribution $\widetilde{\mathbf{B}}\widetilde{\mathbf{L}}$.

The matrices $\widetilde{\mathbf{L}}$, $\widetilde{\mathbf{L}}_e$, and $\widetilde{\mathbf{T}}$ are approximate, both because they are truncated versions of infinite matrices and because their entries are computed numerically.

Choosing a good finite basis $\widetilde{\mathbf{B}}$ (in step 1) would seem to require *a priori* knowledge of the final radiance distribution. Thus, in practice, global illumination computations often require some degree of manual intervention: a coarse solution is computed; the user determines where the solution needs to be more accurate; then a new solution is found; etc. It can take several trials to get an acceptable solution. In contrast, with an automatic, adaptive solution method, as described in the following chapters, this manual intervention can be avoided.

Step 2, computation of the entries of the transport matrix $\widetilde{\mathbf{T}}$ was described in section 3.4.2.

Let us now elaborate on step 3. The truncated radiance equation (3.5) can be solved directly as

$$\widetilde{\mathbf{L}} = (\mathbf{I} - \widetilde{\mathbf{T}})^{-1}\widetilde{\mathbf{L}}_e .$$

However, inversion of the matrix $\mathbf{I} - \widetilde{\mathbf{T}}$ (using Gauss elimination) requires $O(n_b^3)$ operations in general, where n_b is the number of basis functions. This is much too expensive for accurate solutions in nontrivial environments.

Fortunately the energy-absorbing nature of reflections helps us. Since some energy gets absorbed at each reflection, the norm of the transport operator $\mathcal{T}$ is less than 1. This means that the inverse can be expressed as a Neumann series, $(\mathbf{I} - \widetilde{\mathbf{T}})^{-1} = \sum_{i=1}^{\infty} \widetilde{\mathbf{T}}^i$. We can then express the radiance as an infinite sum

$$\widetilde{\mathbf{L}} = \left(\sum_{i=1}^{\infty} \widetilde{\mathbf{T}}^i \right) \widetilde{\mathbf{L}}_e = \sum_{i=1}^{\infty} (\widetilde{\mathbf{T}}^i \widetilde{\mathbf{L}}_e) .$$

This sum can be computed iteratively,

$$\widetilde{\mathbf{L}}^{(0)} \leftarrow \widetilde{\mathbf{L}}_e \qquad \widetilde{\mathbf{L}}^{(k+1)} \leftarrow \widetilde{\mathbf{T}}\widetilde{\mathbf{L}}^{(k)} + \widetilde{\mathbf{L}}_e .$$

Furthermore, only few iterations are required to obtain a good approximation to the solution. The less reflective the surfaces in the scene are, the faster the solution converges. Each multiplication requires $O(n_b^{1.5})$ operations since there are $O(n_b^{1.5})$ nonzero elements in the transport matrix $\mathbf{T}$ (as shown in section 3.6). So the solution time is $O(n_b^{1.5})$ with the constant depending on the reflectivity in the scene. This iterative method is known as Jacobi solution.

We can improve the Jacobi method by using new values as soon as they are computed. This is called the *Gauss-Seidel* method:

$$\widetilde{\mathbf{L}}^{(0)} \leftarrow \widetilde{\mathbf{L}}_e \qquad L_r^{(k+1)} \leftarrow L_{e,r} + \sum_{s=1}^{r-1} T_{rs} L_s^{(k+1)} + \sum_{s=r+1}^{n} T_{rs} L_s^{(k)}$$

This method can be implemented efficiently by substituting new values in place. Golub and van Loan [37] describe these iterative solution methods in more detail.

Step 4, rendering the resulting radiance distribution $\widetilde{\mathbf{B}}\widetilde{\mathbf{L}}$ is straightforward: cast a ray from the eye point through each pixel in the image. For each ray, find the nearest intersection point with a surface in the scene, and evaluate the radiance distribution there in each of the red, green, and blue color bands. These values give the color of the pixel. If an anti-aliased image is desired, several rays can be cast through each pixel, and the radiance values averaged. More advanced rendering techniques can be used instead, for example Gouraud shading [42] or a final gather as described in section 7.9.

The following pseudocode illustrates this solution method using the Gauss-Seidel iterative solution:

```
procedure GlossyGlobalIllumination:
    choose finite basis $\widetilde{\mathbf{B}}$
    $\widetilde{\mathbf{T}} \leftarrow [\langle \overline{\widetilde{\mathbf{B}}} \,|\, \mathcal{T}\widetilde{\mathbf{B}} \rangle]$
    $\widetilde{\mathbf{L}} \leftarrow \widetilde{\mathbf{L}}_e$
    Solve($\widetilde{\mathbf{L}}, \widetilde{\mathbf{L}}_e, \widetilde{\mathbf{T}}$)
    RenderSolution($\mathbf{BL}$)
end procedure

procedure Solve($\widetilde{\mathbf{L}}, \widetilde{\mathbf{L}}_e, \widetilde{\mathbf{T}}$):
    repeat
        for for each basis function coefficient $\ell_r$ do
            $\ell_r \leftarrow \ell_{e,r} + \sum_s T_{rs}\ell_s$
    until convergence of $\widetilde{\mathbf{L}}$
end procedure
```

3.6 Complexity of the Simple Algorithm

The simple algorithm is quadratic in the number of surface patches since transport coefficients are computed between all scaling functions on mutually visible patches. It is also simple to see that the algorithm is $O(n_b^{1.5})$ in the number of basis functions n_b. Let s be the number of basis functions in each dimension on each patch. Then there are $\Theta(s^4)$ four-dimensional basis functions on each patch. Consider a point on a sending patch. Only $O(s^2)$ basis functions will have spatial support at that point, and their directional supports divide the hemisphere above the point. Between them, these $O(s^2)$ basis functions can at most transport light to all $O(s^4)$ basis functions on the receiver. So there are at most $O(s^4)$ transports from basis functions covering a given point to another patch. The sending patch is divided into $O(s^2)$ different spatial supports, so the total number of possible transports between two patches is $O(s^6)$. Since the number of transports is $O(s^6)$ and the number of basis functions n_b is $\Theta(s^4)$, the number of transports is $O(n_b^{1.5})$. This result was also indicated in Immel *et al.* [51].

In the following chapters, we will improve this basic algorithm with two hierarchical approaches (wavelets and clustering), importance-driven refinement, and a final gather in the rendering stage.

“We come now, however, to a point which is of importance.”
Memoirs of Sherlock Holmes

Chapter 4

Importance

For global illumination, we are often interested in generating an image from just a single viewpoint, or a few images from a restricted set of views. Knowledge of the viewpoint can be exploited by computing a *view-dependent* solution, where the accuracy of different parts of the solution is determined by their potential contribution to the image. How can we assure high accuracy of the radiance solution where needed, but avoid wasting effort by computing radiance to high accuracy everywhere? The answer is *importance-driven refinement*.

This chapter describes the theory of importance for glossy global illumination. First we give the background for the use of importance, and further motivate it. Then we introduce a convenient mathematical notation, the concept of adjoints. It follows directly from the definition of adjoints that the radiance equation and Kajiya’s “rendering equation” [54] are adjoint equations. We then define two equivalent types of importance. One is the adjoint of radiance, and the other is related to the adjoint in the same way as radiance is related to two-point transport intensity. Even though these two types of importance are equivalent, the second type is most convenient for glossy global illumination computations. Finally, we describe the simpler case of importance for diffuse global illumination. It turns out that irradiance is the adjoint of radiosity, and since the diffuse importance of Smits *et al.* [89] is also an adjoint of radiosity, it is transported like irradiance.

4.1 Motivation

For refinement, we want to know how much each part of the radiance distribution contributes to the image. This contribution is useful to see if it is worthwhile refining the solution at the (spatial and angular) support of that part. If the contribution is small, there is no need to refine there.

Each part of the radiance distribution contributes to the image through multiple bounces in the scene. To determine this contribution directly, we would have to solve the global illumination problem with each part as the only emitter, and see how much light reached the eye after multiple bounces. But this computation requires solving a global illumination problem for each part of the solution we consider for refinement, hardly a desirable method.

However, there is a surprisingly simple alternative that builds on the reciprocity of light transport: Turn off the light sources, shine light of radiance 1 from the image in the directions that are within the field of view, and let that light bounce multiple times in the scene (reaching equilibrium). Now the contribution of radiance from each part of the solution is proportional to the amount of light reaching that part. The advantage of solving this global illumination problem is that with this solution, we can simply compute the contribution of each part of the solution, without solving a new global illumination problem for each part we consider for refinement. Still, we have to solve one global illumination problem (for importance) in order to make the solution of another global illumination problem (for light) more efficient. Where is the big win? The big win is that we only solve each problem coarsely at first, and then use these solutions together to guide the refinement of both. In this way, light is only refined where important, and importance is only refined where bright.

The advantage of using importance-driven refinement is even larger for the glossy global illumination problem than for diffuse global illumination: directional radiance is only refined if it is in an important part of the scene *and in an important direction*.

4.2 Background and Overview

The first work on importance was done in nuclear physics (neutron transport simulation); see for example Lewins [57]. Lewins mentions that the term "importance" was coined by Soodak [90].

Smits *et al.* [89] introduced the use of importance for diffuse global illumination. Here we develop a directional type of importance for glossy global illumination.

The simplest definition of importance is as the adjoint of radiance. We denote this type of importance Γ^*. By definition, it is transported like two-point transport intensity L^*. We then proceed to define a different, but related, kind of importance that is transported like radiance L. This type of importance is denoted Γ.[1] These two types of importance are equally "valid"; both can be used to compute how much a given radiance distribution contributes to the image. With the first type, the contribution is computed as a simple inner product, while it will turn out that the second type requires multiplication by the geometric

---

[1] We chose the capital letter gamma as the symbol for importance since importance Γ and Γ^* is transported like radiance L and two-point transport intensity L^*, respectively, and since "Γ" is an "L" upside down.

term G in the inner product. We nevertheless prefer the second definition, Γ, since it is algorithmically convenient to transport importance like radiance — the same data structures and the same algorithm can be used.

More formally, we define importance as a dimensionless distribution, emitted from the eye and transported like light. An inner product of importance and a radiance distribution gives the power contributed to the image by that radiance distribution. Importance and radiance are defined as solutions to the same equation; only the boundary conditions are different.

4.3 Adjoints

In order to prove properties about importance, the concept of adjoints is a convenient mathematical formalism. We define adjoint and self-adjoint operators, and also adjoint equations and functions. In addition, we give some algebraic rules for operators. Algebra textbooks, for example Lang [56] and Lipschutz [58], give more complete descriptions of adjoints.

Recall from section 3.1 that the inner product of two functions f and g, both defined on domain D, is

$$\langle f \,|\, g \rangle \equiv \int_D f(x)\, g(x)\, dx\,.$$

Definition *Adjoint operators:*
Two operators O and O^* are adjoint (with respect to the inner product) if

$$\langle f \,|\, Og \rangle = \langle O^* f \,|\, g \rangle$$

for all functions f and g.

The adjoint O^* of an operator O is unique [56, 57]. Using the definition of adjoint operators, it is easy to verify the following three algebraic rules: The adjoint of the adjoint of an operator is the operator itself,

$$(\mathcal{O}^*)^* = \mathcal{O}.$$

The adjoint operator is linear, that is, the adjoint of a sum of operators is the sum of the adjoint operators,

$$(\mathcal{O} + \mathcal{P})^* = \mathcal{O}^* + \mathcal{P}^* .$$

Furthermore, the adjoint of an operator composed of two operators is the composition of the adjoint operators in reverse order,

$$(\mathcal{O} \circ \mathcal{P})^* = \mathcal{P}^* \circ \mathcal{O}^* .$$

Definition *Self-adjoint operator:*
The operator $\mathcal{O}$ is self-adjoint if $\mathcal{O}^* = \mathcal{O}$.

With these definitions, we can now show that the operators for propagation and reflection are both self-adjoint, and that $\mathcal{T}$ and $\mathcal{T}^*$ are adjoint.

Lemma *The propagation operator $\mathcal{G}$ is self-adjoint.*
Proof Utilizing the symmetry of the geometric term G, we get

$$\begin{aligned} \langle L_1 \,|\, \mathcal{G} L_2 \rangle &= \int L_1(y, \vec{\omega}_{yx})\, G(x,y)\, L_2(x, \vec{\omega}_{xy})\, dA_x\, dA_y \\ &= \int G(y,x)\, L_1(y, \vec{\omega}_{yx})\, L_2(x, \vec{\omega}_{xy})\, dA_x\, dA_y \\ &= \langle \mathcal{G} L_1 \,|\, L_2 \rangle \end{aligned}$$

□

Lemma *The reflectance operator $\mathcal{R}$ is self-adjoint.*
Proof Reordering integrals and using Helmholz reciprocity of the BRDF f_r, we get

$$\begin{aligned} \langle L_1^* \,|\, \mathcal{R} L_2^* \rangle &= \int_{zy} L_1^*(z,y) \int_x f_r(\vec{\omega}_{xy}, y, \vec{\omega}_{yz})\, L_2^*(x,y)\, dA_x\, dA_y\, dA_z \\ &= \int_{xy} \int_z f_r(\vec{\omega}_{zy}, y, \vec{\omega}_{yx})\, L_1^*(z,y)\, dA_z\, L_2^*(x,y)\, dA_y\, dA_x \\ &= \langle \mathcal{R} L_1^* \,|\, L_2^* \rangle \end{aligned}$$

□

Theorem *$\mathcal{T}^*$ and $\mathcal{T}$ are adjoint operators.*

Proof Since $\mathcal{T}^* \equiv \mathcal{G} \circ \mathcal{R} = \mathcal{G}^* \circ \mathcal{R}^* = (\mathcal{R} \circ \mathcal{G})^*$ and $\mathcal{T} \equiv \mathcal{R} \circ \mathcal{G}$ it follows that $\mathcal{T}^*$ and $\mathcal{T}$ are adjoint. (This was shown in [17, 79, 96].) □

Adjoint equations and adjoint functions are defined as follows:

Definition *Adjoint equations and adjoint functions:*
For an equation

$$g = \mathcal{O}f$$

the adjoint equations are the set of equations

$$h = \mathcal{O}^* f^*$$

for all functions h. Two functions f and f^* are adjoint functions if they satisfy adjoint equations. (The source term of the adjoint equation, h above, can be arbitrary.)

Since adjoint equations are not unique, neither are adjoint functions. However, given an adjoint equation and a specific source term h, the adjoint function f^* is unique if $\mathcal{O}^*$ is invertible.

Examples The radiance equation (2.13) and the rendering equation (2.15) are adjoint equations. Radiance L and two-point transport intensity L^* are adjoint functions since they satisfy adjoint equations.

4.4 Two-point Importance

Smits *et al.* [89] introduced importance for diffuse global illumination as the adjoint of radiosity. Here we will first show the most straightforward generalization to glossy global illumination, importance as the adjoint of radiance.

Definition *Two-point importance:*

Two-point importance Γ^* is an adjoint function of radiance L that satisfies an adjoint equation of the radiance equation

$$\Gamma^* = \Gamma_e^* + \mathcal{T}^* \Gamma^* \tag{4.1}$$

Equation (4.1) is of the same form as the rendering equation (2.15), but with a different source term.

We want emitted importance to be defined such that the inner product

$$\langle \Gamma_e^* \,|\, L \rangle = \int \Gamma_e^*(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x\, dA_y$$

is the power contributed directly to the image by radiance L. At the same time, we know that the direct ("no-bounce") contribution of radiance $L(x,\vec{\omega}_{xy})$ to the power reaching the image is given by equation (2.3),

$$d\Phi = G(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x\, dA_y$$

if y is on the image and $\vec{\omega}_{xy}$ points towards the eye point e. Hence we get the following definition of the emitted two-point importance.

Definition *Emitted two-point importance:*

$$\Gamma_e^*(y,x) \equiv \begin{cases} G(y,x), & \text{if } y \text{ is on the image and on the line segment } ex, \\ 0, & \text{otherwise.} \end{cases}$$

With this definition, the units of Γ^* are the same as the units of the geometric term, steradian per square meter [sr/m^2]. The inner product $\langle \Gamma_e^* \,|\, L \rangle$ is illustrated in figure 4.1.

With this definition of emitted importance, the importance of directly visible points depends on their distance and orientation. This is desirable since an incorrect radiance is less significant if it covers only a single pixel than if it covers many pixels.

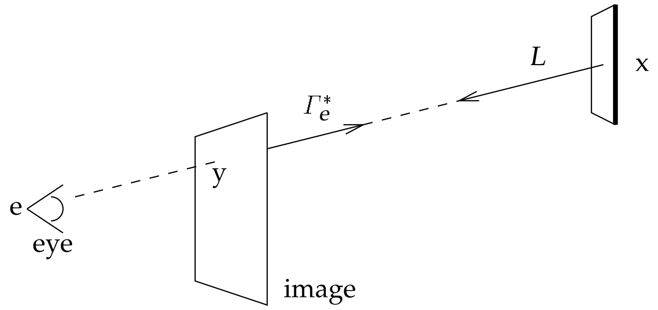


Figure 4.1

The direct contribution of radiance distribution L to the image is the inner product (over points in the scene and points on the image) of emitted importance Γ_e^* and the radiance distribution L.

Now we need to prove that the expression for the direct contribution can be generalized to total contribution through infinitely many bounces.

Theorem *The power that a radiance distribution* L *contributes to the image is the inner product* $\langle \Gamma^* \mid L \rangle$.

Proof Since reflections are energy dissipating, the norm of $\mathcal{T}^*$ is less than unity. Therefore $I - \mathcal{T}^*$ is invertible (I is the identity operator), and the inverse can be expressed as a Neumann series, $(I - \mathcal{T}^*)^{-1} = \sum_{i=0}^{\infty} (\mathcal{T}^*)^i$. This sum can be rewritten using linearity of the adjoint as $\left(\sum_{i=0}^{\infty} \mathcal{T}^i\right)^*$. It follows that we can rewrite the equilibrium equation for two-point importance (4.1) in the form

$$\Gamma^* = (I - \mathcal{T}^*)^{-1}\,\Gamma_e^* = \textstyle\sum_{i=0}^{\infty} (\mathcal{T}^*)^i\,\Gamma_e^* = \left(\sum_{i=0}^{\infty} \mathcal{T}^i\right)^*\,\Gamma_e^*\,.$$

A radiance distribution L bounces infinitely in the scene (getting weaker with each bounce), resulting in the radiance distribution $\sum_{i=0}^{\infty} \mathcal{T}^i L$. The contribution to the image (through all bounces) of distribution L is $\langle \Gamma_e^* \mid \sum_{i=0}^{\infty} \mathcal{T}^i L \rangle$. This contribution can be rewritten as follows:

$$\langle \Gamma_e^* \mid \textstyle\sum_{i=0}^{\infty} \mathcal{T}^i L \rangle = \langle \left(\sum_{i=0}^{\infty} \mathcal{T}^i\right)^* \Gamma_e^* \mid L \rangle = \langle \Gamma^* \mid L \rangle\,.$$

□

Importance defined as the adjoint of radiance was used by Pattanaik [68, 69] for Monte Carlo radiance computations, and by Aupperle and Hanrahan [10] for a finite-element method.

4.5 Directional Importance

It might seem natural to use two-point importance $\Gamma^*(x,y)$ since the inner product $\langle \Gamma^* \,|\, L \rangle$ is exactly the contribution we need to know for refinement. But it is easier to represent and transport importance like radiance (for the same reasons as for using radiance L rather than two-point transport intensity L^*, and also because we already chose to use radiance).

Lewins [57] suggests the use of a weight function to define a function related to the adjoint, but satisfying a more suitable equation — for example the same equation as the primal quantity (neutron density in his case, radiance in our case). For radiance, the weight function turns out to be the geometric term G, and we can define a term, Γ, related to the adjoint Γ^* by an equation similar to the definition of two-point transport intensity (2.4).

Definition *Directional importance:*
Directional importance Γ is defined by the following relation to two-point importance

$$\Gamma^*(x,y) = G(x,y)\,\Gamma(x,\vec{\omega}_{xy})\,. \tag{4.2}$$

Using operator notation, we can write this definition more compactly as

$$\Gamma^* = \mathcal{G}\Gamma\,.$$

We defined two-point transport intensity, in equation (2.4), as the product of the geometric term and radiance. Here the definition works in the opposite direction. Directional importance is defined such that, when multiplied by the geometric term, it becomes two-point importance. In this definition, y is the first point visible along direction $\vec{\omega}_{xy}$. For other points along that direction, the geometric term is zero, and hence the equation does not impose any constraint on the value of Γ. It follows from the definition that directional importance is dimensionless.

Recall that $\langle \Gamma^* \mid L \rangle$ is the power that radiance L contributes to the image. With directional importance, this contribution can be expressed as $\langle \mathcal{G}\Gamma \mid L \rangle$, or equivalently as $\langle \Gamma \mid \mathcal{G}L \rangle$. It follows that $\langle \mathcal{G}\Gamma \mid L \rangle = \langle \Gamma \mid \mathcal{G}L \rangle$ *is the power that radiance distribution* L *contributes to the image*. Now we want to show that this type of importance is transported like radiance. But first a necessary lemma concerning the geometric operator:

Lemma *The geometric operator* $\mathcal{G}$ *has a unique inverse in a closed environment.*

Proof Recall that $\mathcal{G}$ is the operator that converts radiance into two-point transport intensity. In a closed environment, every radiance uniquely determines the two-point transport intensity from its point of origin to the nearest point in its direction. Conversely, each radiance is uniquely determined by the two-point transport intensity from its point of origin to the first point in the direction of the radiance. Therefore, if a radiance distribution L is known for the environment, the two-point transport intensity distribution L^* is uniquely determined, and vice versa. Therefore, $\mathcal{G}$ is invertible. □

Theorem *Directional importance* Γ *satisfies an equilibrium equation like the radiance equation (2.13),*

$$\Gamma = \Gamma_e + \mathcal{T}\Gamma \, . \tag{4.3}$$

Proof Starting from the equilibrium equation for two-point importance (4.1), using the definitions of Γ, $\mathcal{T}^*$, and $\mathcal{T}$, and the lemma above, we get

$$\begin{aligned} & \Gamma^* = \Gamma^*_e + \mathcal{T}^* \Gamma^* \\ \Leftrightarrow \quad & \mathcal{G}\Gamma = \mathcal{G}\Gamma_e + (\mathcal{G} \circ \mathcal{R})(\mathcal{G}\Gamma) \\ \Leftrightarrow \quad & \Gamma = \Gamma_e + \mathcal{R} \circ \mathcal{G}\Gamma \\ \Leftrightarrow \quad & \Gamma = \Gamma_e + \mathcal{T}\Gamma \end{aligned}$$

□

The theorem can also be seen directly: since Γ^* and Γ are adjoint, and L^* and L are adjoint too, and since L^* satisfies the same equation as Γ^*, Γ must satisfy an equilibrium equation like radiance L does.

Since the emitted two-point importance is $G(x, y)$ if y is on the line segment ex, and since directional importance is defined by equation (4.2), the *emitted directional importance* Γ_e has to be 1 in all directions from the eye through the image:

$$\Gamma_e(y, \vec{\omega}_{ey}) \equiv \begin{cases} 1, \text{ if } y \text{ is a point on the image,} \\ 0, \text{ otherwise.} \end{cases}$$

In conclusion, we define directional importance Γ such that the inner product $\langle \mathcal{G}\Gamma \mid L \rangle = \langle \Gamma \mid \mathcal{G}L \rangle$ is the power that radiance L contributes to the image, since this definition makes importance fulfill an equilibrium equation like the radiance equation. Put another way, the contribution of radiance to the image can be found by solving a (light) transport problem with the image emitting light. The fact that importance is propagated like light allows us to use data structures and transport algorithms that treat importance as if it were another set of color channels.

This type of importance was introduced by Christensen *et al.* [17] (there it was called "exitant directional importance"), and used by Christensen *et al.* [19], Schröder and Hanrahan [79] with a spatial parameterization, and Veach and Guibas [96] for a Monte Carlo method.

4.6 Diffuse Importance

The first work on importance in global illumination was done by Smits *et al.* [89] for the diffuse case. They showed that importance, defined as the adjoint function of radiosity, could be used to determine the influence on the image. Here we will show that radiosity and irradiance are adjoint functions and satisfy adjoint equations. Then we show that diffuse importance, as defined by Smits *et al.*, is transported like irradiance, and discuss their choice of source term for importance.

Recall that irradiance E is the integral of all radiance incident on a point,

$$E(y) \equiv \int_{\Omega_+} L^{\text{in}}(y,\vec{\omega}) \cos\theta_y \, d\vec{\omega} = \int_x G(x,y)\, L(x)\, dA_x ,$$

where the change of integration domain was done using equation (2.7) and the definition of the geometric term (2.2). We can write the diffuse radiance equation (2.18) as

$$L(y) = L_e(y) + f_r(y)\, E(y) .$$

Then the irradiance at point z is

$$\begin{aligned} E(z) &= \int_y G(y,z)\, L(y)\, dA_y \\ &= \int_y G(y,z)\, \big(L_e(y) + f_r(y)\, E(y)\big)\, dA_y \\ &= \int_y G(y,z)\, L_e(y)\, dA_y + \int_y G(y,z)\, f_r(y)\, E(y)\, dA_y . \end{aligned}$$

If we consider $\int_y G(y,z)\, L_e(y)\, dA_y$ a source term E_e for irradiance, we get an equilibrium equation equivalent to the diffuse radiance equation (2.18), but expressed in irradiance,

$$E(z) = E_e(z) + \int_y G(y,z)\, f_r(y)\, E(y)\, dA_y . \qquad (4.4)$$

We will now show that the equilibrium equation for irradiance (4.4) is adjoint to the diffuse radiance equation (2.18), and that irradiance and radiosity are adjoint functions.

The transport operator for diffuse radiance (applied to a radiance distribution) is

$$(\mathcal{T}_d L)(y) \equiv f_r(y) \int_x G(x,y)\, L(x)\, dA_x\,.$$

The transport operator for irradiance (applied to an irradiance distribution) is

$$(\mathcal{T}_d^* E)(x) \equiv \int_y G(y,x)\, f_r(y)\, E(y)\, dA_y\,.$$

It is simple to show that these transport operators are adjoint, as already indicated by the choice of notation $\mathcal{T}_d$ and $\mathcal{T}_d^*$:

$$\begin{aligned}\langle E \,|\, \mathcal{T}_d L\rangle &= \int_y E(y)\, f_r(y) \int_x G(x,y)\, L(x)\, dA_x\, dA_y \\ &= \int_x \int_y G(y,x)\, f_r(y)\, E(y)\, dA_y\, L(x)\, dA_x \\ &= \langle \mathcal{T}_d^* E \,|\, L\rangle\,.\end{aligned}$$

Since these operators are adjoint, it follows that equations (2.18) and (4.4) are adjoint equations, and therefore, radiosity and irradiance are indeed adjoint functions.

Smits *et al.* defined diffuse importance, "Ψ", as an adjoint of radiosity. It fulfills an equilibrium equation like (4.4),

$$\Psi(z) = \Psi_e(z) + \int_y G(y,z)\, f_r(y)\, \Psi(y)\, dA_y\,.$$

They defined emitted importance such that everything visible has importance 1, independent of orientation and distance to the eye. With their definition of emitted importance, the inner product $\langle \Psi \,|\, B\rangle$ is the power contributed by a radiosity distribution B to surfaces visible in the image.

Apart from the directionality, the main difference between Ψ and our two types of glossy importance is that we prefer a definition of emitted importance in which orientation and distance to the eye are taken into account. In other words, we prefer a definition that makes the inner product of importance and radiance be the power contributed to the image rather than the power contributed to the visible surfaces.

“I praise thy resolution”
Euripides

Chapter 5

Wavelet Representation

A straightforward method for approximate solution of the discrete radiance equation (3.3) was introduced in chapter 3. It represents the solution with a fixed number of basis functions and transports light between all basis functions to compute the solution. If n_b is the number of basis functions, this method requires $O(n_b^{1.5})$ transports to compute an approximate solution.

Instead, we use a hierarchical, or *multiresolution*, method, which results in only $O(n_b)$ transports. With this method, we first compute a very rough solution, and then refine the representation and transports based on that solution. After the refinement, an improved solution can be computed, new refinements can be performed, and so on. The multiresolution method utilizes two facts: in some parts of the scene, radiance distributions can be represented with sufficient accuracy using only a few basis functions; and even where the radiance distribution is represented with many basis functions, only a few of these basis functions need to transport light to many other basis functions.

In this chapter, we only consider transports between pairs of patches and how these transports can be represented hierarchically. In the following chapter, we extend this hierarchical method to clusters of patches. Importance can be represented in a hierarchical basis and transported discretely in exactly the same way as radiance. In this chapter, we will therefore focus on radiance. We first motivate the use of hierarchical bases and describe the development of these in the area of global illumination. We then present some background on multiresolution analysis, a convenient framework for describing hierarchical bases. We describe one-dimensional wavelet bases and the properties that are desirable when choosing a wavelet basis. The four-dimensional bases necessary for representing radiance distributions can be constructed from one-dimensional wavelet bases. Then, the discrete transport of radiance represented in a wavelet basis is described. Finally, a glossy global illumination algorithm using wavelets and adaptive refinement is described.

Details of our implementation of the algorithm, such as how each transport coefficient is computed efficiently and adaptively, how visibility is computed, etc., are described in chapter 7.

5.1 Hierarchical Bases

For a given function space V, there are many possible sets of basis functions that span V. We are free to choose one that makes computations efficient; for example, a hierarchical basis. Results by Alpert [2, 3, 4], Beylkin *et al.* [13, 14], Gortler *et al.* [41, 78], Hanrahan *et al.* [45, 46] and others indicate that significant performance gains can be achieved by using a hierarchical basis.

Alpert [3] and Beylkin *et al.* [13] discovered that a large class of operators can be approximated with sparse matrices in a wavelet basis. In particular, they studied integral operators where the kernel $K(u, v)$ is singular along the diagonal (when $u = v$), and the kernel is sufficiently smooth away from the diagonal. They showed that for a univariate wavelet basis with n_b basis functions, certain wavelet decompositions result in a matrix with $O(n_b)$ significant entries.

The use of hierarchical bases in global illumination started with Cohen *et al.* [23]. They observed that, in many cases, to get a satisfying result when transporting light between two patches, the accuracy at the receiving patch needs to be higher than at the sending patch. They therefore set up transports from basis functions covering an entire patch to many smaller basis functions at the receiving patch. The large basis function at a patch corresponds to the average of all the smaller basis functions at that patch. This means that there are two levels of representation of the radiosity at all patches.

Hanrahan *et al.* [45, 46] introduced a radiosity algorithm that uses a fully hierarchical approach. Their algorithm uses a hierarchical set of basis functions at both the sending and receiving patch. At the coarsest level, there is a single basis function on each surface patch. If the transport between two such basis functions is too coarse, it is refined to a transport between smaller basis functions. A basis function is allowed to transport to another only if the error in the transport falls below a given threshold. In general, coarse basis functions suffice for long distances and dim light, while more detailed basis functions are required for shorter distances and brighter light. Gortler *et al.* [41, 78] generalized the hierarchical radiosity method to use a wavelet representation.

In the more general context of radiance, Aupperle and Hanrahan [9] used hierarchical piecewise-constant basis functions for a spatial parameterization of radiance; Schröder and Hanrahan [79] and Pattanaik and Bouatouch [70] used wavelets for spatial parameterization of radiance; while Christensen *et al.* [19] used wavelets for a directional parameterization of radiance. In this chapter, we construct a hierarchical basis for efficiently representing radiance distributions along the lines of Christensen *et al.* [19].

5.2 Multiresolution Analysis

Hierarchical bases can be formally described with multiresolution analysis. Before describing multiresolution analysis in detail, it is interesting to compare it to function sampling and Fourier analysis. If a function is sampled at certain points, the value of the function is fully known at these points, but no information about the frequency of the function is known. A Fourier analysis of the function gives information about the frequencies of the function, but no information about the values of the function. In contrast, a multiresolution analysis gives information about local variation of the function, a mix of value and frequency information.

Multiresolution analysis as formulated by Mallat [62] provides a convenient framework for studying hierarchical bases. Chui [20], Daubechies [27], DeRose *et al.* [29], Stollnitz *et al.* [93] and others give more elaborate explanations of multiresolution bases.

There are two basic ingredients for a multiresolution analysis: an infinite chain of nested linear function spaces $V^0 \subset V^1 \subset V^2 \subset \cdots$ and an inner product $\langle f | g \rangle$ of functions f, g in V^j. The basis functions for the spaces V^j are called *scaling functions* and are usually denoted by the symbol ϕ.

Example The simplest multiresolution basis is the *Haar basis* in one dimension. Let us consider the Haar basis on the interval $[0,1]$. The space V^j consists of piecewise-constant functions on $[0,1]$ with discontinuities at $\{0, 1/2^j, 2/2^j, \ldots, 1\}$. The space V^j is spanned by scaling functions $\phi_i^j(u)$. In the Haar basis, the scaling functions are piecewise-constant functions known as the Haar functions or "box" functions,

$$\phi_i^j(u) = \phi(2^j u - i) \qquad \text{for } i = 0, \ldots, 2^j - 1$$

with

$$\phi(u) = \begin{cases} 1, & \text{for } 0 \leq u < 1 \\ 0, & \text{otherwise.} \end{cases}$$

A few Haar scaling functions are shown in figure 5.1. Clearly the condition $V^j \subset V^{j+1}$ holds since we can represent any function in space V^j in space V^{j+1} by copying each coefficient: $\phi_{2i}^{j+1} = \phi_{2i+1}^{j+1} = \phi_i^j$. The Haar scaling functions shown here can be normalized to form an orthonormal, self-dual basis (as discussed in section 3.3).

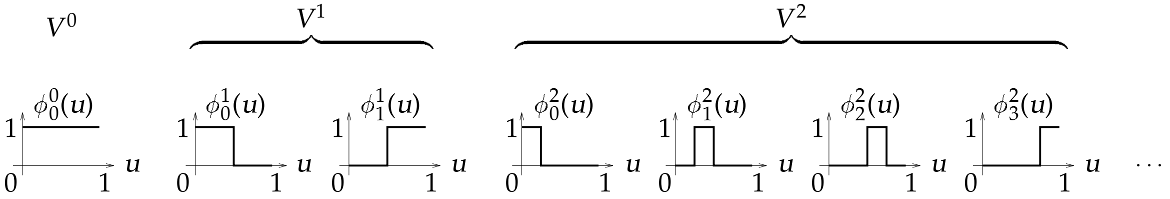


Figure 5.1

Haar scaling functions ϕ_i^j.

As discussed above, a function can be approximated by a weighted sum of scaling functions. Alternatively, we can represent the same approximation as scaling functions in V^0, giving coarse overall shape, along with detail at finer and finer resolution. Detail is accounted for by functions in the *orthogonal complement spaces* W^j defined by

$$W^j \equiv \{f \in V^{j+1} \mid \langle f | g \rangle = 0 \;\; \forall g \in V^j\}.$$

Space W^j is the orthogonal complement of V^j in V^{j+1}. Intuitively, wavelet space W^j contains the functions that are in V^{j+1}, but "missing" from V^j. *Wavelets* refer to basis functions for the orthogonal complement spaces W^j, and are usually denoted by the symbol ψ. The spaces W^j are sometimes called *wavelet spaces*.

Example The wavelet spaces W^j corresponding to the Haar scaling functions are spanned by piecewise-constant wavelets $\psi_i^j(u)$,

$$\psi_i^j(u) = \psi(2^j u - i)$$

with

$$\psi(u) = \begin{cases} 1, & \text{for } 0 \le u < 1/2 \\ -1, & \text{for } 1/2 \le u < 1 \\ 0, & \text{otherwise.} \end{cases}$$

Each wavelet is a linear combination of scaling functions from the next higher space: $\psi_i^j = \phi_{2i}^{j+1} - \phi_{2i+1}^{j+1}$. A few Haar wavelets are shown in figure 5.2.

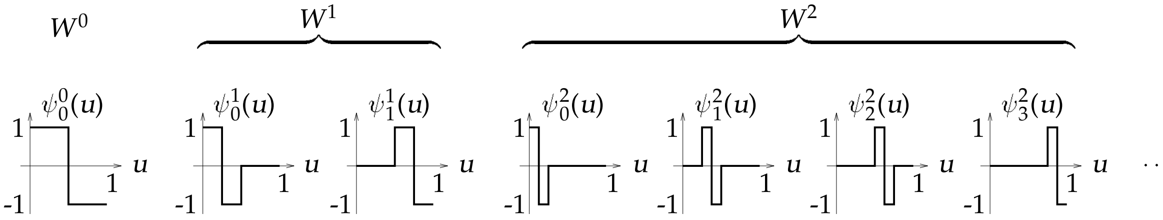


Figure 5.2

Haar wavelets ψ_i^j.

Orthogonal complements are often written as $V^{j+1} = V^j \oplus W^j$. Any function $f^{j+1} \in V^{j+1}$ can be written uniquely as an orthogonal decomposition $f^{j+1} = f^j + g^j$, where $f^j \in V^j$ and $g^j \in W^j$. The space V^j can be fully decomposed as

$$V^j = V^0 \oplus W^0 \oplus \cdots \oplus W^{j-1}.$$

Therefore, a hierarchical basis for V^j can be formed by selecting a scaling function basis for V^0 together with wavelet bases for the spaces $W^0, \ldots, W^{j-1}$. The scaling functions spanning V^0 represent coarse shape, while the wavelets provide detail at increasing resolutions.

In the remainder of this section, we will give examples of some of the most commonly used wavelet bases: spline wavelets, Daubechies's coiflets, multiwavelets and flatlets. In order to appreciate the properties of these wavelet bases, the concept of vanishing moments is necessary.

Definition *Vanishing moments:*
A function f has m vanishing moments if it is orthogonal to all monomials of degree less than m,

$$f \perp x^d \qquad \textit{for } 0 \leq d < m\,.$$

A basis is said to have *m* vanishing moments if some of the basis functions have *m* vanishing moments. In general, the more vanishing moments a basis has, the fewer coefficients it will take to represent a function resembling a low-order polynomial.

The Haar wavelets have only a single vanishing moment; they are orthogonal to constant functions, but not to linear functions. The following families of wavelets have as many vanishing moments as the order of the basis. However, they also have some serious drawbacks, as will be described in section 5.3.2.

Example Another example of a hierarchical basis is the basis consisting of piecewise-linear functions, the linear B-splines. We call the corresponding wavelets *splinelets*. A few of these scaling functions and wavelets (defined on the interval $[0,1]$ for this example) are shown in figure 5.3. The basis shown is continuous, but not smooth. Smooth bases can be constructed from higher-order B-splines [29, 31, 93]. The corresponding wavelets have vanishing moments equal to the order of the spline.

Example Daubechies [27] defined a class of wavelets called "*coiflets*". These basis functions have as many vanishing moments as their order N, and are continuous and even smooth for higher orders. For order $N=2$, the scaling function and wavelet are continuous, but not smooth; they are shown in figure 5.4. The scaling functions and wavelets are orthonormal and therefore self-dual. On the

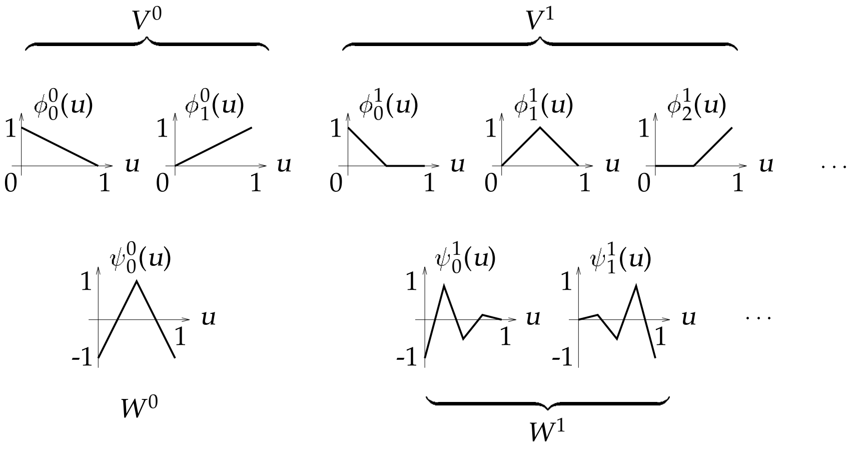


Figure 5.3

Linear B-splines and splinelets.

down side, they have irregular structure and significant overlap, and are defined on the infinite interval. Meyer [63] and Chui and Quak [21] describe a way to define similar (but slightly more complicated) wavelets on a bounded interval.

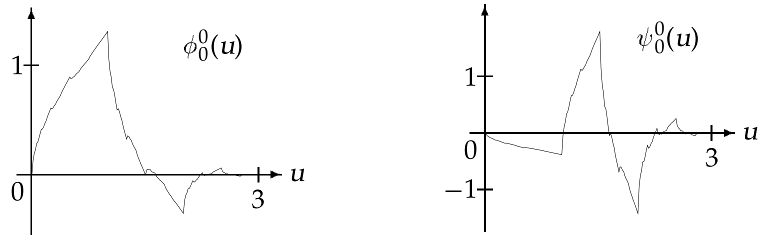


Figure 5.4

Daubechies's scaling function and "coiflet" of order $N = 2$.

Example Gortler *et al.* [41, 78] used wavelet bases to represent radiosity. They used two families of non-overlapping wavelets: *multiwavelets* (defined by Alpert [4]) and *flatlets*, shown in figures 5.5 and 5.6. Multiwavelets of order n are constructed from pieces of monomials of order up to n, and flatlets of order n are constructed from $2n$ piecewise-constant pieces. These wavelets have many vanishing moments and do not overlap translates of themselves — desirable properties for some applications, as will be explained in section 5.3.2. Schröder and Hanrahan [79] used multiwavelets to represent radiance.

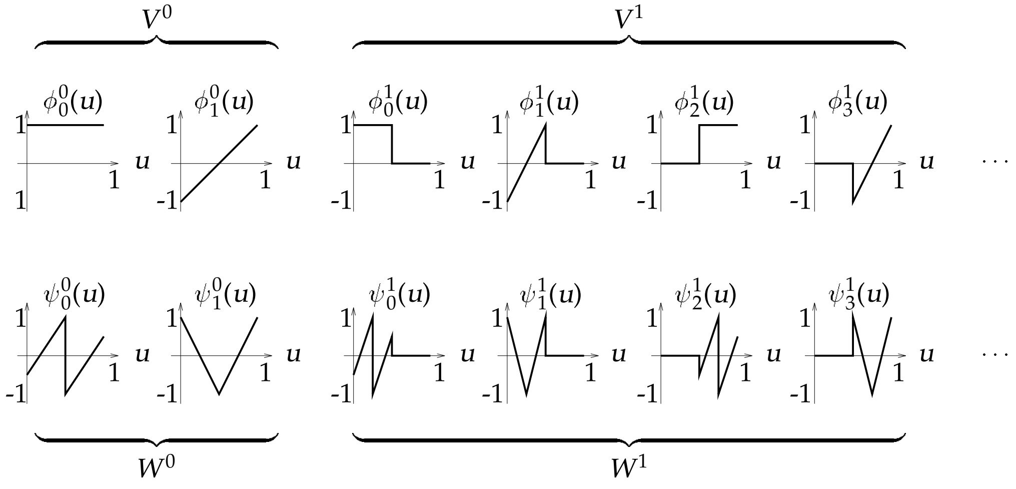


Figure 5.5

Multiwavelets of order 2.

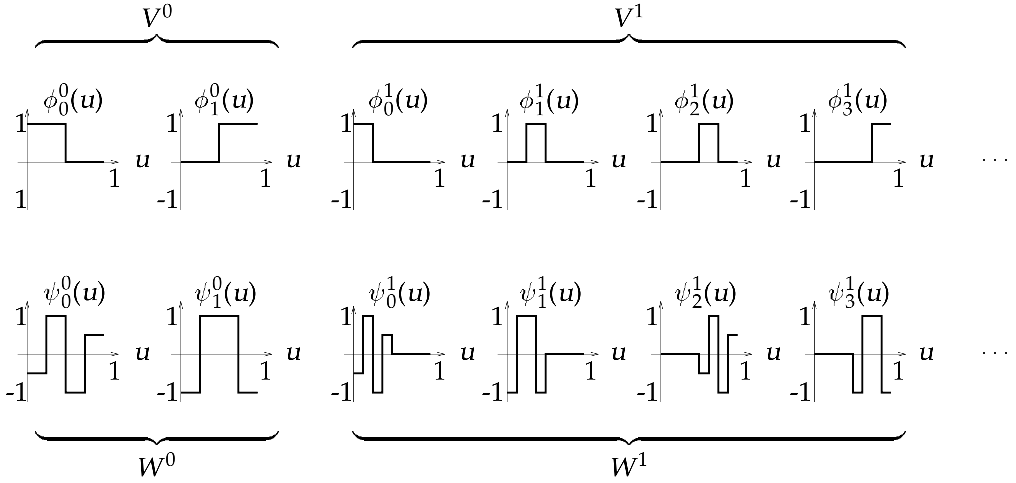
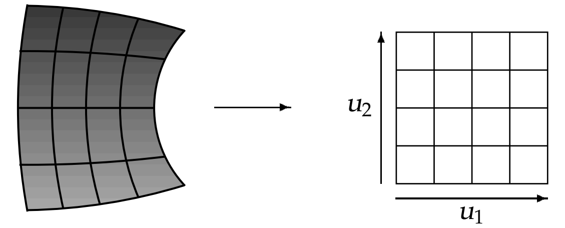


Figure 5.6

Flatlets of order 2.

5.3 Multidimensional Wavelet Bases

With the formalism described in the previous section, we are ready to construct wavelet bases for four-dimensional radiance distributions. As is common, we split the surfaces into patches such that the spatial variables on each patch can be parameterized on the unit square $[0,1]^2$; see figure 5.7. Then three possibilities for parameterization of the remaining two dimensions are described: spatial parameterization, wavelets defined on the hemisphere, and transformation of the hemisphere to a square. We discuss the properties of each of these possibilities.



3D surface patch 2D parameter space

Figure 5.7

Spatial projection: mapping between a 3D surface patch and the 2D parameter space.

Once the domain is chosen, we want to construct bases for it using tensor products of univariate basis functions. We have a choice of how to combine one-dimensional wavelet bases to form the required four-dimensional wavelet bases. These choices are called standard and nonstandard constructions, respectively.

5.3.1 A Convenient Domain for Radiance

A spatial parameterization of radiance, like Schröder and Hanrahan [79] and Pattanaik and Bouatouch [70] used, gives the domain $[0,1]^2$ on both the sending and receiving patch, so the domain of the radiance distribution is the four-dimensional hypercube $[0,1]^4$. As we shall see in section 5.3.3, it is simple to construct bases for $[0,1]^4$ from tensor products of one-dimensional wavelets.

However, we want a directional parameterization of radiance since it leads to a small number of initial light transports between basis functions (see section 2.7). The domain of the radiance distribution on each patch is then $[0,1]^2 \times \Omega+$, where $\Omega+$ is the unit hemisphere. Schröder and Sweldens [80] describe wavelets defined on a spherical domain. Using their wavelets, a basis for radiance can be constructed from a tensor product of bases for the unit square and the unit hemisphere.

We use a simpler method, mapping the hemisphere to a unit square. By mapping $\Omega+$ onto $[0,1]^2$, we get a domain $[0,1]^4$ for directional radiance. This process is described below. We can then use straightforward tensor products of one-dimensional basis functions for both angular and spatial variations.

Mapping the hemisphere to a unit square is done in two steps. First we use *gnomonic projection* to map between points in $\Omega+$ and points on a disc with radius $\pi/2$. As shown in figure 5.8, gnomonic projection maps great circles through the pole of $\Omega+$ to radial lines, and preserves arc length along these curves. We use this map because it is easily computed and introduces only mild distortion. An alternative would be "flat" projection of $\Omega+$ onto a unit disc by simply ignoring the height component, but this projection results in points near the equator being mapped very densely near the circumference of the circle.

The gnomonic projection is followed by a radial "stretch" of the disc to exactly cover the unit square, also shown in figure 5.8. The composition of these mappings is an invertible mapping between the unit hemisphere $\Omega+$ and the unit square $[0,1]^2$.

One might worry that this mapping distorts radiance distributions to an unacceptable degree. However, figure 5.9 shows a typical radiance distribution (resulting from glossy reflection of light from a single point) before and after the angular mapping. After the mapping, the distribution is still continuous, but has a first-derivative discontinuity along the diagonals of the unit square.

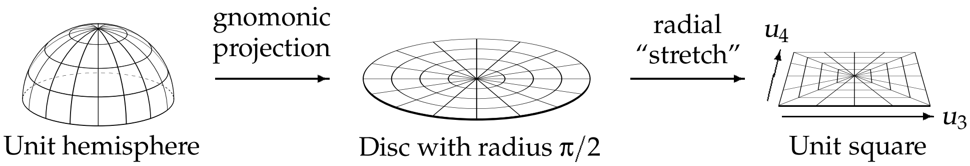


Figure 5.8

Angular projection: gnomonic projection and radial "stretch".

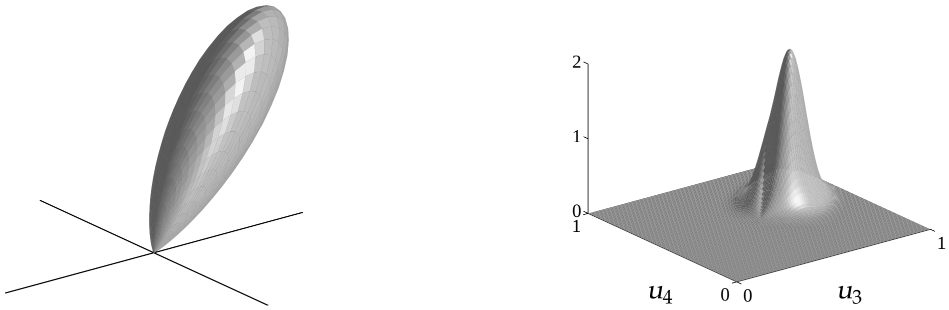


Figure 5.9

Radiance distribution before and after angular mapping.

5.3.2 Choice of Wavelet Basis

There are many desirable properties for a hierarchical basis for radiance, and naturally trade-offs are necessary. Here, we first summarize the properties desirable in a wavelet basis for use in glossy global illumination, and then evaluate the wavelet bases described in section 5.2 with respect to these properties. The most important properties are:

- *Bounded domain.* Multiresolution analysis is usually formulated for functions on the unbounded real line, whereas we wish to represent radiance distributions defined on bounded domains. Unbounded wavelets could be used on a bounded interval by artificially extending the functions of interest [27], but it is unclear how radiance would be extended beyond the geometric extent of surface patches without introducing a number of artifacts. We have found it more convenient to use wavelets defined on a bounded interval.

- *One-dimensional V^0.* The high dimensionality of the radiance transport problem makes it advantageous to have only one scaling function in space V^0: A single scaling function leads to a single light transport between two patches at the coarsest level, while (as shown in section 5.3.3) having two one-dimensional scaling functions leads to 16 four-dimensional scaling functions, requiring 256 transports between two patches at the coarsest level.

- *Sparse matrix.* The sparseness of the transport matrix is related to the number of vanishing moments of the wavelets [3, 4, 41]. The more vanishing moments, the sparser the matrix will be. However, wavelets with more vanishing moments are either of higher polynomial order or have larger support. In both cases, numerical integration required for computing the matrix entries becomes more expensive. Also, as the support of the wavelets increases, nonsmooth features such as singularities and shadow edges will fall under the support of more basis functions and will result in more non-negligible coefficients.

- *Efficient numerical integration.* The Galerkin method requires that we integrate the transport operator with basis functions (and their duals, if the basis is not orthonormal) to compute transport coefficients. It is advantageous to use basis functions for which we can develop inexpensive numerical integration formulas. Such inexpensive formulas require small support and low polynomial order.

We will now judge several wavelet bases in light of the listed requirements.

The Haar basis has several advantages, including a one-dimensional space V^0, orthonormality (and therefore self-duality), compact support, and simple numerical integration formulas. It has two disadvantages: it is discontinuous, and the transport matrix is not as sparse as for bases with more vanishing moments. The discontinuities can be ameliorated by performing a final gathering step during rendering [19, 61, 75, 87]. The non-optimal sparsity is to some extent overcome by the compact support and simple numerical integration.

Splinelets can be defined on a bounded domain as in the example in section 5.2. Also, splinelet bases can be "extended" such that space V^0 becomes one-dimensional by including extra levels in the basis. For an example, see figure 5.10. This basis is both continuous and fulfills the two first requirements above. However, having continuous basis functions on each patch is not sufficient to ensure a continuous solution; continuity must also be enforced across boundaries between adjacent patches, or the basis functions must be defined over complex shapes with arbitrary topology. Both approaches seem unnecessarily complicated compared to the alternative: a discontinuous basis with a final gathering step. but at the same time the splinelets have wider support and more costly integration formulas.

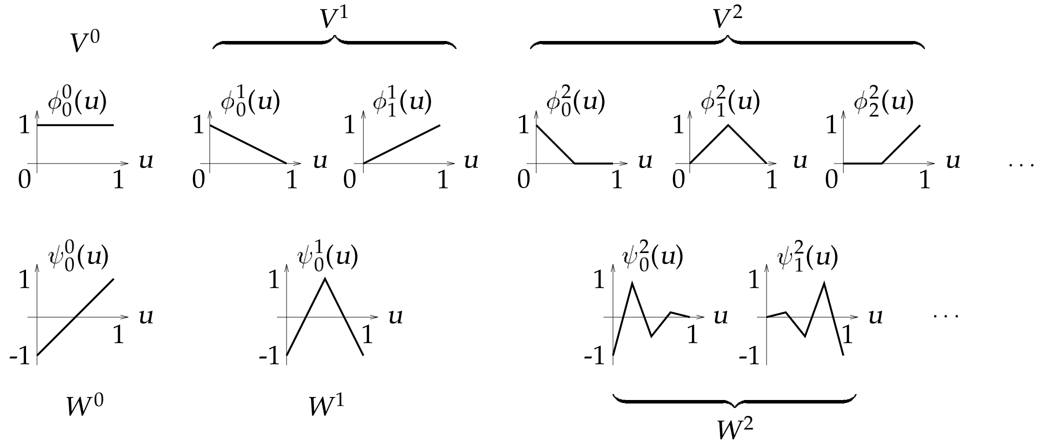


Figure 5.10

A continuous linear wavelet basis with just one scaling function: Linear B-splines and wavelets on domain $[0, 1]$ extended down to a constant function.

The "coiflets" defined by Daubechies are a family of wavelets where both scaling functions and wavelets have many vanishing moments. For coiflets of order N, the scaling function is orthogonal to monomials of degree less than N (except constants), and the wavelets are orthogonal to all monomials of degree less than N. Furthermore, coiflets are smooth: the higher the order N, the smoother the coiflet and scaling function are. The many vanishing moments and smoothness are very desirable properties in a wavelet basis. However, coiflets are defined on the real line and are therefore not directly useful for our purposes. Meyer [63] modified these to be defined on the interval $[0, 1]$. If these wavelet bases were extended to have a one-dimensional space V^0, they could be used for glossy global illumination. They have many vanishing moments, so if the kernel of the transport operator were everywhere-smooth, these wavelets would be ideal. However, the basis functions overlap each other. With a nonsmooth kernel, as the transport operator has, many basis functions will have support on a discontinuity or singularity. This overlap reduces the potential sparsity.

Flatlets are defined on the interval $[0, 1]$ and can easily be extended to have a one-dimensional space V^0. They have more vanishing moments than the Haar basis, increasing the sparsity of the transport matrix. On the other hand, flatlets have wider support and more costly integration formulas. Multiwavelets also require costly integration formulas.

There are more wavelet bases than the ones described here, each with its own set of advantages and disadvantages. We have experimented with splinelets [18], Daubechies's coiflets, and the Haar basis. As a result of these experiments — and weighing the trade-offs above — we have chosen to implement our glossy global illumination algorithm using the Haar basis.

Schröder and Hanrahan [79] presented a comparison of wavelet bases for radiance in which they examined rates of convergence, integration expense and accuracy, and the amount of work required to obtain a solution of a given accuracy. They showed that linear bases are best for reasonable accuracies. Higher-order bases converge at a faster rate, but the gain only shows for very high accuracies.

5.3.3 A Wavelet Basis for Radiance

Now we want to construct basis functions on the four-dimensional hypercube $[0,1]^4$ using tensor products of univariate basis functions. Here, the constructions are done for the Haar basis, but the tensor-product constructions can be used for any wavelet basis.

A uniresolution basis for space $V^j \times V^j \times V^j \times V^j$ consists of all tensor products

$$\phi^j_{i_1}(u_1)\phi^j_{i_2}(u_2)\phi^j_{i_3}(u_3)\phi^j_{i_4}(u_4) \qquad \text{with } 0 \le i_1, i_2, i_3, i_4 < 2^j .$$

For hierarchical bases, there are two alternative methods: the so-called "standard" and "nonstandard" constructions [13].

The *standard construction* forms a basis from all possible tensor products of univariate basis functions: scaling functions in V^0 and wavelets in higher spaces. For the Haar basis, the univariate basis functions are $\phi^0_0, \psi^0_0, \psi^1_0, \psi^1_1, \ldots, \psi^{j-1}_{(j-1)^2-1}$. So the four-dimensional basis consists of the set

$$\left\{\phi^0_0, \psi^0_0, \psi^1_0, \psi^1_1, \ldots, \psi^{j-1}_{(j-1)^2-1}\right\}^4 .$$

In the *nonstandard construction*, each tensor product consists of univariate basis functions in the same space V^j, including scaling functions at all levels. Four univariate scaling functions make a four-dimensional scaling function for space V^0. All other combinations of four univariate scaling functions and wavelets make four-dimensional wavelets for higher spaces W^j.

$$\begin{aligned} &\phi^0_0\phi^0_0\phi^0_0\phi^0_0 \cup \\ &\quad \left\{\phi^j_{i_1}\phi^j_{i_2}\phi^j_{i_3}\psi^j_{i_4},\ \phi^j_{i_1}\phi^j_{i_2}\psi^j_{i_3}\phi^j_{i_4},\ \ldots,\ \psi^j_{i_1}\psi^j_{i_2}\psi^j_{i_3}\psi^j_{i_4} \mid 0 \le j < J;\ 0 \le i_1, i_2, i_3, i_4 < 2^j\right\} . \end{aligned}$$

This construction restricts the support of multivariate Haar basis functions to be square.

Example As four-dimensional bases are hard to visualize, consider the Haar space V^2 for two dimensions. The standard and nonstandard basis constructions are illustrated in figure 5.11. To get a feel for the shape of these basis functions, a few of them are shown in figure 5.12.

$$\begin{array}{c|cccc} & \phi_0^0 & \psi_0^0 & \psi_0^1 & \psi_1^1 \\ \hline \phi_0^0 & \times & \times & \times & \times \\ \psi_0^0 & \times & \times & \times & \times \\ \psi_0^1 & \times & \times & \times & \times \\ \psi_1^1 & \times & \times & \times & \times \end{array} \qquad \begin{array}{c|cccccc} & \phi_0^0 & \psi_0^0 & \phi_0^1 & \phi_1^1 & \psi_0^1 & \psi_1^1 \\ \hline \phi_0^0 & \times & \times & & & & \\ \psi_0^0 & \times & \times & & & & \\ \phi_0^1 & & & & & \times & \times \\ \phi_1^1 & & & & & \times & \times \\ \psi_0^1 & & & \times & \times & \times & \times \\ \psi_1^1 & & & \times & \times & \times & \times \end{array}$$

Figure 5.11

The standard and nonstandard constructions of a two-dimensional Haar wavelet basis for space V^2. The crosses "$\times$" indicate the univariate basis functions that are combined in the two-dimensional basis.

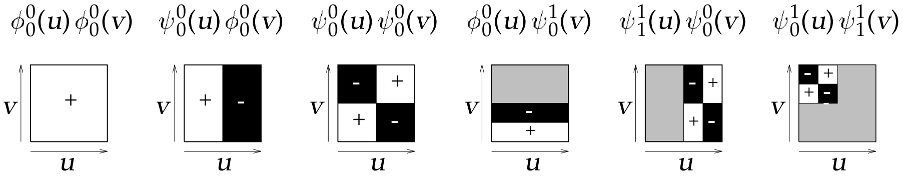

$\phi_0^0(u)\,\phi_0^0(v)$
$\psi_0^0(u)\,\phi_0^0(v)$
$\psi_0^0(u)\,\psi_0^0(v)$
$\phi_0^0(u)\,\psi_0^1(v)$
$\psi_1^1(u)\,\psi_0^0(v)$
$\psi_0^1(u)\,\psi_1^1(v)$
v
u
+
-

Figure 5.12

Examples of two-dimensional Haar basis functions. The functions are 1 in the white areas, -1 in the black areas, and 0 in the gray areas.

We chose the nonstandard basis construction primarily because the required data structures are simpler than for the standard basis construction (see section 7.7). In the remainder of this section, we will describe some of the details of the nonstandard construction.

Let $\mathbf{u} = (u_1, u_2, u_3, u_4)$ denote a point in $[0,1]^4$, and let $\mathbf{i} = (i_1, i_2, i_3, i_4)$ denote a four-component multi-index of integers. The four-dimensional scaling functions for V^j take the form

$$\phi\phi\phi\phi_{\mathbf{i}}^j(\mathbf{u}) \;\equiv\; \phi_{i_1}^j(u_1)\,\phi_{i_2}^j(u_2)\,\phi_{i_3}^j(u_3)\,\phi_{i_4}^j(u_4)\,.$$

That is, the scaling functions for resolution j consist of all possible products of the one-dimensional scaling functions for resolution j. The four-dimensional wavelets spanning the orthogonal complement W^j are formed by taking all other products of scaling functions and wavelets for resolution j. These wavelets consist of 15 types:

$$\phi\phi\phi\psi^j_{\mathbf{i}}(\mathbf{u}),\ \phi\phi\psi\phi^j_{\mathbf{i}}(\mathbf{u}),\ \phi\phi\psi\psi^j_{\mathbf{i}}(\mathbf{u}),\ \ldots\ ,\ \psi\psi\psi\psi^j_{\mathbf{i}}(\mathbf{u})\,.$$

We take as our basis $\mathbf{B}$ the set of basis functions spanning $V^0, W^0, W^1, \ldots$ for each patch in the scene.

As for univariate bases, we need dual basis functions to compute basis function coefficients. Fortunately, multivariate dual functions are simple to construct from univariate dual functions since duals of products are products of duals. For example,

$$\overline{\phi\phi\psi\psi}^j_{\mathbf{i}}(\mathbf{u}) = \overline{\phi}^j_{i_1}(u_1)\,\overline{\phi}^j_{i_2}(u_2)\,\overline{\psi}^j_{i_3}(u_3)\,\overline{\psi}^j_{i_4}(u_4)\,.$$

Gortler *et al.* [41, 78] and Schröder and Hanrahan [79] use a scaling function representation at all levels (for radiosity and radiance, respectively). Their approach therefore requires “pushing” and “pulling” to distribute transported radiance to other levels of the hierarchy before the next transport iteration. By contrast, we use a wavelet representation in which all basis functions are orthogonal to basis functions at other levels.

It seems that there is great potential advantage in separating the spatial from the angular resolution. For instance, a small glossy patch should be more refined in the directional dimensions than in the spatial dimensions; and a large, slightly glossy patch needs more refinement in the spatial dimensions than in the angular. We have not experimented with such standard basis construction, but suggest it as a topic for future research.

5.4 The Hierarchical Galerkin Method

When the basis is nonhierarchical, light has to be transported between all basis functions with appropriate support, meaning that $O(n_b^{1.5})$ transport coefficients have to be computed. The use of hierarchical bases allows the transport operator to be expressed as a sparse matrix with only $O(n_b)$ significant entries. This means that the multiplication with the transport matrix in each Gauss-Seidel iteration (see section 3.5) can be performed in time linear in the number of basis functions. Hanrahan *et al.* [46] used Haar scaling functions for a hierarchical representation of radiosity. Gortler *et al.* [41, 78] also used a hierarchical scaling function representation of radiosity, but used scaling functions corresponding to multiwavelets and flatlets. Schröder and Hanrahan [79] used a similar scaling function representation for radiance while Christensen *et al.* [19] used a wavelet representation.

Here we will describe the choices in operator decomposition, that is, which basis functions are used to represent the radiance distribution and which basis functions can transport light to with each other. We distinguish between three decomposition of an operator: the so-called "standard" and "nonstandard" decompositions [13] and a "HR-style" decomposition [46].

To illustrate the advantages of a wavelet decomposition, let us first consider an example of a matrix in a nonhierarchical basis. To ease illustration of the concepts, it is necessary to reduce the dimensionality of the global illumination problem. We consider glossy reflection in "Flatland" [1, 41, 50, 78], a two-dimensional environment. In Flatland, radiosity only falls off linearly with the distance, and not with the square of the distance (as in three-dimensional environments). Radiance distributions in Flatland are two-dimensional: one parameter corresponds to position and the other to direction.

Example A very simple scene consists of two parallel line segments of unit length with unit distance between them. Figure 5.13 shows this scene. Radiance is transported from point $x = (u, 0)$ on one line segment to point $y = (v, 1)$ on the other line segment. The receiving line segment has a simple Phong-like glossy reflection with exponent 2.

In space V^3, each line segment is discretized into 8 positional scaling functions, and the semicircle of directions at each patch is divided into 8 angular scaling functions. The matrix entries (transport coefficients) of the matrix are

$$T_{rs} = \langle \overline{\phi}^3 \overline{\phi}^3_r \,|\, \mathcal{T} \phi^3 \phi^3_s \rangle = \int \overline{\phi}^3(v)\, \overline{\phi}^3(\vec{\omega})\, f_r(\vec{\omega}_{uv}, v, \vec{\omega})\, G(u, v)\, \phi^3(u)\, \phi^3(\vec{\omega}_{uv})\, du\, dv\, d\vec{\omega}\,.$$

The direct matrix is shown in figure 5.14. It is sparse: 654 coefficients out of $8^4 = 4096$ are nonzero.

We will return to this example in two following examples, to illustrate the advantage of a wavelet decomposition.

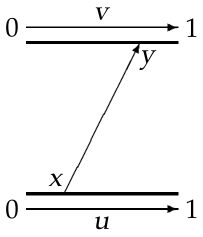


Figure 5.13

Parallel lines in Flatland.

5.4.1 Standard Operator Decomposition

We can change the basis of the radiance distribution into a wavelet basis, as described in section 5.3.3. By doing so, most radiance coefficients — corresponding to fine detail — will be small. Furthermore, many transports will be very weak since small details in the radiance distribution in one part of the scene do not have significant influence on small details elsewhere in the scene.

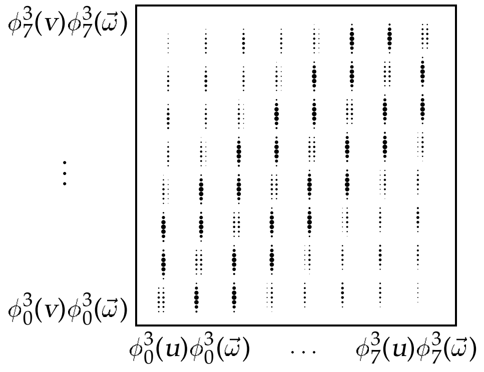


Figure 5.14

Matrix in space V^3 for parallel lines. The radius of each dot is proportional to the magnitude of the corresponding matrix entry.

Utilizing coherency in both spatial and directional dimensions allows better compression — even in Flatland. Therefore, the $n_b \times n_b$ matrix is transformed into a four-dimensional hypercube with side-length $\sqrt{n_b}$. The basis for radiance is transformed similarly: rather than considering it as one long vector of basis functions, it can be more efficiently compressed when considered as a two-dimensional matrix. (For three-dimensional scenes, the matrix can be transformed into an eight-dimensional hypercube with sidelength $\sqrt[4]{n_b}$ and the radiance is transformed into a four-dimensional hypercube with sidelength $\sqrt[4]{n_b}$.) As for a uniform basis, many entries are known *a priori* to be zero.

Example Let us again consider the case of two glossy lines in Flatland. If we perform a standard decomposition of the matrix in figure 5.14, we get the matrix shown in figure 5.15. This matrix is less sparse than the original matrix: in the standard matrix, 1228 entries are larger than 1% of the largest entry.

When the original matrix is sparser than the transformed matrix, why use a wavelet decomposition? The sparsity of the matrix itself does not tell the whole story. It is more relevant to consider the accuracy of multiplying the matrix on a radiance distribution. When a radiance distribution represented in a wavelet

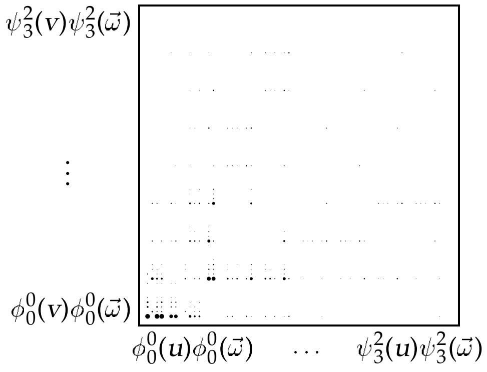


Figure 5.15

Standard Haar matrix in space V^3 for parallel lines.

basis is multiplied by the matrix, few multiplies are required for a given accuracy. The number of multiplies is significantly less than if the radiance and matrix were not in a wavelet representation.

Example To illustrate the reduction in the number of multiplications to reach a given accuracy, we consider again the parallel lines in Flatland. For this example, the sending patch emits directional radiance. The emitted radiance distribution is computed according to Ward's reflection model [98] as if the sending patch was reflecting light from a single point. The receiver reflects light according to Ward's model with parameters $\rho_s = \rho_d = 0.5$ and $\alpha = 0.2$. The specular peak of this BRDF is about 10 times as bright as the diffuse portion.

We evaluate the accuracy of the radiance distribution at the receiving patch as the number of transports increases. The transports are included in order of decreasing magnitude of transport coefficient times sending basis coefficient. The error is computed as $\mathcal{L}^2$ distance between the computed distribution and a reference solution. The reference solution was computed using a nonhierarchical basis in a higher space. The results are shown in figure 5.16. The wavelet matrix decompositions require a much smaller number of transports than the original matrix to achieve a given level of accuracy.

Similar tests with perpendicular lines and occlusion are presented in Christensen *et al.* [16].

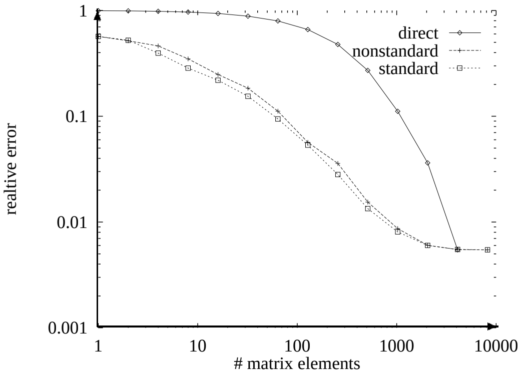


Figure 5.16

Error vs. number of transports: original matrix, standard decomposition, and nonstandard decomposition (described in section 5.4.2).

5.4.2 Nonstandard Operator Decomposition

The nonstandard operator decomposition utilizes an overrepresentation of the basis with both scaling functions and wavelets at all levels. This allows a basis function to only transport to basis functions at the same level. The "missing" transports are taken care of by *pushing* and *pulling*. The pushing and pulling ensure a consistent representation at all levels of the hierarchy. One can informally say that the fine scaling functions act as proxies for the wavelets in coarser spaces and the scaling function in V^0. Beylkin *et al.* [13] showed that the nonstandard operator decomposition gives a linear bound $O(n_b)$. The nonstandard decomposition is not a basis change since the function is overrepresented. When a matrix entry (a transport coefficient) is refined, it is set to zero and new coefficients between finer basis functions are computed.

Since the nonstandard decomposition requires an overrepresentation of the basis, the matrix is larger in each dimension than the original and standard matrices. But only a fraction of the elements can be nonzero at the same time. For example, in a fully refined matrix in space V^3, light is transported between scaling functions in V^2 and wavelets in space W^2, while there are no transports between the scaling functions and wavelets in coarser spaces. In figure 5.16, the accuracy of multiplying by this matrix was compared to the accuracy of the original and standard matrices.

5.4.3 HR-style Operator Decomposition

The wavelet functions can be replaced by scaling functions at a finer resolution since each wavelet is a linear combination of scaling functions at the next finer level. Then the hierarchical representation consists of scaling functions at all levels, and the transport matrix connects scaling functions. We call this operator decomposition "*HR-style*" since it was first used in the hierarchical radiosity method by Hanrahan *et al.* [46]. Gortler *et al.* [41, 78] and Schröder and Hanrahan [79] used this type of decomposition for radiosity transport and radiance transport, respectively. As with the nonstandard decomposition, "pushing" and "pulling" operations [41, 46] are required to make the overrepresentation consistent. Like the nonstandard decomposition, the HR-style decomposition is $O(n_b)$.

5.4.4 Choice of Decomposition

The nonstandard decomposition and HR-style decomposition have been shown to be $O(n_b)$ while the standard decomposition has been shown to be $O(n_b \log n_b)$ [13, 46]. (More precisely, it has been shown that for a given tolerance larger than zero, $O(n_b)$ or $O(n_b \log n_b)$ multiplies are sufficient to apply an operator to a function.) However, experimental results [16, 52, 78] show that the standard decomposition is often more sparse than the nonstandard decomposition. It is on this basis that we informally say that the standard decomposition is $O(n_b)$; there is no formal proof, but it appears to be the case in practice for our type of operator. If one desires a decomposition that is provably $O(n_b)$, the nonstandard or HR-style operator decomposition should be used.

In our implementation, we chose the standard decomposition since it (as mentioned above) is often the most sparse in practice.

5.5 Wavelet Algorithm

The solution method for radiance transport makes use of a wavelet representation and importance-driven refinement. The algorithm computes a view-dependent solution to the radiance equation; that is, the solution is refined most in the parts that contribute most to the image. In some respects, the algorithm is similar to the approach described by Gortler *et al.* [41] for wavelet radiosity. However, there are a number of areas, in addition to the higher dimensionality, in which the algorithm described here differs significantly from this previous work.

In this section, the main algorithm is presented and we discuss which transport coefficients are computed as the refinement proceeds. The primary task is to solve two systems of linear equations, one for radiance and one for importance:

$$\widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}_e + \widetilde{\mathbf{T}}\widetilde{\mathbf{L}} \qquad \text{and} \qquad \widetilde{\mathbf{\Gamma}} = \widetilde{\mathbf{\Gamma}}_e + \widetilde{\mathbf{T}}\widetilde{\mathbf{\Gamma}}.$$

We first compute a small number of entries of the matrix $\widetilde{\mathbf{T}}$ and solve the equations, then compute more entries of $\widetilde{\mathbf{T}}$ and solve again, and so on. The high dimensionality of the global illumination problem makes the entries of $\widetilde{\mathbf{T}}$ very expensive to compute, so we strive to compute as few of these entries as possible while generating a good approximation to the solution. Put briefly, only entries of $\widetilde{\mathbf{T}}$ that are estimated to be large — and that connect large and important basis function coefficients — are computed.

The main part of the algorithm alternates between computing approximate radiance and importance solutions $\widetilde{\mathbf{L}}$ and $\widetilde{\mathbf{\Gamma}}$ and improving the finite representation of the transport operator $\widetilde{\mathbf{T}}$. Quantities with a tilde are approximate, both because they are computed numerically and because they are truncated versions of infinite matrices. Initially, we approximate L_e and Γ_e in space V^0, the space spanned by the coarsest-level scaling functions, to give $\widetilde{\mathbf{L}}_e$ and $\widetilde{\mathbf{\Gamma}}_e$. We also compute the entries of $\mathbf{T}$ for transports between scaling functions in V^0 (as described in section 7.4), giving $\widetilde{\mathbf{T}}$. The algorithm is given in pseudocode below:

```
procedure GlossyGlobalIllumination:
    $\widetilde{\mathbf{T}} \leftarrow [\langle \overline{\widetilde{\mathbf{B}}} \,|\, \mathcal{T}\widetilde{\mathbf{B}} \rangle]$, where $\widetilde{\mathbf{B}}$ consists of all scaling functions in $V^0$
    $\widetilde{\mathbf{L}} \leftarrow \widetilde{\mathbf{L}}_e$
    $\widetilde{\mathbf{\Gamma}} \leftarrow \widetilde{\mathbf{\Gamma}}_e$
    for a series of decreasing tolerances $\varepsilon$ do
        $\widetilde{\mathbf{L}} \leftarrow Solve(\widetilde{\mathbf{T}}, \widetilde{\mathbf{L}}, \widetilde{\mathbf{L}}_e)$
        $\widetilde{\mathbf{\Gamma}} \leftarrow Solve(\widetilde{\mathbf{T}}, \widetilde{\mathbf{\Gamma}}, \widetilde{\mathbf{\Gamma}}_e)$
        if not finest tolerance then $\widetilde{\mathbf{T}} \leftarrow Refine(\widetilde{\mathbf{T}}, \widetilde{\mathbf{L}}, \widetilde{\mathbf{\Gamma}}, \varepsilon)$
    end for
    RenderSolution($\widetilde{\mathbf{B}}\widetilde{\mathbf{L}}$)
end procedure
```

Initially only a diffuse solution is found, the basis functions (scaling functions) are constant over each patch and in all directions on the hemisphere above each patch. Then the spatial and directional resolution of the solution is refined where needed, based on estimates of error and contribution to the image. In parts that contribute very little to the image, the solution is never refined.

The radiance and importance systems are solved simultaneously, with the solution in one system determining the refinements in the other. Importance is used to refine the radiance solution only in parts that are significant to the final image. Likewise, radiance is used to refine the importance solution only in bright parts of the scene. We use Gauss-Seidel iteration to solve the approximate transport equations $\widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}_e + \widetilde{\mathbf{T}}\widetilde{\mathbf{L}}$ and $\widetilde{\mathbf{\Gamma}} = \widetilde{\mathbf{\Gamma}}_e + \widetilde{\mathbf{T}}\widetilde{\mathbf{\Gamma}}$, just as for radiance in the simple algorithm in section 3.5. Refinement is determined by an "oracle", described in the following section.

5.6 Refinement

In many applications of wavelets in numerical analysis [13], the goal is to obtain a sparse representation of a given matrix, thereby making repeated matrix–vector multiplications much faster. The wavelet decomposition of the matrix is done once and for all as a preprocess, and the cost of computing all the matrix elements is amortized by many fast matrix multiplications. In wavelet-based approaches to global illumination, the cost of explicitly constructing an entire transport matrix far outweighs the expense of any matrix–vector multiplications that follow. Therefore, it is essential to restrict the number of computed transport coefficients.

With n_b basis functions in the solution, there are $O(n_b^{1.5})$ nonzero entries in the approximate transport matrix $\widetilde{\mathbf{T}}$. We conjecture from section 5.4 that only $O(n_b)$ entries are necessary to obtain a solution of high accuracy. If we knew which elements were significant, all of them could be computed at the same time, and a solution found. Unfortunately we do not know *a priori* which elements are significant. At the same time, it is too expensive to compute all nonzero matrix elements and then discard the smallest ones. Instead we use a progressive method: Compute a few elements of the transport matrix (corresponding to transports between all scaling functions) and solve the resulting truncated radiance and importance equations. Then based on these solutions decide which new elements of the transport matrix to compute; solve again, and so on.

The goal of the *refinement oracle*[1] is to determine which of the entries of $\mathbf{T}$

---

[1] This procedure is like a "real" oracle in that it predicts the future. It is different in that it is not always right.

missing from $\widetilde{\mathbf{T}}$ should be computed to reduce the visible error in the current radiance solution. The two most important sources of error are:

- *truncation error* due to significant entries missing from $\widetilde{\mathbf{T}}$, and
- *integration error* in computing the entries of $\widetilde{\mathbf{T}}$.

In this section we describe how our oracle reduces truncation error. Section 7.5 outlines a method for simultaneously reducing integration errors.

The refinement oracle uses concepts from the brightness refinement criterion for hierarchical radiosity [46], the oracle used by Gortler *et al.* for wavelet radiosity [41], and the importance-based refinement strategy used by Smits *et al.* [89]. The idea is to estimate the influence on the visible image that would result if a new transport coefficient were to be added to $\widetilde{\mathbf{T}}$. If this quantity falls below some threshold, the expensive computation of the transport coefficient can be avoided without resulting in significant error in the solution.

Consider two basis functions b_s and b_r with no transport coefficient between them yet; see figure 5.17. We compute a new transport coefficient $T_{r\leftarrow s} = \langle b_r \,|\, \mathcal{T} b_s \rangle$ if we estimate that the resulting transport contributes sufficiently to the image, that is, if a sufficiently large value results from the product of

- *radiance*: the magnitude of the sending basis function coefficient $\widetilde{\ell}_s$,
- *estimated transport coefficient*: the estimated new transport coefficient $\widetilde{T}_{r\leftarrow s}$ between the basis functions, and
- *importance*: the contribution $\langle \mathcal{G}\widetilde{\Gamma} \,|\, b_s \rangle$ of the receiving basis function to the image.

The product of the first two quantities estimates the amount of light transported between the two basis functions, that is, the change in the coefficient of the receiving basis function due to this transport. Multiplying by the inner product of the receiving basis function and importance (and the geometric term) gives the contribution to the image. The sending basis function coefficient and approximate importance of the receiving basis function are known from the interim solution. Our estimate of the transport coefficient uses kernel variation. For Haar wavelets, variation is a good estimate of the wavelet coefficient. For higher-order wavelets, it is necessary to estimate how closely the computed solution resembles the true solution [41, 78].

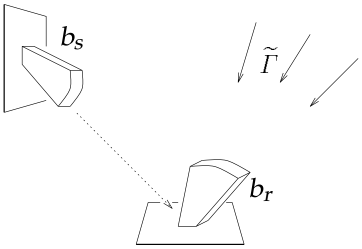


Figure 5.17

Sending and receiving basis functions.

There are infinitely many new transport coefficients to be considered for computation. We need a scheme for considering only some of them in each iteration, while making it possible to eventually consider all. Associate with each basis function b (except for the basis functions in V^0 and W^0) a unique "parent" basis function b' that overlaps b and is in a space one level coarser. Also, let the parent of the wavelets in space W^0 be the scaling function in V^0. For example, for the simple case of the one-dimensional Haar wavelet basis, the parent of ψ_i^j is $\psi_{\lfloor i/2 \rfloor}^{j-1}$ and the parent of ψ_0^0 is ϕ_0^0. In our implementation, we only consider computing a new transport coefficient $\widetilde{T}_{r \leftarrow s}$ if there is already a transport coefficient $\widetilde{T}_{r \leftarrow s'}$ or $\widetilde{T}_{r' \leftarrow s}$.

Since the kernel variation is the same for all fifteen transport coefficients from a given basis function to the fifteen wavelets sharing the same support, the estimated contribution to the image will be the same for all these fifteen wavelets.

We therefore compute all fifteen transport coefficients at once. This approach saves many kernel-evaluations since the same kernel-evaluations can be re-used in computing all 15 transport coefficients. Efficient computation of the transport coefficients is described in section 7.4. At the same time as the computation of the 15 transport coefficients, the transport coefficient between the two scaling functions that share support with the sending and receiving basis functions is computed. This transport coefficient is not used for light transports, but for adaptive improvements of other transport coefficients, as described in section 7.5.

> “Our life is frittered away by detail … Simplify, simplify!”
> *Henry D. Thoreau*

Chapter 6

Clustering

When sufficiently far away from a collection of complex objects, the complex geometry can be approximated as a single point. The advantage of such a simplification is that light can be transported between clusters of surface patches instead of directly between all the individual patches. This simplification reduces the number of transports between two clusters from the product of the number of patches in the two clusters to just one; see the sketch in figure 6.1. Counting all clusters, the number of transports is reduced from quadratic to linear in the number of patches. Realistic environments contain thousands or millions of patches, so in order to compute global illumination solutions, this reduction in the number of transports is essential.

Of course, such approximations are only applicable if we know *a priori* that the impact on the global illumination solution is acceptable. To determine the impact, bounds on the error are used.

The previous chapter described a hierarchical technique for light transports

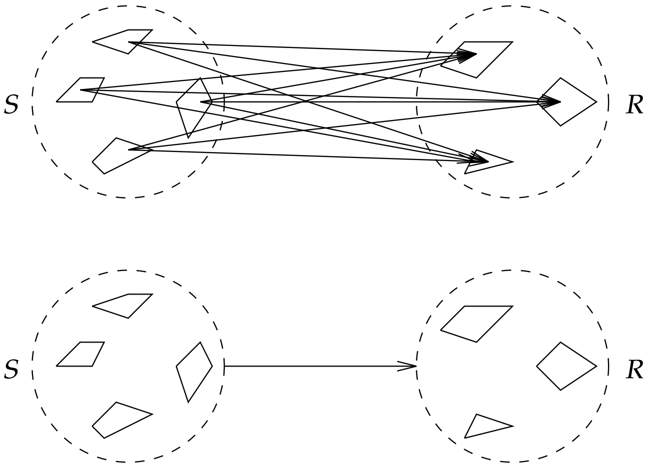


Figure 6.1

Accurate and approximate light transport between two clusters. For the accurate transport, light is transported directly from all patches in cluster S to all patches in cluster R. For the approximate transport, a single transport suffices.

between patches. It works well for scenes with few large patches [9, 19, 79]. We want a similar hierarchical approach for scenes with many small patches. In this chapter we extend the hierarchy upward, to allow light to be transported between many patches with a single transport. We first describe how our method differs from previous work. Then we describe how a hierarchy of clusters is formed from patches, and how approximate light transports take place between clusters. Then importance, as described in chapter 4, is generalized for clusters. We give bounds for each transfer, to use for refinement. The discrete representation of light and importance from and to clusters is described, as well as discrete light and importance transport. Finally, a glossy global illumination algorithm using clustering is described, along with the initial construction of the cluster hierarchy and the refinement of clusters.

6.1 Background

Related work on clustering was done in physics by Appel [5], Barnes and Hut [11], and Greengard [43], among others. They considered the so-called *n-body problem*, where the forces between a set of bodies have to be computed — for example, gravitational forces between stars in a galaxy or electrostatic forces between elementary particles. Instead of letting all $O(n^2)$ pairs of bodies interact, they found it advantageous to cluster nearby bodies together, and compute the approximate force between clusters of bodies, thereby getting a linear algorithm.

However, there are some differences between gravity and light. First, gravity and the electrostatic field from a body are rotationally symmetric (isotropic), while the light leaving a surface can differ significantly in different directions (anisotropic distributions). Second, the superposition principle holds for gravity and electrostatic forces, but not for light. The gravity or electrostatic field from individual bodies can be added to get the total gravity field. In contrast, surface patches might occlude each other. Therefore the equilibrium light distributions resulting from individual patches can not be added without taking occlusion into account. Another difference is that in the n-body problem, each "body" has a very small extent, whereas some of the surfaces in our setting can be quite large, even comparable in size to the entire scene.

Clustering algorithms have recently been proposed for global illumination computations in diffuse environments [55, 76, 83, 88]. The method of Smits *et al.* [88] creates the clusters automatically, uses error bounds to guide the solution process, and has $O(n_p \log n_p)$ complexity. Smits *et al.* use two different types of links between clusters: α-links and β-links. The α-links use information about the orientation of patches inside a cluster for each transport, giving relatively accurate estimates of the transport. In contrast, the β-links ignores this information, and therefore get coarser approximations faster.

It is possible to extend the bounds of Smits *et al.* on the transfer between clusters to account for glossy BRDFs and for directional radiance distributions. Thus, hierarchical glossy global illumination algorithms can benefit from clustering the same way as diffuse global illumination, by replacing the $O(n_p^2)$ initial links with links between clusters costing only $O(n_p \log n_p)$.

Here we go a step further and present a new and more efficient clustering algorithm for glossy environments, with complexity $O(n_p)$. The main idea behind our algorithm is to represent each cluster as a point source (and receiver) with an anisotropic directional distribution of outgoing (and incoming) light. Collapsing clusters into points in this manner produces sufficiently accurate approximations when the sizes of the clusters are small relative to the distance between them. We shall refer to a transport between a pair of clusters with such directional distributions as an Ω-link.

The idea of representing the light leaving complex geometry by a single directional distribution is not new. BRDFs are commonly used to model the reflection of light by complex microfaceted surfaces. Also, real light fixtures are specified in a similar way, with manufacturers providing goniometric diagrams from far-field measurements. Rushmeier *et al.* [76] used a directional representation of the reflectance of clusters of small surfaces. Representing clusters by means of directional distributions was also suggested (but not implemented) by Sillion [83]. Part of the novelty of our approach is in providing estimates of bounds on the errors associated with such approximations. These bounds permit the use of approximations in a controlled fashion, when the errors are acceptable.

An Ω-link (like a β-link in the Smits *et al.* algorithm) can be bounded in constant time. Computing the bound on an Ω-link between two clusters does not require any knowledge of the geometry inside the clusters; directional distributions associated with each cluster encode all the necessary information. The cost of maintaining such a data structure with each cluster depends on the directional resolution of the cluster representation and the number of child patches and subclusters, but not on the number of links to and from the cluster or on how many patches the subclusters contain. As a result, the algorithm has linear time and space complexity in the number of initial surfaces. By providing a bound on the error incurred, we can use the Ω-link approximation to simplify the calculation of light transport between clusters.

Our method not only handles glossy reflections, it also performs asymptotically faster than previous methods — even in diffuse environments. The applicability of our method to diffuse environments is not so surprising if one observes that a collection of diffuse surfaces, when considered as a cluster, is likely to have an overall reflectance that is highly directional. Thus, representing clusters as anisotropic point sources is advantageous for both glossy and diffuse environments, yielding a much more accurate representation than either Smits *et al.*'s β-links or Sillion's density volumes [83], both of which treat clusters as isotropic entities.

6.2 Light Transfer between Clusters

We want an approximation such that light can be transported between small distant clusters with a single transport, rather than transporting between all pairs of patches. Bounds, derived in section 6.4, will ensure that the error introduced by the approximation is acceptable. In this section, it will be shown that a cluster can be approximated as a point with an anisotropic distribution of light. Likewise, light incident on a cluster can be approximated as an anisotropic distribution of incident light.

6.2.1 Far-field Approximation

First we describe the *far-field approximation*, a cluster approximation of light transport between patches. The far-field light transport is divided into three steps: approximating light distributions on patches with distributions on clusters, transporting between clusters, and reflecting light incident on clusters to light distributions on patches.

Let $S = \cup s$ be the surfaces of all patches in a sending cluster, and $R = \cup r$ the surfaces in the receiving cluster. The radiance coming from cluster S that is reflected by a point y on a surface in cluster R is:

$$\begin{aligned} L_{SR}(y,\vec{\omega}) &= \int_{x \in S} f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, G(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x \\ &= \int_{x \in S} f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, \frac{\cos\theta_x \cos\theta_y}{\|x-y\|^2}\, V(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x\,. \end{aligned}$$

If the clusters are far apart compared to their size, we can approximate $\vec{\omega}_{xy}$ with $\vec{\omega}_{SR}$, the direction from the center of S to the center of R. We can also substitute the exact θ_x and θ_y with the angle between the direction $\vec{\omega}_{SR}$ and the normal at x and y. And we can approximate $\|x-y\|$ with the distance between the centers of the two clusters, d_{SR}. With these approximations we get

$$L_{SR}(y,\vec{\omega}) \approx \int_{x \in S} f_r(\vec{\omega}_{SR}, y, \vec{\omega})\, \frac{\cos\theta_x \cos\theta_y}{d_{SR}^2}\, V(x,y)\, L(x,\vec{\omega}_{SR})\, dA_x\,.$$

Since $\vec{\omega}_{SR}$ is independent of x, we can move the BRDF f_r and $\cos\theta_y$ outside the integral. The distance d_{SR} is also independent of x and can be moved:

$$L_{SR}(y,\vec{\omega}) \approx f_r(\vec{\omega}_{SR}, y, \vec{\omega})\, \frac{\cos\theta_y}{d_{SR}^2} \int_{x \in S} L(x,\vec{\omega}_{SR}) \cos\theta_x\, V(x,y)\, dA_x\,.$$

Now we split the visibility V into three parts: V_S, V_{SR}, and V_R. V_{SR} is the average visibility between the clusters S and R, and V_S and V_R accounts for self-occlusion within each cluster. The visibility within S, the term $V_S(x,\vec{\omega})$, equals 0 if the ray leaving x in direction $\vec{\omega}$ intersects one of the surfaces inside the cluster, and 1 otherwise. The visibility $V_R(y,\vec{\omega})$ within the receiver is similar. Using this approximation we get

$$L_{SR}(y,\vec{\omega}) \approx \underbrace{f_r(\vec{\omega}_{SR},y,\vec{\omega})\, V_R(y,\vec{\omega}_{RS}) \cos\theta_y}_{\text{receiving cluster}} \, \frac{V_{SR}}{d_{SR}^2} \underbrace{\int_{x\in S} L(x,\vec{\omega}_{SR})\, V_S(x,\vec{\omega}_{SR}) \cos\theta_x \, dA_x}_{\text{sending cluster}} \, .$$

This separation of visibility is a very rough approximation. However, in section 6.4 we introduce bounds that ensure that this approximation is acceptable.

In the following, we will show how this approximate transport can be divided into three steps: converting radiance on patches into radiant intensity on a cluster, transporting radiant intensity on a cluster to incident radiance on other clusters, and finally reflecting incident radiance on a cluster to radiance on patches.

6.2.2 Exitant Approximation

We wish to approximate the light leaving a cluster as light leaving a single point. In order to do so, we need the following definition of light leaving a point:

Definition *Radiant intensity:*

Radiant intensity $I(\vec{\omega})$ is the power emanating from a point in direction $\vec{\omega}$ per unit solid angle in that direction.

The unit of radiant intensity is [W/sr]. Radiant intensity is two-dimensional since a direction can be specified by two parameters.

How do we compute the radiant intensity corresponding to a cluster? In the case of children that are patches, we need to convert the radiance leaving each patch into radiant intensity. For each direction $\vec{\omega}$ the radiant intensity leaving a patch s is given by integrating over its area:

$$I(\vec{\omega}) = \int_s L(x,\vec{\omega})\, V_S(x,\vec{\omega}) \cos\theta_x \, dA_x \, .$$

where $V_S(x,\vec{\omega})$ accounts for occlusion within the cluster. The angle θ_x is the angle between the normal at x and direction $\vec{\omega}$. Summing contributions from all patches in the cluster we get

$$I(\vec{\omega}) = \sum_{s \in S} \int_s L(x,\vec{\omega})\, V_S(x,\vec{\omega}) \cos\theta_x \, dA_x = \int_S L(x,\vec{\omega})\, V_S(x,\vec{\omega}) \cos\theta_x \, dA_x \,.$$

We approximate this spatial and angular distribution with a point with the same directional light-distribution, see figure 6.2.

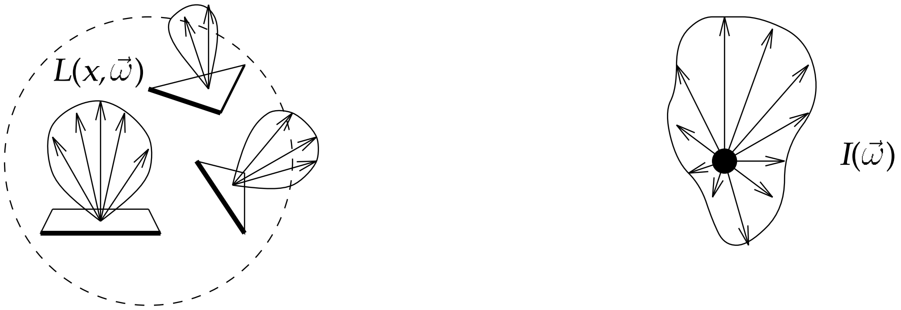


Figure 6.2

Cluster of patches and exitant point approximation.

Suppose we are given a cluster containing subclusters. In this case, the radiant intensity function $I(\vec{\omega})$ of the cluster is given by the sum of the radiant intensities of the subclusters in direction $\vec{\omega}$, except for internal occlusion. The internal occlusion is explained in section 7.6.3.

Adding contributions from both patches and subclusters, the radiant intensity of a cluster is:

$$I(\vec{\omega}) = \int_S L(x,\vec{\omega})\, V_S(x,\vec{\omega}) \cos\theta_x \, dA_x + \sum_{\text{children}} V_{\text{child}}(\vec{\omega})\, I_{\text{child}}(\vec{\omega}) \,.$$

This distribution is stored with each cluster to make transports faster.

6.2.3 Incident Approximation

Light is transported from one cluster to another in the following way: First the radiant intensity distribution I_S of the sending cluster is evaluated in the direction of the receiving cluster. Then this radiant intensity is multiplied by the average visibility V_{SR} between the clusters and divided by the square of the average distance d_{SR} between the clusters. This gives an incident radiance at cluster R, which is added to the incident radiance already at R. See figure 6.3 for an illustration of this approximation.

This transport is done between all pairs of clusters that are interacting. The radiance incident on a cluster R from all clusters in direction $\vec{\omega}$ is then

$$L^{\text{in}}(\vec{\omega}) \approx \sum_{S} \frac{V_{SR}}{d_{SR}^2} I_S(\vec{\omega}).$$

Radiance (including incident radiance) is defined per unit area. In the point approximation we only parameterize incident radiance with a direction — the area is implicitly assumed to be perpendicular to the direction. The distribution of incident radiance is stored with each cluster.

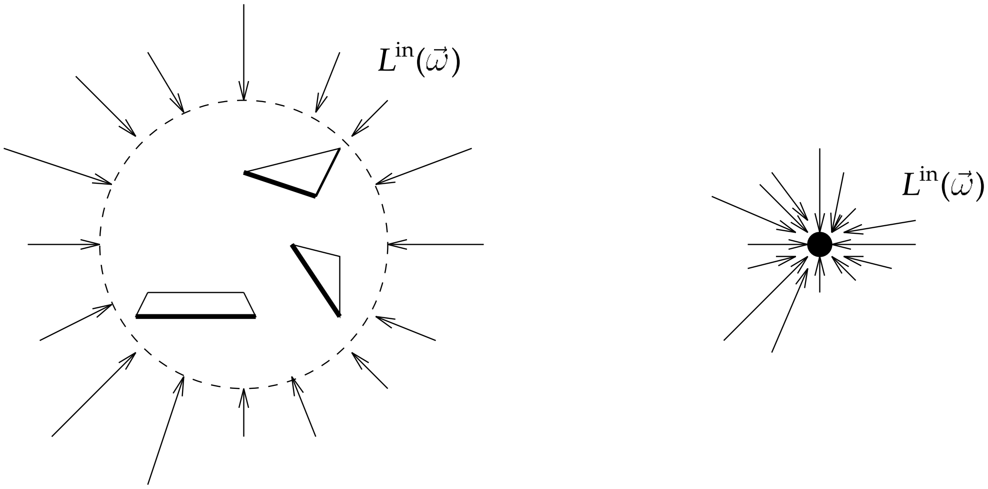


Figure 6.3

Cluster of patches and incident point approximation.

6.2.4 Reflection

To push the incident radiance of a parent cluster to its cluster children, we add incoming radiance L^{in} of the parent (attenuated by internal visibility) to the incident radiance of each child.

The incident radiance on a cluster is reflected off each patch in the cluster. For a point y on a patch, this is done by integrating the incident radiance over the hemisphere $\Omega+$ above y, weighting each incoming direction $\vec{\omega}$ by the corresponding value of the BRDF at y:

$$L(y,\vec{\omega}') = \int_{\Omega+} f_r(\vec{\omega},y,\vec{\omega}')\, V_R(y,\vec{\omega}) \cos\theta_y\, L^{\text{in}}(\vec{\omega})\, d\vec{\omega}\,.$$

6.3 Importance

Here the concept of importance is extended from being between points on surfaces to being between clusters. We show that also in this setting, importance can be transported and represented just like light.

Recall from chapter 4 that directional importance is defined such that

$$\langle \mathcal{G}\Gamma \mid L \rangle = \iint \Gamma(y,\vec{\omega}_{yx})\, G(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x\, dA_y$$

is the power that radiance distribution $L(x,\vec{\omega})$ contributes to the image. The total contribution of all light from cluster S to cluster R is the integral over contributions from all pairs of points x in S and points y in R:

$$\int_{y\in R} \int_{x\in S} \Gamma(y,\vec{\omega}_{yx})\, G(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x\, dA_y\,.$$

Approximating $\vec{\omega}_{xy}$ with $\vec{\omega}_{SR}$ and splitting visibility into three as in section 6.2.1 we get the approximation

$$\left(\int_{y\in R} \Gamma(y,\vec{\omega}_{RS})\, V_R(y,\vec{\omega}_{RS}) \cos\theta_y\, dA_y\right) \frac{V_{SR}}{d_{SR}^2} \left(\int_{x\in S} L(x,\vec{\omega}_{SR})\, V_S(x,\vec{\omega}_{SR}) \cos\theta_x\, dA_x\right)\,.$$

The right integral is the radiant intensity I of cluster S. We define the left integral to be a similar quantity for importance:

Definition *Importance intensity:*

Importance intensity $\Upsilon(\vec{\omega})$ is the importance emanating from a point in direction $\vec{\omega}$,

$$\Upsilon(\vec{\omega}) \equiv \int_{y \in R} \Gamma(y, \vec{\omega})\, V(y, \vec{\omega}) \cos\theta_y \, dA_y \,. \tag{6.1}$$

The units of importance intensity are [m^2].

Using the definitions of radiant intensity and importance intensity, we can write the approximate contribution to the image as

$$\Phi_{SR} \approx \Upsilon(\vec{\omega}_{RS}) \frac{V_{SR}}{d_{SR}^2} I(\vec{\omega}_{SR}) \,. \tag{6.2}$$

Thus, to estimate the contribution of the transfer between S and R we need to approximate the distance and visibility between the clusters as well as $I(\vec{\omega}_{SR})$ and $\Upsilon(\vec{\omega}_{RS})$. Note that the time required to compute this contribution does not depend on the number of patches within each of the participating clusters if we already have a representation of $I(\vec{\omega})$ and $\Upsilon(\vec{\omega})$ at each cluster.

Importance and importance intensity are approximated, stored, and transported in exactly the same manner as radiance and radiant intensity, respectively.

6.4 Bounds

Bounds are used to determine where light can be transported between clusters without introducing significant error. Here we give an overview of the previous work on bounds for clustering and present our bounds. Also, the distributions of bounds that need to be stored at each cluster are introduced.

6.4.1 Previous Work on Bounds

Following Smits *et al.* [88] we denote the maximum value of a function f (of two variables) over some domain $A \times B$ as

$$\lceil f \rceil_{A,B} \equiv \max_{x \in A,\, y \in B} f(x,y)\,.$$

Smits *et al.* present both L_1 norms and L_∞ norms. Here we will concentrate on L_1 norms since they correspond to energy, while L_∞ norms represent maxima.

Alpha Links

When all reflection is purely diffuse, the total flux from cluster R caused by light from cluster S is

$$\Phi_{SR} = \int_\Omega \int_R \int_S f_r(y)\, G(x,y)\, L(x)\, dA_x\, dA_y\, d\vec{\omega} = \pi \int_R \int_S f_r(y)\, G(x,y)\, L(x)\, dA_x\, dA_y\,.$$

The symbol Ω is the sphere of directions. For α-links, the upper bound on this flux is

$$\Phi_{SR} \;\leq\; \pi \left(\sum_r A_r \lceil f_r(y) \cos\theta_y \rceil_{S,r} \right) \left(\sum_s \left\lceil \frac{\cos\theta_x}{\|x-y\|^2} \right\rceil_{s,R} \int_s L(x)\, dA_x \right)\,.$$

This upper bound is a product of two terms: the upper bound of the radiance leaving S reaching R, and an upper bound of the fraction of this radiance that is actually reflected by R. For each patch s in cluster S, the radiance is integrated (this radiance is direction-independent since all reflections are diffuse), and multiplied by an upper bound on the solid angle of cluster R seen from patch s. All these upper bounds are summed to give an upper bound on the radiance leaving S in the direction of R. To compute the fraction that is reflected by cluster R, the upper bounds on cosine of the angle of incidence times reflectance times area of each patch r are computed and added.

In short, this upper bound takes into account area, orientation, radiance, minimum distance, and reflectance. The upper bound on visibility is set to 1, corresponding to full visibility everywhere.

The time to compute a bound is $O(m+n)$, where m and n are the numbers of patches in the sending and receiving clusters, respectively, since all patches in the sending cluster have to be evaluated and all patches in the receiving cluster have to reflect. Smits *et al.* show that transporting light within a fixed tolerance in a hierarchy of clusters containing n_p initial patches results in $O(n_p)$ transports with total transport cost of $O(n_p \log n_p)$.

Beta Links

An asymptotically lower total link cost is achieved using coarser bounds. These bounds are obtained by ignoring the orientations of the patches in each cluster. Thus, for a given cluster-to-cluster transport, it is assumed that each source patch is directly facing and visible to all the receivers, that each receiving patch is highly reflective, and that all the patches are as close as possible (within the bounding box) to the other cluster:

$$\Phi_{SR} \leq \pi \left(\sum_r A_r \right) \left\lceil \frac{1}{\|x-y\|^2} \right\rceil_{S,R} \left(\int_S L(x)\, dA_x \right) .$$

If $\sum A$ and $\int L(x)\, dA_x$ are stored with each cluster, this bound can be evaluated in constant time. Thus the total link cost is $O(n_p)$. Links using this bound are referred to as β-links. Because this bound is so crude, β-links are only used to represent the most negligible transports between clusters.

In their implementation, Smits *et al.* used β-links where possible, α-links where higher accuracy is required, and patch-to-patch transports for high accuracy. Since most transports require the use of α-links, the total cost of the clustering algorithm is $O(n_p \log n_p)$.

6.4.2 Bounds using Omega links

Consider first a transport between two patches, r and s. The amount of power that is reflected by patch r due to illumination by patch s and that ultimately reaches the image is given by

$$\Phi_{sr} = \int_r \int_s \Gamma(y,\vec{\omega}_{yx})\, G(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x\, dA_y . \tag{6.3}$$

Replacing the integration over patches r and s in equation (6.3) with the product of their areas times an upper bound on the integrand, we obtain the bound

$$\begin{aligned} \Phi_{sr} &\leq A_r A_s \left\lceil \Gamma(y,\vec{\omega}_{yx})\, G(x,y)\, L(x,\vec{\omega}_{xy}) \right\rceil_{s,r} \\ &= A_r A_s \left\lceil \Gamma(y,\vec{\omega}_{yx}) \frac{\cos\theta_y \cos\theta_x}{\|x-y\|^2}\, V(x,y)\, L(x,\vec{\omega}_{xy}) \right\rceil_{s,r} . \end{aligned}$$

Splitting this upper bound into several parts yields

$$\begin{aligned} \Phi_{sr} &\leq A_r A_s \left\lceil \Gamma(y,\vec{\omega}_{yx}) \cos\theta_y \right\rceil_{s,r} \left\lceil \frac{1}{\|x-y\|^2} \right\rceil_{s,r} \left\lceil V(x,y) \right\rceil_{s,r} \left\lceil L(x,\vec{\omega}_{xy}) \cos\theta_x \right\rceil_{s,r} \\ &= A_r \left\lceil \Gamma \cos\theta_y \right\rceil_{s,r} \left\lceil \frac{1}{\|x-y\|^2} \right\rceil_{s,r} A_s \left\lceil L \cos\theta_x \right\rceil_{s,r} . \end{aligned}$$

Here the upper bound on visibility $V(x,y)$ is simply set to 1.

This bound easily extends to a bound on the transfer between two clusters R and S. We only need to replace maxima over the areas of the patches with maxima over the bounding volumes of the clusters and sum over the patches in each of the clusters:

$$\Phi_{SR} \leq \left(\sum_{r \in R} A_r \left\lceil \Gamma \cos\theta_y \right\rceil_{S,r} \right) \left\lceil \frac{1}{\|x-y\|^2} \right\rceil_{S,R} \left(\sum_{s \in S} A_s \left\lceil L \cos\theta_x \right\rceil_{s,R} \right) . \tag{6.4}$$

The term $\lceil \Gamma \cos\theta_x \rceil_{S,r}$ is the importance (weighted by a cosine) of the receiving patch r maximized over all directions towards cluster S. The term $\lceil L \cos\theta_y \rceil_{s,R}$ is the radiance (weighted by a cosine) of the sending patch s maximized over all directions towards cluster R.

Unfortunately, we do not have the true, exact radiance and importance from each patch, only the approximate solutions computed so far. So we have to use an approximate upper bound, where the true radiance and importance is substituted by the approximate quantities:

$$\Phi_{SR} \leq \left(\sum_{r \in R} A_r \left\lceil \widetilde{\Gamma} \cos\theta_y \right\rceil_{S,r} \right) \left\lceil \frac{1}{\|x-y\|^2} \right\rceil_{S,R} \left(\sum_{s \in S} A_s \left\lceil \widetilde{L} \cos\theta_x \right\rceil_{s,R} \right) . \tag{6.5}$$

6.4.3 Bounds Computable in Constant Time

We could use the bound in equation (6.4) as an alternative to the α-links of Smits *et al.* [88]. This bound involves a sum over source patches and a sum over receiving patches, yielding a clustering algorithm of complexity $O(n_p \log n_p)$. Here it is described how storing directional information with each cluster yields bounds that are computable in constant time, leading to a clustering algorithm of complexity $O(n_p)$.

As mentioned earlier, we would like to treat clusters as point sources with angular distributions. With this goal in mind, we define a maximum outgoing radiant intensity function $\overline{I}(\vec{\omega})$, which gives an upper bound on the radiant intensity leaving a source cluster S in direction $\vec{\omega}$:

$$\overline{I}(\vec{\omega}) = \sum_{s \in S} A_s \max_{x \in s} \left[L(x, \vec{\omega}) \cos\theta_x \right].$$

The units of radiant intensity are watts per steradian [W/sr]. Similarly, we define a maximum outgoing importance intensity function $\overline{\Upsilon}(\vec{\omega})$, which gives the importance intensity leaving a receiving cluster R in direction $\vec{\omega}$:

$$\overline{\Upsilon}(\vec{\omega}) = \sum_{r \in R} A_r \max_{y \in r} \left[\Gamma(y, \vec{\omega}) \cos\theta_y \right].$$

Importance intensity has units of square meters [m^2]. It follows from these definitions that if we maximize over directions between clusters S and R we get the bounds

$$\sum_{s \in S} A_s \left\lceil L \cos\theta_x \right\rceil_{s,R} \leq \left\lceil \overline{I}(\vec{\omega}_{xy}) \right\rceil_{S,R}$$

and

$$\sum_{r \in R} A_r \left\lceil \Gamma \cos\theta_y \right\rceil_{S,r} \leq \left\lceil \overline{\Upsilon}(\vec{\omega}_{yx}) \right\rceil_{S,R}.$$

Finally, substituting the previous two inequalities into equation (6.4)), we get

$$\Phi_{SR} \leq \left\lceil \overline{\Upsilon}(\vec{\omega}_{yx}) \right\rceil_{S,R} \frac{1}{d_{SR}^2} \left\lceil \overline{I}(\vec{\omega}_{xy}) \right\rceil_{S,R} \tag{6.6}$$

Thus, to estimate a bound on the transfer between R and S we need to approximate the minimum distance between clusters as well as the maxima of $\overline{\Upsilon}(\vec{\omega}_{yx})$ and $\overline{I}(\vec{\omega}_{xy})$ over the directions between the two clusters. Note that the time required to perform this computation does not depend on the number of patches within each of the participating clusters since a precomputed representation of $\overline{\Upsilon}(\vec{\omega})$ and $\overline{I}(\vec{\omega})$ is stored with each cluster.

6.5 Discretization

In this section we describe how the six directional distributions on each cluster (radiant intensity, incident radiance, importance intensity, incident importance, upper bound on radiant intensity, and upper bound on importance intensity) are represented in a discrete basis. We also describe how radiance and importance are transported using these discrete basis functions on clusters.

6.5.1 Discrete Bases for Cluster Distributions

The six directional distributions can be approximated using any finite set of basis functions $[b_1(\vec{\omega}), \ldots, b_n(\vec{\omega})]$ defined over the sphere. Thus, each of the six directional distributions described above is represented as an array of coefficients, one coefficient for each basis function. For example, the radiant intensity $I(\vec{\omega})$ is represented as a linear combination

$$I(\vec{\omega}) = \sum_{i=1}^{n} I_i \, b_i(\vec{\omega}).$$

In our implementation we use a piecewise-constant basis. Thus, each coefficient describes the average magnitude of the represented quantity over the solid angle corresponding to the support of the basis function. Since we use a discontinuous basis, an extra discontinuity at the sphere's equator does not matter, so we represent the sphere as two hemispheres. The distribution on each hemisphere is represented on a square domain — like radiance from a patch, see section 5.3.1.

If we wanted a continuous basis for the distributions, a split of the sphere into two hemispheres would be unacceptable since it gives a discontinuity at the equator. Instead a more general framework must be applied: wavelets defined directly on a spherical domain; see Schröder and Sweldens [80].

The storage required for a cluster is proportional to the number of directional basis functions used, but does not depend on the number of patches in the cluster. Each cluster therefore requires a constant amount of storage, for a fixed directional resolution. Because the number of clusters depends linearly on the number of input patches n_p, the entire cluster hierarchy requires $O(n_p)$ storage.

6.5.2 Discrete Transport

As for the continuous case, we divide the transport into three phases: pulling of radiance and importance from patches up through the cluster hierarchy, transport between clusters and pushing (giving incident distributions), and reflection of incident radiance and importance at the surfaces.

Discrete Exitant Approximation

To pull the radiance from a patch, we sample the radiance distribution on the patch. The integral

$$I(\vec{\omega}) = \int_S L(x,\vec{\omega})\, V(x,\vec{\omega}) \cos\theta_x \, dA_x \,.$$

is estimated numerically by point-sampling the patch.

Pulling radiance and importance from subclusters to a cluster is just adding coefficients (attenuated by precomputed internal visibility; see section 7.6.3).

Discrete Incident Approximation

To perform a transfer along a link from a source cluster S to a receiving cluster R, we need to convert the radiant intensity leaving S into incoming radiance arriving at R. Transport between basis functions is simply multiplication by the average visibility and division by the square of the average distance.

If the support of the i-th basis function contains the direction $\vec{\omega}_{SR}$, and the support of the j-th basis function contains the direction $\vec{\omega}_{RS}$, the transfer is performed by updating the j-th receiving coefficient:

$$L_j^{\text{in}} \leftarrow L_j^{\text{in}} + \frac{V_{SR}}{d_{SR}^2} I_i \, b_i(\vec{\omega}_{SR}) \, ,$$

where the visibility V_{SR} is estimated in constant time as described in section 7.6.2.

The transfer of importance from S to R is performed analogously,

$$\Gamma_j^{\text{in}} \leftarrow \Gamma_j^{\text{in}} + \frac{V_{SR}}{d_{SR}^2} \Upsilon_i \, b_i(\vec{\omega}_{SR}) \, .$$

Pushing incident radiance and importance from a cluster to its subclusters is simply adding coefficients attenuated by internal visibility.

Discrete Reflection

For each patch, we need to convert the incoming radiance of the parent cluster into radiance reflected off the patch. The radiance reflected from point x on the patch in direction $\vec{\omega}$ is given by the following integral of the incoming radiance over the hemisphere $\Omega+$ above x:

$$L(x, \vec{\omega}) = \int_{\Omega+} f_r(\vec{\omega}', x, \vec{\omega}) \cos\theta' \, V(x, \vec{\omega}') \, L^{\text{in}}(\vec{\omega}') \, d\vec{\omega}' \, .$$

In our implementation, incoming radiance that gets pushed down to a patch from its parent cluster is only used to update the coarsest basis function on the patch. The coarse basis function represents the average radiance of the patch over all points and directions. In terms of directional distribution coefficients, this update is accomplished by increasing the patch's coefficient by

$$\frac{1}{\pi}\sum_i \rho_{\mathrm{ch}}(x, \Delta\vec{\omega}_i)\, V_i\, L_i^{\mathrm{in}}\, \Delta\vec{\omega}_i\,.$$

Here ρ_{ch} is the conical-to-hemispherical reflectance of the patch [24], and $\Delta\vec{\omega}_i$ is the solid angle corresponding to the support of the i-th basis function on the sphere.

The quantities derived from importance are pushed and pulled in exactly the same manner as those derived from radiance.

6.5.3 Discrete Bounds

Equation (6.6) tells us that in order to bound the transfer between R and S we need to determine the minimum distance between clusters as well as the maxima of $\overline{\Upsilon}(\vec{\omega}_{yx})$ and $\overline{I}(\vec{\omega}_{xy})$, for $x \in S$ and $y \in R$. Similarly to Smits *et al.* [88], we estimate this bound by sampling. A fixed number of pairs (x, y) are chosen randomly, where x is a point in the bounding volume of R and y is a point in the bounding volume of S. For each pair (x, y), we evaluate $\overline{\Upsilon}(\vec{\omega}_{yx})$, $\overline{I}(\vec{\omega}_{xy})$, and $\|x - y\|^{-2}$, recording the maximum value for each of these three quantities. The product of the three maxima is taken as the estimate of the bound. No visibility checks are performed since our upper bound assumes full visibility.

The upper bounds on radiant intensity and importance intensity leaving a patch are also estimated by point sampling. These upper bounds are pulled up the cluster hierarchy similarly to the radiant intensity but not transported on to other clusters.

6.6 Clustering Algorithm

The algorithm starts by constructing a cluster hierarchy containing all the initial patches in the environment. Each cluster has several directional distributions describing its outgoing and incoming radiance and importance, upper bounds on the radiance and importance leaving the cluster in each direction, and visibility in each direction. The outgoing cluster distributions are initialized by pulling emitted radiance and initial importance from the patches towards the root of the cluster hierarchy. Initially, a single link is established from the root cluster to itself. The algorithm then alternates between refining the links (transports) and solving the resulting linear system, until reaching some desired tolerance.

This process is summarized in the following pseudocode:

```
procedure GlossyGlobalIllumination:
    Preprocess
    for a series of decreasing tolerances $\varepsilon$ do
        Solve
        if not finest tolerance $\varepsilon$ then Refine($\varepsilon$)
    end for
    Render
end procedure

procedure Preprocess:
    Construct cluster hierarchy
    Initialize approximate visibility data-structures
    Set up initial links
end procedure
```

```
procedure Solve:
    repeat
        Pull radiance and importance from patches to clusters
        Transport radiance and importance along links
        Push radiance and importance from clusters to patches
    until convergence
end procedure

procedure Refine($\varepsilon$):
    for each link $\ell$ do
        if ErrorBound($\ell$) > $\varepsilon$ then RefineLink($\ell$)
    end for
end procedure
```

In procedure "*Solve*", the transport of radiance along links is both between clusters, between clusters and patches, and between radiance basis functions on patches.

In the rest of this chapter we describe the remaining major components of this algorithm: the creation of clusters and the refinement process. In sections 7.6.2, 7.6.3, and 7.9 the more implementation-dependent stages of the algorithm are described: visibility computation between and within, and rendering using final gather.

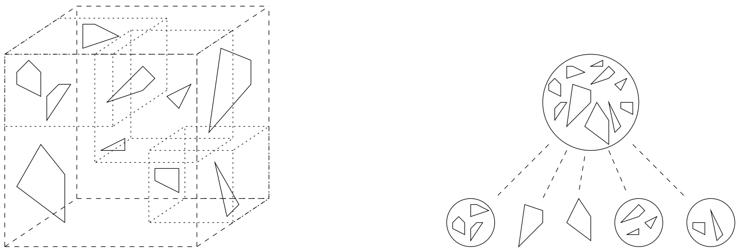
6.7 Creation of Clusters

We start by creating a single root cluster containing all surface patches in the scene. The bounding box of the cluster is divided into eight octants. For each patch in the cluster, one of two things happens:

1. If the size of the patch is smaller than an octant, the patch is assigned to the octant containing its centroid.
2. Otherwise, the patch becomes a direct child of the cluster.

Then every octant containing more than a few patches is made into a child cluster. (Patches in the remaining nearly empty octants become direct children of the parent cluster.) The above process is repeated recursively on any newly created clusters — the recursion terminates for a cluster when there are only a few patches in it.

The above algorithm assigns each patch to a single cluster, without splitting any of the patches. The construction results in a hierarchy of clusters whose bounding boxes may overlap. The bounding box of each child cluster is strictly contained within that of its parent. The children of a cluster can be smaller clusters as well as patches, as illustrated in figure 6.4.



(a) (b)

Figure 6.4

A cluster with cluster children and patch children: (a) geometry; (b) schematic view.

If we assume a relatively uniform distribution of n_p patches, the cluster construction described above will result in a hierarchy with $O(\log n_p)$ levels. At each level of the hierarchy, the construction has to consider $O(n_p)$ patches to distribute them into octants. The construction of the entire hierarchy therefore takes $O(n_p \log n_p)$ time. Note that a similar hierarchy could also be constructed from the bottom up, by first associating each input patch with a cell on some fine uniform grid. Then adjacent non-empty cells could be grouped together into larger cells. This approach would construct a hierarchy of depth $O(\log n_p)$ in $O(n_p)$ time. Our experience has been that the hierarchy construction takes only a small fraction of the total running time, so we have not found it necessary to improve the complexity of this stage.

6.8 Refinement

If the bound corresponding to a link exceeds the current tolerance, the link is refined. In our implementation, a link between two clusters is refined by splitting the "largest" cluster, linking its children to the "smaller" cluster, and recursively considering those new links for refinement. We define the size of a cluster as the sum of the areas of the patches contained within it. For a link between a patch and a cluster, the cluster end of the link is refined. A link between two patches is refined as described in chapter 5. New links that are established in the refinement process will be used later to transfer energy, so an approximate visibility term is computed and stored with each link when it is created, as described in section 7.6.2.

Another possible type of refinement is to increase the accuracy of the directional distributions stored with each cluster, rather than replace links between two clusters with links between their subclusters. We avoid increasing the angular resolution of directional distributions in order to maintain constant space requirements for clusters (and time requirements for updating them). It is important to emphasize that using a fixed resolution for clusters does not limit the accuracy of the simulation: so long as we bound the error in each transfer, all significant transfers will eventually be refined down to the patches. At this point, the algorithm of chapter 5 will resolve these transfers to the required accuracy.

Following Greengard [43], Hanrahan *et al.* [46], and Smits *et al.* [88], we assume that the number of links from each cluster to other clusters in the hierarchy is a constant that depends on the current tolerance, but not on the number of patches n_p. Because the hierarchy contains $O(n_p)$ clusters, there are $O(n_p)$ cluster-to-cluster links that can possibly be refined. As mentioned earlier, the transfer can be bounded in constant time, and visibility is ignored for refinement. As a result, each refinement stage requires only $O(n_p)$ time. The corresponding cost in the algorithm described by Smits *et al.* is $O(n_p \log n_p)$, because their α-links cannot be bounded in constant time.

Our approach to clustering, as well as that of Smits *et al.*, might raise concerns regarding the accuracy of the resulting solutions. In both clustering algorithms, transfers between clusters are approximated very coarsely. Furthermore, refining a transport between two clusters by breaking it into several smaller pieces does not in itself guarantee an improvement in the accuracy of the resulting approximation: this will only be the case if the sum of the errors corresponding to these pieces is smaller than the error of the original transport.

Clustering provides a reliable way of determining where we can get away with coarse approximations. All significant transfers, on the other hand, are refined until they take place between patches, where transfers are treated much more accurately. The philosophy behind clustering algorithms is that clusters need not be treated particularly accurately. In a typical complex environment, many of the transfers between clusters are very small because energy falls off with the square of the distance. These transfers have little or no impact on the solution, and even a very coarse approximation suffice.

“The three dots ’…’ here suppress a lot of detail
— maybe I should have used four dots.”
Donald Knuth

Chapter 7

Implementation

Implementing the hierarchical techniques described in the previous chapters requires a number of practical considerations. In this chapter, the features of the implementation are described. The specification of the scene defines geometry, emission, and reflection. The geometry consists of surface patches that are quadrilaterals and Bézier patches. The emission from a light source can have both spatial and angular variation. The eye emits importance in the directions within view. The reflection is a combination of an anisotropic glossy reflection model and texture maps. Transport coefficients are computed with adaptive accuracy to make the initial computations fast, while still ensuring that the solution converges to the correct solution. There are three types of visibility being computed. One is the visibility between patches, which has to be computed to high accuracy. The second is visibility between clusters, where a fast, coarse approximation suffices. The third is visibility inside each cluster. The data structures to represent basis function coefficients and transport coefficients are also described. Finally, rendering using a final gather pass is described. A final gather improves the visual quality of the image by computing smooth radiance, exact shadows, and textures at pixel-resolution.

7.1 Surface Geometry

Since the hierarchical techniques are independent of the geometry of the patches, as long as they can be parameterized on a unit square, it is natural to incorporate curved surfaces. In our implementation, a patch can be either a Bézier surface or a quadrilateral. It would also be straightforward to add non-uniform rational B-spline surfaces [30]. In fact, the only requirements are that we need to be able to quickly

1. compute a position, surface normal, and differential area associated with a given parametric point (u_1, u_2), and
2. determine the intersection of a ray with the surface.

The images in figure 8.9 show a sphereflake consisting of 728 Bézier patches. The images in figure 8.11 show a scene containing several curved objects. For example, the teapot consists of 28 Bézier patches, the mug consists of 20, and the doorknob consists of 16.

7.2 Light Sources

By storing the wavelet decomposition of an image as coefficients on a patch, we can model a light source that emits a spatially-varying radiance (like a television screen). In general, not all coefficients of the emitting image will have links from them, but the coefficients are ready to be transported into the scene if the refinement procedure so decides. This technique allows a complex environment to be displayed using simple geometry.

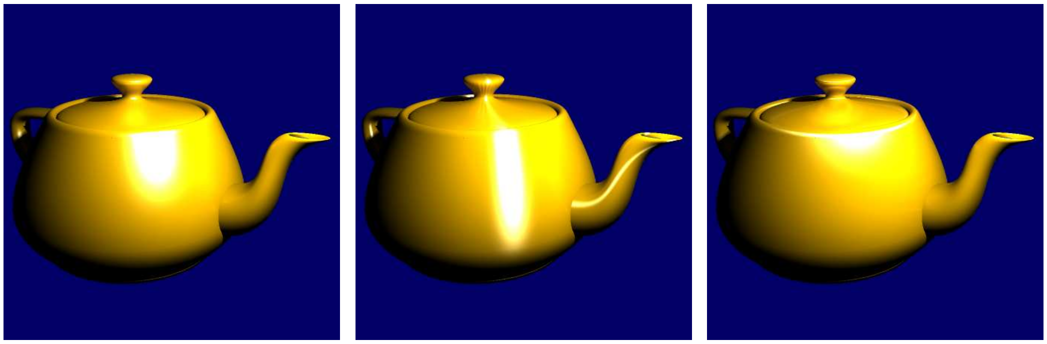
A simple approach to angular variation is to let the emission depend on direction. For example, we model "spotlights" using a Phong-like function in which emission depends on some power of the cosine of the angle between the emission direction and the surface normal of the patch. Spotlights appear dark from most directions because of the very narrow distribution of light they emit. The eye is emitting importance within a frustum, so that the source term for importance is 1 in directions within the view.

We demonstrate the use of spotlights and a spatially-varying emitter (the outdoor environment seen through the window) in figure 8.5. More complex effects such as a slide projector or sunlight through a stained-glass window could be modeled by combining spatial and angular variations in an emitter.

7.3 Reflection Models and Texture Maps

As mentioned in section 2.6.1, we use the Ward isotropic and anisotropic reflection models [98] since they are physically valid and fast to evaluate. Examples of this reflection model can be seen in figure 7.1 and also in figures 8.5, 8.9, and 8.11. In addition to angular variation in reflectance, we use spatially varying reflectances to simulate details of the materials in the scene. We take the BRDF to be the product of a texture and the Ward model. Figures 8.5 and 8.11 demonstrate both texture (on the floor, brick walls, etc.) and an anisotropic reflectance function (on the teapot).

In the course of numerically approximating a transport coefficient, the geometric term and the BRDF are sampled at a number of quadrature points. The reflectance for each quadrature point is determined by a look-up in a texture map, multiplied by the angular value given by Ward's model. Multiresolution tex-



(a) (b) (c)

Figure 7.1

Ward's reflection model: (a) isotropic reflection, $\alpha_u = \alpha_v = 0.2$; (b) anisotropic reflection, $\alpha_u = 0.1, \alpha_v = 0.5$; (c) anisotropic reflection, $\alpha_u = 0.5, \alpha_v = 0.1$.

tures could be incorporated in our method by using a pyramid of texture averages instead of sampling. This approach would eliminate sampling errors from sampling the texture. Gershbein *et al.* [33] present an alternative approach, using wavelet decompositions of textures for radiosity.

7.4 Efficient Computation of Transport Coefficients

The algorithm requires computation of transport coefficients between basis functions, as described in section 3.4.2. Each transport coefficient is defined in equation (3.4) as a six-dimensional integral, which we approximate using numerical integration. For example, the influence of wavelet $\psi\phi\psi\phi^{j}_{\mathbf{i}}(\mathbf{u}_s)$ on another wavelet $\psi\phi\phi\phi^{j'}_{\mathbf{i}'}(\mathbf{u}_r)$ is

$$T_{r\leftarrow s} = \langle \overline{\psi\phi\phi\phi}^{j'}_{\mathbf{i}'} \,|\, \mathcal{T}\psi\phi\psi\phi^{j}_{\mathbf{i}} \rangle \,.$$

The domain of radiance is position and direction, for example x and $\vec{\omega}$, while the domain of our tensor-product basis functions is the four-dimensional hypercube $[0,1]^4$. The transformations were described in section 5.3.1. For convenience, we make the spatial and angular transformations implicit, and write the arguments of the basis functions as points and directions: let the sending position x correspond to the two parameters u_1 and u_2, and let the direction $\vec{\omega}_{xy}$ correspond to parameters u_3 and u_4 (and similarly for the parameters y and $\vec{\omega}$ of the receiving basis function). Then the inner product above takes the form

$$\begin{aligned} T_{r\leftarrow s} &= \langle \overline{\psi\phi\phi\phi}^{j'}_{\mathbf{i}'} \,|\, \mathcal{T}\psi\phi\psi\phi^{j}_{\mathbf{i}} \rangle \\ &= \int_{\vec{\omega}y} \overline{\psi\phi\phi\phi}^{j'}_{\mathbf{i}'}(y,\vec{\omega}) \int_{x} f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, G(x,y)\, \psi\phi\psi\phi^{j}_{\mathbf{i}}(x,\vec{\omega}_{xy})\, dA_x\, dA_y\, d\vec{\omega} \\ &= \int_{xy} \left[\int_{\vec{\omega}} \overline{\psi\phi\phi\phi}^{j'}_{\mathbf{i}'}(y,\vec{\omega})\, f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, d\vec{\omega} \right] G(x,y)\, \psi\phi\psi\phi^{j}_{\mathbf{i}}(x,\vec{\omega}_{xy})\, dA_y\, dA_x \,. \end{aligned} \tag{7.1}$$

Note that only the BRDF and the receiving basis function depend on the direction $\vec{\omega}$. Our numerical integration routine samples these two functions in its innermost loop, while the remaining functions are evaluated only as the positional variables change.

We approximate integrals such as (7.1) using slightly jittered uniform sampling of the integrand. More accurate rules such as Gauss-Legendre or Gauss-Kronrod quadrature could be employed instead [41, 73, 105].

7.5 Adaptive Numerical Integration

If we always use a numerical integration rule of high accuracy to compute transport coefficients, time is wasted evaluating the kernel for many transports that have little effect on the final image. On the other hand, the significant transport coefficients must be computed to high precision; otherwise, the solution will not converge to the correct value. It is therefore advantageous to use an adaptive numerical integration technique that reduces error in transport coefficients, particularly on transport coefficients that are refined by the oracle. We have implemented such an adaptive integration as part of the refinement procedure.

At the time a transport coefficient is computed, the numerical integration technique described in section 7.4 is used. Later, if more detailed transport coefficients are computed between the same supports, the kernel is sampled more densely. These samples are re-used to recompute the coarse transport coefficient more accurately.

As described in section 5.6, the transport coefficient between two scaling functions is computed at the same time as the transport coefficients between other basis functions with the same support (at practically no extra cost since the necessary kernel-evaluations have been done already). Since wavelets in a certain space can be expressed as a linear combination of scaling functions in higher spaces, coarse-level transport coefficients between wavelets can be recomputed by taking linear combinations of the transport coefficients between finer-level scaling functions. In this way, transport coefficients are adaptively recomputed where the kernel is sampled densely.

7.6 Visibility

As pointed out many times already, solving the glossy global illumination problem is much more demanding than solving the diffuse global illumination problem. However, visibility computations are an exception: the visibility computations required to solve the glossy global illumination problem are exactly the same as for diffuse global illumination.

7.6.1 Visibility between Patches

When computing the transport coefficient between two basis functions using numerical integration as described above, the visibility between two points has to be determined. To avoid checking all patches in the entire environment for occlusion, we use the hierarchy of clusters and their bounding boxes. Starting with the top-level cluster, it is recursively determined which cluster bounding-boxes the line segment intersects, and only patches withing these clusters are checked for occlusion. This hierarchical method requires order $O(\log n_p)$ patches to be checked for occlusion instead of $O(n_p)$ without the hierarchy.

7.6.2 Visibility between Clusters

For transports between clusters, a more approximate estimate of visibility is used. This is necessary since sampling the visibility would require tests for intersection with objects on the path between the two clusters, which would cost $O(\log n_p)$ tests for each transport between two clusters.

We would like to compute visibility by dividing the scene into voxels, computing visibility for each voxel, and then compositing the visibilities. But exact visibility can not be computed in such a simple way because visibilities can not simply be composited to give an exact composite visibility. Instead, we can give a bound on the composite visibility. Consider first composition of occlusion, for example the composition of two occlusions o_1 and o_2. if the objects causing occlusions o_1 and o_2 overlap completely, the composite occlusion o_{12} equals the maximum occlusion $\max(o_1, o\,2)$. If the objects do not overlap, the composite occlusion is the sum of the occlusions. For partial overlap the composite occlusion is between these two extreme cases, so a bound on the composite occlusion is:

$$\max(o_1, o_2) \;\leq\; o_{12} \;\leq\; \min(1, o_1 + o_2)\,.$$

For visibility $v = 1 - o$, these bounds correspond to

$$\max(v_1 + v_2 - 1, 0) \;\leq\; v_{12} \;\leq\; \min(v_1, v_2)\,.$$

Following Sillion [83], we compute approximate composite visibility by multiplication of visibilities,

$$v_{12} \;\approx\; v_1 v_2\,.$$

This product is within the bounds above. Also following Sillion [83], we approximate visibility for a voxel with an isotropic volume density. As part of the preprocessing, the environment is divided into a fixed number of voxels, and the total area of the patches within each voxel is computed. This preprocessing can be done in time linear in the number of patches. The area within a voxel corresponds to a density (extinction coefficient) κ inside the voxel. This is a good approximation for many small, randomly distributed patches; however, it is a very rough approximation for large patches with nonrandom orientations. The visibility between any two points x and y can be approximated by casting a ray from x to y and visiting each voxel intersected by the ray. The visibility of a ray traveling a distance d through a voxel is given by $e^{-\kappa d}$. The visibility of the entire ray is approximated by the product of the visibilities through each voxel. The average visibility for a link between clusters is estimated by tracing a fixed number of rays and averaging the values. The way visibilities are computed and composited makes this visibility computation method very approximate. The one redeeming quality about this method is that visibility can be approximated in constant time independent of the complexity of the geometry.

There are some more sophisticated methods for visibility computations that have near-constant expected time per visibility ray. Among them are the ray classification scheme proposed by Arvo and Kirk [7] and the ray coherence scheme of Ohta and Maekawa [65]. These schemes are nontrivial to implement, but could perhaps improve the accuracy of the visibility computations without interfering with the linear complexity of our algorithm.

7.6.3 Visibility within Clusters

The pulling and pushing operations described in section 6.5.2 redistribute quantities from a cluster to its children and vice versa. To account for internal occlusion within a cluster, we need to know the fraction of radiance leaving each child of the cluster that is unoccluded by other children of the cluster. There is no need to compute visibility within a cluster every time we pull or push, as the geometry and the hierarchy remain fixed. Therefore, once the cluster hierarchy has been created, we use a preprocessing step that computes the internal visibility function $V(\vec{\omega})$ for each child. This function gives the probability that a ray leaving the child in direction $\vec{\omega}$ exits its parent cluster without occlusion. This function is estimated by shooting a fixed number of rays for each directional basis function. Each ray is tested for intersection with all the siblings of the child.

Since the patches within each cluster are organized hierarchically, the expected time per ray is $O(\log n)$, for a parent cluster whose subtree contains a total of n patches. A cluster hierarchy containing p patches has $\log p$ levels. It follows that a cluster at level k (the root being level 0) is the root of a subtree with $\log p - k$ levels containing $8^{\log p - k}$ patches (assuming a branching factor of 8). Therefore, a ray intersection test takes time proportional to $\log(8^{\log p - k}) \sim \log p - k$. There are 8^k clusters at level k, so the total time for the visibility preprocessing within clusters is given by

$$c \sum_{k=0}^{\log p - 1} 8^k (\log p - k) \leq cp\,.$$

The constant c depends on the time it takes to perform a single ray–patch intersection test, on the number of children per cluster, on the number of rays shot per directional basis function (four in our implementation), and on the number of directional basis functions (thirty-two in our implementation). All of the these quantities are independent of n_p, and therefore the total time taken by this stage of the algorithm is $O(n_p)$.

7.7 Data Structure for Basis Function Coefficients

As in hierarchical radiosity algorithms [24], the matrices $\widetilde{\mathbf{T}}$, $\widetilde{\mathbf{L}}$, $\widetilde{\mathbf{L}}_e$, $\widetilde{\mathbf{\Gamma}}$, and $\widetilde{\mathbf{\Gamma}}_e$ are never formed explicitly. Entries of $\widetilde{\mathbf{L}}$, $\widetilde{\mathbf{L}}_e$, $\widetilde{\mathbf{\Gamma}}$, and $\widetilde{\mathbf{\Gamma}}_e$ are associated with the surface patches, while entries of $\widetilde{\mathbf{T}}$ are stored as "links" between radiance coefficients (and between importance coefficients). The coefficients and links are allocated dynamically as the solution is refined.

A hierarchy of basis function coefficients is associated with each patch. We have implemented the hierarchy as a tree where each node contains all coefficients ℓ with the same indices (space j and translations $i_1, \ldots, i_4$). Each node contains the 15 wavelet coefficients and one scaling function coefficient for each of six "color bands": red, green, and blue radiance; and red, green, and blue importance. Only the scaling function coefficients within root nodes (corresponding to space V^0) are part of the hierarchical representation. The remaining scaling functions are used for adaptive numerical integration as explained in section 7.5. Each node contains 16 pointers to child nodes that contain the coefficients in the next, more refined, space. The child nodes correspond to coefficients with all 16 combinations of indices $2i_1$ or $2i_1 + 1$, ..., $2i_4$ or $2i_4 + 1$. The pointers between nodes are illustrated in figure 7.2. Initially, each patch has only a single node containing a scaling function coefficient in space V^0 for each color band.

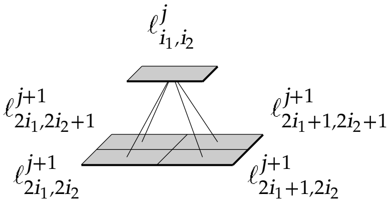

$\ell^{j}_{i_1,i_2}$
$\ell^{j+1}_{2i_1,2i_2+1}$
$\ell^{j+1}_{2i_1+1,2i_2+1}$
$\ell^{j+1}_{2i_1,2i_2}$
$\ell^{j+1}_{2i_1+1,2i_2}$

Figure 7.2

Tree of basis function coefficients on a patch (simplified to 2D Flatland, where each node has only 4 children). Each node contains all coefficients of wavelets with the same support, and the root node also contains scaling function coefficients.

7.8 Data Structure for Transport Coefficients

The transport coefficients that describe the transport between radiance (and importance) basis functions at different patches are stored on *links*. As described in section 5.6, the transport coefficients from a sending basis function to all 15 wavelets and the scaling function sharing support are computed at the same time. In our implementation, these 16 transport coefficients are stored on the same link. Alternatively, each transport coefficient could be stored on a separate link, but the extra storage overhead makes this impractical since each link needs to point to the two basis functions between which it is transporting light. As another alternative, the transport coefficients between all 15^2 possible combinations of wavelets on sender and receiver could be stored on the same link. This method would also waste memory since it sets up links with room for many transport coefficients that are never computed, for example, if the sending coefficient are too low.

Each link contains:

- a pointer to the node from which it is transporting radiance and importance,
- information about what type of basis function it is transporting from,
- 15 entries of $\widetilde{\mathbf{T}}$ for each of the three color bands,
- the sample variation encountered while computing those transport coefficients (used for refinement as described in section 5.6),
- a scaling-function-to-scaling-function transport coefficient for each of the three color bands (used for adaptive improvement of transport coefficients as described in section 7.5), and
- a pointer to the next link to the same receiving basis functions.

Note that there can be several links between the same pair of nodes, each connecting different basis functions with the same support, as illustrated in figure 7.3. All links pointing to a given node are organized in a linked list. Initially links are set up between root nodes for all pairs of patches that are mutually visible.

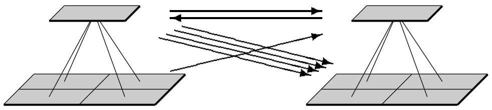


Figure 7.3

Example of links between basis functions (in Flatland).

We only consider creating new links between basis functions b_s and b_r where there is a link from b_s to b_r's parent, or where there is a link from b_s's parent to b_r; see figure 7.4. This restriction reduces the number of new links that have to be considered for refinement at one time, while still allowing all possible links to be created eventually. The existing link contains information about the kernel variation encountered while computing that transport coefficient; this variation is used as an estimate of the yet uncomputed transport coefficient to or from a child basis function. The one exception to this scheme is root nodes since they have no parent. Here the link between the two scaling functions is used for information about kernel variation for the wavelets in W^0.

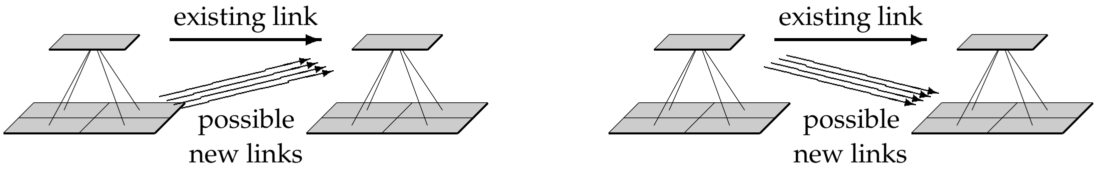


Figure 7.4

New links to be considered (in Flatland).

Links are never destroyed in our algorithm. By contrast, the approach described by Gortler *et al.* [41] removes a link at one level of the hierarchy and replaces it with multiple links at a finer level of detail (because they use a scaling function representation at all levels of detail).

The adaptive numerical integration takes place after new links have been set up. The integration involves a bottom-up traversal of all links. For each link, we first check whether the two nodes this "parent link" connects have children, and if so, whether there are links between some of the children. If there are links between the children, the transport coefficients on the parent link are recomputed by linear combination of the scaling-function-to-scaling-function transport coefficients on links between the children.

7.9 Final Gather

The finite-element solution eventually converges to a physically accurate and visually pleasing image. Unfortunately, it takes too much time to compute the solution and too much space to represent it in this way. Instead, we can compute an overall approximation to the solution with the finite-element method, and then use a *final radiance gathering* pass to get a visually pleasing image.

Following the ideas that Reichert [75], Lischinski *et al.* [61], and Smits [87] used for radiosity, we have implemented a final radiance gathering pass. For each pixel in the image, we perform a final gathering of light to the surface point that corresponds to the midpoint of the pixel. If supersampling is desired, a final gather is done to the center of each subpixel.

The computed radiance solution is used to solve the rendering equation for each point corresponding to a pixel center. This avoids the last reprojection onto the finite-element basis. Let y be the point in the scene corresponding to the center of a pixel, let p be the patch y is on, and let e be the eye point. For each patch s with a link to p, we evaluate the radiance equation (2.13) with sending radiance over patch s,

$$L(y,\vec{\omega}_{ye}) = \int_s f_r(\vec{\omega}_{xy}, y, \vec{\omega}_{ye})\, G(x,y)\, L(x,\vec{\omega}_{xy})\, dA_x\,.$$

Since the receiving position y is fixed and the radiance is reflected towards the eye e, the integration is only over sending positions x on patch s.

For patch-to-cluster links to the cluster containing patch p, we perform the same final gather as for links between patches. However, visibility can be computed less accurately since a patch-to-cluster link supposedly carries less important radiance — otherwise it would have been refined in the solution.

For a cluster-to-patch link to p, or a cluster-to-cluster link to the cluster containing p, a different final gather is performed. A small number of points x in the cluster (ten in our implementation), are selected at random. For each point, the distribution of radiant intensity of the cluster is evaluated in the direction to point y. That value is multiplied by the average visibility from the cluster, the cosine term at the receiving point, and the inverse square of the distance:

$$L(y,\vec{\omega}_{ye}) = f_r(\vec{\omega}_{xy}, y, \vec{\omega}_{ye})\, \frac{\cos\theta_y}{d_{xy}^2}\, V(x,y)\, I(\vec{\omega}_{xy})\,.$$

Formally, the final gather corresponds to changing to a piecewise-constant basis, where the support of each basis function is the projection of a pixel onto a surface in the scene. Intuitively, this basis is tailored to be visually pleasing. The final gather smooths the discontinuities in the wavelet representation, makes highlights and textures crisper, and creates the correct soft shadows from area light sources. The improvement brought about by the final gather can be seen by comparing figures 8.5(e–f) and 8.11(a–f).

Another way of describing the final gathering step is in the context of distribution ray tracing [26]. When a ray cast from the eye intersects a surface in the scene, a group of reflected rays are traced from the intersection point to points on other surfaces in the scene. A constant number of rays are cast to the support of each selected basis function in the radiance solution. In this way, the directions of the rays are guided by the solution. Thus, the most refined areas of the radiance solution are sampled the most by the distribution of reflected rays. Note that the costly “explosion” of the number of recursive bounces used in distribution ray tracing is avoided since only a single bounce is followed.

We utilize the different characteristics of direct and global illumination to speed up the computation. Direct illumination has high intensity and fast variation, so it needs to be computed to high accuracy. Fortunately, it is cheap to compute when there are few light sources in the scene, so we can sample it 16 times for each pixel. Global illumination, on the other hand, often has low intensity and slow variation. At the same time, it is expensive to compute, so we save time by computing it only once per pixel, or even reuse values over several pixels where the variation is low. The texture is evaluated 16 times per pixel, just like the direct illumination.

What is the complexity of the final gather pass? Consider the cost for one ray cast from the eye. First we need to find the nearest intersection with a patch (point y), which takes $O(\log n_p)$ time on the average. Then we need to gather radiance across all the links that contribute to the basis functions whose support includes y. Each cluster and basis function have roughly the same constant number of links, so the total number of links to gather from is proportional to the depth of the hierarchy. There are $O(\log n_p)$ levels of the cluster hierarchy to gather from, and for these links visibility is computed in constant time. There are d levels of the patch basis function hierarchy to gather from, where d depends on the accuracy of the solution but not on n_p, and each link from a patch basis function requires $O(\log n_p)$ time for computing visibility. The total cost per ray from the eye is therefore $O(\log n_p + \log n_p + d \log n_p) = O(\log n_p)$. Note that the final gather, too, benefits from clustering: Without clustering the solution at a surface point y is influenced by as many as $O(n_p)$ links from patches. Each link needs to be checked accurately for visibility, taking $O(\log n_p)$ time. So without clustering, the final gather would spend $O(n_p \log n_p)$ time per ray from the eye.

Once a finite-element radiance solution has been computed, the final gather requires no additional memory. We often use a strategy of computing as accurate a solution as the memory of our computer can store, and then perform a final gather pass to improve the visual quality. However, some of our results (see section 8.7) suggest that perhaps a coarser solution would suffice for creating plausible images with global illumination effects. If so, there are many interesting trade-offs involved in deciding when to stop the finite-element solution.

“Look here, upon this picture, and on this.”
“Hamlet” by Shakespeare

Chapter 8

Results

In this chapter, tests of different aspects of the algorithm are shown. The algorithm is first tested on a very simple scene consisting of three patches. For the simple scene, a reference solution is easy to obtain, so convergence and convergence rates can be tested. Next, the algorithm without clustering is tested on a scene consisting of 152 patches. Then we experimentally verify the theoretical predictions regarding the $O(n_p)$ asymptotic complexity of our clustering method, and compare its performance with the α-links of Smits *et al.* Next, we examine the accuracy and effectiveness of our cluster representation. Then, we examine the results produced by our method on a highly glossy environment and make qualitative comparisons with Ward’s RADIANCE system [99]. Finally, the effectiveness of the method for complex glossy environments is demonstrated using an architectural interior containing nearly 8000 initial surfaces.

8.1 Three Patches

To test convergence and convergence rates, we tried a simple scene consisting of two tiny patches and a large patch. The geometry is shown in figure 8.1. Patch 1 is emitting radiance L_e, and this light is reflected by patch 2 according to Ward's glossy reflection model [98] with $\alpha = 0.2$. This glossy reflection results in a directional radiance distribution on patch 2. The light from patch 2 is received at patch 3, which is a diffuse reflector.

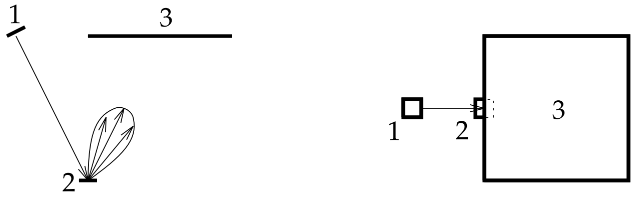


Figure 8.1

Three patches seen from the side and from above.

The angular variation of the radiance distribution on patch 2 is shown in the rightmost image in figure 8.2. This reference solution was computed as

$$L_2(y,\vec{\omega}) \;=\; f_r(\vec{\omega}_{xy}, y, \vec{\omega})\, G(x,y)\, L_e(x,\vec{\omega}_{xy})\, A_1\,,$$

for directions $\vec{\omega}$ on the hemisphere (transformed from the unit square by the transformation described in section 5.3.1). Here x is the midpoint of patch 1, y is the midpoint of patch 2, and A_1 is the area of patch 1. Converging finite-element approximations of this angular variation are shown in figure 8.2, along with difference images illustrating the difference between the approximations and the reference image. The corresponding convergence is shown in the graph in figure 8.4(a).

The spatial variation of the radiance on the large diffuse receiver is shown in the rightmost image of figure 8.3. This reference solution was computed as

$$L_3(z,\vec{\omega}) \;=\; f_r(\vec{\omega}_{yz}, z, \vec{\omega})\, G(y,z)\, L_2(y,\vec{\omega}_{yz})\, A_2\,,$$

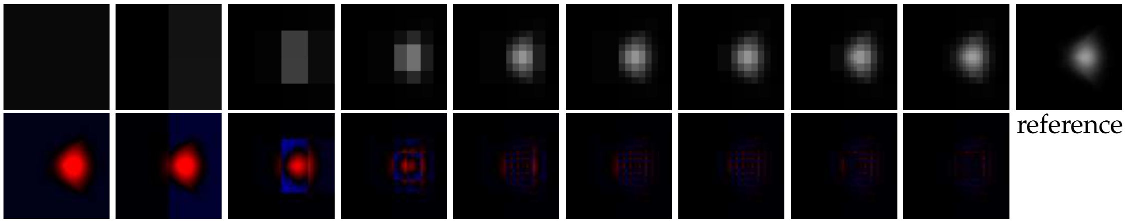


Figure 8.2

Refinement of radiance distribution on patch 2. The distribution is shown as a function of angular parameters for a fixed position. The rightmost image is the reference solution. The bottom row shows the difference between the images and the reference image, red is used where the reference is highest, and blue where the approximation is highest.

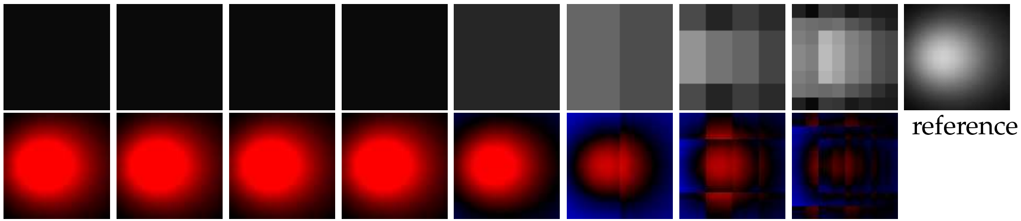
for points z on patch 3, where A_2 is the area of patch 2. Here the direction $\vec{\omega}$ is unimportant since patch 3 is a purely diffuse reflector. The top row of images shows the convergence of the wavelet representation. The first four images are identical, because all refinements take place between patches 1 and 2 (since larger radiance is involved in that transport, and importance is not taken into account). From the fifth image on, the light transports to patch 3 are also refined. Difference images are shown directly below each wavelet solution.

The middle pair of rows shows converging solutions (and difference images) when there is importance at the receiving patch. The transports are refined more at the receiver than in the previous test.

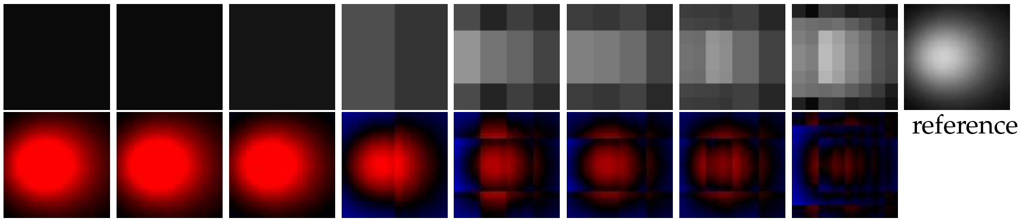
The bottom pair of rows of images in figure 8.3 shows the solution with a final gathering step, but without importance. Here the rendering takes advantage of the refinements of the transports to patch 2 even before the transports to patch 3 are refined.

The convergence of the radiance distribution on patch 3, with and without importance, and with final gathering, is shown in the graph in figure 8.4(b).

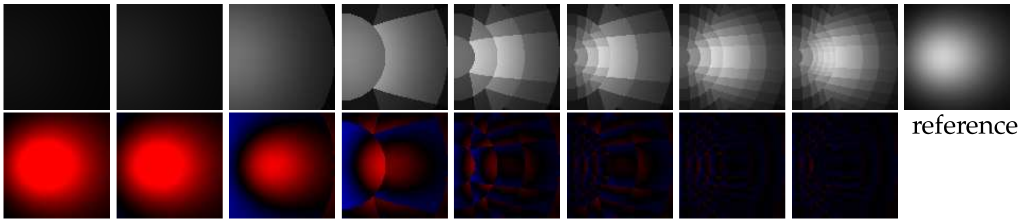
As these results show, a final gather improves the solution, and gives a bet-



Top row: the solution. Bottom row: the difference between the solution and the reference image, red is used where the reference is highest, and blue where the approximation is highest.



Top row: solution with importance at patch 3. Bottom row: the difference between the solution with importance and the reference.



Top row: solution with final gather (without importance). Bottom row: the difference between the solution with final gather and the reference.

Figure 8.3

Refinement of radiance distribution on patch 3. The three rightmost images are identical, they are the reference solution for patch 3.

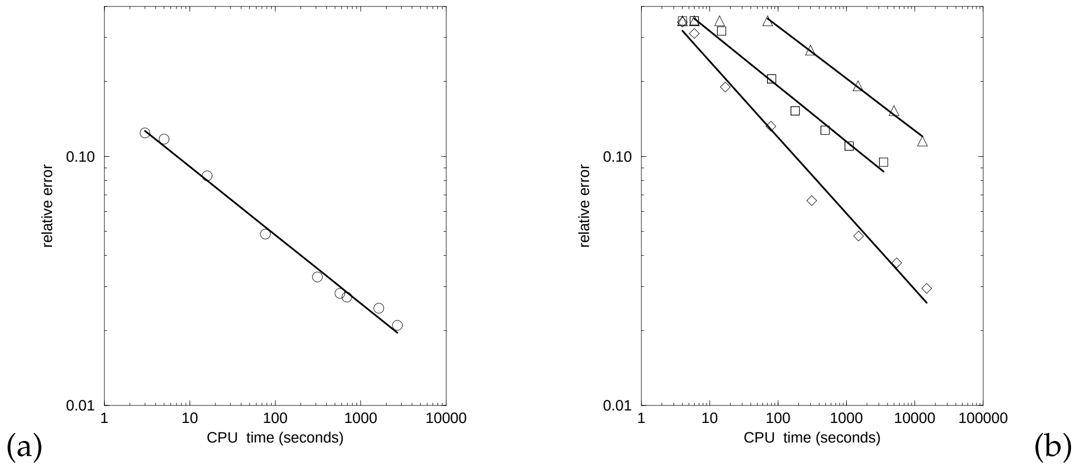


Figure 8.4

(a) Convergence of the radiance distribution on patch 2. (b) Convergence of the radiance distribution on patch 3: The top curve is the solution without importance, the middle curve is the solution with importance, and the bottom curve is the solution with final gather (without importance). The CPU times were measured on a DEC 3000/400 "Alpha" computer.

ter solution in the same amount of CPU time. However, the final gathering step does not appear to improve the convergence rate significantly. The final gather is only useful for display since the result is an image and not a set of basis functions that can be used for further refinement and solution. Future research could examine how far the solution would have to proceed before the final gather is performed, if a given accuracy in the solution is required.

Faster convergence may be obtained by several means:

1. Selecting wavelets with more vanishing moments (but with acceptable numerical integration complexity) would make the transport matrix more sparse.

2. In the example of figure 8.1, a standard construction of the basis would eliminate some basis functions and transports since patch 2 has little spatial variation and patch 3 has no angular variation.

8.2 Scene of Medium Complexity

As a more complex test scene, we used a maze of hallways with a glossy Bézier-patch teapot in the center (see figure 8.5). The scene consists of 152 patches, including 28 Bézier patches, and has 12,603 mutually visible pairs of patches. The teapot's reflectance function uses Ward's reflection model [98], and is anisotropic with specularities $\alpha_u = 0.2$ and $\alpha_v = 0.5$, specular reflectivity $\rho_s = (0.1, 0.1, 0.1)$ and diffuse reflectivity $\rho_d = (0.2, 0.15, 0)$. The illumination consists of 24 "spotlights", patches that emit directional radiance. There is a patch outside the window that emits light according to a scanned image of an outdoor scene, giving the appearance of a full environment beyond the window. The radiance emitted by the lights and reflected in the scene is shown in figure 8.5(a). The objective is to generate an image of this complex scene as seen from the eye, a small patch in the hallway in front of the teapot. All back faces, where no radiance is computed, are rendered gray.

Importance is emitted from the eye and reflected to the important parts of the scene, as shown in figure 8.5(b). This picture demonstrates how small a fraction of the model significantly influences the solution visible from the eye. Figure 8.5(c) is a gray-scale encoding of the number of links between the basis functions on each surface patch. This "refinement image" verifies that most work is performed in areas that are bright and important. Note that we could get arbitrarily large speed-ups, compared to a solution obtained without using importance, by choosing a sufficiently complex scene where many parts do not contribute significantly to the final image.

The program begins by creating 12,603 links between scaling functions, and then solves for the equilibrium distribution. This initial solution can be seen in figure 8.5(d). After six iterations of refinement and solution, there are 126 scaling functions in V^0, 1,518 wavelets in W^0, 18,852 wavelets in W^1, 160,248 wavelets in W^2, 165,495 wavelets in W^3, and approximately 1.73 million links. This solu-

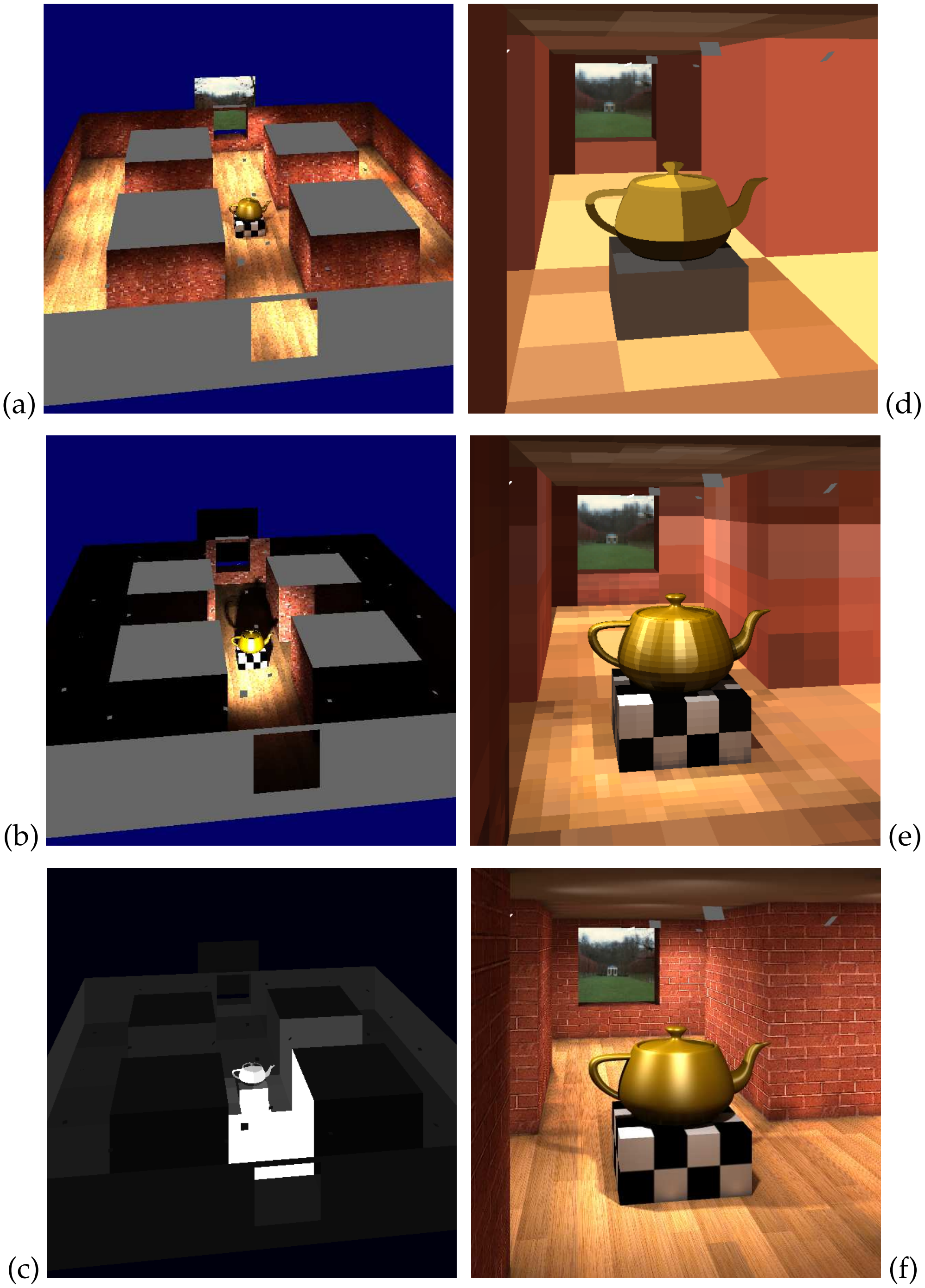


Figure 8.5

Solutions for scene of medium complexity: (a) radiance seen from above; (b) importance seen from above; (c) refinement; (d) initial radiance solution; (e) refined radiance solution; (f) final gather of (e).

tion can be seen in figure 8.5(e). In some refinements, new links are only created within already existing spaces, so the solution space reaches V^4 after six iterations. Running times on a DEC 3000/400 "Alpha" machine were approximately 5 minutes to compute the initial solution, then 100 minutes to iterate the main algorithm and refine as far as V^4 in important parts of the scene. Given this solution, an image of it needs to be created, either by evaluating the solution directly or using a final gather step. It takes 15 minutes to render a 600×600 image using ray casting and evaluation of the solution (see figure 8.5(e)). Alternatively, using a final gathering step for the rendering takes approximately two hours, making it comparable to the solution process itself, and the result is shown in figure 8.5(f). Note the significant color bleeding from the brick walls to the dim ceiling. Also note the glossy highlights on the teapot.

8.3 Asymptotic Behavior

The following experiment is designed to examine the observed asymptotic time complexity of our clustering method and to compare it to that of Smits *et al.* [88]. Since their algorithm was designed for diffuse global illumination, we perform the comparison using a diffuse environment. Our test environment consist of two tessellated spheres: a hollow sphere of radius 2 containing a sphere of unit radius. Triangles on each sphere emit the same constant radiance. This test case is similar to the one used by Smits *et al.*, except the interior sphere added to test the effects of occlusion.

We timed our method on tessellations containing 256, 1024, 4096, 16,384, and 65,536 triangles. Each of the runs involved constructing the cluster hierarchy, initializing internal visibilities for each cluster, pulling, refining to a specified tolerance, transporting energy through the links, and pushing. The graph in figure 8.6 shows how the times spent in the different stages of our method grow with the complexity of the environment, for a fixed tolerance. As a reference, we also plotted the function $y = 0.002x$. This graph demonstrates that all the stages exhibit growth that is roughly linear in the environment size.

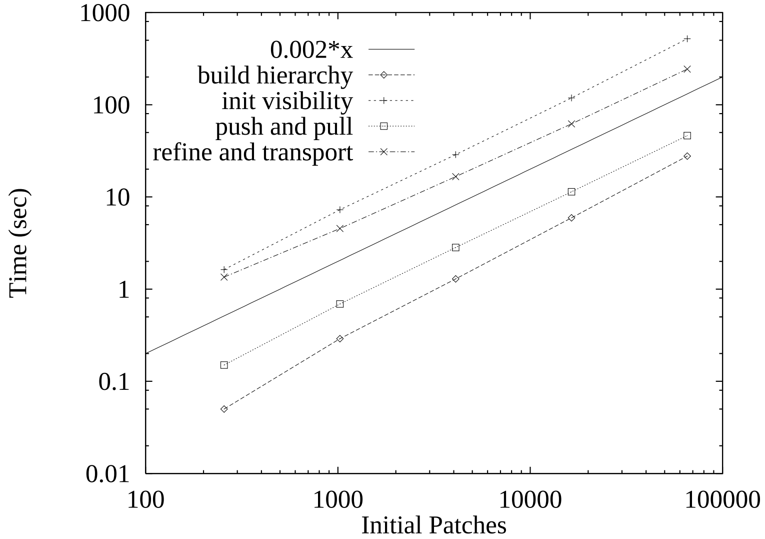


Figure 8.6

The time spent in various stages of the new method as a function of the number of initial patches.

The graph in figure 8.7 compares the total running time of our algorithm to that of Smits *et al.* [88]. Since their algorithm ignores internal occlusion within clusters, we omitted the time of the internal visibility precomputation from the total time for our algorithm. Both methods ran on the same machine, with the same tolerance, resulting in roughly the same number of links in each case. The difference in the asymptotic time complexities of the two methods is revealed by the difference in the slopes of the two log-log plots. Note that for the tessellation with 65,536 triangles, Ω-links exhibit a speedup by a factor of 2.5 over α-links. Since Ω-links are asymptotically more efficient than α-links, the improvement in performance should become even larger for more complex environments.

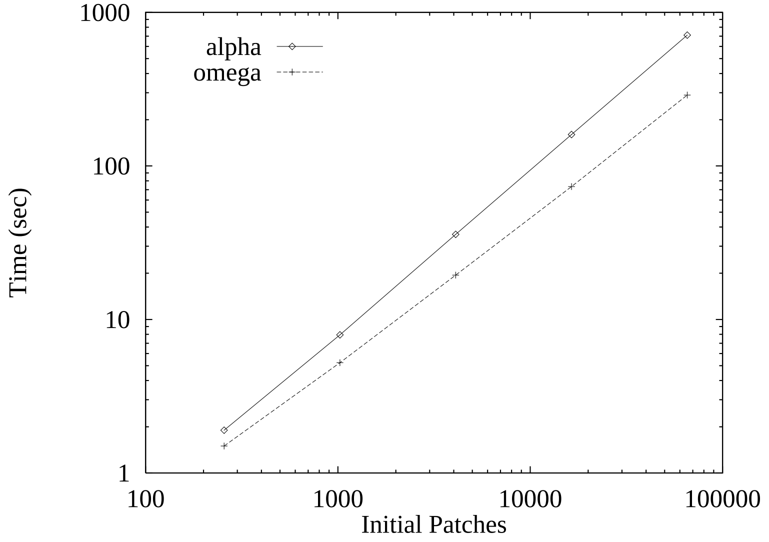


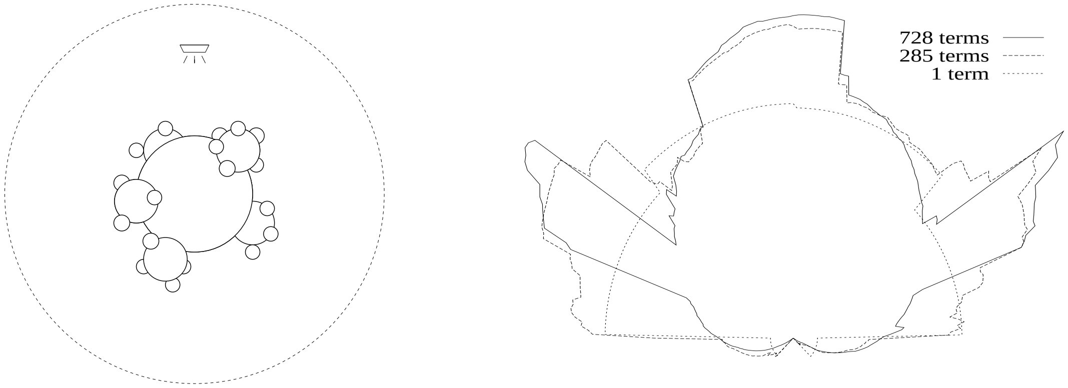
Figure 8.7

The time spent by different clustering strategies as a function of the number of initial patches.

8.4 Accuracy of Cluster Approximation

To test the accuracy of the cluster approximation employed by our method we used a “sphereflake” [44] made of 91 spheres and illuminated by a single square light source. A schematic view of the model is shown in figure 8.8(a). A color image of the same sphereflake (with a wood-textured polygon added behind it) is shown in figure 8.9. The gray, cyan, magenta, and yellow spheres are diffuse; the red, green, and blue spheres are isotropically glossy; and the black, copper, and chrome spheres are anisotropically glossy. In our implementation each sphere is approximated by eight Bézier patches. In total, the sphereflake consists of 728 surface patches, of which 432 are glossy.

A cluster hierarchy containing all of the sphereflake’s patches was constructed as described in section 6.7. The sphereflake was directly illuminated by a single light source, and the reflected radiance was pulled up through the cluster hierarchy. The reflected light was evaluated in several different ways at densely-



(a) (b)

Figure 8.8

Glossy sphereflake: (a) sketch; (b) polar plots of cluster approximations. The solid line represents radiance evaluated on all 728 patches. The dashed line represents the evaluation of 285 terms: radiance from 189 patches, and radiant intensity from 96 clusters. The dotted line shows the coarsest approximation, the radiant intensity from a single cluster.

spaced locations on an orbit around the sphereflake. The ratio of the orbit radius to the sphereflake radius was about 10:1. The results are shown in the form of polar plots in figure 8.8(b). The solid curve corresponds to results obtained by directly integrating the radiance from each of the 728 patches. This curve serves as our reference solution since no clustering approximations were used in its computation. The coarsest approximation is obtained by evaluating the radiant intensity distribution leaving the cluster that contains the entire sphereflake. Also plotted is an approximation of intermediate accuracy, where the transfer between the sphereflake and the receiver on the orbit is refined into 285 transports: 189 with patches, and 96 with clusters.

As the polar plots in figure 8.8 show, the approximation of the entire sphereflake as a single cluster is fairly coarse, yet it captures the most prominent characteristics of the radiant intensity distribution. As the approximation is refined by evaluating more terms, more details are captured and the approximation becomes more accurate.

8.5 Results for a Highly Glossy Scene

The sphereflake model described in the previous section was also used to test the ability of our method to correctly handle different types of interreflections in a scene with highly glossy surfaces. One solution computed by our method is shown in figure 8.9. The finite-element solution shown in figure 8.9(a) took 42 CPU minutes to compute on an IBM RS6000 POWERstation. Of this time, 2 minutes were spent building the hierarchy and performing visibility preprocessing, and the remaining 40 minutes were spent refining and solving. The solution has 1,389 links between clusters, 4,445 links between a patch and a cluster, and 4,052 links between patch scaling functions. There are also 411,645 transports between 313,918 wavelet coefficients representing the radiance distributions on patches.

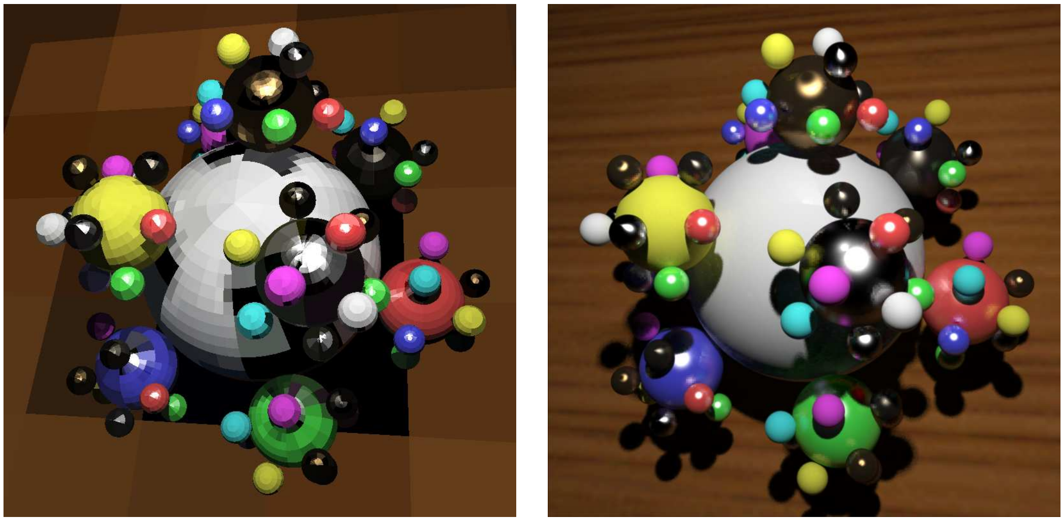

(a) (b)

Figure 8.9

A sphereflake computed using our method: (a) finite-element solution; (b) after final gather.

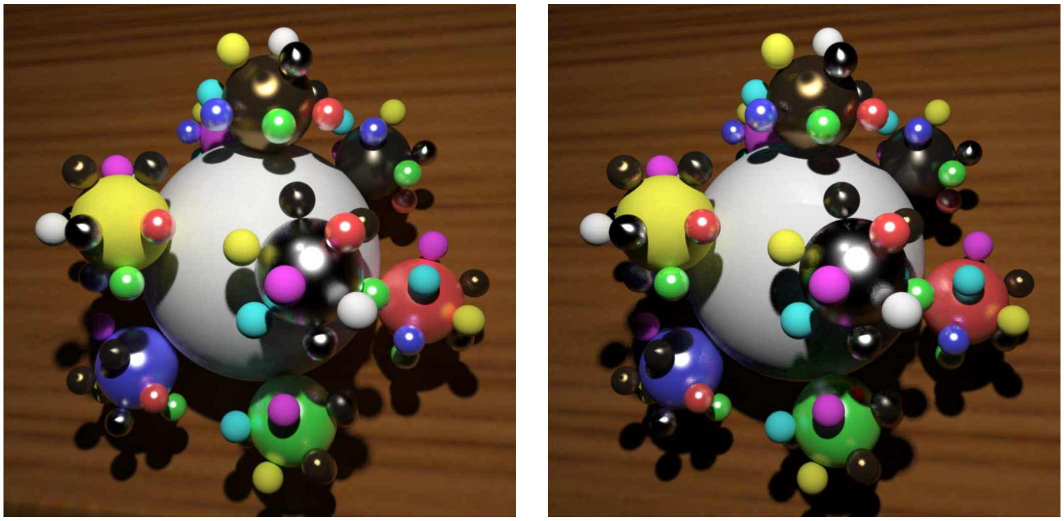
The image in figure 8.9(b) was computed using a final gather at a resolution of 3200 × 3200, and then was reduced to 800 × 800 pixels using a Gaussian filter. The final gather took 240 minutes in addition to the finite-element solution time, resulting in total computation time of 4.7 hours.

A careful examination of figure 8.9 reveals that all of the possible combinations of diffuse and glossy transport mechanisms are present. Diffuse-to-diffuse transport creates subtle yellow, blue, and red color bleeding on the large gray sphere. Diffuse-to-specular transport creates the reflection of the diffuse spheres in the shiny ones. Specular-to-specular transport creates the reflections of highlights in the shiny spheres. Finally, specular-to-diffuse transport illuminates the base of the small, yellow, diffuse sphere near the top of the image.

8.6 Comparison with a Monte Carlo Method

A qualitative comparison of our method and the RADIANCE system [99] was performed using the sphereflake scene described above. Both methods were used on the same machine to compute images at 3200 × 3200 resolution, and then the images were reduced to 800 × 800 pixels using a Gaussian filter. The RADIANCE system took 6.2 CPU hours, resulting in the image shown in figure 8.10(a). After 4.7 CPU hours, our method produced the image in figure 8.10(b). (This is the same image as in figure 8.9(b).) Note that the RADIANCE system had the advantage of performing all its computations with 91 spheres, while in our solution these spheres were represented as 728 Bézier patches. We did, however, use spheres for visibility rays.

Figure 8.10 shows that our solution and that of RADIANCE converge toward the same final result. While RADIANCE is a mature product that has been debugged and optimized over the past decade, clustering and wavelet techniques for glossy global illumination are still in their infancy. Undoubtedly, our method could benefit enormously from further algorithmic refinement and fine-tuning, and our implementation could benefit from further optimization. Our conclusion from this experiment is that hierarchical finite-element methods are a viable



(a) (b)

Figure 8.10

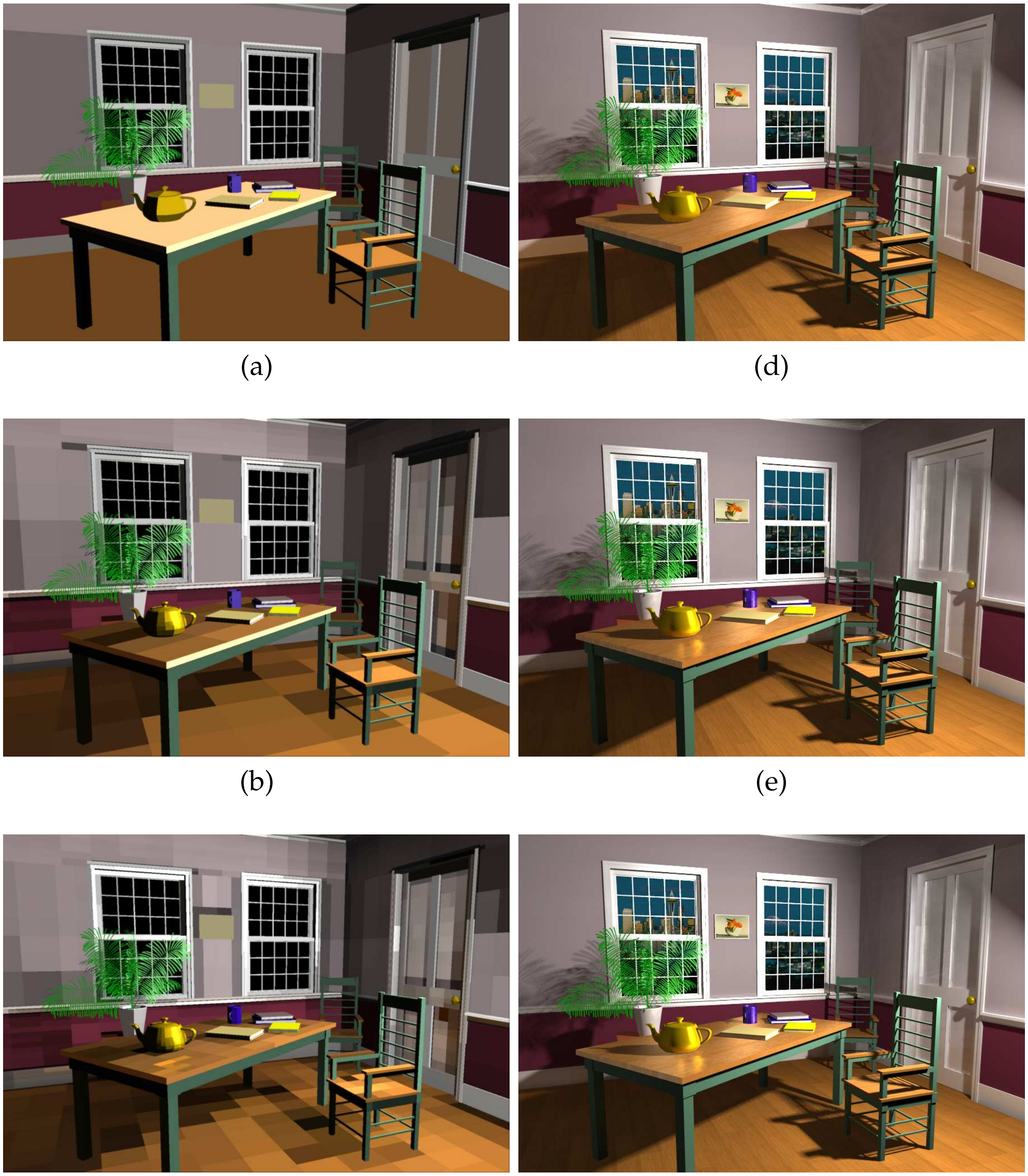
A comparison of sphereflake images. (a) Image computed in 6.2 hours using RADIANCE. (b) Image computed in 4.7 hours using our method (same as figure 8.9(b)).

and promising alternative to Monte Carlo for efficiently simulating glossy global illumination.

8.7 A Complex Interior

To test the effectiveness of our clustering method for complex glossy environments, we experimented with the architectural interior environment shown in figure 8.11. This environment consists of 7,721 patches: 7,653 quadrilaterals and 68 Bézier patches. The teapot, mug, tabletop, door, doorknob, window frames, plant, and pot all have reflectances that are partially glossy and partially diffuse, giving a total of 6,629 glossy surfaces. The remaining surfaces are purely diffuse. The room is illuminated by a single diffuse area light source.

A solution without clustering requires $7721^2 \approx 60,000,000$ potential transport coefficients to be considered in the initial linking stage. An optimistic estimate



(a) (d)

(b) (e)

(c) (f)

Figure 8.11

An interior with both glossy and diffuse surfaces. (a-c) Finite-element solutions computed by our method. (d-f) The same solutions after a final gather (local pass).

says that one would need 4 gigabytes of memory and nearly five days to compute the initial links.

By contrast, a coarse solution was obtained after just 11 minutes with our clustering algorithm (on a DEC 3000/500X "Alpha" workstation with 512MB virtual memory). A cluster hierarchy consisting of 1629 clusters was constructed during the first 4 minutes, and the remaining 7 minutes were spent creating initial links and computing the solution shown in figure 8.11(a). Of 111,983 total links in this solution, only 14,087 links are between two patches. Two more refined solutions are shown in figures 8.11(b) and (c). The statistics for each of these solutions are:

image	coefficients	transports	solution time	final gather time
(a,d)	7,694	111,983	11 min	135 min
(b,e)	63,237	218,354	31 min	271 min
(c,f)	686,048	1,313,296	109 min	290 min

A final gather pass was performed on these solutions; the resulting images are shown in figure 8.11(d-f). The images were rendered at a resolution of 900 × 600 pixels with supersampling using sixteen rays per pixel. The final gather pass is currently the most time consuming stage of the simulation Nevertheless, the final gather is very beneficial. Even for the coarsest finite-element solution the final gather is able to produce an image of high quality: all the direct illumination, including shadows and glossy highlights, and the textures are reproduced correctly. However, since this finite-element solution has no directional variation, the glossy reflection of the teapot-highlight in the table top is not present in figure 8.11(d). As the representation of the radiance leaving the teapot is refined in the second solution (figure 8.11(b)), the reflection appears in the corresponding image after the final gather, even though the table top has not yet been refined enough to reveal the reflection in the solution. Eventually the table top is refined as well, and evidence of the reflection becomes visible even without the final gather (figure 8.11(c)).

Note that for this particular environment, there is no significant difference between the images in figure 8.11(e) and (f). Thus, for most practical purposes, the finite-element solution has effectively converged after only 31 minutes. The progression of solutions demonstrates that our method lends itself well to progressive image generation, with useful images obtained early in the process.

> “O you who have borne even heavier things,
> God will grant an end to these too.”
>
> *Virgil*

Chapter 9

Conclusion

Synthetic images of existing or virtual scenes are useful in many areas including architectural design, interior design, illumination engineering, industrial design, virtual reality, and special effects for movies. To generate realistic images, it is essential to simulate the effects of global illumination. Efficient computation of global illumination is challenging because all light in a scene is interdependent. For scenes with glossy reflections, the problem is even more challenging because of the high dimensionality of the unknown light distributions.

9.1 Contributions

This dissertation describes efficient hierarchical techniques for simulating light transport in complex scenes with glossy and diffuse reflections. The efficiency comes from using a wavelet representation of radiance, a clustering method for approximate representation of radiant intensity from groups of patches, and importance-driven refinement for a view-dependent solution.

- We first showed that the radiance equation and Kajiya's rendering equation are adjoint equations, and that radiance and two-point transport intensity are adjoint functions. This relationship is particularly helpful for describing the properties of importance. The most obvious type of importance for glossy global illumination is an adjoint of radiance. Instead, a type of importance that is algorithmically more convenient than the adjoint of radiance was introduced. This type of importance is a dimensionless distribution emitted from the eye and transported like radiance.
- Wavelet representation of the radiance at each surface reduces the number of transports from $O(n_b^{1.5})$ to linear in the number of basis functions n_b. We used the simplest possible wavelet basis, the Haar basis, since it is orthonormal and has compact support and simple numerical integration formulas. To reduce the number of initial transports, we found it advantageous to use a parameterization of radiance and importance with two spatial parameters specifying a position on a surface, and two directional parameters specifying a direction.
- Clustering groups of surface patches reduces the number of transports from quadratic to linear in the number of surface patches. Light leaving a collection of patches is estimated as radiant intensity emanating from a point. When transported, this radiant intensity turns into incident radiance on other clusters, and this radiance is reflected off the patches in these clusters. A bound on the error associated with the point approximation is given.

Our method is the first finite-element method capable of handling complex nondiffuse environments. The results prove that the finite-element method is a viable alternative to the Monte Carlo method for solving the glossy global illumination problem. With better wavelet bases and more advanced strategies for cluster refinement, the discrete method can be improved even further.

9.2 Future Directions

There are many possible extensions to the present method.

- *Refinement strategy.* One area of interest is the refinement strategy. For example, which end of a link between two clusters should be refined to reduce the error most?
- *Multiresolution representation of radiant intensity.* It could be interesting to develop an adaptive multiresolution representation for directional distributions on clusters. At present, we use fixed angular resolution, and when that resolution is insufficient, the transport from the cluster is substituted by transports from its subclusters. Perhaps spherical wavelets [80] would be beneficial.
- *Better bases.* The basis functions for radiance always have the same level of detail for the directional and spatial parameters. But for a small glossy patch, more refinement is required in the angular domain than in the spatial. In contrast, a large slightly glossy patch usually requires more spatial refinement than angular. So it seems that using basis functions with independent resolution for spatial and angular parameters could potentially reduce the number of basis functions and transports considerably. It would also be interesting to test other wavelet bases. Higher-order wavelets [41, 78] have more vanishing moments than the Haar basis, which will give fewer significant transport coefficients. But higher-order wavelets also have wider support, which might reduce the advantage of using

them.

- *Specular reflection.* Wavelet representation of radiance is not suited for specular (mirror) reflection. Instead, a ray tracing step for ideal specular reflection could be incorporated in the same fashion as in Sillion *et al.* [85].
- *Transmission.* Surfaces that transmit light in addition to reflecting it could be incorporated into our algorithm by using basis functions defined on the entire sphere of directions.
- *Participating media.* In scenes with participating media, radiance has to be computed everywhere in the volume containing the scene — computing the radiance at surfaces is insufficient.
- *More accurate visibility.* Our method is only linear in the number of patches when visibility is computed in linear time. The approximation of visibility between clusters we use is linear, but very coarse. More accurate linear-time algorithms would improve the accuracy of the algorithm without increasing the algorithmic complexity.
- *Incremental final gather.* Is there a way to improve the final gather image incrementally, to get a more progressive algorithm?
- *Plausible global illumination.* For some applications, a physically accurate solution is not necessary: a solution that *looks* plausible is sufficient. To generate such pseudorealistic images, the effects of global illumination are necessary, but the effects do not have to be correct as long as they look correct.
- *Fast walk-throughs.* Fast walk-throughs of glossy scenes by postprocessing global illumination solutions.

- *Incremental refinement.* Updating the solution incrementally for moving viewpoint and moving objects.

- *Parallelism.* The method seems well suited for a parallel implementation to get images faster. For applications such as virtual reality, the goal is real-time computation, but many applications could also benefit from more modest speed-ups. A large fraction of the computation time is spent computing transport coefficients. These computations are completely independent (but all depend on the geometry of the scene). Another large contribution to the computation time is the gathering of radiance along links. This requires many reads of coefficient values.

Bibliography

[1] Edwin A. Abbott. *Flatland: A Romance of Many Dimensions*. Dover, New York, 1884.

[2] Bradley Alpert, Gregory Beylkin, Ronald Coifman, and Vladimir Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM Journal on Scientific Computing*, 14(1):159–184, January 1993.

[3] Bradley K. Alpert. *Sparse Representations of Smooth Linear Operators*. PhD thesis, Yale University, New Haven, Connecticut, 1990.

[4] Bradley K. Alpert. A class of bases in $\mathcal{L}^2$ for the sparse representation of integral operators. *SIAM Journal on Mathematical Analysis*, 24(1):246–262, January 1993.

[5] Andrew W. Appel. An efficient program for many-body simulation. *SIAM Journal on Scientific and Statistical Computing*, 6(1):85–103, January 1985.

[6] James Arvo. Linear operators and integral equations in global illumination. *ACM SIGGRAPH '93 course notes (course 42)*, August 1993.

[7] James Arvo and David Kirk. Fast ray tracing by ray classification. In *Proceedings of ACM SIGGRAPH '87*, pages 55–64, July 1987.

[8] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In *Proceedings of ACM SIGGRAPH '94*, pages 75–84, July 1994.

[9] Larry Aupperle and Pat Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Proceedings of ACM SIGGRAPH '93*, pages 155–162, August 1993.

[10] Larry Aupperle and Pat Hanrahan. Importance and discrete three point transport. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 85–94, June 1993.

[11] Josh Barnes and Piet Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(4):446–449, December 1986.

[12] Petr Bechmann and Andre Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. Pergamon Press, New York, 1963.

[13] Gregory Beylkin, Ronald Coifman, and Vladimir Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, 44:141–183, 1991.

[14] Gregory Beylkin, Ronald R. Coifman, and Vladimir Rokhlin. Wavelets in numerical analysis. In Mary Beth Rushkai et al., editors, *Wavelets and Their Applications*, pages 181–210. Jones and Bartlett, Boston, Massachusetts, 1992.

[15] Phong Bui-Tuong. Illumination for computer-generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.

[16] Per Christensen, Eric Stollnitz, David Salesin, and Tony DeRose. Global illumination of glossy environments using wavelets and importance. Technical Report 94-10-01, Department of Computer Science and Engineering, University of Washington, Seattle, Washington, October 1994.

[17] Per H. Christensen, David H. Salesin, and Tony D. DeRose. A continuous adjoint formulation for radiance transport. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 95–104, June 1993.

[18] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Importance-driven wavelet radiance. Technical Report 94-01-05, Department of Computer Science and Engineering, University of Washington, January 1994.

[19] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Wavelet radiance. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 287–302, June 1994.

[20] Charles K. Chui. *An Introduction to Wavelets*. Academic Press, London, UK, 1992.

[21] Charles K. Chui and Ewald Quak. Wavelets on a bounded interval. In Dietrich Braess and Larry L. Schumaker, editors, *Numerical Methods of Approximation Theory*, volume 9, pages 53–75. Birkhäuser Verlag, Basel, Schwitzerland, 1992.

[22] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Proceedings of ACM SIGGRAPH '88*, pages 75–84, August 1988.

[23] Michael F. Cohen, Donald P. Greenberg, David S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(2):26–35, March 1986.

[24] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, Massachusets, 1993.

[25] Robert L. Cook. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, January 1982.

[26] Robert L. Cook. Distributed ray tracing. In *Proceedings of ACM SIGGRAPH '84*, pages 137–146, July 1984.

[27] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, Pennsylvania, 1992.

[28] L. M. Delves and J. L. Mohamed. *Computational Methods for Integral Equations*. Cambridge University Press, Cambridge, UK, 1985.

[29] Tony D. DeRose, David H. Salesin, and Eric J. Stollnitz. Wavelets for computer graphics: A primer. In *SIGGRAPH '94 Computational Representations of Geometry, Course Notes 23*, pages 113–141, July 1994.

[30] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, Boston, third edition, 1993.

[31] Adam Finkelstein and David H. Salesin. Multiresolution curves. In *Proceedings of ACM SIGGRAPH '94*, pages 261–268, July 1994.

[32] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics, Principles and Practice*. Addison-Wesley Publishing Company, Reading, Massachusetts, second edition, 1990.

[33] Reid Gershbein, Peter Schröder, and Pat Hanrahan. Textures and radiosity: Controlling emission and reflection with texture maps. In *Proceedings of ACM SIGGRAPH '94*, pages 51–58, July 1994.

[34] Andrew S. Glassner, editor. *An Introduction to Ray Tracing*. Academic Press, San Diego, California, 1989.

[35] Andrew S. Glassner. A model for flourescence and phosphorescence. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 57–68, June 1994.

[36] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, San Francisco, California, 1995.

[37] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, second edition, 1989.

[38] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman. Wavelength dependent reflectance functions. In *Proceedings of ACM SIGGRAPH '94*, pages 213–220, July 1994.

[39] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. In *Proceedings of ACM SIGGRAPH '84*, pages 213–222, July 1984.

[40] Steven J. Gortler, Michael F. Cohen, and Phillip Slusallek. Radiosity and relaxation methods: Progressive refinement is Southwell relaxation. *IEEE Computer Graphics and Applications*, 14(6):48–58, November 1994.

[41] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *Proceedings of ACM SIGGRAPH '93*, pages 221–230, August 1993.

[42] Henri Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, C-20(6):623–628, June 1971.

[43] Leslie F. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. PhD thesis, Yale University, New Haven, Connecticut, 1987. (Available as ACM Distinguished Dissertation 1987, MIT Press, Cambridge, Massachusetts).

[44] Eric A. Haines. A proposal for standard graphics environments. *IEEE Computer Graphics and Applications*, 7(11):3–5, November 1987.

[45] Pat Hanrahan and David Salzman. A rapid hierarchical radiosity algorithm for unoccluded environments. In *Proceedings of Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 151–172, June 1990.

[46] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In *Proceedings of ACM SIGGRAPH '91*, pages 197–206, July 1991.

[47] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. In *Proceedings of ACM SIGGRAPH '91*, pages 175–186, July 1991.

[48] Paul S. Heckbert. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, University of California, Berkeley, California, June 1991.

[49] Paul S. Heckbert. Discontinuity meshing for radiosity. In *Proceedings of the Third Eurographics Workshop on Rendering*, pages 203–216, May 1992.

[50] Paul S. Heckbert. Radiosity in Flatland. In *Proceedings of Eurographics '92*, September 1992.

[51] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. In *Proceedings of ACM SIGGRAPH '86*, pages 133–142, August 1986.

[52] S. Jaffard and Ph. Laurençot. Orthonormal wavelets, analysis of operators, and applications to numerical analysis. In Charles K. Chui, editor, *Wavelets: A Tutorial in Theory and Applications*, pages 543–602. Academic Press, London, UK, 1992.

[53] James T. Kajiya. Anisotropic reflection models. In *Proceedings of ACM SIGGRAPH '85*, pages 15–22, July 1985.

[54] James T. Kajiya. The rendering equation. In *Proceedings of ACM SIGGRAPH '86*, pages 143–150, August 1986.

[55] Arjan J. F. Kok. Grouping of patches in progressive radiosity. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 221–231, June 1993.

[56] Serge Lang. *Algebra*. Addison-Wesley, New York, second edition, 1984.

[57] Jeffery Lewins. *Importance, The Adjoint Function: The Physical Basis of Variational and Perturbation Theory in Transport and Diffusion Problems*. Pergamon Press, New York, 1965.

[58] Seymour Lipschutz. *Linear Algebra*. Schaum's outline series. McGraw-Hill, second edition, 1991.

[59] Dani Lischinski, Brian Smits, and Donald Greenberg. Bounds and error estimates for radiosity. In *Proceedings of ACM SIGGRAPH '94*, pages 67–74, July 1994.

[60] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Discontinuity meshing for accurate radiosity. *IEEE Computer Graphics and Applications*, 12(6):25–39, November 1992.

[61] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Proceedings of ACM SIGGRAPH '93*, pages 199–208, August 1993.

[62] Stephane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[63] Yves Meyer. Ondelettes sur l'intervalle. *Rev. Mat. Iberoamericana*, 7:115–143, 1991.

[64] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. *Geometrical Considerations and Nomenclature for Reflectance*. National Technical Information Service. U.S. Department of Commerce, National Bureau of Standards, Washington, D.C., October 1977.

[65] M. Ohta and M. Maekawa. Ray coherence theorem and constant time ray tracing algorithm. In *Proceedings of CG International '87*, pages 303–314, 1987.

[66] Michael Oren and Shree K. Nayar. Generalization of lambert's reflectance model. In *Proceedings of ACM SIGGRAPH '94*, pages 239–746, July 1994.

[67] S. N. Pattanaik and S. P. Mudur. Computation of global illumination by Monte Carlo simulation of the particle model of light. In *Proceedings of the Third Eurographics Workshop on Rendering*, pages 71–83, May 1992.

[68] S. N. Pattanaik and S. P. Mudur. Efficient potential equation solutions for global illumination computation. *Computers and Graphics*, 17(4):387–396, 1993.

[69] Sumanta N. Pattanaik. *Computational Methods for Global Illumination and Visualisation of Complex 3D Environments*. PhD thesis, Birla Institute of Technology and Science, Bombay, India, 1993.

[70] Sumanta N. Pattanaik and Kadi Bouatouch. Haar wavelet: A solution to global illumination with general surface properties. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 273–286, June 1994.

[71] Mark S. Peercy. Linear color representations for full spectral rendering. In *Proceedings of ACM SIGGRAPH '93*, pages 191–198, August 1993.

[72] Mark S. Peercy, Benjamin M. Zhu, and Daniel Baum. Interactive full spectral rendering. In *Symposium on Interactive 3D Graphics*, pages 67–68. ACM, April 1995.

[73] Robert Piessens, Elise de Doncker-Kapenga, Christoph W. Überhuber, and David K. Kahaner. *QUADPACK: A subroutine package for automatic integration*. Springer Verlag, Berlin, 1983.

[74] Pierre Poulin and Alain Fournier. A model for anisotropic reflection. In *Proceedings of ACM SIGGRAPH '90*, pages 273–282, August 1990.

[75] Mark C. Reichert. A two-pass radiosity method driven by lights and viewer position. Master's thesis, Program of Computer Graphics, Cornell University, Ithaca, New York, January 1992.

[76] Holly E. Rushmeier, Charles Patterson, and Aravindan Veerasamy. Geometric simplification for indirect illumination calculations. In *Proceedings of Graphics Interface '93*, pages 227–236, May 1993.

[77] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *Proceedings of ACM SIGGRAPH '87*, pages 293–302, July 1987.

[78] Peter Schröder, Steven J. Gortler, Michael F. Cohen, and Pat Hanrahan. Wavelet projections for radiosity. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 95–104, June 1993.

[79] Peter Schröder and Pat Hanrahan. Wavelet methods for radiance computations. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 303–312, June 1994.

[80] Peter Schröder and Wim Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In *Proceedings of ACM SIGGRAPH '95*, August 1995. To appear.

[81] Min-Zhi Shao, Qun-Sheng Peng, and You-Dong Liang. A new radiosity approach by procedural refinements for realistic image synthesis. In *Proceedings of ACM SIGGRAPH '88*, pages 93–102, August 1988.

[82] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Publishing Corp., Washington D.C., 1981.

[83] François Sillion. Clustering and volume scattering for hierarchical radiosity calculations. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 105–117, June 1994.

[84] François Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. In *Proceedings of ACM SIGGRAPH '89*, pages 335–344, July 1989.

[85] François X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. In *Proceedings of ACM SIGGRAPH '91*, pages 187–196, July 1991.

[86] François X. Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, San Francisco, California, 1994.

[87] Brian Smits. *Efficient Hierarchical Radiosity for Complex Environments*. PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1994.

[88] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In *Proceedings of ACM SIGGRAPH '94*, pages 435–442, July 1994.

[89] Brian E. Smits, James R. Arvo, and David H. Salesin. An importance-driven radiosity algorithm. In *Proceedings of ACM SIGGRAPH '92*, pages 273–282, July 1992.

[90] Harry Soodak. *The Science and Engineering of Nuclear Power*. United Nations, New York, 1948.

[91] E. M. Sparrow and R. D. Cess. *Radiation Heat Transfer*. Hemisphere Publishing Corp., Washington D.C., 1978.

[92] Ephraim M. Sparrow. On the calculation of radiant interchange between surfaces. In Warren E. Ibele, editor, *Modern Developments in Heat Transfer*, pages 181–212. Academic Press, New York, 1963.

[93] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer. *IEEE Computer Graphics and Applications*, 15(3,4), May–July 1995. (Tutorial in two parts).

[94] Kenneth E. Torrance and Ephraim M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, September 1967.

[95] Roy Troutman and Nelson L. Max. Radiosity algorithms using higher order finite elements. In *Proceedings of ACM SIGGRAPH '93*, pages 209–212, August 1993.

[96] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 147–162, June 1994.

[97] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. In *Proceedings of ACM SIGGRAPH '87*, pages 311–320, July 1987.

[98] Gregory J. Ward. Measuring and modeling anisotropic reflection. In *Proceedings of ACM SIGGRAPH '92*, pages 265–273, July 1992.

[99] Gregory J. Ward. The RADIANCE lighting simulation and rendering system. In *Proceedings of ACM SIGGRAPH '94*, pages 459–472, July 1994.

[100] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In *Proceedings of ACM SIGGRAPH '88*, pages 85–92, August 1988.

[101] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.

[102] Lawrence B. Wolff. Diffuse reflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 472–478, June 1992.

[103] Lawrence B. Wolff and David J. Kurlander. Ray tracing with polarization parameters. *IEEE Computer Graphics and Applications*, 10(6):44–55, November 1990.

[104] Hau Xu, Qun-Sheng Peng, and You-Dong Liang. Accelerated radiosity method for complex environments. In *Proceedings of Eurographics '89*, pages 51–61, September 1989.

[105] Harold R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In *Proceedings of ACM SIGGRAPH '93*, pages 213–220, August 1993.

Vita

Per Henrik Christensen was born on July 21st, 1966 in Charlottenlund, north of Copenhagen, Denmark. Besides computer graphics and computer vision, his interests include drawing, paragliding, hang-gliding, diving, classic cars, Lego bricks, and desserts.

Education

- 1990 – 1995: University of Washington, Seattle, Washington.
  Ph.D. in Computer Science. The research was done with Professors David Salesin and Tony DeRose as advisors. The topic was efficient simulation of light transport in glossy environments using adaptive, hierarchical finite-element methods.

- 1985 – 1990: Technical University of Denmark (DTU), Lyngby, Denmark.
  Master of Science in Electrical Engineering. The masters project was done at the Department of Computer Science with Professor Jørgen Staunstrup as advisor. The topic was automated synthesis of delay-insensitive circuits.

Work

- Summer 1993: Industrial Light and Magic (Lucasfilm), San Rafael, California.
  Programmer, wrote C++ procedures for image processing and improved a program prototype for automatic computation of camera position from an image.

- 1993 – 1995: University of Washington, Seattle, Washington.
  Research assistant in the Department of Computer Science.

- Winter 1992: University of Washington, Seattle, Washington.
  Teaching assistant for a senior-level computer graphics course in the Department of Computer Science. Planned exercises, helped students, and graded.

- Summer 1989: Technical University of Denmark, Lyngby, Denmark
  Research assistant in the Department of Computer Science. Worked with design and layout of a "Viterbi decoder", a VLSI chip for reliable (error-correcting) satellite communication.

Publications

Computer graphics

- Per Christensen, Dani Lischinski, Eric Stollnitz, and David Salesin.
  "Clustering for Glossy Global Illumination".
  Submitted to *ACM Transactions on Graphics*.
  (Also technical report UW-CSE-95-01-07.)

- Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose.
  "Global illumination of glossy environments using wavelets and importance".
  Submitted to *ACM Transactions on Graphics*.
  (Also technical report UW-CSE-94-10-01.)

- Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose.
  "Wavelet radiance".
  In *Proceedings of the Fifth Eurographics Workshop on Rendering*,
  pages 287–302, June 1994.

- Per H. Christensen, David H. Salesin, and Tony D. DeRose.
  "A continuous adjoint formulation for radiance transport".
  In *Proceedings of the Fourth Eurographics Workshop on Rendering*,
  pages 95–104, June 1993.

Computer vision

- Per H. Christensen and Linda G. Shapiro.
  "Three-dimensional shape from color photometric stereo".
  International Journal of Computer Vision,
  pages 213-227, volume 13, number 2, October 1994.

- Per H. Christensen and Linda G. Shapiro.
  "Determining the shape of multi-colored dichromatic surfaces using color photometric stereo".
  In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*,
  pages 767–768. June 1993.

Chip design

- Per H. Christensen, Henrik Hulgaard and Jørgen Staunstrup.
  "Synthesis of delay insensitive circuits from verified programs".
  In *Research Directions in High-level Parallel Programming Languages*,
  editors J.-P. Banatre and D. Le Metayer.
  Lecture Notes in Computer Science, volume 574, pages 326–337.
  Springer Verlag, 1991.