HIERARCHICAL ALGORITHMS FOR ILLUMINATION

Larry Aupperle

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
COMPUTER SCIENCE

November 1993

© Copyright by Larry Aupperle 1993
All Rights Reserved

Abstract

This dissertation is a discussion and development of hierarchical algorithms for illumination. These algorithms operate through recursive, adaptive refinement of the environment into hierarchical meshes — rather than computing light transport only between elements at the finest level of refinement, the algorithms allow computation of transport between higher level subpatches, as controlled by user specified error bounds. As discussed in this dissertation, employment of hierarchical methods yields significant savings in computation.

The initial work in hierarchical methods was that of Hanrahan and Salzman for the computation of radiosity over unoccluded environments. In this dissertation, we discuss extension of the algorithm to occluded environments, incorporating visibility heuristics and acceleration via radiosity weighting. Given an environment consisting of k polygonal patches and n elements at the finest level of refinement, the algorithm requires at most $O(n + k^2)$ transport interactions; traditional methods require $O(n^2)$.

Application of hierarchical transport to nondiffuse reflection is developed through the derivation of a radiance formulation for discrete three point transport, incorporating a new measure and description of reflectance: *area reflectance*. This formulation and associated reflectance allow an estimate of error in the computation of radiance across triples of surface elements, and lead directly to a hierarchical refinement algorithm for global illumination.

We have implemented and analyzed this algorithm over surfaces exhibiting glossy and diffuse reflection. Theoretical growth in transport computation is shown to be $O(n + k^3)$ — this growth is exhibited in experimental trials. Naive application of three point transport would require computation over $O(n^3)$ element triples.

Global illumination within nondiffuse environments is ideally suited for computation under importance and radiance driven refinement: a transport interaction is of significance

only if it lies within paths of directional reflection of both radiance originating at a light source, and importance originating at the eye. We have thus derived the adjoint to the radiance transport formulation, and present preliminary results of application of this adjoint in the form of an importance driven version of our implementation. These results show significant reduction in computation, and indicate that importance and radiance driven hierarchical techniques possess great potential for efficient evaluation of global illumination over general reflection.

Acknowledgements

I would like to extend my sincere thanks to my advisor, Dr. Pat Hanrahan, with whom it has been an honor to work. Thanks, also, to the readers of this dissertation, Dr. Michael Cohen, and Dr. David Dobkin, for their helpful comments and suggestions, and to the members of the fpo committee, Dr. Elisha Sacks, and Dr. Robert Sedgewick.

I would like to thank Dr. Przemyslaw Prusinkiewicz for generously providing me with access to the graphics research facilities at the University of Calgary during the final stages of this work.

The research presented in this dissertation was partially supported by equipment grants from Apple and Silicon Graphics Computer Systems, and a research grant from the National Science Foundation (CCR 9207966). All support is gratefully acknowledged.

Thanks to my office-mates at Princeton, especially those who had to put up with me longest: Jenny, John, and Krishnan, for all of their help over the years. I would also like to thank Lefteris Koutsofios for his friendly and frequent assistance while I was first getting acquainted with the graphics lab.

I would like to thank my parents for being there for me, even when I was mostly absent, particularly toward the end of this work.

Finally, to Debbie Fowler, for her love, encouragement, and understanding, my thanks, and all of my love.

Contents

Chapter 1

Introduction

1.1 A Brief Overview of Image Synthesis and Illumination

Computer generated images have become increasingly realistic and complex over the last two decades. They have also become much more numerous — in the early seventies such images could be found only in computer graphics laboratories, and in technical journals; they are now nearly commonplace on television, in popular magazines, and in motion pictures, and show great promise as tools for design and visualization in science, engineering, architecture, communications and art. This combination of progress and growth has been realized both through advances in hardware technology producing more powerful and less expensive processors, displays, and memory, and through the development of algorithms that allow the modeling of complex and interesting environments, the computation of illumination over these environments, and the rendering of the resulting images for display.

Image synthesis allows accurate, controlled generation of images containing environments, illumination, and effects that may be difficult or impossible to create or manipulate by other means. It is this capability that motivates research in the field, and the growing employment of computer generated images in other venues. As the field has developed, the demand and ambition for progressively more complex and realistic images and effects, along with the development of increasingly sophisticated environments, have encouraged more and more accurate simulation of the behavior of light, both in models of reflection, and in algorithms for illumination.

The earliest simulation of light and reflection for image synthesis modeled only simple, direct illumination — commonly referred to in computer graphics literature as *local illumination*. Local illumination takes into account only light leaving an emitter, arriving directly at some surface in the given environment, and reflected or transmitted immediately to the eye.

Initially under local illumination only simple, diffuse reflection was simulated — surfaces thus appeared as though they were covered in extremely nonglossy matte paint. Light sources were located at infinity, inducing parallel, directional "rays", or, if local, were modeled as simple point sources.

As image synthesis has developed, local illumination effects have become increasingly sophisticated. Models of reflection allowing the simulation of a greater variety of materials, including metals and plastics, have been developed. At first such models were largely empirical; later reflection models have been based on geometric or wave optics, and allow more realistic and accurate simulation. Reflection models permitting controlled specular, glossy reflection have been widely implemented and applied, allowing surfaces to exhibit shininess and highlights. Anisotropic reflection has been modeled, permitting simulation of brushed or polished surfaces such as varnished wood, or brushed steel. Methods employing texture, "bump", and displacement maps have been developed to model the changes in composition, orientation, and geometry necessary for complex and realistic materials and surfaces. Techniques have been developed to simulate area and directional light sources. The development of shading languages has allowed the simple, structured integration and amplification of all of the above techniques, forming a powerful tool for the expression of illumination.

In many environments, as a first order approximation, local illumination can capture much of the behavior of light with respect to the eye. For example, local illumination will provide a good indication of the shape and highlights over a single bicubic patch, or quadric surface, or more complex object in isolation. However, as local illumination ignores secondary reflection and transmission and the resulting indirect illumination, it tends to yield overly stark, "computery" images, with relatively harsh lighting. These effects may be somewhat alleviated through the addition of a constant "ambient" term to computed local illumination; however, as modeled objects, surfaces, and reflection functions become

Figure 1: Radiosity Interior

more realistic, the lack of global effects in illumination becomes increasingly pronounced. Given a highly realistic model of an oak table within a detailed architectural interior, it seems odd to see realistic glossy highlights induced directly by light sources on the table surface, but no reflection taking into account the rest of the environment. In general, much of the nuance of reality may be due to the influence of such secondary illumination effects.

Three methods have been widely studied and applied in efforts to overcome the above problems: radiosity, ray tracing, and reflection/environment maps. All three attempt to capture, at least partially, more global effects in illumination.

Basic radiosity methods operate by first subdividing the given environment into a mesh of relatively small elements. Light transport is then estimated from one element to another, iterating until all significant rereflection has been taken into account, and computed illumination has converged. Radiosity provides a complete solution for global illumination — the illumination at each element is computed under all relevant interreflection within the environment. Radiosity solutions thus exhibit subtle illumination effects present within actual environments, including indirect lighting, color bleeding from surfaces to others in close proximity, and relatively dim illumination in constricted areas such as interior corners (Figure 1). The basic assumption underlying the radiosity algorithm, however, is that all

Figure 2: Ray Traced Scene

surfaces within the environment exhibit only diffuse reflection. Radiosity thus provides only a highly restricted solution to global illumination, and will not compute light transport over environments containing varying surfaces.

Under basic ray tracing, rays are traced from the eye point through "pixels" on a virtual image plane and out into the environment. When a ray intersects a surface, local illumination is computed at the intersection point, and additional light gathered by recursively tracing rays in the directions of perfect specular reflection and transmission. Ray tracing is thus equivalent to the computation of local illumination over the given environment, combined with specular paths leading directly to the eye (Figure 2). Note that all illumination due to specular reflection is not simulated — although in Figure 2 we see the reflection of the checkerboard in the large sphere, there is no indication of light reflected from the sphere onto the checkerboard. Note also that only perfect specular reflection is modeled in global terms — although local illumination provides fuzzy or glossy reflection of the three light sources in each sphere, the reflection of the checkerboard is that of a perfect mirror. Ray tracing thus provides both a highly restricted, and only partial, solution for global illumination.

There has been much work combining radiosity and ray tracing or extending radiosity to account for specular reflection, or ray tracing to account for diffuse reflection and complete specular effects. This research has produced very realistic and impressive images, but does not provide a full solution for global illumination. One problem is that it is difficult to efficiently capture and account for the reflection of light from a specular surface onto a diffuse — the representation and particularly the controlled reconstruction of this illumination is not trivial. Furthermore, even a solution that correctly simulates all possible light transport under radiosity and ray tracing is still restricted to perfect diffuse and specular reflection, fully modeling only those environments consisting of matte paint and mirrors.

The final method for incorporation of global effects that has received widespread study and application is the employment of reflection/environment maps. A reflection map, called also an environment map, is created by computing illumination as seen in all directions by some central point in the environment (it is possible to employ multiple points and environment maps in more sophisticated applications of the method). This illumination describes the light incident at the chosen point from the rest of the environment, and may be stored in a cubical map "surrounding" the point, or as a two dimensional map indexed by polar coordinates. The computed illumination is usually simple local illumination over the environment, with shadowing. The resulting environment map is employed to approximate incoming secondary illumination at *all* points in the environment. When computing illumination for the eye, local illumination may be combined with secondary illumination from the environment map, as reflected at the local point (simple specular reflection is usually employed).

Reflection mapping can thus give an indication of global effects — specifically those induced by immediate secondary reflection, which are often dominant. However, the method is restricted to secondary reflection — capture of tertiary reflection would require recomputation of the map, and the method has usually been restricted in practice to perfect specular reflection. In addition, the use of a reflection map is inherently inaccurate at any point other that that originally chosen. The invariance of directional indexing into the map simulates light incident from an infinite sphere, and the employment of reflection mapping over a finite environment will induce distortion in computed illumination. For the computation of global illumination, environment mapping is thus restricted, partial, and potentially inaccurate.

What we desire is an algorithm for global illumination that is both unrestricted — allowing true reflection and transmission of *all* received light, not just that computed locally, by arbitrary reflection/transmission functions, and complete — efficiently capturing all significant illumination, interreflection, and transmission within the environment. To develop an algorithm for such illumination we require a means of expressing it — such a means was provided in a general formulation for true global illumination developed by James Kajiya in 1986 [K86]:

$$I(x, x') = g(x, x')(e(x, x') + \int_X r(x, x', x'')I(x', x'')dx'')$$

This equation is known as the *rendering equation.* $I(x, x')$ is a measure of the light transported from point x' to point x within the environment. $e(x, x')$ expresses the light emitted at x' in the direction of x. Integration is over X, all surfaces within the environment, and $r(x, x', x'')$ is a measure of the light reflected or transmitted at x' from x'' and toward x. Finally, $g(x, x')$ is 0 if visibility from x' to x is occluded, otherwise it expresses geometric attenuation of light over distance.

The rendering equation states that the light that arrives at point x from point x' is equal to the light emitted at x' in the direction of x, plus, for all points x'' in the environment, the light from x'' that arrives at x' and is then reflected/transmitted to x, as modulated by occlusion and geometric attenuation.

The rendering equation is a relatively complex integral equation — such equations do not admit tractable analytic solution, and are generally solved numerically by Monte Carlo or finite element methods.

Within computer graphics, Monte Carlo methods have most commonly been applied to the rendering equation, in the form of distributed or stochastic ray and path tracing [CRMT91, C86, K86, S90, WRC88]. These methods compute illumination in a fashion similar to standard ray tracing, save that when a ray intersects a surface, the resulting recursively traced rays are not necessarily oriented in the directions of perfect specular reflection and transmission. Rather, the direction of secondary rays is distributed stochastically under the reflection function at the surface, each intersection yielding one or more such distributed rays, as determined by the algorithm. Monte Carlo algorithms thus capture the scattering inherent in all but perfect reflection. Furthermore, the techniques are relatively easy to implement and employ for complex geometries and reflection functions. Unfortunately, they are notoriously inefficient — requiring large numbers of distributed rays for convergence to an accurate and acceptably artifact-free solution.

Finite element methods have received less application to the rendering equation than Monte Carlo techniques. They have, however, been very successfully applied under the assumption that all reflection is diffuse, in the form of the radiosity algorithm. In general, finite element methods approximate the desired solution function, in our case illumination $I(x, x')$, by a linear combination of chosen basis functions of finite support. The methods operate by subdividing the integration domain(s) into finite subelements, where one or more such elements form the domain of support for the bases. A variety of techniques may then be employed to discretize the integral equation over the elements, deriving a linear system of equations that may be solved to yield the desired approximating function.

Under radiosity, discretization of the environment into a mesh corresponds to the subdivision of domains of integration into finite elements. Radiosity methods assume that incoming, and thus outgoing, illumination (radiant flux density) is constant across each element; this is equivalent to the employment of simple piecewise constant/box basis functions. Discretization of the integral equation corresponds under radiosity to discretization of light transport over all pairs of elements in the environment. This discretization yields a system of linear equations, each equation expressing transport over all element pairs containing a given element as the receiver. For n elements, this linear system may be written as an $n \times n$ matrix, and solved to approximate the illumination function. The solution methods commonly employed under the radiosity algorithm — successive iterations simulating either light transport from one element to all others, or from all to one, correspond to variations on Jacobi methods.

Finite element methods have only begun to be applied to less restricted formulations of the rendering equation. This work has focussed on extension of the radiosity method to more general reflection functions — specifically those simulating rough specular reflection. Given nondiffuse reflection, the primary task in extending the radiosity method is the representation of the directionally distributed illumination reflected from each element, and the computation of discrete light transport over this representation. There have been two major studies extending the finite element method to more general reflection: Immel et al. [ICG86] and Sillion et al. [SAWG91].

Immel et al. [ICG86] discretize light transport over a lattice of cubical environment maps. Each element in the environment possesses its own map, constructed about a central point on the element — all maps, however, are oriented with the same global axes. The map at a given element subdivides the set of incoming (and outgoing) directions at that element into a finite number of discrete subunits. As all maps within the environment are aligned, they induce an identical, global, directional subdivision.

The maps are employed both to determine from which other elements a given element may receive illumination, and to store the resulting distributions of reflected light over the directional subdivision. Each "pixel" within a map induces a subunit within this directional subdivision, the environment map providing the identity of the element within the environment that the map's element "sees" through the pixel, and from which it thus receives illumination in the induced discrete direction. The magnitude of this illumination may be found by querying the "seen" element's map for outgoing light in the given direction. Incoming illumination may then be passed through the receiving element's reflection function, and the resulting scattered light distributed and stored over the directional subdivision as represented by the receiving element's map.

The elements and associated environment maps thus allow discrete expression of all interelement light transport under general reflection. The resulting linear system may then be solved to approximate global illumination. Immel et al. compute this solution by employing a gathering iteration (successive transport of light from all elements to one) analogous to the Gauss-Seidel method for solution over linear systems.

Unfortunately, the resources necessary under [ICG86] to store and compute over a sufficiently fine lattice of environment maps and elements are extreme. For an environment consisting of two boxes within a simple enclosure, computation time required was on the order of days. The method thus appears prohibitively expensive, though it captures completely all illumination within the environment.

The method of Sillion et al. [SAWG91] does not employ discretized directional subdivision. Rather, the reflected light at each element is approximated by a finite sum of spherical harmonics. Spherical harmonics form an infinite basis over all functions on the unit sphere — the use of a truncated series of harmonics allows the representation of relatively smooth distributions of reflected light, such as induced by gently directional reflection functions.

The employment of spherical harmonics rather than an arbitrarily and universally imposed directional subdivision yields several advantages:

1. The cubical environment maps employed under [ICG86] are very susceptible to aliasing, whereas the inherently smooth, nondiscrete, functional surfaces of Sillion et al. are not subject to the problem.

2. Spherical harmonics are easily manipulated — rotated, summed, and scaled, and may be efficiently evaluated [SAWG91].

3. Finally, for the directional diffuse reflection functions over which Sillion et al. have implemented their method, representation of light distributions by spherical harmonics is relatively compact — acceptably accurate approximations requiring less than 1 K bytes at each element vertex.

Given the element/spherical harmonic representation, Sillion et al. employ a *shooting* algorithm to solve the resulting system. With each iteration, light is transported (shot) from a chosen element to all others in the environment. The intensity of each interelement shoot is determined by the chosen element's associated spherical harmonic/light distribution function. Note that shooting over all elements is equivalent to gathering over all elements — the advantage of shooting is that the "brightest" elements with the greatest potential for light contribution to the environment may be selected for shooting first, allowing relatively rapid generation of intermediate images in which the most significant illumination and reflection have been accurately simulated. (This progressive scheme for illumination was developed originally for radiosity [CCWG88].) The method, in combination with a ray tracing pass, has generated realistic and attractive images, and requires moderate computation time — for environments consisting of around 30,000 elements, the reported average time for a shooting iteration (ie. one element to all) is five minutes [SAWG91].

1.2 Hierarchical Algorithms for Illumination

1.2.1 Global Illumination

In this dissertation we present a new application of the finite element method to global illumination under nondiffuse reflection. The most important difference between our algorithm and previous methods is that transport under our technique is discretized and computed in an error driven, fully hierarchical fashion. That is, we employ a multilevel hierarchy of elements in which each transport computation is calculated at a level sufficient to satisfy user specified error bounds. Thus, transport calculations requiring highly refined elements for accuracy are carried out relatively deep within the hierarchy, whereas those sufficiently accurate over more coarse elements are carried out at higher levels, allowing concentration of computation where it is required, without incurring excess computation over less demanding transport.

We shall examine this algorithm and methodology in considerable detail in the following chapters; the sketch provided here is intended as a brief introduction and discussion of the work:

Application of the finite element method requires discretization of transport across chosen environmental elements. In order to achieve this discretization we have developed a radiance formulation for three point transport, and a new measure and description of reflectance: *area reflectance*. Consider an environment of k polygons discretized into n sufficiently small elements — let A_j and A_k be two such elements. Then, as shown in Chapter 4, the following radiance formulation for discrete transport may be obtained:

$$L_{jk} = E_{jk} + \sum_{i=1}^{n} R_{kji} L_{ij} \tag{1}$$

This equation states that the radiance (angular radiant flux density, radiance corresponds to the intuitive notion of light intensity) L_{jk} at area A_j in the direction of area A_k is equal to the radiance emitted at A_j in the direction of A_k, plus, for every area A_i, the radiance at A_i in the direction of A_j, multiplied by the area reflectance, R_{kji}. Area reflectance, R_{kji}, is the fraction of the radiant flux transported from A_k incident to A_j that is reflected in the direction of area A_i. As discussed in Chapter 4, the accuracy of estimators for R_{kji} is improved as the sizes of A_i, A_j, and A_k decrease relative to their distances apart.

Equation (1) is based on, and motivated by, three point transport as expressed in Kajiya's original rendering equation. There are several important differences between our formulation and the rendering equation however:

1. Equation (1) expresses transport over discrete areas, rather than at simple points, as is necessary for the application of the finite element method.

2. Equation (1) applies and expresses *radiance*, a highly useful, accepted and fundamentally important radiometric measure (see Chapter 2), whereas the rendering equation employs relatively novel units (the transport intensity employed under the rendering equation is flux$/m^4$, rather than angular flux density).

3. The appearance of area reflectance within the radiance formulation has a natural and satisfying, though perhaps surprising, physical significance not evident in the rendering equation: transport over reflection depends only on the reflectance between the transport patches, "reversed" through the use of a reciprocity relation to eliminate area dependence (Chapter 4).

Readers familiar with radiosity will note the similarity of equation (1) to the radiosity formulation.

This similarity of the radiance formulation to the well understood radiosity equation is the major reason for the employment of discretization entirely over surface elements. The formulation and resulting method do not explicitly represent or discretize over direction, unlike the directional subdivision of [ICG86] or directional representation under [SAWG91]. Rather, we apply the finite element approach directly to the surface over which transport occurs. In Chapter 4 we will briefly discuss the possible application of hierarchical techniques over directional subdivision or spherical harmonics.

The radiance formulation, combined with its improvement under subdivision of areas A_i, A_j, and A_k, leads directly to a hierarchical refinement algorithm for discrete three point transport. The algorithm operates over patch to patch transport *interactions*. This was suggested by equation (1), which describes the radiance across the interaction from area A_j to area A_k in terms of interactions from areas A_i to area A_j. The algorithm operates by refining all pairs of interactions that share the same central patch (corresponding to A_j

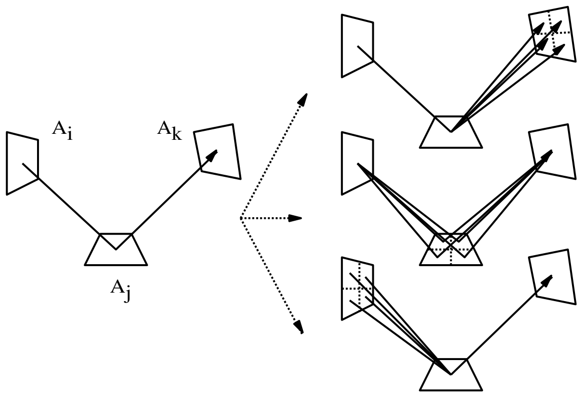


Figure 3: Refinement and Subdivision

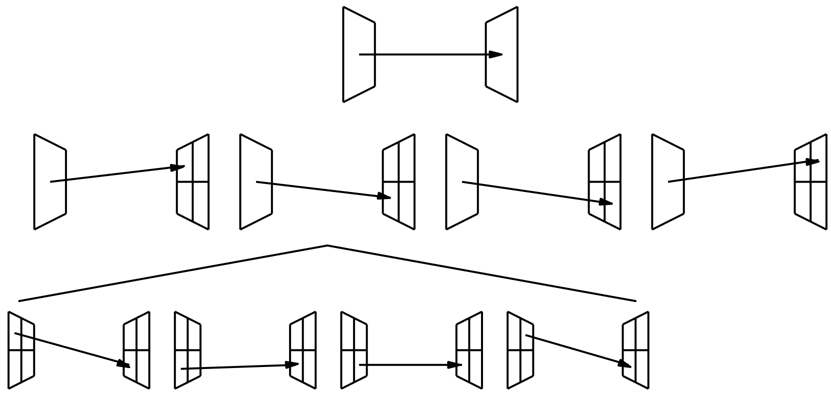


Figure 4: Interaction Hierarchy

in the formulation) to ensure accuracy of computed transport. For a given interaction pair, and associated patch triple, an estimate of error in computed R_{kji} across the interactions is determined: if estimated error is greater than user-specified bounds, then one or both interactions are subdivided, and refinement recursively applied to their descendants (Figure 3); otherwise, the interactions are *linked* to indicate that transport may be computed across the pair within desired accuracy.

Interaction subdivision is induced by the subdivision of component patches; in implementation we have employed quadtree subdivision over patches, thus inducing a corresponding quadtree hierarchy over interactions (Figure 4). Only one patch is subdivided with respect to a given interaction, ensuring a well-formed hierarchy.

Figure 5: Illumination and Meshing

The error driven adaptive refinement procedure described above creates a set of interlinked interaction hierarchies, each link joining pairs of interactions over which transport may be computed. The actual computation of transport is carried out via a gathering iteration, each interaction gathering radiance from interactions to which it has been linked, as governed by estimated area reflectance under equation (1). These gathered radiances are then distributed within each receiving hierarchy (as described in Chapter 4), and subsequent iterations computed until satisfactory convergence is achieved.

We have implemented and analyzed this algorithm over surfaces exhibiting glossy reflection, under a highly simplified Torrance-Sparrow model (Chapters 2 and 4). Figure 5 shows illumination and meshing under this model over surfaces of differing glossiness — within each image, the reflecting surface is perpendicular to the diamond-shaped light source, and we see the resulting reflection in the direction of the eye. Note the conformation of meshing to the highlight over each surface. We have shown that theoretic growth in light transport triples (linked interaction pairs) under the model is $O(n + k^3)$ for sufficient refinement, where n is the number of elements at the finest level of refinement over an environment of k polygonal patches – this growth is exhibited in experimental trials. Naive application of three point transport to global illumination would require $O(n^3)$ element

Figure 6: Cube and Slabs

triples.

The hierarchical methods described above may be greatly accelerated by weighting estimated transport error by the radiance and importance across each linked interaction pair. Importance may be viewed as a quantity emitted by the eye and interreflected within the environment much as radiance emitted by light sources. Global illumination within nondiffuse environments is ideally suited for computation under importance and radiance driven refinement: a transport interaction is significant to the view from the eye only if it lies within paths of directional reflection of both radiance originating at a light source, and importance originating at the eye.

We have thus derived the adjoint to our original radiance formulation, yielding an equation describing the transport of importance under discrete three point computation. In Chapter 5, we describe the preliminary results of application of this adjoint, in the form of an importance and radiance driven version of our glossy implementation. Figure 6 shows an image created under the implementation: there are potentially 222 billion discrete transport triples within the shown environment at full refinement (15,114 elements); the application employs 71,000, a reduction to 3 hundred-thousandths of 1 percent. The solution was computed after seven complete iterations, each iteration corresponding to shoots or gathers at *all* interactions, and total time just over three minutes.

The general difficulty in applying finite element methods to global illumination is that the transport interaction matrix implicitly or explicitly employed under such methods grows prohibitively large over interesting environments. The algorithm presented in this dissertation shows that illumination may be computed over a relatively compact, hierarchical representation of this matrix, constructed through error driven adaptive refinement. Results of application of this algorithm under glossy reflection show a significant reduction in transport computation, and indicate that an importance and radiance driven hierarchical technique possesses great potential for the development of efficient global illumination algorithms for general reflection.

1.2.2 Radiosity

The global illumination methods for glossy reflection described in the previous section were inspired by, and are an extension of, hierarchical techniques for the computation of radiosity. Recall that radiosity methods compute illumination and transport over environments containing only surfaces exhibiting diffuse reflection. The radiosity equation expresses this transport:

$$B_i = E_i + \rho_i \sum_j B_j F_{ij} \tag{2}$$

where B_i is the radiosity (outgoing radiant flux density – see Chapter 2) at element A_i — the discretization underlying this equation thus assumes that radiosity across each element is (at least nearly) constant. E_i is the radiosity emitted by element A_i itself, and ρ_i expresses A_i's reflectivity. The summation is over all elements A_j within the environment, accounting for the contribution after transport of all radiosities B_j to B_i. These contributions are expressed by the *form factor* F_{ij} — for a given A_i, A_j, F_{ij} is the percentage of the flux leaving A_i that arrives at A_j. The form factor has thus been "reversed" similarly to area reflectance under equation (2). Assuming visibility between A_i and A_j is unoccluded, the form factor depends only on the geometry, orientation and interdistance of the elements — as discussed in Chapter 3, the accuracy of estimators for the form factor is improved as the sizes of A_i and A_j decrease relative to their distance apart.

In 1990, Hanrahan and Salzman [HS90] showed that this dependence of computed transport accuracy on size and distance could be employed in the development of rapid hierarchical algorithms for illumination under radiosity. Under their method, input polygonal patches are adaptively refined within quadtree structures — the radiosity at each surface is then represented in a hierarchical fashion: each subpatch within the quadtree has an associated radiosity representing the flux density over the subpatch at that level of refinement.

The adaptive refinement underlying these hierarchies is carried out over all pairs of input patches: the error in diffuse transport over each given pair is computed (as discussed in Chapter 3), if this error exceeds user specified bounds, then one of the patches is subdivided and refinement recursively applied, otherwise the patches are *linked*. Actual transport may then be computed by a gathering iteration in which each patch gathers radiosity, as governed by equation (2), from those patches with which it has been linked.

The relationship between this algorithm and its extension to three point transport should be clear: under three point transport patch to patch interactions play the role analogous to patches under hierarchical radiosity.

As discussed in [HS90] (and also in Chapter 3), for sufficient refinement the algorithm will generate $O(n + k^2)$ interpatch links, each patch interacting only with a constant number of other patches within a limited local neighborhood. Recall that n is the number of elements within the environment at the finest level of refinement, and k the number of input polygonal patches. Previous radiosity algorithms required $O(n^2)$ transport interactions.

Figure 7 shows the hierarchical meshing and transport links generated by adaptive refinement over a simple enclosure. The links shown are those indicating transport involving the rear wall of the enclosure — note that relatively larger subpatches interact at a distance, whereas those more closely linked have been further refined.

The algorithm presented in [HS90] may be applied only over unoccluded environments — that is, the algorithm does not consider the visibility between patches when computing transport. The work of Hanrahan et al., presented in [HSA91], and in Chapter 3 of this dissertation, describes the extension of the algorithm to computation over visibility, and the application, proposed in [HS90], of radiosity driven refinement.

Visibility is estimated under [HSA91] through a relatively straightforward application of ray casting — however, the employed visibility technique introduces and incorporates two optimizing heuristics particularly suited for hierarchical meshing over illumination.

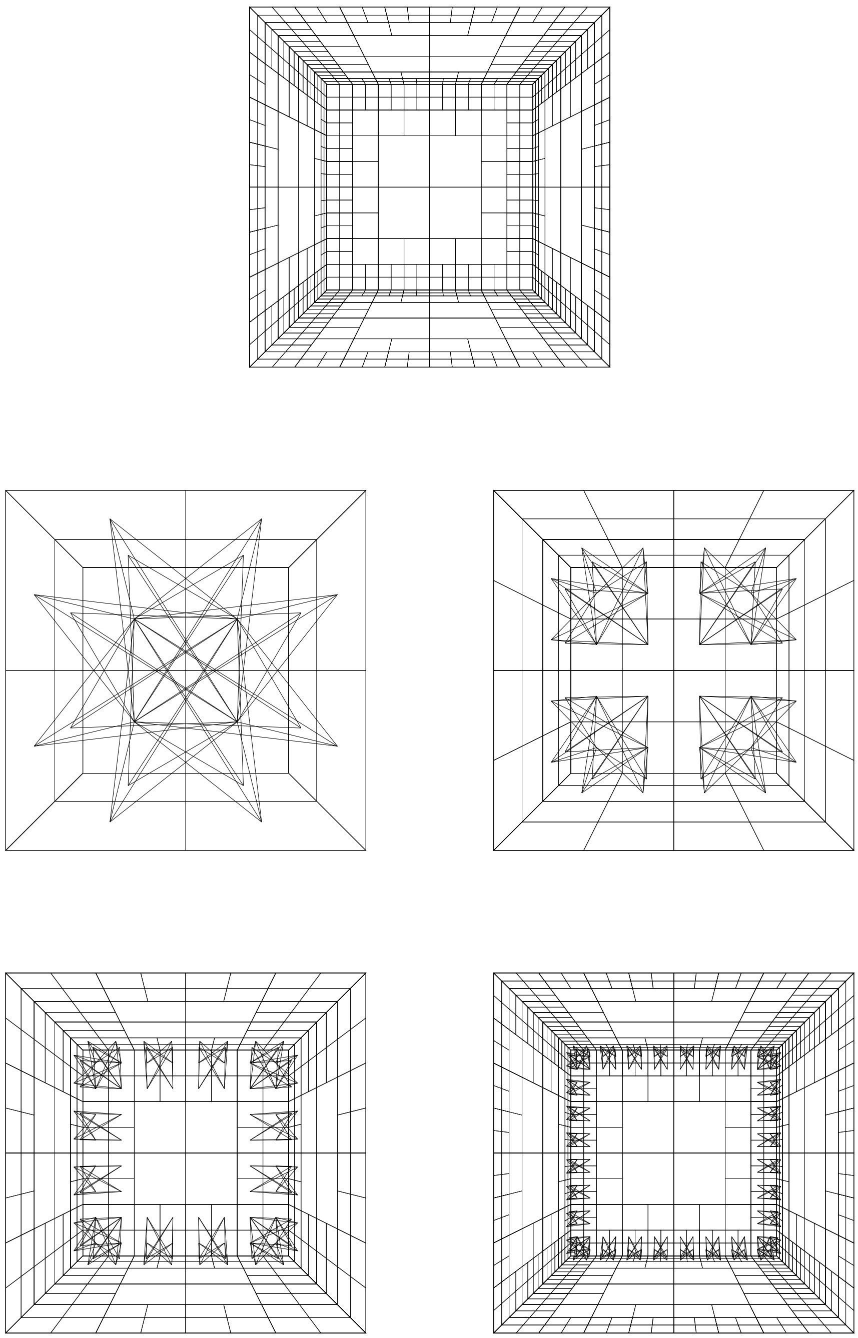


Figure 7: Hierarchical Meshing and Transport

Given two patches, visibility between the patches is estimated by first subdividing each patch into a 2D grid (typically 4×4), and assigning the cells in each grid numbers from two differing, grid-sized magic squares; rays are then cast between jittered points within corresponding cells, the fraction of rays not intersecting an intervening surface within the environment indicating the transport visibility between the patches.

Note that if the visibility between two patches is completely unoccluded, then the visibility between all pairs of their descendent subpatches must be similarly unoccluded. This *visibility inheritance* applies equally to totally occluded visibility, although given such visibility direct transport is not possible, and thus no subdivision or subpatches are required. Visibility inheritance may be exploited in the computation of visibility over refinement: consider refinement over two patches — if visibility between the patches is completely unoccluded, then no further visibility calculation is required over further refinement, whereas if visibility is occluded refinement may be immediately terminated, as no transport is possible. It is only in the case of partial visibility that both additional refinement and additional visibility computation may be required.

We have found that in practice over our trial environments (simple architectural interiors) most patch interactions exhibit either complete visibility or occlusion, with fringes of "penumbral" partial interactions comprising a smaller fraction of the total (in the neighborhood of 20%). Application of the above heuristic has thus yielded a significant savings in visibility computation and thus total computation time (Chapter 3).

Partial visibility interactions are of particular interest, however — they indicate possible discontinuities or reductions in illumination due to transitions from visibility to occlusion; that is, the presence of shadow lines and penumbral regions. It is extremely important to capture such phenomena in the computation of illumination, particularly as perceived by the eye. The relative scarcity of partial interactions allows the concentration of additional computation over such regions, without a prohibitive increase in computing time. Thus, in our adaptive refinement procedure, we increase computed transport error if visibility between patches is estimated as partial — generating increased subdivision and allowing the capture of possible induced intensity gradients.

In addition to visibility weighting, the algorithm of [HSA91] allowed the application of weighting by transport *radiosity*. That is, the error in refinement between a given pair of patches is computed as estimated transport error multiplied by the total radiosity to be transported between the patches. Application of radiosity weighting allows very significant savings in computation over interactions with relatively little transport energy, and a corresponding concentration of computation over the most significant light transport. It allows rapid and accurate computation over environments under hierarchical subdivision equivalent to very exacting refinement over significant illumination. Figure 1 shows an example image created by the algorithm under radiosity weighting and visibility heuristics; illumination was computed in one minute and eleven seconds. Under classic radiosity this would require 1.4 billion interactions, whereas the algorithm employs 20150.

Hierarchical radiosity techniques have been further improved and accelerated by the work of Smits et al. [SAS92] introducing importance driven refinement. Under their algorithm, the dual systems governing transport of radiosity from light sources and importance from the eye are simultaneously solved — hierarchical refinement is controlled by error computed as transport error multiplied by both transport radiosity and transport importance (see Chapter 5). Refinement thus occurs only over interactions exhibiting appreciable radiosity important to the view from the eye, resulting in an extremely significant reduction in computation of viewed radiosity within complex environments [SAS92]. The effectiveness of this approach directly motivated the derivation of the adjoint to the discrete three point transport equation, and subsequent application of importance described in the previous section and presented in Chapter 5.

1.3 This Dissertation

This dissertation is a discussion and development of hierarchical algorithms for illumination.

In Chapter Two we provide a brief overview of some of the concepts, nomenclature, and models useful in the description and computation of illumination and reflection.

In Chapter Three we present in detail the hierarchical radiosity algorithm described in the previous section. This chapter is derived directly from [HSA91].

In Chapter Four we derive and discuss the radiance formulation for discrete three point transport given in Section 1.2.1, and present the resulting adaptive refinement algorithm for hierarchical computation of global illumination over transport triples. We describe the application and analysis of hierarchical three point transport over glossy reflection, and present the results of this implementation. This chapter is derived directly from [AH93a].

In Chapter Five we derive the adjoint to our original radiance formulation, and describe its application in the form of an importance driven hierarchical refinement algorithm. This chapter is directly derived from [AH93b].

In Chapter Six, we provide some general discussion of hierarchical algorithms, their application, and potential for improvement and further work.

Finally, in two appendices we describe in detail the error analysis underlying the implementation over glossy reflection, and discuss conservation of energy over our reflection model.

Chapter 2

Illumination and Reflection

In the following chapters we will express and manipulate illumination over both individual surfaces and relatively complex environments. This work requires an understanding of the measurement and description of radiated energy; that is, the field of *radiometry*. In this chapter we discuss some of the basics of radiometry: the nomenclature and quantities employed in light transport, reflection, and the computation of local and global illumination.

This chapter is based on three excellent reviews of radiometry, reflection, and illumination: [NRHGL77], [K90], and [H92]. If the reader is not familiar with these topics, s/he is very much encouraged to read over the given references — the review provided here is sketched in specifically to meet the needs of the following chapters (particularly Chapter 4), and does not adequately introduce a highly rewarding subject fundamental to image synthesis.

2.1 Radiometry and Reflection

As discussed in [K90], the primary difficulty in the measurement and description of light is that light, as electromagnetic radiation, is not a static and easily measured quantity such as mass. Rather, light is fundamentally a *flow*, a radiant flux of energy per unit time:

$$\Phi = \frac{dQ}{dt}$$

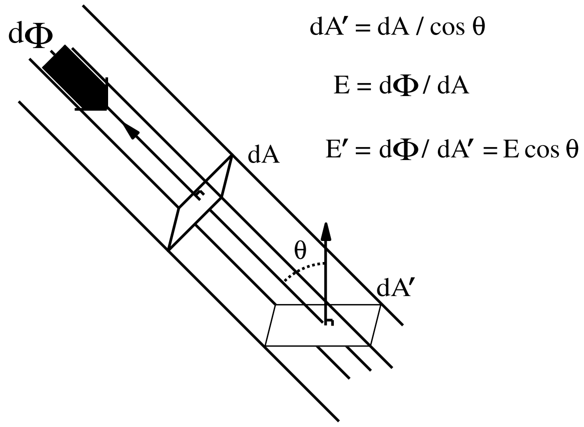


Figure 8: Irradiance and Orientation

where Q is energy, measured in joules, and t is time, in seconds. Recall that a joule is the unit of energy required for the application of a force of one newton over a distance of one meter, where a newton is the unit of force inducing acceleration of one meter per second per second over a mass of one kilogram.

Radiant flux, Φ, is thus measured in watts $=$ joules$/$second.

The illumination within a given environment may be characterized by the radiant flux (in the visible wavelengths) through every point in the environment. We may determine the radiant flux at a given point by computing the flux through a differential area dA at that point:

$$E = \frac{d\Phi}{dA} (= \frac{dQ}{dAdt})$$

the result is *flux density*, that is, the flux per unit area at dA: energy per unit time per unit area, measured in w/m^2. In radiometry, the flux density at a given (real or imagined) surface is called the *irradiance* at the surface. As discussed in Section 2.2.1, radiosity is simply outgoing radiant flux density.

Note that the irradiance at a given differential area depends on the orientation of the area relative to the incident flux (Figure 8). Consider two differential areas dA, dA' such that each area exactly spans a narrow "beam" of incident, parallel flux. If dA is perpendicular to this beam, the irradiance incident at dA is simply $E = d\Phi/dA$. Let θ be the angle of

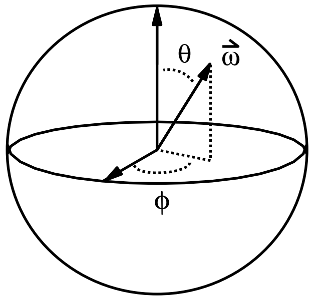


Figure 9: Direction Vector $\vec{\omega}$

incidence at dA', then:

$$dA' = \frac{dA}{\cos\theta}$$

and thus:

$$E' = \frac{d\Phi}{dA'} = E\cos\theta$$

Irradiance at a surface thus varies as the cosine of the angle of incidence.

When computing illumination it is useful to be able to capture directly this directionality of light. That is, we wish to describe the light incident at a given surface from a particular direction or directions — to parameterize radiant flux explicitly in terms of direction. In order to achieve this we must first derive a more exact and tractable definition of direction in three dimensions.

2.1.1 Solid Angles

Consider a unit direction vector $\vec{\omega}$ (Figure 9). This vector may be represented as a point (θ, ϕ) on the surface of the unit sphere. We may generalize this notion to a cone or cluster of directions represented by a finite area on the sphere's surface (Figure 10).

The *solid angle* subtended by an area A on the surface of sphere of radius r is defined to be:

$$\omega = \frac{A}{r^2} \text{ steradians}$$

The solid angle is thus a dimensionless quantity analogous to the angle as defined in radians: recall that an arc of length a and radius r subtends $\alpha = a/r$ radians. Note that the solid angle subtended by the entire sphere is 4π steradians, and that subtended by the hemisphere

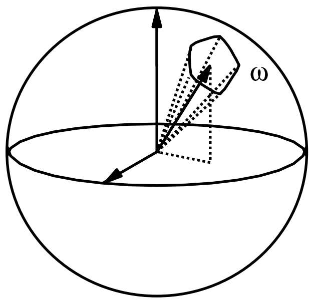


Figure 10: Solid Angle ω

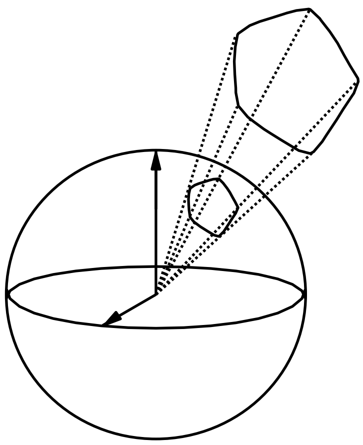


Figure 11: Subtended Solid Angle

above a given point is 2π. Finally, the solid angle at a given point p subtended by a given object is equal to that subtended by the projection of the object onto a sphere centered about p (Figure 11).

Computation of light transport will frequently demand integration over the directions of incidence and reflection at some point of interest. The differential distributions of directions that this integration requires may be expressed as differential solid angles. Consider Figure 12: differential solid angle $d\omega$ is subtended by a differential area of "width" $r \sin\theta d\phi$, and "height" $r d\theta$, and thus:

$$d\omega = \sin\theta d\theta d\phi$$

Consider also the solid angle subtended by a differential area dA at distance r from the viewpoint. The area of dA projected in the direction of the viewpoint is $\cos\theta_p dA$, giving

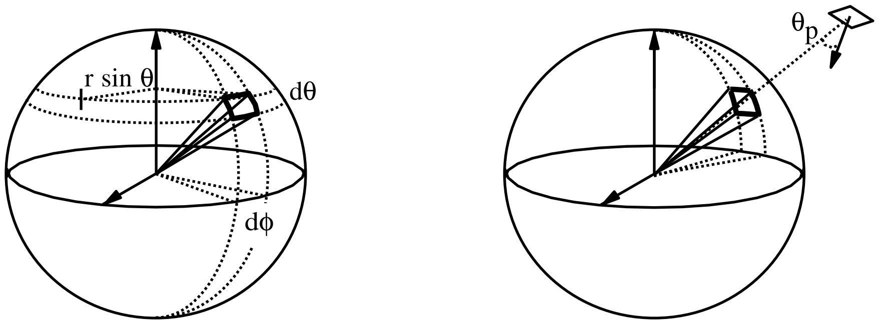


Figure 12: Differential Solid Angles

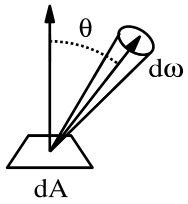


Figure 13: Radiance

induced differential solid angle:

$$d\omega = \frac{\cos\theta_p dA}{r^2}$$

2.1.2 Radiance

We wish to characterize the light incident to (or reflected from) a surface along a given direction. Consider a direction $\vec{\omega}$ at differential area dA (Figure 13); let $L(\vec{\omega})$ be a quantity such that:

$$d\Phi = L(\vec{\omega})\cos\theta dA d\omega$$

where $d\Phi$ is the differential flux incident to dA through $d\omega$. This implies:

$$L(\vec{\omega}) = \frac{d\Phi}{\cos\theta dA d\omega}$$

We call the quantity L *radiance*, that is power per unit solid angle per unit projected area. Radiance is thus angular flux density, and is measured in w/m^2sr. An understanding of radiance, its properties, and manipulation, will be of extreme importance in Chapter 4.

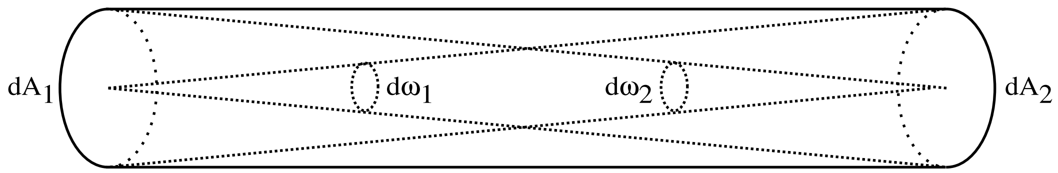


Figure 14: Invariance of Radiance

As is discussed in [H92] and [N63], radiance is arguably the fundamental quantity underlying light transport and image synthesis. The distribution of radiance within an environment completely characterizes all illumination — all radiometric quantities may be computed from it. For example, given solid angle ω_i, the irradiance incident to a surface element dA through ω_i may be computed as:

$$E_i = \int_{\omega_i} L(\vec{\omega}_i) \cos\theta_i d\omega_i$$

The *total* irradiance due to light incident above surface element dA is thus:

$$E_i = \int_{2\pi} L(\vec{\omega}_i) \cos\theta_i d\omega_i$$

where the integral is over the entire illumination hemisphere above dA (recall that the solid angle subtended by a hemisphere is 2π).

An important property of radiance, and one employed implicitly throughout the following chapters, is that the radiance along a given ray is invariant throughout that ray. That is, at any point along a given unobstructed light ray, the radiance in the direction of the ray remains constant. Consider a thin pencil of light (Figure 14) spanned by two differential areas dA_1, dA_2. By conservation of energy, the total flux must be identical at both differential areas. That is:

$$\Phi_1 = \Phi_2$$

and thus:

$$L_1 d\omega_1 dA_1 = L_2 d\omega_2 dA_2$$

Recall that $d\omega_1 = dA_2/r^2$, and $d\omega_2 = dA_1/r^2$, giving:

$$L_1 \frac{dA_1 dA_2}{r^2} = L_2 \frac{dA_1 dA_2}{r^2}$$

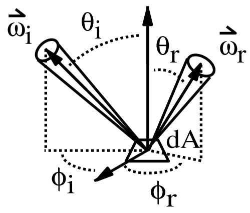


Figure 15: Reflection

and thus:

$$L_1 = L_2$$

showing the desired invariance.

Finally, as discussed in [H92], the response of a light sensor, such as a human eye or exposure meter, is proportional to the radiance visible to the sensor (response may vary, however, with the wavelength of the received radiation). Radiance thus corresponds roughly with our intuitive notion of the "intensity" or "brightness" of light. As a consequence of the above, it is the *radiance* at visible surface in the direction of the eye that should be output to image display devices.

2.1.3 Reflection

Consider differential solid angles at $\vec{\omega}_i$ and $\vec{\omega}_r$ (Figure 15). Experiments show that the radiance reflected along $\vec{\omega}_r$ due to light incident through $\vec{\omega}_i$ is proportional to the irradiance at dA induced through $\vec{\omega}_i$. That is:

$$dL(\vec{\omega}_r) \propto dE(\vec{\omega}_r)$$

We define the ratio of reflected radiance to incident differential irradiance to be the *bidirectional reflectance-distribution function* or BRDF:

$$f_r(\vec{\omega}_i, \vec{\omega}_r) = \frac{L(\vec{\omega}_r)}{L(\vec{\omega}_i) \cos \theta_i d\omega_i}$$

By definition, the radiance along $\vec{\omega}_r$ due to illumination through solid angle ω_i is thus:

$$L(\vec{\omega}_r) = \int_{\omega_i} f_r(\vec{\omega}_i, \vec{\omega}_r) L(\vec{\omega}_i) \cos \theta_i d\omega_i \tag{3}$$

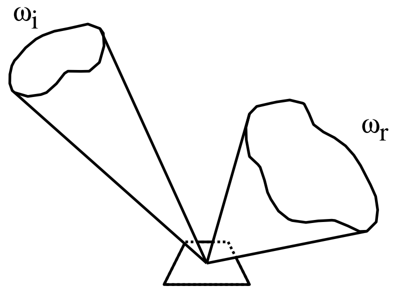


Figure 16: Biconical Reflectance

Note that if x' is the point at dA, and x and x'' are points along $\vec{\omega}_i, \vec{\omega}_r$, we may reparameterize f_r as $f_r(x, x', x'')$. Equation (3) and its reparameterization form the basis of the three point transport formulation derived in Section 4.2.

As discussed in [H92], the ratio of incident irradiance to reflected radiance remains unchanged if reflection is reversed. That is:

$$f_r(\vec{\omega}_i, \vec{\omega}_r) = f_r(\vec{\omega}_r, \vec{\omega}_i)$$

This symmetry of the BRDF with respect to $\vec{\omega}_i$ and $\vec{\omega}_r$ is known as the Helmholtz Reciprocity Principle.

2.1.4 Reflectance

The BRDF at any surface is always greater than or equal to 0 for any given directions of incidence and reflection. Note, however, that the function is not necessarily bounded between 0 and 1 — it is a ratio of radiance to irradiance, and radiance is normalized by solid angle: for reflection into extremely small solid angles, such as under specular or near specular reflection, radiance, and thus the BRDF, may achieve arbitrarily large magnitude. Under perfect specular reflection the BRDF is infinite at the angle of exact reflection, as governed by employment of the Dirac delta function [NRHGL77][H92].

The radiometric quantity *reflectance* does satisfy the above bounds and may correspond more closely to our intuitive understanding of the reflection of light. Consider two finite solid angles ω_i and ω_r (Figure 16). The radiant flux incident through ω_i onto differential area dA is:

$$d\Phi_i = dA \int_{\omega_i} L_i(\vec{\omega}_i) \cos\theta_i d\omega_i$$

The resulting flux reflected from dA through ω_r is:

$$d\Phi_r = dA \int_{\omega_r} L_r(\vec{\omega}_r) \cos\theta_r d\omega_r$$

The ratio of reflected to incident flux over these solid angles is thus:

$$\begin{aligned} \frac{d\Phi_r}{d\Phi_i} &= \frac{\int_{\omega_r} L_r(\vec{\omega}_r) \cos\theta_r d\omega_r}{\int_{\omega_i} L_i(\vec{\omega}_i) \cos\theta_i d\omega_i} \\ &= \frac{\int_{\omega_r}\int_{\omega_i} f_r(\vec{\omega}_i, \vec{\omega}_r) L_i(\vec{\omega}_i) \cos\theta_i \cos\theta_r d\omega_i d\omega_r}{\int_{\omega_i} L_i(\vec{\omega}_i) \cos\theta_i d\omega_i} \end{aligned}$$

If we assume that L_i is uniform and isotropic over ω_i, it may be brought out of the integrals and cancelled. The resulting quantity is *biconical reflectance:*

$$\rho(\omega_i, \omega_r) = \frac{\int_{\omega_r}\int_{\omega_i} f_r(\vec{\omega}_i, \vec{\omega}_r) \cos\theta_i \cos\theta_r d\omega_i d\omega_r}{\int_{\omega_i} \cos\theta_i d\omega_i}$$

By conservation of energy, $0 \le \rho(\omega_i, \omega_r) \le 1$.

Biconical reflectance is the ratio of (uniform and isotropic) incident to reflected flux over specified finite solid angles. Other varieties of reflectance are defined over the other possible pairings of finite (“conical”), differential (“directional”), and hemispherical solid angles: ω, $d\omega$, 2π. Of particular interest is directional-hemispherical reflectance:

$$\rho(d\omega_i, 2\pi) = \int_{2\pi} f_r(\vec{\omega}_i, \vec{\omega}_r) \cos\theta_r d\omega_r$$

as this reflectance expresses conservation of energy over the BRDF for the given angle of incidence — an examination of $\rho(d\omega_i, 2\pi)$ under glossy reflection is presented in Appendix B.

Finally, we note that in Chapter 4 we derive and discuss a new variety of reflectance, *area reflectance*, closely related to biconical reflectance above.

2.2 Diffuse Reflection and Radiosity

2.2.1 Diffuse Reflection and Reflectance

A given surface is said to exhibit diffuse or Lambertian reflection if, for any incident light, reflected radiance is equal in all directions. That is, diffuse surfaces exhibit the same perceived brightness from any viewing angle — the brightness of a wall covered with matte paint does not change appreciably when viewed from different locations.

Let f_d be a BRDF expressing diffuse reflection. Reflected radiance along $\vec{\omega}_r$ due to flux incident through solid angle ω_i is thus:

$$L(\vec{\omega}_r) = \int_{\omega_i} f_d(\vec{\omega}_i, \vec{\omega}_r) L(\vec{\omega}_i) \cos\theta_i d\omega_i$$

Reflected radiance is equal in all directions, $L(\vec{\omega}_r) = L_d$ is thus independent of the given $\vec{\omega}_r$. By the above equation this implies that f_d is also independent of $\vec{\omega}_r$. Furthermore, Helmholtz reciprocity indicates that f_d must be independent of $\vec{\omega}_i$. We thus have:

$$\begin{aligned} L_d &= \int_{\omega_i} f_d L(\vec{\omega}_i) cos\theta_i d\omega_i \\ &= f_d \int_{\omega_i} L(\vec{\omega}_i) cos\theta_i d\omega_i \\ &= f_d E_i \end{aligned}$$

where E_i is the irradiance at dA induced through ω_i.

We define *diffuse reflectance* ρ_d to be the ratio of the flux reflected into the hemisphere above the reflecting surface at dA to that incident through solid angle ω_i. As shown below, $\rho_d(\omega_i, 2\pi)$ is independent of ω_i — the solid angle of incidence may be any finite solid angle, differential solid angle, or the entire illumination hemisphere.

$$\begin{aligned} \rho_d(\omega_i, 2\pi) = \frac{d\Phi_r}{d\Phi_i} &= \frac{\int_{2\pi}\int_{\omega_i} f_d L(\vec{\omega}_i)\cos\theta_i \cos\theta_r d\omega_i d\omega_r}{\int_{\omega_i} L(\vec{\omega}_i)\cos\theta_i d\omega_i} \\ &= f_d \int_{2\pi} \cos\theta_r d\omega_r \\ &= \pi f_d \end{aligned}$$

We thus have:

$$\rho_d = \pi f_d$$

and:

$$f_d = \rho_d / \pi \tag{4}$$

Note that $0 \leq \rho_d \leq 1$ implies that f_d may be of value no greater than $1/\pi$ ($f_d = 1/\pi$ indicates an ideal surface exhibiting perfect diffuse reflection with no absorption of incident light).

Equation (4) gives an alternative formulation for diffuse reflection:

$$L_d = \rho_d E_i / \pi \tag{5}$$

Finally, consider the outgoing irradiance due to diffuse reflection from a given differential area dA. Following usage in radiative heat transfer, and its adaption to computer graphics, we call this quantity *radiosity*, and represent it by the symbol B:

$$\begin{aligned} B &= \int_{2\pi} L_d \cos\theta_r d\omega_r \\ &= \pi L_d \end{aligned}$$

The radiosity at a given diffuse (and nonemitting) surface is thus simply π times the reflected radiance at that surface. Furthermore, given equation (5), we may express diffuse reflection in terms of radiosity and reflectance as:

$$B = \rho_d E_i \tag{6}$$

2.2.2 The Radiosity Formulation

In this section we derive the radiosity equation presented briefly in Section 1.2.2. Consider an environment consisting entirely of opaque surfaces exhibiting diffuse reflection; the outgoing radiance, both emitted and reflected, at a given differential area dA' (Figure 17) may be written:

$$L_d = L_d^e + \int_{2\pi} f_d L(\vec{\omega}_i') \cos\theta_i' d\omega_i'$$

where L_d^e is the radiance emitted at dA' (we assume that emitted as well as reflected radiance is distributed equally in all directions). We may reparameterize the equation over all surfaces X within the environment, replacing integration over the illumination hemisphere:

$$L_d = L_d^e + \int_X f_d L(x) \frac{\cos\theta_r \cos\theta_i'}{|x - x'|^2} dx$$

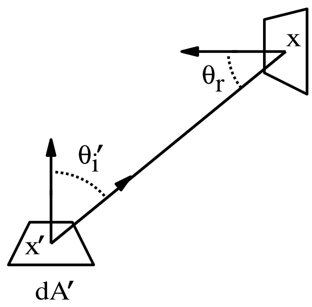


Figure 17: Radiosity Discretization

where x' is the point at dA'. Note that within this section we neglect occlusion: integration over X should include a factor accounting for visibility between points x' and x.

We wish to express in our formulation the illumination at discrete surfaces rather than at differential areas: we thus reparameterize and integrate both sides of the above equation over all surfaces X' at each point x':

$$\int_{X'} L_d(x')dx' = \int_{X'} L_d^e(x')dx' + \int_{X'}\int_X f_d L_d(x)\frac{\cos\theta_r \cos\theta_i'}{|x-x'|^2}dxdx'$$

We assume that X' may be subdivided into subareas A_i such that L_d is nearly constant over each A_i. Similarly, we subdivide X into areas A_j each exhibiting near constant illumination. We then have:

$$\int_{A_i} L_d(x')dx' = \int_{A_i} L_d^e(x')dx' + \sum_j \int_{A_i}\int_{A_j} f_i L_d(x)\frac{\cos\theta_i \cos\theta_j}{|x-x'|^2}dxdx'$$

where f_i is the diffuse BRDF at area A_i

Under our assumption of near constant illumination we may rewrite the above as:

$$L_i A_i = L_i^e A_i + f_i \sum_j L_j \int_{A_i}\int_{A_j} \frac{\cos\theta_i \cos\theta_j}{|x-x'|^2}dxdx'$$

where L_i is the outgoing radiance at A_i, and L_i^e is the emitted component of this radiance. We may divide through by A_i, yielding an expression for L_i:

$$L_i = L_i^e + f_i \sum_j L_j \frac{1}{A_i}\int_{A_i}\int_{A_j} \frac{\cos\theta_i \cos\theta_j}{|x-x'|^2}dxdx'$$

This equation may be rewritten in terms of reflectance, giving:

$$\begin{aligned} L_i &= L_i^e + \rho_i \sum_j L_j \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi |x - x'|^2} dx dx' \\ &= L_i^e + \rho_i \sum_j L_j F_{ij} \end{aligned}$$

where:

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi |x - x'|^2} dx dx'$$

is termed the *form factor* from A_i to A_j (see Section 2.2.3, below).

Finally, this formulation is traditionally written in terms of outgoing irradiance (radiosity) rather than radiance. The identity $B_i = \pi L_i$ may be applied across the equation above, yielding the simple substitution:

$$B_i = E_i + \rho_i \sum_j B_j F_{ij} \tag{7}$$

where $E_i = \pi L_i^e$. The employment of E_i for emitted irradiance is standard usage under radiosity — in the remainder of this section, and within Chapter 3, we shall employ E_i to indicate this quantity rather than the radiometric usage as simple irradiance.

Equation (7), the radiosity formulation, states that the radiosity at area A_i is equal to the radiosity actually emitted by A_i, plus, for every area A_j, the radiosity at A_j times form factor F_{ij}, modulated by the reflectance at A_i. This equation provides a succinct and relatively tractable expression of global illumination at any surface area, under discretization over environments exhibiting only diffuse reflection.

2.2.3 An Alternative Derivation

The Form Factor

Given two areas A_i, A_j (Figure 18), consider the fraction of the outgoing flux at A_i that arrives at A_j. The total flux leaving A_i is:

$$\begin{aligned} \Phi_i &= \int_{A_i} \int_{2\pi} L(x') \cos\theta_i d\omega dx' \\ &= \int_{A_i} L(x') \pi dx' \end{aligned}$$

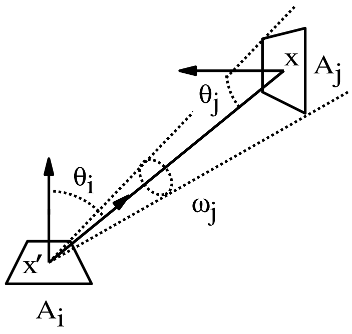


Figure 18: The Form Factor

If we assume, as in Section 2.2.2 above, that the outgoing radiance $L(x') = L_i$ is nearly constant over A_i, we have:

$$\Phi_i = \pi L_i A_i$$

Similarly, the received flux is:

$$\begin{aligned} \Phi_j &= \int_{A_i} \int_{\omega_j} L(x') \cos\theta_i d\omega_j dx' \\ &= L_i \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{|x - x'|^2} dx dx' \end{aligned}$$

The desired ratio is thus:

$$\frac{\Phi_j}{\Phi_i} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi |x - x'|^2} dx dx'$$

Therefore, by definition (Section 2.2.2, again ignoring visibility):

$$F_{ij} = \frac{\Phi_j}{\Phi_i} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi |x - x'|^2} dx dx'$$

That is, the form factor from A_i to A_j is the fraction of the diffusely reflected and emitted energy leaving A_i that reaches A_j. As a ratio of outgoing to received flux, $0 \leq F_{ij} \leq 1$.

Form factors satisfy a reciprocity relation that will be employed in derivations given in the following subsection, and in Section 4.2. Consider the form factor from A_j to A_i:

$$F_{ji} = \frac{1}{A_j} \int_{A_j} \int_{A_i} \frac{\cos\theta_i \cos\theta_j}{\pi |x - x'|^2} dx' dx$$

Then:

$$A_i F_{ij} = \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi |x - x'|^2} dx dx' = A_j F_{ji}$$

Giving the desired reciprocity:

$$A_i F_{ij} = A_j F_{ji}$$

We have defined the form factor from finite area to finite area. We may also define the differential area to finite area form factor from dA_i to A_j:

$$F_{dij} = \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{|x - x'|^2} dx$$

In practice, the computation of a full area to area form factor is usually difficult or impossible; the algorithm and implementation described in Chapter 3 employs a differential area to area form factor to estimate the area to area quantity.

Similarly, we may define the differential to differential form factor as:

$$F_{didj} = \frac{\cos\theta_i \cos\theta_j}{\pi |x - x'|^2} dx$$

Note that F_{ij} is simply the integrated differential form factor, area averaged over A_i.

Finally, note that the form factor F_{ij} is very close to a variety of *reflectance*. Consider incidence from the entire hemisphere above A_i and reflected into ω_j, the solid angle induced by A_j at each point x' on A_i; we may imagine reflectance integrated over the surface at A_i — that is, the fraction of all flux incident at A_i from any direction in the illumination hemisphere that is reflected to A_j:

$$\rho(2\pi, A_i, A_j) = \rho_d F_{ij}$$

If the diffuse reflectivity $\rho_d = 1$, then F_{ij} is exactly this reflectance. The significance of the observation may be more apparent in Section 4.2.2: Area Reflectance.

The Radiosity Formulation

Given the expression of the form factor as the ratio of transported to outgoing flux, we may derive the radiosity equation directly in terms of energy balance. Consider the total outgoing flux at a given surface A_i. We assume, as above, that the radiosity over each discrete surface element is relatively constant. We thus have:

$$B_i A_i = E_i A_i + \rho_i \sum_j B_j A_j F_{ji}$$

That is, the total flux at A_i is equal to the total flux emitted by A_i, plus, for each area A_j, the total flux at A_j times F_{ji}, the fraction of this flux that arrives at A_i, reflected at A_i under reflectance ρ_i.

We may then apply form factor reciprocity $A_j F_{ji} = A_i F_{ij}$ to yield:

$$B_i A_i = E_i A_i + \rho_i \sum_j B_j A_i F_{ij}$$

and divide through by A_i giving an expression for B_i, and the radiosity equation:

$$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

Note that our use of reciprocity to eliminate area dependence results in a somewhat counterintuitive reversal in the sense of the applied form factor.

2.2.4 Matrix Formulation

The radiosity formulation given above provides a system of equations describing the illumination at every surface element in the environment — each equation requiring solution for a given B_i, the radiosity at element A_i, for $1 \leq i \leq n$. We may rewrite these equations as:

$$B_i - \rho_i \sum_j^n F_{ij} B_j = E_i$$

yielding a matrix formulation for the system under which we may solve for B:

$$\begin{pmatrix} 1 & -\rho_1 F_{12} & -\rho_1 F_{13} & \ldots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 & -\rho_2 F_{23} & \ldots & -\rho_2 F_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & -\rho_n F_{n3} & \ldots & 1 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}$$

It is this system that underlies all solution for radiosity within diffuse environments.

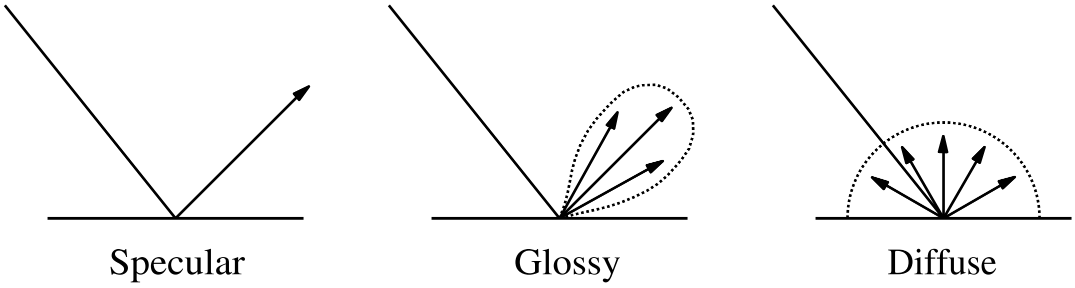


Figure 19: Variation in Reflection

Note that the form factor matrix above is diagonally dominant, as $\rho_i \sum_j F_{ij} < 1$ for all i. The system may thus be solved efficiently under Jacobi iterative methods and variations. The gathering technique employed in Chapter 3 is equivalent to this method, applied over a hierarchical construction.

2.3 Glossy Specular Reflection

Perfect specular and diffuse reflection represent opposite extremes over a range of possible reflection functions (Figure 19) scattering received light with varying degrees of directionality. In this section we discuss one variety of reflection function exhibiting such directional scattering: the general model for "glossy"/rough specular reflection developed by Torrance and Sparrow [TS67], and introduced and adapted to image synthesis by Blinn [B77] and Cook and Torrance [CT82]. It is this BRDF that we will employ in our implementation of discrete three point transport over directional reflection (Chapter 4).

Under the Torrance-Sparrow model it is assumed that surfaces are composed of small, planar *microfacets* of varying orientation — the average slope over microfacets giving the orientation of the represented surface. These microfacets account for the scattering of reflected light (Figure 20) incident to facets of differing orientation. The variance in the distribution of orientation over microfacets indicates the degree of scattering in reflection: widely varying microfacets of arbitrary orientation induce a large degree of scattering and thus a surface exhibiting near diffuse reflection; microfacets with little variation in orientation from the mean surface normal induce highly directional reflection and near

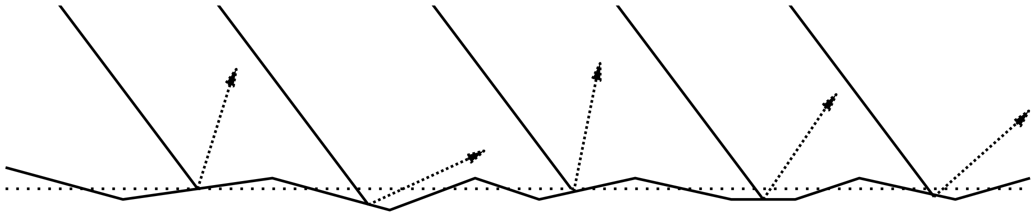


Figure 20: Scattering over Microfacets

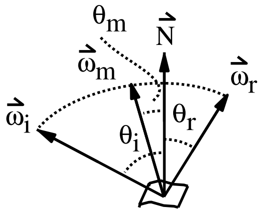


Figure 21: Geometry of Reflection

specular surfaces.

Under the microfacet assumption, the following BRDF modeling glossy reflection may be derived [TS67]:

$$f_{TS}(\vec{\omega}_i, \vec{\omega}_r) = \frac{D S_h F_r}{4 \cos\theta_i \cos\theta_r}$$

The function D describes the distribution of microfacet orientation over the modeled surface. A variety of such functions have been proposed; we employ a simple facet distribution function presented by Blinn [B77]:

$$D_B(\theta_m, \kappa) = \cos^\kappa \theta_m$$

where θ_m is the angle between the mean surface normal $\vec{N}$, and $\vec{\omega}_m$, the *microfacet mirror orientation normal*. Vector $\vec{\omega}_m$ is the normal to the microfacet allowing perfect mirror reflection from $\vec{\omega}_i$ to $\vec{\omega}_r$ (Figure 21); $\vec{\omega}_m$ thus lies halfway between the given directions of incidence and reflection.

Consider a given differential area dA: note that the total area of all microfacets at all orientations over the differential area exceeds dA (Figure 22); however, for any physically realizable surface, the area of the projection of these facets *onto* the differential area must

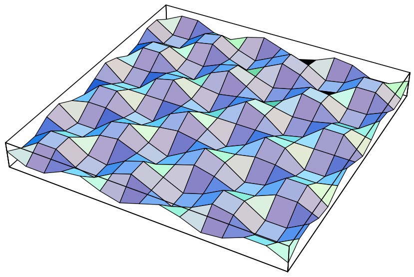


Figure 22: Microfacet Area

be *exactly* dA. As discussed in [M92], we must normalize D_B to account for projected area.

By definition of $D(\theta_m)$ the total projected area of all facets over a given dA is:

$$\int_{2\pi} D(\theta_m) \cos\theta_m d\omega_m dA = dA$$

Thus, for a given $D(\theta_m) = c\cos^\kappa \theta_m$ (c being a normalization coefficient) we have:

$$c\int_{2\pi} \cos^{\kappa+1} d\omega_m = 1$$

And thus:

$$c = \frac{\kappa+2}{2\pi}$$

The normalized microfacet distribution function we employ is thus:

$$D_M(\theta_m, \kappa) = \frac{\kappa+2}{2\pi}\cos^\kappa \theta_m$$

Note that, for large κ, this function has value near 0 over most of $(0, \pi/2)$, but is of great magnitude over a small interval containing 0. For infinite κ, $D(\theta_m)$ collapses into a type of delta function, of infinite value over an infinitesimal interval at 0. This behavior is exactly as expected for near specular and specular reflectors.

The function $S_h(\theta_i, \theta_r)$ expresses *self shadowing* and *masking* over microfacets . The presence of a given microfacet may prevent incident light from reaching other facets over a modeled surface (Figure 23) — the degree to which such shadowing occurs is dependent upon the angle of incidence. Similarly, light may be masked after reflection by

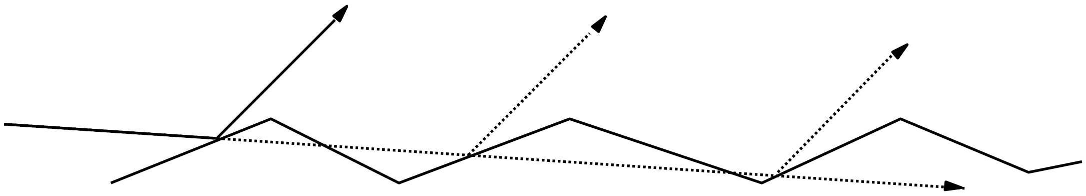


Figure 23: Self Shadowing

occluding microfacets as determined by surface characteristics and the angle of reflection. As discussed in Appendix B, it is necessary to take into account microfacet shadowing and masking in order to preserve conservation of energy over modeled BRDFs.

As with distribution functions, a number of shadowing functions have been proposed; however, as discussed in Appendix B, for near specular surfaces shadowing and masking do not become critical until relatively high θ_i or θ_r. We thus employ the function:

$$S_h(\theta_i, \theta_r) = \text{step}(\theta_i - \theta_c, \theta_r - \theta_c)$$

simply clamping S_h from 1 to 0 when θ_i or θ_r exceeds a preset limit θ_c near the horizon.

Finally, F_r is the *Fresnel reflectance function*. As discussed in [H89], Fresnel reflectance expresses the fraction of incident energy at a given surface that is reflected as opposed to transmitted (the fraction $(1 - F_r)$ is thus transmitted/absorbed at the surface). This fraction is a function of the angle of incidence and polarization of received light, and the indices of reflection of the media and reflecting materials. Within this dissertation, we shall assume $F_r \equiv 1$, ignoring Fresnel effects such as increased reflectivity as the angle of incidence approaches grazing.

Chapter 3

A Rapid Hierarchical Radiosity Algorithm

3.1 Introduction

In this chapter we present efficient techniques for solving the transport equations that arise in the computation of global illumination under radiosity. These techniques draw from recent insights into fast numerical algorithms for solving the N-body problem [A85, BH86, G88]. Computational efficiency is achieved by carefully analyzing the error in performing form factor integrals. Without such error analysis, pictures may contain artifacts where the form factors have large error. More importantly, many form factor computations are done at much higher precision than is necessary. Error analysis, in combination with a multi-resolution representation, can be used to reduce significantly the number of interactions that are considered.

In previous work, Hanrahan and Salzman [HS90] have analyzed the form factor calculation between two unoccluded polygonal patches discretized into n finer polygonal elements. They have shown that the form factor matrix may always be approximated to within some preset numerical tolerance with at most $O(n)$ terms, and often many fewer. The work presented in this chapter extends the previous radiosity algorithm to handle scenes with many polygons, where occlusion plays an important role. Occlusion, although costly to detect, reduces the number of interactions even further. The form factor matrix is therefore sparser, allowing faster solution for equilibrium radiosities. The technique used for determining visibility is based on ray tracing, but two important optimizing heuristics are introduced.

One takes advantage of visibility coherence between different levels of detail; the other is based on the observation that most interactions between patches are either totally visible or totally invisible with respect to each other. Finally, we show how to use multigridding in combination with a brightness-weighted error estimate. This leads to a faster progressive radiosity algorithm.

This chapter is derived directly from the original presentation of this work appearing in the proceedings of Siggraph 91 [HSA91].

3.2 A Brief Review of Previous Work

3.2.1 Radiosity

As discussed in Chapter 2, radiosity algorithms assume the environment has been discretized into small elements which have constant brightness. In this dissertation we shall employ the term "element" to describe the subpatches at the finest level of surface subdivision, and the term "patch" for all larger, higher level subpatches, including the original input polygons. We shall assume k input polygons, and n elements at the finest level of permitted subdivision. Enforcing an energy balance at every element yields a system of equations of the form (Section 2.2.3):

$$B_i = E_i + \rho_i \sum_j^n F_{ij} B_j$$

where B_i is the radiosity, E_i is the emissivity, ρ_i is the diffuse reflectance, and F_{ij} is the form factor (the percentage of light leaving element i that arrives at element j). Similar equations exist for all elements, yielding a linear system of equations (Section 2.2.4):

$$\begin{pmatrix} 1 & -\rho_1 F_{1,2} & \dots & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & 1 & \dots & -\rho_2 F_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n,1} & -\rho_n F_{n,2} & \dots & 1 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}$$

This system of equations can be efficiently solved using iterative algorithms such as the Gauss-Seidel method. Physically, the Gauss-Seidel method is equivalent to successively *gathering* incoming light. An alternative iteration scheme is to reverse this process by successively *shooting* light from patches in order of their brightness [CCWG88]. This has the advantage that the solution converges more quickly, and if the scene is drawn during the iteration, successive images gradually improve as the computation proceeds [BFGS88].

The most expensive part of the calculation is computation of form factors. Recall the form factor over areas A_i and A_j:

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r_{ij}^2} dA_i dA_j$$

This formula does not take into account occlusion. To do so requires that differential form factors be accumulated only if the two infinitesimal elements are mutually visible. The first practical approach to integrating visibility into form factor computations was the *hemi-cube* algorithm developed by Cohen and Greenberg [CG85]. The hemi-cube algorithm is simple and fast, and can be accelerated using current workstation graphics hardware. Algorithms based on ray tracing also have been proposed for form factor calculation [M88, WRC88, WEH89, SP89].

There are two major sources of error when computing form factor integrals. First, the integral is evaluated by sampling the patches in some way; since the results of uniform sampling process are subject to aliasing, early methods had noticeable aliasing errors. However, more recent methods [WEH89] have overcome sampling errors by incorporating stochastic sampling into a ray tracer [C86]. Second, the form factor between two surface samples can be approximated by the differential form factor only if the distance separating the two samples is large compared to their size. This condition frequently occurs along edges and in corners where polygons meet. To avoid this problem, Baum et al. [BRW89] switch to an analytically calculated form factor in these situations. Another approach, employed by Wallace et al. [WEH89], is to adaptively supersample the integral.

The form factor matrix is n by n, where n is the number of elements. This n^2 growth causes time and memory problems for complex scenes, The first method to reduce the computational costs was motivated by the method of *substructuring* used in finite element calculations. The polygons comprising the scene are discretized at two levels [CGIB86]. One level contains the patches into which input polygons are broken, and the other level contains the elements into which each patch is broken. Normally, the number of patches and elements are determined *a priori*, but the number of elements can also be determined by recursive subdivision based on radiosity gradients [CGIB86]. Other attempts to utilize adaptive subdivision are described in [CF90] and [H90].

3.2.2 N-Body Problem

The hierarchical subdivision algorithm proposed in this chapter is inspired by methods recently developed for solving the N-body problem. In the N-body problem, each of n particles exerts a force on all the other $n-1$ particles, implying $n(n-1)/2$ pairwise interactions. The fast algorithms compute all the forces on a particle in less than quadratic time, building on two key ideas:

1. Numerical calculations are subject to error, and therefore, the force acting on a particle need only be calculated to within the given precision.
2. The force due to a cluster of particles at some distant point can be approximated, within the given precision, with a single term—cutting down on the total number of interactions.

Appel was the first to develop a hierarchical algorithm for solving the N-body problem, by approximating the forces between particles in two clusters with a single force, when the separation between the clusters significantly exceeded their sizes. A top-down traversal of a hierarchical k-d tree representing the clusters yielded an $O(n \log n)$ algorithm [A85]. More recently, Esselink analyzed Appel's algorithm and showed that time required to calculate the forces is linear [E89], and that the observed $O(n \log n)$ running time is a consequence of the preprocessing time required to build the hierarchical data structures. Barnes and Hut developed a similar algorithm based on octrees [BH86]. Greengard and Rokhlin devised the first $O(n)$ algorithm, using a p-term multipole expansion for the potential due to any cluster, along with algorithms for splitting, merging, and translating the resulting multipole expansions [G88]. The algorithm proposed in this chapter is most closely related to Appel's and Barnes and Hut's algorithms; it should be mentioned that these two algorithms are very easy to implement in a few hundred lines of code.

The radiosity problem shares many similarities with the N-body problem which suggest that these ideas can be used to increase its efficiency. In both the N-body and the radiosity problem, there are $n(n-1)/2$ pairs of interactions. Moreover, just as gravitational or electromagnetic forces fall off as $1/r^2$, the magnitude of the form factor between two patches also falls off as $1/r^2$. Finally, according to Newton's Third Law, gravitational forces are equal and opposite, and, according to the reciprocity principle, form factors between two polygons are related.

One major difference between the two problems is the manner in which the hierarchical data structures are formed. The N-body algorithms begin with n particles and cluster them into larger and larger groups. Our radiosity algorithm, however, begins with a few large polygons and subdivides them into smaller and smaller patches. Subdividing based on the error of a potential interaction provides an automatic method for discretizing the scene within the given error bounds. The specifics of our subdivision algorithm is discussed in Section 3.3. The separate problem of building clusters out of individual patches is not dealt with in this dissertation.

Another difference is that the N-body algorithms take advantage of linear superposition; the principle of superposition states that the potential due to a cluster of particles is the sum of the potentials of the individual particles. This principle does not always apply to the radiosity problem, because of occlusion: intervening opaque surfaces can block the transport of light between two other surfaces, which makes the system non-linear. Occlusion thereby introduces an additional cost to the radiosity problem. This is discussed in Section 3.4.

Finally, the N-body problem is based on a differential equation, whereas the radiosity problem is based on an integral equation. The integral equation arising from the radiosity problem can, however, be solved efficiently using iterative matrix techniques. Fortunately, the hierarchy of interactions produced by our subdivision is equivalent to a block structured matrix, and the iteration can be efficiently computed. This is discussed in Section 3.5.

3.3 Form Factor Matrix Approximation

This section describes a recursive refinement procedure which simultaneously decomposes a polygon into a hierarchy of patches and elements, and builds a hierarchical representation of the form factor matrix by recording interactions at different levels of detail. We begin by describing the procedure and its results, and then proceed to analyze the error in the resulting form factors, and the number of interactions that need to be considered. This section is very similar to [HS90].

Consider the procedure `Refine`:

```
Refine(Patch *p, Patch *q, float Feps, float Aeps)
{
    float Fpq, Fqp;

    Fpq = FormFactorEstimate(p, q);
    Fqp = FormFactorEstimate(q, p);

    if (Fpq < Feps && Fqp < Feps)

        Link(p, q);

    else {

        if (Fpq > Fqp) {

            if (Subdiv(q, Aeps)) {

                Refine(p, q->ne, Feps, Aeps);
                Refine(p, q->nw, Feps, Aeps);
                Refine(p, q->se, Feps, Aeps);
                Refine(p, q->sw, Feps, Aeps);

            }
            else Link(p, q);
        }
        else {

            if(Subdiv(p, Aeps)) {

                Refine(q, p->ne, Feps, Aeps);
                Refine(q, p->nw, Feps, Aeps);
                Refine(q, p->se, Feps, Aeps);
                Refine(q, p->sw, Feps, Aeps);

            }
            else Link(p, q);
        }
    }
}
```

`Refine` first estimates the form factor between two patches, and then either subdivides the patches and refines further, or terminates the recursion and records an interaction between the two patches. If the form factor estimate is less than `Feps`, then the true form factor (not taking into consideration occlusion) can be approximated accurately by the estimate (see below), and the patches are allowed to interact at this level of detail. (The procedure `Link` records the interaction between the two patches.) However, if either of the form factor estimates is larger than `Feps`, then the form factor estimate is not accurate, and so the patch with the larger form factor is subdivided, and `Refine` is called recursively with the smaller subpatches.

`Subdiv` subdivides a patch into subpatches. In our implementation, a patch is a planar quadrilateral, and it is subdivided equally into four new quadrilaterals by splitting it at its center. The subdivision hierarchy is stored in a quadtree; the pointers to the four children are stored in the fields `nw`, `ne`, `sw`, and `se`. (This data structure is similar to adaptive radiosity textures proposed in [H90], although information is stored at all levels of the hierarchy, not just at the leaf nodes, and each level also stores a list of its interactions.) `Subdiv` returns *false* if the patch cannot be split; this condition occurs if the area of the patch is smaller than some absolute predetermined area `Aeps`, and is necessary to prevent infinite recursion in corners and along edges. If subdivision is not possible, we force the two patches to interact. Note that a patch may be refined against many patches, and so the actual subdivision of a patch may have been performed previously. When this occurs `Subdiv` need do no other work and simply returns *true*.

The procedure `FormFactorEstimate` returns an upper bound on the form factor from the first patch to the second patch, assuming the first patch has infinitesimal size and the second patch has finite size. The form factor can be estimated by either calculating the solid angle subtended by a disk with cross sectional area equal to the surface area of the patch [WEH89], or by circumscribing a sphere around the patch and estimating the solid angle subtended by the sphere.

An example of a tree that might be produced by `Refine` and its associated form factor matrix is shown in Figure 24. For simplicity, the figure illustrates the interactions between two hypothetical 1D patches; in this case the hierarchy can be represented with a binary rather than quaternary tree. The two binary trees representing the induced subdivision, and are drawn side by side along the edges of the form factor matrix. Since in this example each

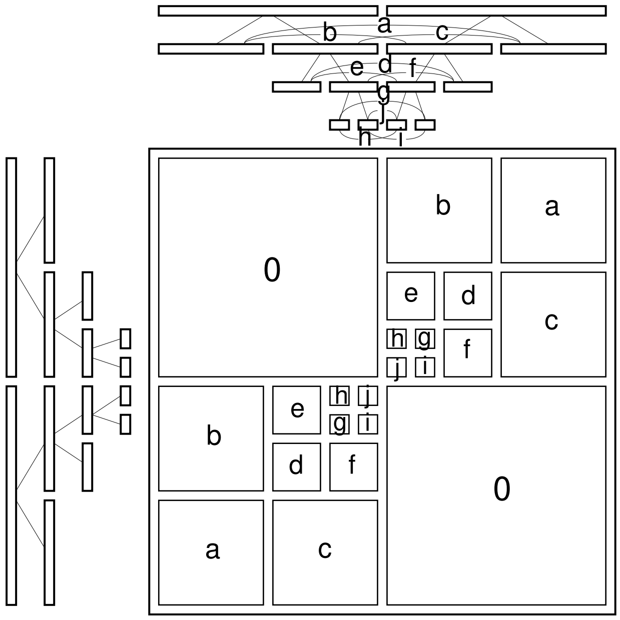


Figure 24: The block form factor matrix for a particular binary tree example. Each labelled block corresponds to a labelled arc connecting nodes in the hierarchical subdivision. Although the blocks are all square in this example, that is not the case in general.

binary tree represents a polygon, no interactions are shown with itself. The leaves of the tree are the elements in the discretization. The combination of all the leaf nodes completely cover the input patch. Interactions between patches at different levels are represented by labelled blocks in the form factor matrix, and by labelled arcs between nodes in the trees. Notice that the size of the block in the form factor matrix depends on the level in the tree the patches interact at. The higher the level, the bigger the block.

The first point in the analysis is the relationship between the termination criteria and the accuracy of the computed form factors. Obviously, the termination criteria causes the form factor corresponding to each interaction to have approximately the same magnitude, because, if an estimated form factor were larger, the patches would be subdivided, otherwise, they are allowed to interact. More importantly, the termination criteria also places an upper bound on the error associated with the form factor integral between the two interacting patches. This can be verified by examining the form factor from a point to a disk of radius r:

$$F_{disk} \approx \frac{r^2}{R^2 + r^2} = \left(\frac{r}{R}\right)^2 \left(1 - \left(\frac{r}{R}\right)^2 + \left(\frac{r}{R}\right)^4 + \cdots\right)$$

where R is the distance from the point to the center of the disk. Thus, the error due to the finiteness of the geometry is given by terms involving powers of (r/R). Because F goes as $(r/R)^2$, when F is small (implying that the size of the patch is small compared to the distance separating the patches), the differential form factor is also a good estimate of the true form factor. A more rigorous proof of this result may be obtained by forming the Taylor expansion of the form factor integral. In the N-body problem, this expansion is the multipole expansion. However, one need not ever calculate the expansion explicitly to use this algorithm.

Note that, under this algorithm and analysis, estimated error is bound over individual form factor computations. As discussed in Sections 4.6 and A.2, bounding estimated error over individual transport does not necessarily provide a measure of over all error in the total system. The development and application of such a metric remains an important open problem.

The second point in the analysis is that the resulting form factor matrix has fewer than n^2 blocks. To a certain extent this is obvious, because every time an interaction occurs at some higher level of detail, the number of interactions is reduced, but we wish to count the interactions more precisely. For simplicity, again consider the 1D problem of n equally spaced patches along a line. Later we will consider what happens if the patches are 2D and non-uniformly distributed. Let us construct a binary tree above the patches by merging adjacent contiguous patches recursively. This is shown in Figure 25. The error criterion says that two patches can interact directly only if $(r/R)^2 < \texttt{Feps}$. In other words, two patches of size r can interact only if the distance R between them is greater than $r/\sqrt{\texttt{Feps}}$. For concreteness, let us fix `Feps` so that this criterion is equivalent to saying that two patches at the same level in the binary tree can interact only if at least one other patch at that level is between them: Otherwise, they would subtend too large a solid angle and would subdivide, pushing the interaction down a level in the tree. Now consider the interactions of a patch p in the interior of the tree. At any level in the tree, the rule forbids the patch p from interacting with its immediate neighbors. These immediate neighbor interactions, therefore, must be handled by p's children. In the same way, p is only responsible for

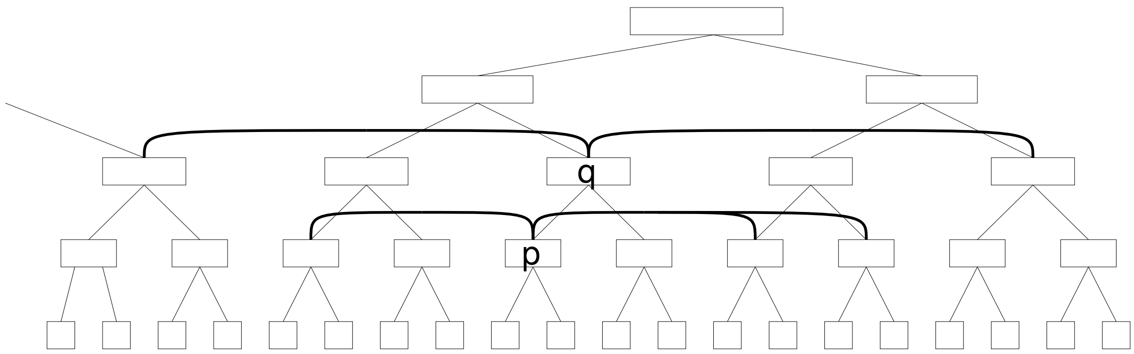


Figure 25: Interactions of the node p with neighboring nodes in a one-dimensional subdivision.

handling the interactions from its parent q's immediate neighbors. Therefore, p need only interact with the children of q's immediate neighbors. Figure 25 shows the node p and its parent q. The above considerations imply that p need only make three connections to nodes at its level. This argument applies to all levels of the tree (except the top and the bottom, but these levels result in fewer interactions), and therefore each node in the tree connects to a constant number of other nodes. Thus, the total number of interactions is proportional to the number of nodes in the tree, which is $O(n)$. A similar analysis has been derived independently by [E89].

Figure 26 shows the quadtree subdivision and the interactions at each level in the hierarchy computed by `Refine` between a pair of perpendicular polygons. This figure shows that each interior patch has a constant number of interactions with other patches regardless of the level in the tree. Note also that the choice of hierarchical structure, in this case a quadtree, can lead to a degree of asymmetry in refinement and linking.

Figure 27 plots the actual number of interactions versus the number of potential interactions at a fixed uniform level of discretization. The number of interactions for perpendicular polygons goes, surprisingly, as $O(\sqrt{n})$. The subdivision induced between two perpendicular polygons is comparable to a binary tree turned on its side with its leaf nodes along the common edge, and the total number of nodes in such a sideways binary tree will be $O(\sqrt{n})$. The worst case for `Refine` is two parallel polygons whose size is much larger than the distance separating them. In this case, there will be $O(n)$ interactions. As the polygons move

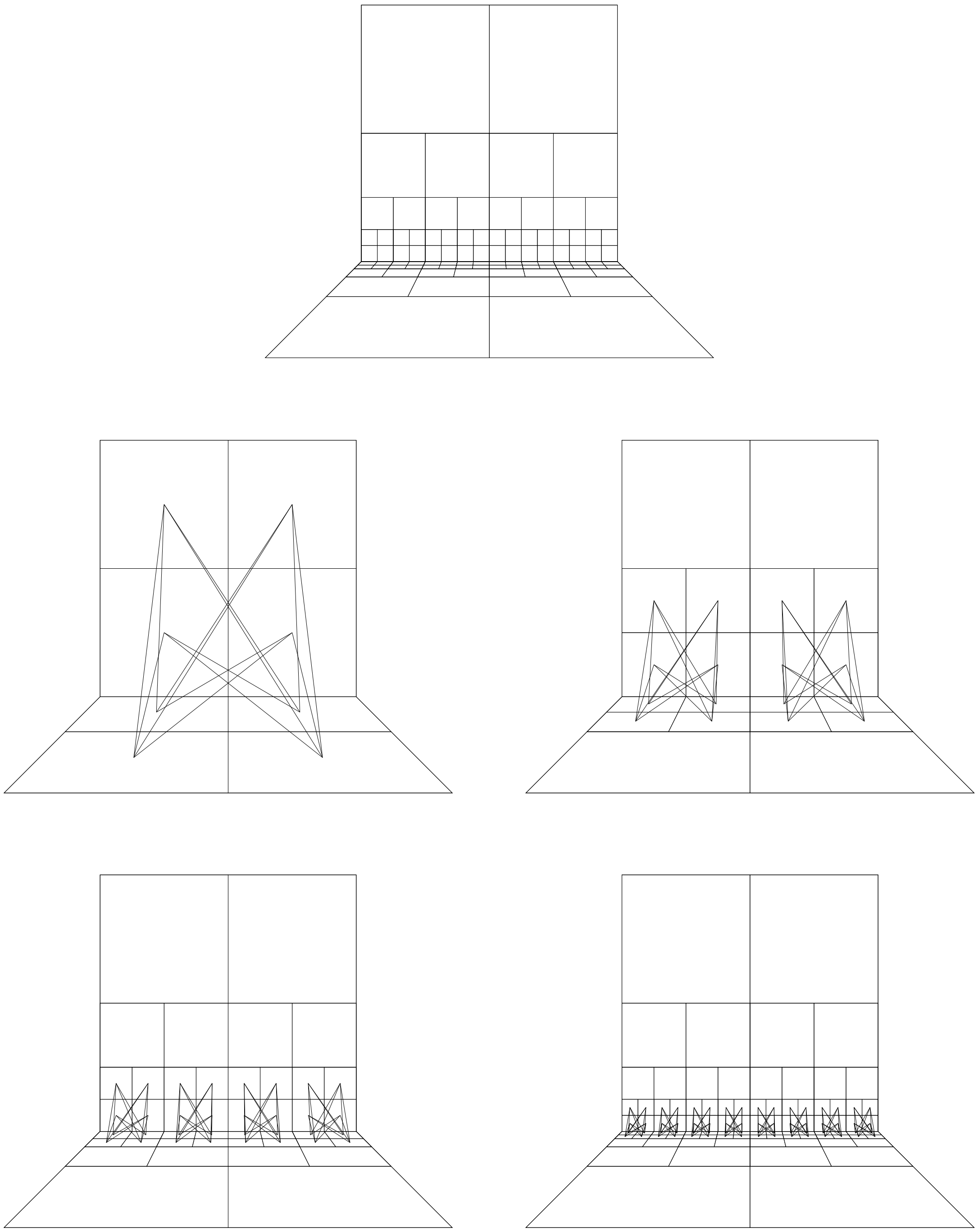


Figure 26: Interactions between a pair of perpendicular polygons.

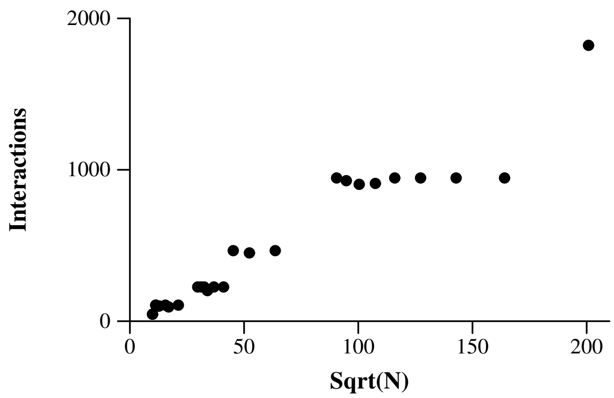


Figure 27: Number of interactions vs. number of elements for a pair of perpendicular polygons.

further apart, or are tilted relative to each other as in the case of perpendicular polygons, the number of interactions is reduced. Finally, as the two polygons move still farther apart, eventually only a single interaction is required.

To verify the accuracy of the form factors generated by our method, we compared the computed form factors with the analytical form factors which are available for the parallel and perpendicular geometries (see, for example, [SH81]). To compute the form factor between two finite areas, `Refine` can be modified to return the sum of the form factors of a patch's children or, if the patch is a leaf, the product of the patch's area and the differential form factor to the other patch. Figure 28 shows the measured relative error between the computed and the analytical form factors as a function of `Feps`. As expected, the actual error in the form factor is proportional to the `Feps` given to `Refine`, as predicted by the theory. Note that the self symmetry in form factor computation over these configurations indicates that similar error is exhibited at differing levels of refinement. The plateaus in these figures are due to the discrete nature of subdivision.

Finally, note that initial refinement over an environment requires examination of all pairs of input polygons, at a cost of $O(k^2)$ operations.

In summary, our hierarchical refinement method estimates the form factor matrix between two unoccluded patches to within a fixed error tolerance automatically. In the process it reorganizes the form factor matrix into $O(n + k^2)$ or fewer blocks; the estimated form

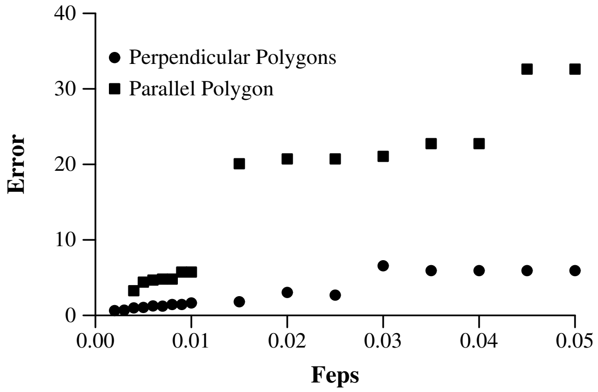


Figure 28: Measured relative percentage error vs. `Feps`.

factor associated with each block has the same value and error as other blocks.

3.4 Visibility

The pairwise method for computing form factors described in the previous section is accurate as long as each patch is completely visible with respect to the other patch. Unfortunately, occlusion exists in all realistic environments, and so this idealization is not very useful in practice. In this section we modify the algorithm to take into consideration visibility.

Intervening occluding surfaces can only decrease light transport between two patches, thus, the true form factor in the presence of occlusion is never greater than the form factor estimate described above. The effect of occlusion can be modeled by multiplying the estimated form factor by a visibility correction factor which estimates the percentage each patch sees of the other.

$$F = V_e F_e$$

where F_e is the estimated form factor without considering occlusion, and V_e is the estimated visibility. If $V_e = 1$ then the two patches are totally visible; if $V_e = 0$ then they are completely occluded; and otherwise they are partially visible. Thus, assuming no visibility error, the level of detail for the interaction between two patches need never be finer than that computed by the procedure `Refine`.

Recall that all the form factor estimates computed by `Refine` have approximately

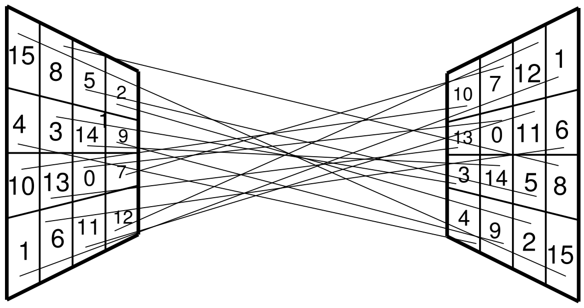


Figure 29: Jittered rays fired between two polygons to determine the percentage visibility.

the same error. As the form factor is not precise, the calculation of V_e need only be estimated to similar precision. Ideally, the visibility module should take into account the precision required; in reality, current visibility modules probably compute visibility much more accurately than is necessary. Under our implementation, we employ the same visibility tests over all potential interactions. The total number of such tests is thus at worst proportional to the number of interactions. That is, the total work performed is:

$$T(n) = F(n)V(n)$$

where $F(n)$ is the number of computed form factors and $V(n)$ is the cost of performing the visibility test for a given number of elements. As has been shown, $F(n)$ varies at most linearly with n, so many fewer visibility tests need be done than with conventional radiosity algorithms.

Two types of visibility tests are performed in implementation. The first test determines whether two polygons face each other, face away from each other, or if the support plane of one polygon splits the other. This test considers only the two polygons and not the environment, and therefore can be done in constant time. The second visibility test checks how much of each polygon is visible from the other polygon given the global environment. The test fires a fixed number of rays between the two patches, and computes the percentage of rays not blocked by intervening surfaces. Each patch is subdivided into a 2D grid (typically 4×4), and the cells in the grid are assigned numbers from a magic square [C86]. Each ray is really a line segment formed by joining jittered points within corresponding cells with the same number as shown in Figure 29. A naive ray intersection test would require $O(k)$ time. In order to accelerate the visibility test, we use a modified version of the

BSP-tree algorithm, described in [TN87], which in principle reduces the intersection cost to $O(\log k)$. This ray intersection module recalls shadow testing, because it returns only whether the ray is blocked or not, and not the closest intersection along the ray direction.

Figure 30 shows a simple scene. Induced subdivision is indicated for each polygon, and interactions between subpatches are illustrated as links. The series shows all links, ordered by increasing height, within the hierarchy, of the linked subpatches. Note that large patches interact with other large patches that are far away, whereas smaller patches interact with similarly sized patches at closer distances. This hierarchy is most clearly shown in the corners and along the edges of the room.

A tentative list of interactions was computed by refining patches relative to each other irrespective of visibility. Each interaction was then tested for visibility and the form factor adjusted. In the figure, links between interacting patches that are completely visible relative to each other are colored white. Green links interacting patches that are partially visible with respect to each other, pink links those for which one patch's supporting plane splits the other, and dark blue links those patches whose interactions are found to be completely occluded (invisible).

Table 1 shows some basic statistics for this scene. Note that the total number of interactions is only 15526, compared with over 10^9, assuming all elements interacted directly. The total time to compute this picture was approximately one minute (all times quoted in this chapter are on a SiliconGraphics 210 GTX unless otherwise noted).

The interactions in scenes such as those shown in Figure 30 exhibit a great deal of *visibility coherence*. Fringes of partial and splitting links tend to surround larger areas of complete visibility or occlusion (see also Figure 31, below). Partial visibility corresponds to penumbral areas generated along the silhouettes of an object, which occur less often than the interior or exterior of an object, although pathological exceptions may be easily constructed. In the scene shown in Figure 30, (see Table 1), we find that 52.6% of the interactions are totally visible, 28.8% are totally invisible, and only 18.5% are partially visible.

The recursive refinement procedure can exploit visibility coherence in a natural fashion to prune out unnecessary refinement and visibility calculation. If, in the course of subdivision, two subpatches become totally invisible relative to each other, then the refinement

Figure 30: Hierarchical subdivision, interactions, and visibility. This sequence shows the hierarchical subdivision of each polygon, from the smallest elements in the upper left image, up the hierarchy to the largest subpatches shown in the image at lower right. Line segments link interacting patches, shown at the hierarchical level of the smaller patch. Segment color indicates visibility: white – completely visible, green – partially visible, pink – cut by supporting plane, and dark blue – relatively invisible. Note that there are many more visible and invisible links than partial.

Polygons	98	
Potential elements	44773	
Potential interactions	1002288378	
Without visibility coherence		
Patches	7286	
Elements	5489	
Interactions	15526	
Totally-invisible	4477	28.8%
Totally-visible	8171	52.6%
Partially-visible	2878	18.5%
Tests		
Refinement tests	19117	
Visibility tests	11123	
Ray tests	177968	
With visibility coherence		
Patches	7350	
Elements	5537	
Interactions	15598	
Totally-invisible interactions	4495	28.8%
Totally-visible interactions	8249	52.9%
Partially-visible interactions	2854	18.3%
Tests		
Refinement tests	19213	
Totally-invisible refines	3600	18.7%
Pre-Totally-invisible refines	0	0.0%
Totally-visible refines	10487	54.6%
Pre-Totally-visible refines	9700	50.5%
Partially-visible refines	5126	26.7%
Partial visibility tests	9513	
Ray tests	20527	
Visibility tests	4296	
Ray tests	68736	

Table 1: Statistics for Figure 30

between them can be immediately terminated, significantly reducing the number of calls to `Refine`. If two patches become totally visible, there is no need for further visibility tests between them, although further refinement may still need to occur. Finally, if two patches are partially visible with respect to each other, further refinement would require additional visibility computation. However, as patches are subdivided, their visibility will tend to fall into the visible or invisible category, only those on the fringes remaining partial. Employing these optimizations on the scenes shown in Figure 30 cuts in half the number of rays fired in visibility tests, reducing running time by fifty percent as well.

The nature of visibility coherence, and the refinement procedure's use of this coherence, should be reflected in the visibility test. Total visibility or invisibility are the most common visibility interactions, and result in immediate pruning of computation, thus the visibility test should quickly detect these cases. Blinn calls this principle *triage* [B90]. A partial visibility result simply indicates that further visibility computation is necessary lower in the hierarchy. An precise estimate of percentage visibility is not required until refinement is terminated and the patches are to be linked. Thus, for the purposes of refinement, the visibility test need only detect partial visibility situations, not completely analyze them. Furthermore, it is acceptable to return a partial visibility result if the visibility situation is complex, as additional subdivision will tend to reduce the complexity of the visibility calculation. This is very similar to Warnock's visible surface algorithm [W69].

The methods used to detect visibility are likely to be more accurate when patches are totally visible or totally invisible. When two patches are partially visible, we assume there is more likely to be an error in visibility and increase the error in the form factor estimate. This causes increased subdivision in regions of partial visibility; the cost of this is minor because they occur so infrequently, however, the benefits are great because these often arise at shadow boundaries where there are sharp intensity gradients.

3.5 Solution Techniques

Once the form factors have been determined, the next step is to solve for the radiosities. The most efficient way to do this is to invert the matrix iteratively. Each iteration involves multiplying a matrix times a vector, which normally takes $O(n^2)$ operations. However, because the form factor matrix is represented with $O(n)$ blocks, each matrix multiplication can be done in linear time. In this section we give program fragments that implement the technique of gathering and briefly explain how to implement shooting. These techniques are quite similar to the unoccluded case, and we refer the reader to [HS90] for more details.

3.5.1 Shooting and Gathering

The classic Jacobi iteration (which differs from the Gauss-Seidel in that the brightnesses are not updated in-place) can be implemented using the following simple recursive procedure:

```
Gather(Patch *p)
{
    Patch *q;
    float Fpq;

    if (p) {
        p->Bg = 0.0;
        ForAllElements(q, p->interactions) {
            Fpq = FormFactor(p, q);
            p->Bg += Fpq * p->Cd * q->B;
        }
        Gather(p->sw);
        Gather(p->se);
        Gather(p->nw);
        Gather(p->ne);
    }
}
```

The average brightness of each patch is stored in `B` and its diffuse color is stored in `Cd`. The brightness gathered is stored in `Bg`, and is computed by receiving energy from all the patches `q` stored on the list of interactions of `p` (`p->interactions`).

The total amount of energy received by an element is the sum of the energy received by it directly, plus the sum of all the energy received by its parent subpatches. To update the energies for the next iteration, all the energy gathered is *pushed* down to the leaf nodes, and then *pulled* upward towards the root polygon. During this upward pass, the radiosity of interior subpatches are set equal to the area weighted average of its children's radiosities. Both these operations can be done in a single depth-first traversal of the quadtree, which takes time proportional to the number of nodes in the hierarchy.

The radiosity equation can be solved by shooting instead of gathering. All patches in the hierarchy are sorted into a priority queue based on their brightness. A patch at a time is taken off the queue, and its energy shot to the patches that interact with it. This version of shooting, however, has a much smaller granularity then the classic method of shooting used in progressive refinement. This is because in our algorithm each patch shoots light to a constant number of other patches, whereas in the previous algorithms a patch shoots light to the entire scene.

3.5.2 Multigridding and Radiosity Driven Refinement

An interesting variation of shooting or gathering refines the hierarchy as the iteration proceeds. This is similar to the idea of multigridding, where a finite difference equation is solved first at a coarse resolution, and then at successively finer resolutions. The advantage of multigridding is that the coarse solution involves a low resolution iteration that can be performed cheaply. This coarse solution provides a better starting point for the costlier iterations at the finer resolutions, resulting in fewer expensive iterations before convergence. Multigridding allows for an even more progressive radiosity algorithm: Shooting is performed in the early stages at coarse resolutions to get a rough idea of the image, and then at successively finer and finer resolutions as the calculation proceeds.

Multigridding is easily incorporated into the algorithm by successively refining the mesh with smaller and smaller `Feps`'s. The procedure `Refine` is extended to delete the link indicating a previous interaction at a given level of detail, if subdivision is required. `Refine` is then called between iterations to increase the resolution of the grid.

A final improvement to the algorithm bases the refinement of two patches on BF; that is, on the total amount of energy potentially transported between the patches. The procedure `Refine` is extended to use this test for subdivision rather than F alone. This causes refinement of the mesh to be put off until energy is actually available to be transported, thus saving even more work on early iterations. This works particularly well in corners which normally contain a large number of interactions because of their proximity, but tend to be dark because light must reflect off several surfaces to reach the inner recesses.

With radiosity driven refinement and multigridding, shooting has no advantage over

Figure 31: Radiosity driven refinement. Refinement is based on total energy transport between patches. Thus there are many interactions with the light sources, and a reduced number of interactions in areas exhibiting less energy transport, such as corners. White – total visibility, green – partial visibility, pink – cut by supporting plane, and dark blue – total occlusion.

gathering. Since all interactions carry approximately the same amount of energy, there is no advantage to sorting them based on brightness.

Figure 31 shows the subdivision and interactions based on a BF error criteria for the same scene as shown in Figure 30. As in the previous figure, the series shows all links, ordered by increasing height in the hierarchy. To accentuate the effect, we have made the two lamps small and very bright. Note that this causes many more interactions with the lights than between other parts of the room; all the interactions at the finest level of detail in the corners are eliminated because they are inconsequential compared to the light interactions.

Another effect clearly shown in this figure is the increased subdivision along shadow boundaries. This is exhibited in the first image of the series in the subdivision of the right

Polygons	98	
Potential elements	175964	
Potential interactions	15481576666	
Patches	5674	
Elements	4280	
Interactions	11800	
Totally-invisible	4605	39.0%
Totally-visible	4519	38.3%
Partially-visible	2676	22.7%
Tests		
Refinement tests	14149	
Totally-invisible refines	3901	27.6%
Pre-Totally-invisible refines	0	0.0%
Totally-visible refines	5414	38.3%
Pre-Totally-visible refines	4128	29.2%
Partially-visible refines	4834	34.2%
Partial visibility tests	10021	
Ray tests	53187	
Visibility tests	3545	
Ray tests	56720	

Table 2: Statistics for Figure 31

wall behind the biplane, and on the floor in the neighborhood of the desk. The second and third images in the series illustrate the corresponding links; note the spray of interactions ranging in from white (visible) to green (partial) to dark blue (occluded).

Table 2 gives basic statistics for the scene in Figure 31. The number of interactions is approximately 10,000, which is less than the number in Figure 30, even though the number of potential interactions in this scene is sixteen times greater.

Figure 32 shows a series of iterations using multigridding coupled with radiosity driven refinement. At the bottom are the induced subdivisions, and above the resulting images. The first image shows the initial mesh, the next two images show the mesh after iterations in which the error has been decreased, and the final iteration improves the final image, but does not involve decreasing the error in the mesh. Note the gradual refinement of the ceiling and right wall, as they are illuminated by the lamps and light reflected from the desktop. In the last image, enough light has been transported into the neighborhood of the near end of

Figure 32: Multigridding and radiosity driven refinement.

the desk to induce its subdivision. Table 3 gives the error bound for each iteration, as well as the number of patches, elements and interactions. Note that the last iteration in which error bound was not changed does not involve refining, and hence takes much less time than the other iterations.

Patches	Elements	Interactions	Error	Time(s)
5674	4280	11800	1.000	45
7646	5759	16216	0.707	34
10462	7871	24091	0.500	62
10714	8060	24886	0.500	10

Table 3: Statistics for Figure 32

Figure 33:

3.6 Results

Figure 33 shows an example image created by the algorithm. At the maximum level of detail, it contains potentially 52841 elements, of which 12635 patches are actually created by refinement. Using classical radiosity, this would require 1.4 billion interactions, whereas the algorithm requires only 20150. The solution shown in this image was computed in 1 minute and 11 seconds, employing a SiliconGraphics Indigo workstation with a single 50MHz R4000 processor.

3.7 Summary and Discussion

The radiosity algorithm presented in this chapter drastically reduces the number of interactions that need to be considered while maintaining the precision of the form factors that are calculated. This reduction in the number of form factors allows much higher-quality imagery to be generated within a given amount of time or memory. Successively refining the environment using a radiosity-weighted error criteria leads to an algorithm in which the granularity of each step in the progression is much smaller than in the standard progressive refinement algorithm. This allows for more control and faster updates in interactive situations.

The algorithm proposed works best for environments with relatively few large polygons with high brightness gradients that require the polygon to be broken into many elements. This is very common in architectural environments, but there are situations where this assumption is not valid. The general principles outlined in this chapter are still valid in these situations, but the methods for producing the hierarchy and estimating visibility would be quite different. Useful applications for such algorithms are for rendering volumes and participating media.

One of the emerging themes of realistic image synthesis is that the geometric aspects of the problem are becoming subservient to the optical aspects. The optical portion involves numerically solving an integral equation; the geometric portion involves primarily determining visibility between the finite elements used to discretize the equation. Unfortunately, most visibility algorithms developed in computer graphics were not developed with these numerical calculations in mind. What are needed are fast algorithms that compute visibility to within a given precision. Ideally, the less the precision, the faster the algorithm. Visibility algorithms also need to be developed that consider patch-to-patch interactions and not just point-to-patch interactions, as are almost exclusively the case. Finally, in order to exploit the visibility coherence found in typical environments, such algorithms should allow the rapid determination of complete visibility or occlusion between patches.

Chapter 4

A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection

4.1 Introduction

As discussed in Chapter 1, a major open problem in image synthesis is the efficient solution of the rendering equation. Radiosity methods have been quite successful over environments containing surfaces that exhibit only diffuse reflection. Unfortunately, very few materials are purely Lambertian reflectors, and efficient solution techniques have not yet been developed for more general specular or glossy reflection functions.

The rendering equation is an integral equation, and the solutions to complicated integral equations are generally obtained using either Monte Carlo or finite element techniques. Monte Carlo algorithms sometimes go under the name of distributed or stochastic ray tracing and are the most commonly employed in computer graphics (for examples, see [CRMT91, C86, K86, S90, WRC88]). Monte Carlo techniques have the advantage that they are easy to implement and can be used for complicated geometries and reflection functions. Unfortunately, their disadvantage is that they are notoriously inefficient. The second approach, the finite element method, has been very successfully applied to the rendering equation under the radiosity assumption, but has only begun to be employed in the general case, and with limited success. Immel et al. [ICG86] have developed a technique in which radiance is discretized into a lattice of cubical environment maps, allowing solution of the resulting system. More recently, Sillion et al. [SAWG91] employed a mesh of spherical harmonic functions to represent radiance, and solved the resulting system using a shooting algorithm.

There are many ways to parameterize the rendering equation, and each leads to a different choice of basis functions. In the transport theory community two techniques are common: directional subdivision (the method of discrete ordinates or S_N), and spherical harmonics (P_N). These two techniques roughly correspond to the methods of Immel et al. and Sillion et al., although many interesting variations are possible. Our approach is somewhat different, and based on Kajiya's original formulation of the rendering equation [K86]. Under this formulation, the rendering equation is expressed in terms of three point transport. That is, the kernel of the integral expresses the transport of light from a point on the source to a point on the receiver, via a point on a reflector. Given this formulation, the three point rendering equation can be discretized over pairs of elements to form a linear system of equations. Solving this system yields the radiance transported between elements. Note that this approach is very similar to the radiosity formulation.

The problem with finite element methods is that the matrix of interactions is very large for interesting environments. For a given environment of k input polygonal patches containing n elements at the finest level of refinement, the three point discretization that we are proposing generates an n^3 matrix of interactions. However, in this chapter we show that we may accurately approximate the n^3 reflectance matrix with $O(n + k^3)$ blocks, in a way very similar to the hierarchical radiosity algorithm presented in Chapter 3. Although the results presented in this chapter are preliminary, we believe a hierarchical finite element approach along these lines will ultimately lead to a fast, efficient algorithm.

In the following section we describe our application of the finite element element method to the three point rendering equation, yielding a radiance formulation for discrete transport. In Section 4.3 we present a simple adaptive refinement algorithm for computation over this formulation, and the iterative solution technique employed for the actual calculation of transport. In Section 4.4 we discuss our implementation of the algorithm over glossy reflection, and in Section 4.5 we present some experiments and results. Appendix A to this dissertation contains details of our error analysis for discrete transport under the glossy model.

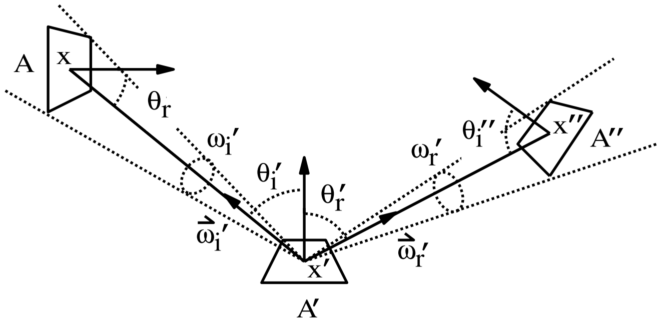


Figure 34: Geometry of Reflection

4.2 Discrete Three Point Transport

The algorithm presented in this chapter operates through two functions: refinement of the environment to form a hierarchy of discrete interactions, patches and elements, and the actual computation of illumination over this hierarchy.

In this section we develop the basis for both discretization and transport. We derive a radiance formulation for three point transport, and a new measure and description of reflectance, *area reflectance*. This radiance formulation and associated reflectance provide a natural criterion for discretization under illumination and reflection, and allow both the computation of radiance across triples of individual surface elements, and the expression and computation of all light transport over all surfaces.

4.2.1 A Radiance Formulation for Three Point Transport

When computing and rendering illumination within an environment, we are interested in the transport of light from surface to surface — it is this interaction of surfaces that characterizes illumination, in the absence of participatory media. Reflection within an environment may thus be naturally expressed over triples of surfaces. Consider surfaces A, A', and A'' (Figure 34) — we will examine the transport of light incident at A' originating at A and reflected toward A''.

Let ω_i' and ω_r' be the solid angles subtended at point x' by A and A'', respectively. Consider differential solid angles at $\vec{\omega}_i'$ and $\vec{\omega}_r'$ — by definition of the bidirectional reflectance-distribution function (BRDF), f_r [NRHGL77], the radiance $L(\vec{\omega}_r')$ along $\vec{\omega}_r'$ due to illumination through solid angle ω_i' is:

$$L(\vec{\omega}_r') = \int_{\omega_i'} f_r(\vec{\omega}_i', \vec{\omega}_r') L(\vec{\omega}_i') \cos\theta_i' d\omega_i'$$

Integrating this expression over ω_r', and introducing $\cos\theta_r'$, we have:

$$\int_{\omega_r'} L(\vec{\omega}_r') \cos\theta_r' d\omega_r' = \int_{\omega_r'} \int_{\omega_i'} f_r(\vec{\omega}_i', \vec{\omega}_r') L(\vec{\omega}_i') \cos\theta_i' \cos\theta_r' d\omega_i' d\omega_r'$$

We may then reparameterize over A and A'' to yield:

$$\begin{aligned}&\int_{A''} L(x', x'') G(x', x'') dx'' = \\ &\quad \int_A \int_{A''} f_r(x, x', x'') L(x, x') G(x, x') G(x', x'') dx'' dx\end{aligned}$$

where:

$$G(x, x') = \frac{\cos\theta_r \cos\theta_i'}{|x - x'|^2} v(x, x')$$

where $v(x, x')$ is 1 if points x, x', are mutually visible, and 0 otherwise. Note that G is very similar to a differential form factor.

We integrate over A', thus introducing all three areas into the formulation:

$$\begin{aligned}&\int_{A'} \int_{A''} L(x', x'') G(x', x'') dx'' dx' = \\ &\quad \int_A \int_{A'} \int_{A''} f_r(x, x', x'') L(x, x') G(x, x') G(x', x'') dx'' dx' dx\end{aligned} \tag{8}$$

We may now rewrite the equation in discrete form. Let A_j and A_k be subareas of A' and A'' such that $L(x', x'')$ is nearly constant over their surfaces. The left side of equation (8) may then be rewritten, bringing radiance out of the integral as L_{jk}:

$$L_{jk} \int_{A_j} \int_{A_k} G(x', x'') dx'' dx' = \pi L_{jk} A_j F_{jk}$$

by definition of the diffuse form factor, F_{jk}.

We may similarly discretize A as A_i, and rewrite the right side of equation (8) as:

$$L_{ij} \int_{A_i} \int_{A_j} \int_{A_k} f_r(x, x', x'') G(x, x') G(x', x'') dx'' dx' dx = \sum_i \pi L_{ij} A_i F_{ij} R_{ijk}$$

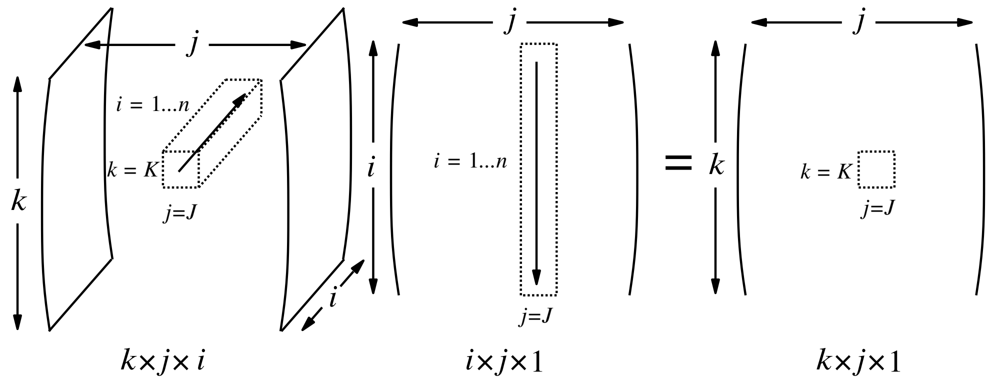


Figure 35: A Reflection Product

where R_{ijk} is defined such that:

$$\pi A_i F_{ij} R_{ijk} = \int_{A_i} \int_{A_j} \int_{A_k} f_r(x, x', x'') G(x, x') G(x', x'') dx'' dx' dx \tag{9}$$

Note that, by the symmetry of f_r and G:

$$A_i F_{ij} R_{ijk} = A_k F_{kj} R_{kji}$$

We thus have:

$$\begin{aligned} \pi L_{jk} A_j F_{jk} &= \pi \sum_i L_{ij} A_k F_{kj} R_{kji} \\ &= \pi \sum_i L_{ij} A_j F_{jk} R_{kji} \end{aligned}$$

by the reciprocity of form factors, and thus:

$$L_{jk} = \sum_i L_{ij} R_{kji}$$

The three dimensional character of R_{kji} over indices i, j, k leads naturally to a three dimensional matrix formulation for the above system. Consider a product over an $n \times n \times n$ R_{kji} "matrix" and an $n \times n \times 1$ L_{ij} matrix producing an $n \times n \times 1$ matrix of reflected radiances, as shown in Figure 35. Note that the R_{kji} matrix is of size $O(n^3)$ — the hierarchical method discussed in subsequent sections of this chapter addresses more tractable representation of this matrix.

Taking into account emission, we have derived a radiance formulation for three point transport:

$$L_{jk} = E_{jk} + \sum_i L_{ij} R_{kji} \tag{10}$$

This formulation states that:

> *The radiance at Area j in the direction of Area k is equal to the radiance emitted by j in the direction of k, plus, for every Area i, the radiance at i in the direction of j multiplied by the area reflectance R_{kji}.*

Note that equation (10) is very similar to the radiosity formulation:

$$B_j = E_j + \rho_j \sum_i B_i F_{ji}$$

4.2.2 Area Reflectance

The quantity R_{kji} has a natural and satisfying physical significance — it is an expression of reflectance over areas A_i, A_j, and A_k.

Consider the fraction of the radiant flux transported from A_i incident to A_j that is reflected in the direction of area A_k:

$$\frac{\int_{A_i}\int_{A_j}\int_{A_k} f_r(x,x',x'')L(x,x')G(x,x')G(x',x'')dx''dx'dx}{\int_{A_i}\int_{A_j} L(x,x')G(x,x')dx'dx}$$

If we assume that incident radiance is uniform and isotropic over both ω_i' (as induced by A_i) and A_j, we may divide through by $L(x,x')$, yielding:

$$\rho(A_i, A_j, A_k) \equiv \frac{\int_{A_i}\int_{A_j}\int_{A_k} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx}{\int_{A_i}\int_{A_j} G(x,x')dx'dx}$$

We define $\rho(A_i, A_j, A_k)$ to be *area reflectance*. Note that area reflectance is similar to biconical reflectance [NRHGL77], save that it is also integrated over the reflecting surface.

By definition of R_{ijk}:

$$R_{ijk} = \rho(A_i, A_j, A_k)$$

Conservation of energy over reflection, and the reciprocity relation derived for R_{ijk} above, constitute fundamental properties of area reflectance:

1. $\sum_k R_{ijk} \leq 1$, for fixed i, j.
2. $A_i F_{ij} R_{ijk} = A_k F_{kj} R_{kji}$.

where equality is achieved in property 1 over complete enclosures and perfect reflectivity.

Note that area reflectance and the form factor play corresponding roles under discrete three point transport and radiosity:

F_{ji} — the fraction of the radiant flux leaving A_j that arrives at A_i.

R_{kji} — the fraction of the radiant flux transported from A_k to A_j reflected toward A_i.

4.2.3 Evaluation of R_{kji}

In this section we examine the evaluation of R_{kji} over given patches A_i, A_j, A_k.

Recall:

$$R_{kji} = \frac{\int_{A_k}\int_{A_j}\int_{A_i} f_r(x'', x', x) G(x'', x') G(x', x) dx dx' dx''}{\int_{A_k}\int_{A_j} G(x'', x') dx' dx''}$$

We assume that discrete areas A_i, A_j, A_k are of small enough scale that f_r and G are relatively constant over their surfaces. Then:

$$\begin{aligned} R_{kji} &= \frac{S_{kji} G_{kj} G_{ji} A_k A_j A_i}{G_{kj} A_k A_j} \\ &= S_{kji} G_{ji} A_i \end{aligned}$$

where S is the discretized value of f_r, $S_{ijk} = S_{kji} = S_{x_k x_j x_i}$.

Note that the average value of $G(x', x)$ over A_i and A_j is $\pi F_{ji}/A_i$ — we thus estimate $G_{ji} A_i$ by πF_{ji}, and compute R_{kji} as:

$$R_{kji} = \pi F_{ji} S_{kji}$$

In practice, it will not be possible to compute the exact values of F_{ji} and S_{kji} over A_i, A_j, A_k. We assume that we are able to estimate these values, along with error bounds for each estimation. Let ΔF_{ji} and ΔS_{kji} be error estimates for computed F_{ji} and S_{kji}, respectively. We then have an estimate for area reflectance in the form:

$$\begin{aligned} R_{kji} &= \pi(F_{ji} + \Delta F_{ji})(S_{kji} + \Delta S_{kji}) \\ &= \pi(F_{ji} S_{kji} + \Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji} + \Delta F_{ji} \Delta S_{kji}) \\ &\approx \pi(F_{ji} S_{kji} + \Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji}) \end{aligned}$$

Assuming $\Delta F_{ji} < F_{ji}$, $\Delta S_{kji} < S_{kji}$, we have neglected the last term and estimate the error in R_{kji} as $\pi(\Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji})$.

In general, and as is shown for glossy reflection in Section 4.4, the accuracy of estimators for F_{ji} and S_{kji} is dependent on the size of the patches over which reflectance is computed, relative to their distance apart. As relative size decreases, so does error in computation, leading directly to the adaptive refinement strategy for illumination presented in Section 4.3 below.

4.3 Algorithms for Three Point Transport

4.3.1 Introduction

Recall equation (10):

$$L_{jk} = E_{jk} + \sum_i L_{ij} R_{kji}$$

This equation suggests both a solution strategy for radiance under three point transport, and a natural representation for illumination within the solution system.

We may interpret equation (10) as a gathering iteration similar to that employed for radiosity under diffuse reflection: the radiance L_{jk} at patch A_j in the direction of patch A_k is found by gathering radiances L_{ij} in the direction of A_j at patches A_i (Figure 36). We may solve for transport by gathering radiance for each L_{jk}, and successively iterating to capture all significant re-reflection.

We are left with the question of what structure we are gathering over and iterating upon. Note that all illumination is expressed as the radiance at a given patch in the direction of another — it is these patch-patch interactions that form the primary structure within the solution system. All operation is over interactions: both the representation and transport of radiance, and the iteration and solution for illumination.

Consider the following structure:

```
typedef struct _interaction {

    Patch *from;
    Patch *to;

    Color  L;
    Color  Lg;

    List  *gather;

    struct _interaction *nw, *sw, *se, *ne;

} Interaction;
```

A given interaction ij is defined by two patches `ij->from` and `ij->to`, and represents the radiance at `from` in the direction of `to`. This radiance is stored within the interaction as attribute `L`. `Lg` is radiance gathered during the current solution iteration from interactions contained in the list `gather`. Subinteractions `nw`, `sw`, `se`, `ne` are the children of ij, induced by subdivision over either `from` or `to`. The structure assumes quadtree refinement, leaving northwest, southwest, southeast, and northeast descendents (Figure 37).

In the following sections we will present an algorithm for the refinement and computation of illumination over a hierarchy of interactions. The algorithm will operate by refining pairs of interactions ij, jk (such that `ij->to == jk->from`), to ensure that computed reflectance across the interaction pairs, and associated patch triples, satisfies user specified error bounds. If a given interaction pair ij, jk is satisfactory, the interactions are linked to record that radiance may be gathered from ij to jk, otherwise one or both interactions are subdivided and refinement applied to their descendents.

After refinement, a gathering iteration may be carried out, each interaction gathering radiance from interactions to which it has been linked. The gathered radiances are then distributed within each receiving interaction hierarchy, and subsequent iterations computed until satisfactory convergence has been achieved.

Note that, within this system, the eye may be regarded as simply another object with which patches may interact. The radiance along interactions to the eye provides the resulting view.

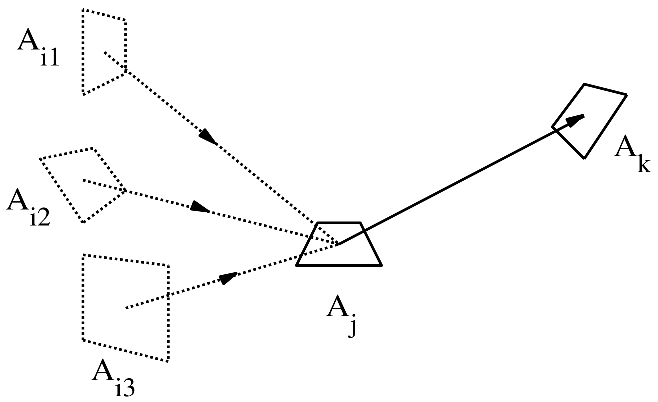


Figure 36: Gathering over Interactions

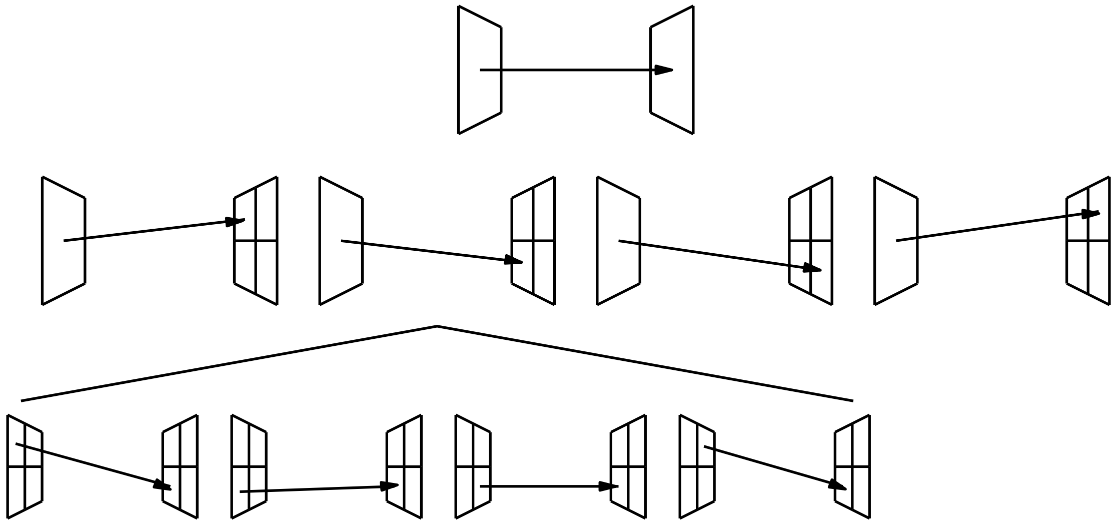


Figure 37: An Interaction Hierarchy

4.3.2 Adaptive Refinement

Consider the following procedure:

```
Refine(Interaction *ij, Interaction *jk,
       float Feps, float Seps, float Aeps)
{
    float feps, seps;

    feps = GeometryErrorEstimate(ij);
    seps = ReflectionErrorEstimate(ij, jk);

    if (feps < Feps && seps < Seps)
        Link(ij, jk);

    else if (seps >= Seps) {

        switch(SubdivS(ij, jk, Aeps)) {

        case PATCH_I:

            Refine(ij->nw, jk, Seps, Feps, Aeps);
            Refine(ij->sw, jk, Seps, Feps, Aeps);
            Refine(ij->se, jk, Seps, Feps, Aeps);
            Refine(ij->ne, jk, Seps, Feps, Aeps);

            break;

        case PATCH_J:
            /* refine over children of ij and jk */
        case PATCH_K:
            /* refine over children of jk        */
        case NONE:
            Link(ij, jk);
        }
    }

    else {    /* feps >= Feps */

        switch(SubdivG(ij, jk, Aeps)) {

        /* refine over children, or link, as    */
        /* directed by PATCH_I, J, K, or NONE.  */

        }
    }
}
```

This procedure computes over pairs of interactions, and associated patch triples, subdividing and recursively refining if estimated error exceeds user specified bounds, linking the interactions for gathering if the bounds are satisfied, or if no further subdivision is possible. `Feps` and `Seps` are the bounds for geometric and reflection error, respectively; `Aeps` specifies the minimum area a patch may possess and still be subdivided. `GeometryErrorEstimate` and `ReflectionErrorEstimate` provide estimations for $\pi \Delta F_{ji} S_{kji}$ and $\pi \Delta S_{kji} F_{ji}$.

`SubdivS` and `SubdivG` control refinement for reflection and geometry error, respectively. Both routines select a patch for refinement, subdividing the patch and associated interaction(s) if required. An identifier for the selected patch is returned — if no patch may be subdivided, then `NONE` is passed back. Note that a given interaction/patch may be refined against many different interactions within the system, and thus may have already been subdivided when selected by a `Subdiv` routine — in this case, the routine simply returns the proper identifier.

The `Subdiv` routines should select for refinement patches that are of large size relative to their distance from their partner(s) in the transport triple. Form factor estimation is a convenient criterion for the determination of such patches — a large differential to area form factor F_{dpq} indicates that patch q is of large relative size. Care must be taken in subdivision, however, to ensure that each interaction is always subdivided in the same way for all refinements involving that interaction.

The `Subdiv` routines thus choose for refinement a patch that is of large form factor within ij and/or jk, always selecting for subdivision the same patch over any given interaction. If patch p_j is of greatest form factor over both ij and jk, and of area greater than `Aeps`, then it is chosen for refinement (Figure 38 at middle). Otherwise, if p_j is selected over one interaction, but p_i or p_k is selected over the other, then the "outside" patch is chosen for refinement. Given two selected outside patches, `SubdivS` selects the one of greater form factor relative to p_j; `SubdivG` selects p_i over p_k, as p_k has no direct effect on geometric accuracy. Note, however, that even under `SubdivG`, if only p_j and p_k are allowed subdivision, p_k will be selected, although with further subdivision the triple will eventually balance sufficiently to allow refinement over p_j.

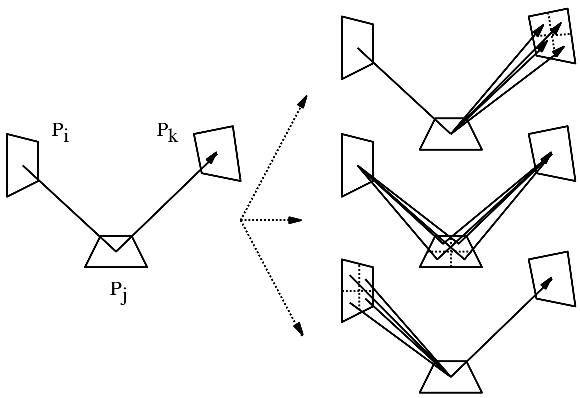


Figure 38: Refinement and Subdivision

4.3.3 Gathering Radiance

Gathering radiance over interactions may be written as a simple procedure:

```
Gather(Interaction *jk)
{
    Interaction *ij;

    if (jk) {

        jk->Lg = 0;

        ForAllElements(ij, jk->gather)
            jk->Lg += ij->L * Reflectance(ij, jk);

        Gather(jk->nw);
        Gather(jk->sw);
        Gather(jk->se);
        Gather(jk->ne);
    }
}
```

We gather radiance into `jk->Lg` rather than directly into `jk->L` to avoid the necessity of a push/pull with every invocation of the procedure (see Section 4.3.4). The solution method is thus simple Jacobi iteration, as opposed to Gauss-Seidel, as the hierarchical structure imposes simultaneous rather than successive displacement.

4.3.4 Radiance within a Hierarchy

A gathering iteration results in received radiance scattered throughout each interaction hierarchy. This gathered radiance must be distributed and accounted for over all ancestors and descendents of each receiving interaction, in order to maintain the consistency and correctness of the hierarchical representation of radiance between patches.

We employ a distribution algorithm similar to that presented in Chapter 3 for radiosity over patch/element hierarchies: gathered radiance is "pushed" to the leaf interactions within each hierarchy to ensure propagation to all descendents, and then "pulled" and distributed back up from the leaves through all higher level interactions to their common ancestor at the root. To correctly implement this algorithm, we must understand how the radiance at a given interaction is related to the radiance at its parent and at its children.

Consider a transport interaction from patch A_j to patch A_k. Under equation (10), the radiance at A_j in the direction of A_k is determined by area reflectance R_{kji}. The dependence of R_{kji} upon subdivision of A_j and A_k thus determines the distribution of received radiance within the interaction hierarchy.

By definition of area reflectance, we have:

$$\pi A_k F_{kj} R_{kji} = \int_{A_i}\int_{A_j}\int_{A_k} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx$$

Consider subdivision of A_k, yielding children $A_{k_1}, A_{k_2}, \ldots, A_{k_c}$. We may rewrite the above expression for R_{kji} as:

$$\begin{aligned}
\pi A_k F_{kj} R_{kji} &= \int_{A_i}\int_{A_j}\int_{A_k} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx \\
&= \int_{A_i}\int_{A_j}\int_{A_{k_1}} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx \\
&\quad + \cdots + \int_{A_i}\int_{A_j}\int_{A_{k_c}} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx \\
&= \pi A_{k_1} F_{k_1 j} R_{k_1 ji} + \cdots + \pi A_{k_c} F_{k_c j} R_{k_c ji}
\end{aligned}$$

Under the subdivision criteria described in Section 4.3.2, we may assume that if the interaction from A_j to A_k is employed in transport, then areas A_j and A_k are well separated, and thus [SAS92]:

$$F_{kj} \approx F_{k_d j}, \text{ for } d = 1 \ldots c$$

We then divide through by $\pi A_k F_{kj}$, yielding:

$$R_{kji} = \frac{A_{k_1}}{A_k} R_{k_1 ji} + \cdots + \frac{A_{k_c}}{A_k} R_{k_c ji}$$

Furthermore, as $A_j F_{jk} R_{kji} = A_k F_{kj} R_{kji}$, the above argument may be applied over subdivision of A_j as well.

Thus, given distribution by area reflectance, the radiance at a parent interaction is simply the area average of the radiance at its children, under subdivision of either A_k or A_j.

In a similar fashion, it may be shown that radiance received at a given parent interaction may be pushed unchanged to its children under subdivision of either of its defining patches.

4.4 Application over Glossy Reflection

In this section we discuss our implementation of the above algorithms over glossy reflection.

4.4.1 The Reflection Function

As discussed in Section 2.3, we employ a highly simplified Torrance-Sparrow [TS67] model for our glossy reflection function:

$$f_g(\vec{\omega}_i, \vec{\omega}_r) = \frac{\kappa + 2}{8\pi} \frac{\cos^\kappa \theta_m}{\cos \theta_i \cos \theta_r} S_h(\theta_i, \theta_r)$$

This function incorporates the facet distribution function $\cos^\kappa \theta_m$ developed by Blinn [B77], normalized for projected facet area under [M92]. Angle θ_m is that made to the mean surface normal by $\vec{\omega}_m$, the microfacet mirror orientation normal lying halfway between $\vec{\omega}_i$ and $\vec{\omega}_r$.

Function $S_h(\theta_i, \theta_r)$ expresses self shadowing over microfacets — for near specular surfaces, such self shadowing or masking does not become critical until relatively high θ_i or θ_r (see Appendix B). The implemented system thus simply clamps S_h from 1 to 0 when θ_i or θ_r exceeds a preset θ_c near the horizon. This scheme serves as a crude approximation to the shadowing function; however, a better strategy would be to employ a much fuller tabulation of the function, incorporated into the error analysis presented below. A more complete discussion of shadowing and conservation of energy over f_g is presented in Appendix B.

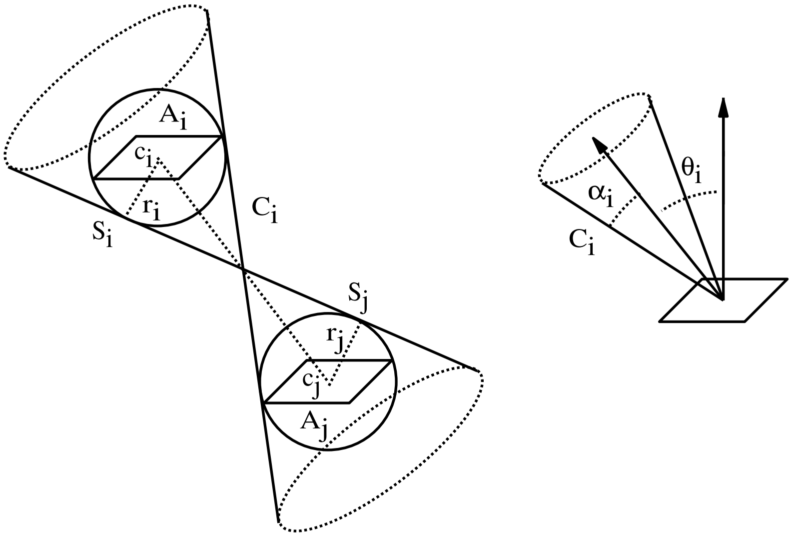


Figure 39: Estimating Cones

4.4.2 Error Estimation

Recall the general expression for error derived in Section 4.2.3:

$$\pi(\Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji})$$

In implementation we have estimated the form factor F_{ji} by F_{dji}, the form factor from a differential area at A_j to a disk of area A_i centered at A_i, as was employed in Chapter 3. As discussed in Chapter 3, the relative error in this estimate is proportional to the estimate itself. In our implementation we have thus estimated absolute error ΔF_{ji} as at most proportional to F_{dji}^2. A brief discussion of relative and absolute error over hierarchical methods is presented in Appendix A.

We now consider the error estimate ΔS_{kji}. As discussed in Appendix A to this dissertation, we may compute bounding cones C_i, C_r, and C_m over all possible incident, reflected, and mirror orientation directions induced at A_j by A_i and A_k (Figures 39 and 40 — these figures are discussed more fully in Appendix A). We may then compute maximum and minimum $\cos^\kappa \theta_m$, $\cos \theta_i$, $\cos \theta_r$ over these cones, and estimate error by interval width. The full expression for estimated error over transport is given in Appendix A.

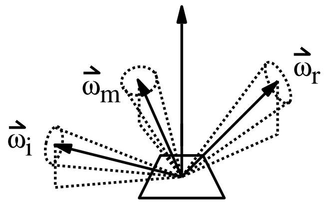


Figure 40: C_i, C_r, and C_m

4.4.3 Clamping and Visibility

Evaluation of glossy reflectance over three surface areas, as required by the gather iteration, may be difficult, particularly if surface subdivision has been limited by `Aeps` rather than satisfaction of error bounds, and if κ, the facet distribution exponent, has high value. In this case we must estimate the integral of a spikey function over a relatively broad area.

Our solution is to band limit the BRDF in a fashion similar to that presented by Amanatides [A92]. We employ the cone estimation techniques of the previous section to determine if the BRDF varies significantly over the given patches — if this variance exceeds a set bound, we "roughen" the reflecting surface, lowering κ to broaden the resulting reflection over the estimated cones. We then renormalize the resulting blurred function, as described in [A92], to prevent amplification of its low frequency components. The resulting antialiasing is relatively aggressive, significantly dimming or eliminating reflections requiring overmuch blurring. Note that transport computation under κ may thus depend on the fineness of meshing.

In implementation, we have computed visibility via jittered ray casting and inheritance similar to that of Chapter 3, storing visibility data in interactions as it is computed.

4.5 Results

4.5.1 Growth in Transport

We have measured the growth in transport triples (linked interactions) versus n, the maximum number of elements at the finest level of subdivision, over parallel, perpendicular, and "oriented" patches (Figure 41). The corresponding geometries are shown in Figure 42.

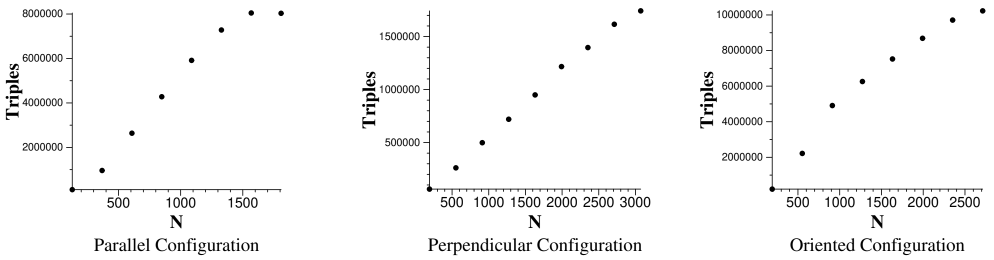


Figure 41: Triples vs. N over Geometry. Error bounds $e = 0.1$. Glossy exponent κ = 25. For oriented case $e = 0.005$.

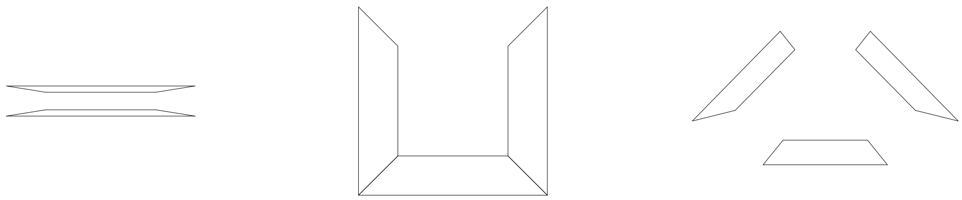


Figure 42: Geometric Configurations

The graphs show linear or near linear behavior over each range — the graph of triples vs. n for the perpendicular case is slightly concave over the lower data points, but subsides to linear with further refinement.

In Chapter 3, with respect to hierarchical refinement for radiosity, it was shown that for error estimate proportional to F_{dji}, and sufficient refinement, each subpatch may only interact with other patches in a limited local neighborhood. Each patch may thus participate in at most c interactions, for some constant c independent of n and k. Adaptive refinement thus generates at most $O(n)$ transport interactions. We will show a similar bound for discrete three point transport under glossy reflection.

Recall that the estimate for error in computed transport is proportional to

$$\Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji}$$

Our argument depends on two assumptions:

1. We may bound both ΔS_{kji} and S_{kji} by some S_{max}.

   As discussed below, the lower this S_{max}, the smaller the magnitude of the leading coefficient underlying the resulting bound.

Note that our argument thus does not apply to perfect specular reflection, as the corresponding BRDF incorporates the Dirac delta function [NRHGL77]. Equivalently, the argument does not hold over f_g for $\kappa = \infty$ (inducing mirror reflection), as we can not provide a finite bound for S in this case.

For finite κ, however, the desired bound over glossy reflection is achieved by:

$$\frac{\kappa + 2}{8} \max(\cos^\kappa \theta_m) \max(\sec \theta_i) \max(\sec \theta_r)$$

The maxima over the secant terms are bounded by microfacet self shadowing.

2. ΔF_{ji} and F_{ji} within our error estimate are at most proportional to F_{dji}.

   Recall that we estimate F_{ji} as F_{dji}, and ΔF_{ji} as F_{dji}^2, thus satisfying this assumption.

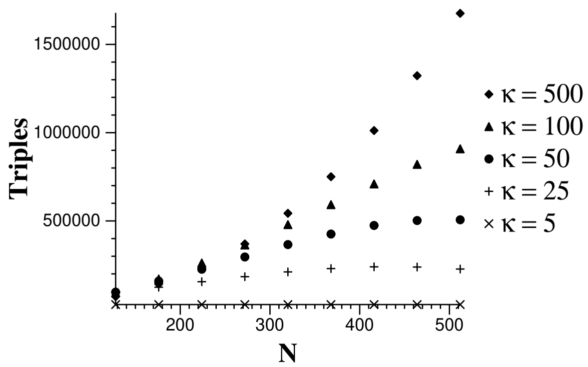
Given assumptions 1 and 2, estimated error in individual transport computation is at most proportional to $S_{\max} F_{dji}$.

We may now show $O(n)$ growth, for sufficient refinement. Consider refinement over interaction ij under an error estimate at worst proportional to $S_{\max} F_{dji}$. The error estimate is thus proportional to F_{dji}, and therefore, for sufficient refinement, there are at most $O(n)$ such interactions, as discussed in Chapter 3.

Consider now an error satisfied link from ij to an interaction jk. For sufficient refinement under our subdivision scheme, we may assume that form factors F_{ij}, F_{ji}, F_{jk}, F_{kj} over A_i, A_j, and A_k are roughly equal. Furthermore, these satisfying form factors depend only on the error estimate, reflection function, and error bounds, not on n or k.

At worst the above form factors are such that $F_{**} S_{\max} < \texttt{Eps}$, where `Eps` is the most restrictive error bound. Note that, as stated above, F_{**} depends only on the error estimate, reflection function (ie. $S_{\max}$), and error bounds. Only some constant number of such form factors may be fitted over the directional hemisphere above A_j, and thus ij may only be linked to some constant number of interactions jk. The total number of linked interactions, and corresponding transport triples, is thus $O(n)$.

Note that the above argument, although it establishes the desired bound, may overstate the potential for links at a given interaction. For a given ij, much of the directional reflection into the hemisphere over A_j may not achieve $S_{\max}$, and may even be of maximum 0. That



Specular Configurations

Figure 43: Triples vs. N over κ. The graph is over parallel polygons for which the error bounds and interpolygon distance have been doubled.

is, the analysis ignores the modulation between the paired error and value terms within the error estimate.

As κ increases in magnitude, the corresponding bound $S_{\max}$ must increase as well. We may thus expect greater growth in transport computation with higher specular exponent, as shown in Figure 43. Within this graph, growth is superlinear for $\kappa = 500$, though further trials over a higher range of $n = 500 \ldots 2000$ have shown that the rate subsides to linear as n increases, allowing sufficient refinement for the local neighborhood property to obtain.

Finally, we note that under specular reflection each element is reflected across every other element perfectly, and to a first approximation is visible from a constant number of other elements in the environment (at least in the case of a convex enclosed room; the analysis is complicated by occlusion and certain worst case alignments). Thus, the number of interactions is at least $O(n^2)$ — we conjecture that it is no worse than this bound.

4.5.2 Illumination and Refinement

Figure 44 shows illumination and meshing over surfaces of varying glossiness (specular exponent). Within each image, the reflecting surface is perpendicular to the diamond shaped light source, and we see the resulting reflection in the direction of the eye. Note the conformation of meshing to the highlight over each surface. The "stretched" nature of the highlight along the axis to the eye is characteristic of Torrance-Sparrow reflection over

Figure 44: Illumination and Refinement

Figure 45: Eye and Offset Views

Figure 46: Meshing for glossy and diffuse reflection

fairly oblique angles, and accounts for the increased sensitivity of meshing along this axis. The rightmost three images in the figure show the meshing from above. The illumination shown in these images is somewhat unusual — it shows the reflection to the eye as though it had been painted on the reflecting surface, and then viewed from a different location, directly above. The images in Figure 45 show similar eye/offset views for the reflection of a garish checkerboard.

The image in Figure 46 shows contrasting illumination and meshing induced by diffuse and glossy reflection. Note the distinct meshing for each highlight. Glossy reflection is at a less oblique angle, and thus both the highlight and meshing exhibit less distortion in the

Figure 47: Cube and Slabs

direction of the eye.

Note that these scenes are extremely simple — application to more complex environments is still very expensive, despite the employment of hierarchical methods. Motivated by the work of Smits et al. [SAS92] in hierarchical radiosity, we are currently experimenting with importance and radiance weighting over three point transport — preliminary results of this work are shown in Figure 47, and are discussed in the following chapter. The given environment contains four reflectors: the broad face of each of the three "slabs" and the top of the central cube. In addition to the reflections seen in the slabs, note the play of light originating at the lamp at left, reflected off the cube top, and over the upper part of the green wall at right. Total potential transport triples over this environment at the finest level of subdivision is just over 222 billion — our system, under importance and radiance weighting, employs 70,995, a reduction to 3 hundred-thousandths of 1 percent.

Table 4 provides further statistics for the images. Timings are given for a Silicon-Graphics Indigo workstation with a single 50 MHz R4000 processor. The image shown in Figure 47 was generated after seven complete iterations (gathers to all interactions), and total time just over three minutes.

Figure 44				
Max Elements		4160		
Max Triples		262144		
Computed	patches	elements	triples	time
κ = 25	790	593	6706 (2.6%)	2.2s
κ = 100	1290	968	24214 (9.2%)	8.0s
κ = 500	874	656	12106 (4.6%)	4.1s
Figure 46				
Max Elements		16448		
Max Triples		1048576		
Computed	patches	elements	triples	time
κ = 500	1578	1184	7834 (0.75%)	5.0s
Figure 47				
Max Elements		15138		
Max Triples		222385209344		
Computed	patches	elements	triples	time
κ = 500	6479	4866	70995 (0.00003%)	3m13s

Table 4: Image Statistics

4.6 Discussion

Recall the matrix formulation shown in Figure 35. For any n of reasonable size, the resulting n^3 matrix will be unmanageable — we have shown, however, that for sufficient refinement the n^3 entries in the matrix may be approximated to within user specified bounds by $O(n)$ subblocks. The gather and push/pull procedures described in preceding sections allow manipulation and solution over this representation. As discussed in the following chapter, the resulting system may be shown to converge.

Growth in transport is more accurately described as $O(n + k^3)$, where k is the number of input polygonal patches within the environment, as opposed to elements. The k^3 term is generated by the initial examination of all polygon triples for reflection, and is subsumed by n as the number of elements increases. As the number of polygons in an environment grows, however, the k^3 term will become prohibitively large.

As discussed in [SAS92] with respect to the related problem under hierarchical radiosity, the capability to cluster as well as refine polygons would reduce the difficulty of unnecessary initial interactions. Employing nested bounding spheres over groups of polygons, and the resulting cone tests, would represent a partial approach to this problem — such clustering would not be viable for the actual computation of transport, however, only for the elimination of triples for which transport is not possible. Clustering is arguably the most important open problem in the computation of global illumination.

The hierarchical approach described in this chapter was derived by writing the rendering equation in a three point transport formulation. Another option would be to parameterize radiance by position and direction – we believe that a similar hierarchical approach could be employed with the method of discrete ordinates or spherical harmonics.

The current technique for estimation of error in area reflectance requires relatively complex geometric arguments combined with interval techniques over a fairly unsophisticated kernel. As discussed in [GSCH93], wavelet theory provides a very general means for error estimation when approximating any smooth kernel, and has great potential for employment in hierarchical algorithms for more complex reflection functions.

Finally, we note that, similarly to other algorithms for hierarchical illumination [HSA91, SAS92], the algorithm described in this chapter bounds estimated error over individual transport computations. As discussed in [SAS92], bounding estimated error over individual transport does not easily or necessarily provide a rigorous bound for overall error in the solution. An analysis and means of computing such a bound over hierarchical illumination remains an interesting open problem.

Chapter 5

Importance and Discrete Three Point Transport

5.1 Introduction

Global illumination within nondiffuse environments is ideally suited for computation under importance and radiance driven hierarchical refinement: a transport interaction is significant to the view from the eye only if it lies within paths of directional reflection of both radiance originating at a light source, and importance originating at the eye.

In the previous chapter, we developed a hierarchical algorithm for global illumination based on a radiance formulation for discrete three point transport; we now derive the adjoint to our transport formulation, and discuss preliminary results of the application of this adjoint in the form of an importance driven version of our glossy implementation. These results show a significant reduction in transport computation, and indicate that an importance and radiance driven hierarchical technique possesses great promise for the development of efficient global illumination algorithms for general reflection.

In the following section we will provide a brief overview of previous work examining importance and hierarchical illumination. In Section 5.3 we derive the adjoint to our transport formulation, and discuss interpretation of the resulting equation. In Section 5.4 we describe the application of the adjoint to our algorithm, and derive the distribution of importance within transport interaction hierarchies. Finally, in Section 5.5 we discuss preliminary results of importance and radiance driven computation over glossy reflection.

Our derivation of the adjoint and discussion of importance is specifically oriented toward discrete three point transport, and in particular to the iterative solution technique we employ for solution over this transport. A very interesting and more general discussion of the adjoint formulation for transport is presented by Christensen et al. [CSD93] in a paper appearing, along with the original publication of this work [AH93b], in the Fourth EuroGraphics Workshop on Rendering (1993).

5.2 Importance and Hierarchical Illumination

The brief sketch provided here is based on the excellent discussion presented in [SAS92].

The matrix formulation given for radiosity in Section 2.2.4 is a discretized form of a more general transport equation:

$$\int T(x, y)L(x)dx = E(y) \tag{11}$$

where T is the transport kernel, L the transported quantity (for which the system is to be solved), and E initial emission. Under radiosity we have $T = I_d - F$, where I_d is the identity operator, and F is a continuous version of the form factor relation. Similarly, as shown below, under discrete three point transport we have $T = I_d - R$.

We may rewrite equation (11) as:

$$TL = E$$

where T is the integral operator inducing the above relation.

The actual generation of an image requires the application of some view function v over L to compute the scene as seen from the eye. That is, given a solution for light transport, we apply v to determine the contribution of solved illumination to the view at the eye; for example, v may compute the average radiance seen through a given pixel or other aperture under perspective or orthogonal projection [SAS92]. We assume we may express v as an inner product over L and some function N:

$$v(L) = \langle N, L\rangle = \int N(x)L(x)dx$$

Just as E specifies the emittance of the transport quantity from (light) sources, N may be understood as a specification of emittance of some new quantity from the eye. We may

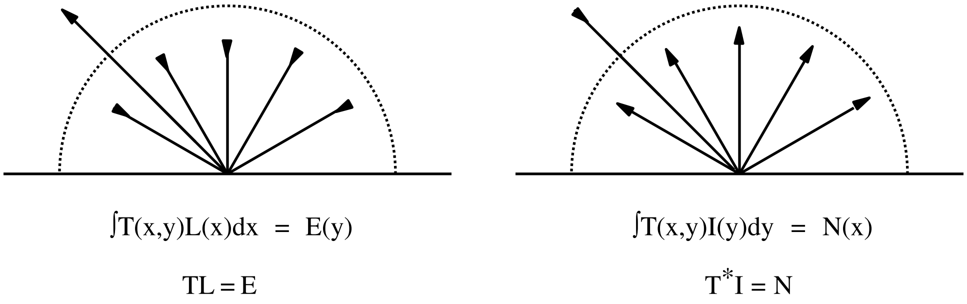


Figure 48: Transport and Adjoint Formulation

write an equation for the transport of this quantity, dual to equation (11) above:

$$\int T(x,y)I(y)dy = N(x) \tag{12}$$

This equation is referred to as the adjoint transport equation, and may be written as:

$$T^*I = N$$

where T^* is the operator adjoint to T.

Equations (11) and (12) express forward and backward duality over transport (Figure 48). Under equation (11), the transport quantity (in image synthesis "light", for example, radiosity or radiance) is expressed forward from x to y: gathered over the domain of integration, each $L(x)$ is transported via $T(x,y)$ into y, and is solved relative to initial emission E. In contrast, transport of quantity I is expressed in "reverse", each $L(y)$ under the domain of integration is transported backward via $T(x,y)$ into a given x, and is solved relative to the quantity N determined by v at the eye.

Furthermore, we may computed the desired view function employing either "forward" or "backward" computation:

$$v(L) = \langle N, L\rangle = \langle T^*I, L\rangle = \langle I, TL\rangle = \langle I, E\rangle$$

Thus the desired function may be determined either via the solution for illumination L and the quantity N at the eye, or the solution for the quantity I and the initial emission E at the sources.

The quantity I expresses the contribution of the emission at each element to the view function v. We will call I "importance", as it describes the importance of each element to the view from the eye. N, the dual to E, is thus the initial "emitted" importance, as determined by v. Importance may be understood as a quantity emitted by the eye and interreflected within the environment in the reverse of illumination; the resulting transport system may be solved just as the forward system for light, solution for either of the dual systems allowing computation of the view function v. Note that a given transport interaction is significant only if at least some importance and radiosity/radiance is transported across the interaction, in opposite directions.

In practice it is not possible to determine the the transport operator T exactly. As discussed in [SAS92], assume we employ computed approximation:

$$\tilde{T} = T + \Delta T$$

We may then compute solution $\tilde{L}$ under $\tilde{T}$:

$$\tilde{T}\tilde{L} = E$$

The resulting error as viewed from the eye may be written as:

$$\begin{aligned} v(L - \tilde{L}) &= \langle I, T(L - \tilde{L})\rangle \\ &= \langle I, E - (\tilde{T} - \Delta T)\tilde{L}\rangle \\ &= \langle I, \Delta T\tilde{L}\rangle \end{aligned}$$

Given approximation $\tilde{I}$ for transported importance, we may estimate this error as:

$$\tilde{v}(L - \tilde{L}) = \langle \tilde{I}, \Delta T\tilde{L}\rangle \tag{13}$$

Smits et al. have developed an importance and radiosity driven hierarchical radiosity algorithm that simultaneously solves for transport of both illumination and importance. As predicated by equation (13), hierarchical refinement is controlled by error computed as transport error weighted by estimated importance and radiosity. The algorithm thus allows accurate computation of radiosity significant to the view from the eye, and avoids wasted refinement that does not influence this view.

5.3 Importance and the Adjoint to the Transport Equation

In the paper of Smits et al. [SAS92] on importance driven radiosity, refinement and illumination are computed within a maze — within such an environment, many light interactions contribute to the view only after multiple reflections, and are thus relatively "unimportant", requiring little refinement. The preferred directionality of more specular reflection creates an *inherent* maze of interactions within transport space, in which only a limited number of paths substantially contribute to, or even lead to, the eye. Imagine importance radiating from the eye and interreflecting among surfaces in a near specular fashion: only interaction paths that lie along specular reflections will be strongly "illuminated" by importance, and thus require any degree of refinement. Under directional reflection, even within a fairly simple environment, there is a very high fraction of unimportant transport paths, much higher than exhibited under diffuse reflection. We can thus expect importance weighting to be even more significant for three point transport over glossy reflection than for diffuse reflection under radiosity.

5.3.1 Derivation of the Adjoint

Recall the system of equations that describe the radiance formulation for three point transport:

$$E_{jk} = L_{jk} - \sum_{i=1}^{n} R_{kji} L_{ij}$$

we would like to rewrite this system in the form:

$$TL = E$$

Under the given system we may decompose T into the operators $T = I_d - R$, where R is a "reflectance" operator accounting for the R_{kji} terms. Given the discrete form of our equation and algorithm, we will develop corresponding discrete operators. We first examine the derivation of R: the three dimensional character of R_{kji} over indices i, j, k leads (a computer scientist at least) naturally to a sort of three dimensional matrix or array. One could imagine a product over an $n \times n \times n$ R_{kji} "matrix" and an $n \times n \times 1$ L_{ij} matrix producing an $n \times n \times 1$ matrix of reflected radiances, such as shown in Figure 49.

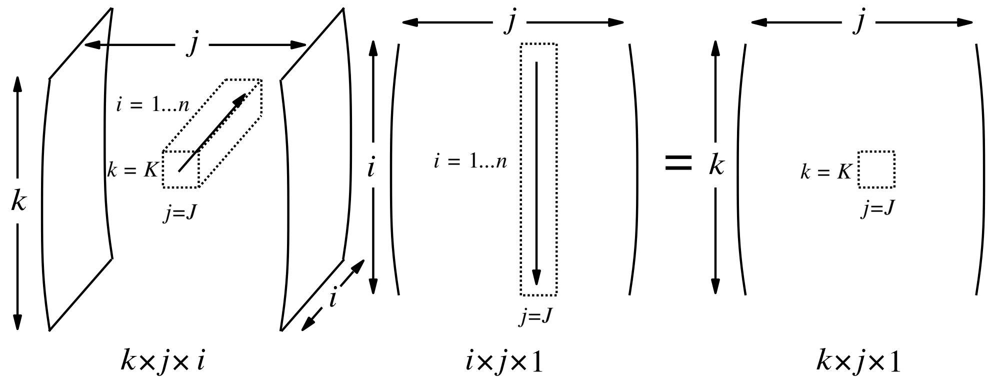


Figure 49: A 3D Matrix "Product": the row vector (into the page) at $j = J$ and $k = K$ in the 3D matrix multiplies the column at $j = J$ in the 2D matrix, to give the entry at $k = K$, $j = J$ in the result.

How one might manipulate such a matrix, however, is not be immediately clear. Furthermore, it is not possible to incorporate the identity operator into the shown "product" — each entry R_{kji} in the matrix multiplies some L_{ij} to yield result E_{jk}: such multiplication may not yield identity unless $i = j = k$.

These problems do not arise if the operator R is rewritten as a standard two-dimensional matrix. The $O(n^3)$ entries in R_{kji} may be placed within an $n^2 \times n^2$ matrix, the remaining entries set to zero. Given an $n^2 \times n^2$ transport matrix, we employ an $n^2 \times 1$ column vector for L, the resulting matrix-vector product yielding a similar $n^2 \times 1$ column vector for E. We choose to order the entries in L and E with the first index as most significant, that is, $(L_{11}, L_{12}, \ldots, L_{1n}, L_{21}, L_{22}, \ldots\ldots, L_{nn})$.

Consider the entries in the $n^2 \times n^2$ matrix representation for R (Figure 50). For a given $i = I$, the consecutive L_{Ij} entries in the L_{ij} vector determine a column of width n within the reflectance matrix. Similarly, for a given $j = J$, the consecutive E_{Jk} entries in E_{jk} determine a reflectance row, also of width n. Indices i and j together partition the matrix into n^2 $n \times n$ submatrices.

A given J-I submatrix represents the operator's action upon the element L_{IJ} in L_{ij}. Note that, within the submatrix, only elements in the J^{th} column multiply L_{IJ}. This column should thus contain $R_{kJI} = \{R_{1JI}, R_{2JI}, \ldots, R_{nJI}\}$, the multipliers of L_{IJ} within the system of equations (Figure 51). A complete reflectance matrix for the system under

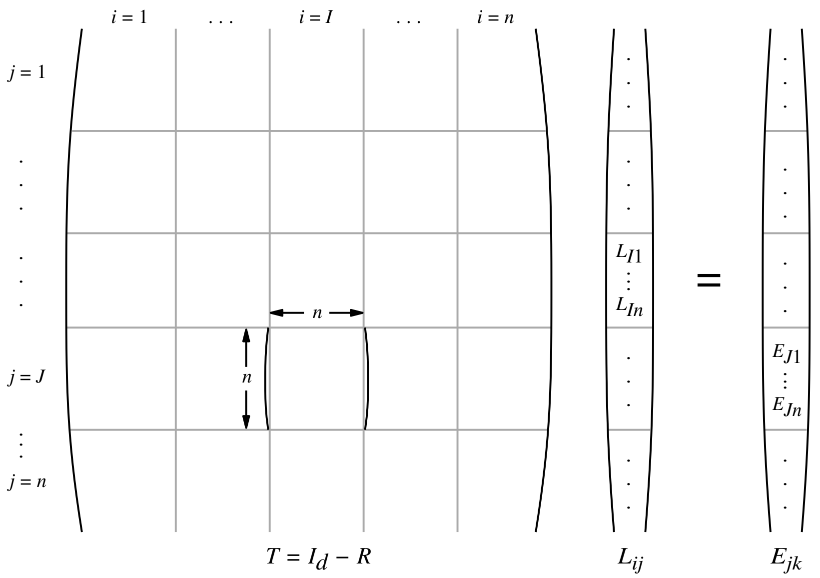


Figure 50: A Matrix Representation for R

$$
\begin{array}{cc}
 & \boldsymbol{I} \\
 & \begin{array}{cccccccc} 1 & 2 & \dots & & J & \dots & & n \end{array} \\
\boldsymbol{J} & \left(\begin{array}{cccccccc}
0 & 0 & \dots & 0 & R_{1JI} & 0 & \dots & 0 \\
0 & 0 & \dots & 0 & R_{2JI} & 0 & \dots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \dots & 0 & R_{nJI} & 0 & \dots & 0
\end{array}\right)
\end{array}
$$

Figure 51: A J-I Submatrix

$$\begin{pmatrix} \begin{pmatrix} R_{111} & 0 & 0 \\ R_{211} & 0 & 0 \\ R_{311} & 0 & 0 \end{pmatrix} & \begin{pmatrix} R_{112} & 0 & 0 \\ R_{212} & 0 & 0 \\ R_{312} & 0 & 0 \end{pmatrix} & \begin{pmatrix} R_{113} & 0 & 0 \\ R_{213} & 0 & 0 \\ R_{313} & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & R_{121} & 0 \\ 0 & R_{221} & 0 \\ 0 & R_{321} & 0 \end{pmatrix} & \begin{pmatrix} 0 & R_{122} & 0 \\ 0 & R_{222} & 0 \\ 0 & R_{322} & 0 \end{pmatrix} & \begin{pmatrix} 0 & R_{123} & 0 \\ 0 & R_{223} & 0 \\ 0 & R_{323} & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & R_{131} \\ 0 & 0 & R_{231} \\ 0 & 0 & R_{331} \end{pmatrix} & \begin{pmatrix} 0 & 0 & R_{132} \\ 0 & 0 & R_{232} \\ 0 & 0 & R_{332} \end{pmatrix} & \begin{pmatrix} 0 & 0 & R_{133} \\ 0 & 0 & R_{233} \\ 0 & 0 & R_{333} \end{pmatrix} \end{pmatrix}$$

Figure 52: A Complete Reflectance Matrix

$n = 3$ is given in Figure 52. Given a two dimensional representation for R, the identity operator immediately follows, and we thus have the desired $T = I_d - R$.

T is a real matrix, and thus the adjoint to T is simply its transpose, giving the adjoint equation:

$$T^{\mathrm{T}} I = (I_d - R^{\mathrm{T}})I = N$$

Consider the action of this operator to produce a given N_{JK}. Transposition of R induces a transposition of its submatrices; N_{JK} is thus determined by the transpose of the submatrix at $j = K$ and $i = J$ in the original R. By the construction of this submatrix, we have:

$$N_{JK} = I_{JK} - R_{1KJ} I_{K1} - R_{2KJ} I_{K2} - \cdots - R_{nKJ} I_{Kn}$$

this gives us the equation for transport of importance:

$$I_{jk} = N_{jk} + \sum_{i=1}^{n} R_{ikj} I_{ki}$$

5.3.2 Discussion

Recall the equation for three point transport of radiance:

$$L_{jk} = E_{jk} + \sum_{i=1}^{n} R_{kji} L_{ij}$$

This equation is relatively straightforward: the radiance at A_j in the direction of A_k depends on the radiance A_j receives from all other areas A_i, multiplied by a function of reflectance over the three areas. This reflectance has been "reversed", as with the form factor under radiosity, through the use of a reciprocity relation to eliminate area dependence.

In contrast, an intuitive understanding of the adjoint equation may not be immediately obvious: consider three areas A_I, A_J, and A_K; we shall assume that I_{IJ} is of great magnitude, but there is no importance between any other pairing of areas, and that $N_l = 0$ for $l = I, J, K$. Then, by the adjoint equation, we have:

$$\begin{aligned} I_{JK} &= N_{JK} + R_{IKJ} I_{KI} &= 0 \\ I_{KJ} &= N_{KJ} + R_{IJK} I_{JI} &= 0 \\ I_{IK} &= N_{IK} + R_{JKI} I_{KJ} &= 0 \end{aligned}$$

but

$$I_{KI} = N_{KI} + R_{JIK} I_{IJ} = R_{JIK} I_{IJ} > 0$$

assuming R_{JIK} is of nonzero value.

This result may seem counterintuitive, as one might expect importance I_{IJ} to be reflected from A_J toward A_K, and thus contribute to I_{JK} rather than I_{KI}. Quantity I_{IJ}, however, should be understood as the importance that *arrives* at area A_I *from* area A_J; that is, it is expressed in reverse of L_{IJ}, the radiance at A_I outgoing in the direction of A_J. Importance I_{IJ} is thus reflected at A_I under R_{JIK} toward A_K, contributing to I_{KI}.

As discussed in Section 5.2, radiance and importance are represented and transported in opposite directions; both, however, are reflected under discrete three point transport as governed by area reflectance. The dual nature of these equations may be more easily seen by writing both in terms of R_{kji}:

$$\begin{aligned} L_{jk} &= E_{jk} + \sum_i R_{kji} L_{ij} \\ I_{ij} &= N_{ij} + \sum_k R_{kji} I_{jk} \end{aligned}$$

A given R_{kji} thus governs transport of radiance L_{ij} and importance I_{jk}, in opposing directions. Furthermore, as discussed in Section 5.2, the estimate of error in such transport may be written as:

$$\tilde{v}(L - \tilde{L}) = \tilde{L}_{ij} \Delta R_{kji} \tilde{I}_{jk}$$

Finally, we note the the matrix formulations for both T and its adjoint T^* are diagonally dominant, and thus Jacobi methods applied over the dual systems will converge.

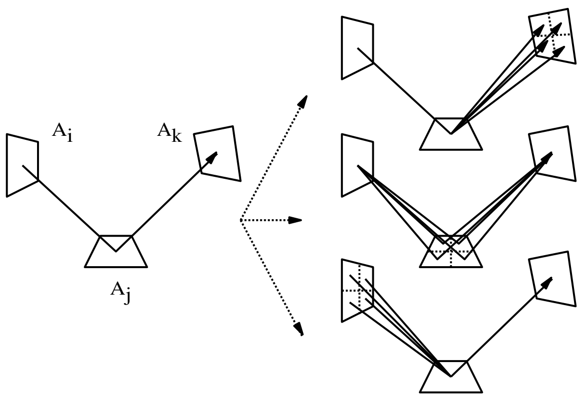


Figure 53: Refinement and Subdivision

5.4 Application

Given the adjoint equation, we have developed an importance driven version of our hierarchical algorithm for glossy reflection under three point transport (Chapter 4), directly analogous to the importance driven radiosity algorithm presented in [SAS92]. Refinement over interaction pairs (and corresponding area triples) is governed by transport error weighted by estimated transport radiance and importance. If this error exceeds user specified bounds, one or both interactions are subdivided and refinement is recursively applied (Figure 53); otherwise the interactions are linked. Actual transport is computed via dual gather/give iterations, in which each interaction gathers radiance from, and gives importance to, those interactions with which it has been linked.

A gather/give iteration results in received radiance and importance scattered throughout each interaction hierarchy. These quantities must be distributed and accounted for over all ancestors and descendents of each receiving interaction, in order to correctly maintain the hierarchical representation of the dual systems. We employ a distribution algorithm similar to those presented in Chapters 3 and [SAS92] for radiosity and importance over patch/element hierarchies: received radiance and importance is "pushed" to the leaf interactions within each hierarchy to ensure propagation to all descendents, and then "pulled" and distributed back up from the leaves through all higher level interactions to their common ancestor at the root. To correctly implement this algorithm, we must understand how the importance at a given interaction is related to the importance at its parent and at its children.

The following section is very similar to Section 4.3.4 of Chapter 4: Radiance within a Hierarchy; the resulting importance distribution, however, may be somewhat surprising.

5.4.1 Importance within a Hierarchy

As is shown in Chapter 4, radiance may be pushed unchanged from parents to children within an interaction hierarchy, and area averaged as it is pulled from child to parent. In this section we examine the distribution of importance within interaction hierarchies.

Consider a transport interaction from patch A_j to patch A_k. Under the adjoint equation the importance at A_j in the direction of A_k is determined by area reflectance R_{ikj}. The dependence of R_{ikj} upon subdivision of A_j and A_k thus determines the distribution of received importance within the interaction hierarchy.

By definition of area reflectance, we have:

$$\pi A_k F_{ki} R_{ikj} = \int_{A_i}\int_{A_k}\int_{A_j} f_r(x, x', x'')G(x, x')G(x', x'')dx''dx'dx$$

We first consider subdivision of A_k, yielding children $A_{k_1}, A_{k_2}, \ldots, A_{k_c}$. We may rewrite the above expression for R_{ikj} as:

$$\begin{aligned}
\pi A_k F_{ki} R_{ikj} &= \int_{A_i}\int_{A_k}\int_{A_j} f_r(x, x', x'')G(x, x')G(x', x'')dx''dx'dx \\
&= \int_{A_i}\int_{A_{k_1}}\int_{A_j} f_r(x, x', x'')G(x, x')G(x', x'')dx''dx'dx \\
&\quad + \cdots + \int_{A_i}\int_{A_{k_c}}\int_{A_j} f_r(x, x', x'')G(x, x')G(x', x'')dx''dx'dx \\
&= \pi A_{k_1} F_{k_1 i} R_{ik_1 j} + \cdots + \pi A_{k_c} F_{k_c i} R_{ik_c j}
\end{aligned}$$

Under the subdivision criteria employed in Section 4.3.2, we may assume that if the interaction from A_j to A_k is employed in transport, then areas A_i, A_j, and A_k are well separated, and thus [SAS92]:

$$F_{ki} \approx F_{k_d i}, \text{ for } d = 1 \ldots c$$

We then divide through by $\pi A_k F_{ki}$, yielding:

$$R_{ikj} = \frac{A_{k_1}}{A_k} R_{ik_1 j} + \cdots + \frac{A_{k_c}}{A_k} R_{ik_c j}$$

Thus, given distribution by area reflectance, the importance at a parent interaction is simply the area average of the importance at its children, under subdivision of A_k.

In contrast, subdivision of A_j induces a simple sum rather than an average:

$$\begin{aligned}
\pi A_k F_{ki} R_{ikj} &= \int_{A_i}\int_{A_k}\int_{A_j} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx \\
&= \int_{A_i}\int_{A_k}\int_{A_{j_1}} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx \\
&\quad + \cdots + \int_{A_i}\int_{A_k}\int_{A_{j_c}} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx \\
&= \pi A_k F_{ki} R_{ikj_1} + \cdots + \pi A_k F_{ki} R_{ikj_c}
\end{aligned}$$

dividing through by $\pi A_k F_{ki}$ we have:

$$R_{ikj} = R_{ikj_1} + \cdots + R_{ikj_c}$$

In a similar fashion, it may be shown that importance received at a given parent interaction may be pushed unchanged to its children under subdivision of A_k, but should be distributed by proportional area given subdivision of A_j.

5.4.2 Discussion

Note that while radiance is averaged/inherited from and to children over subdivision of either areas A_j and A_k, and the same relation holds for importance over subdivision of A_k, subdivision of A_j under importance induces summation/proportional distribution over children.

Averaging/inheritance over radiance follows from the definition of radiance as angular flux density. Flux density is inheritly an area average, and, for a surface at a given constant flux density, identical over all subregions. The assumption that A_j and A_k are well separated, and that $f_r(x,x',x'')$, $G(x,x')$, and $G(x',x'')$ are relatively constant (within error bounds), reduce the angular dependence of this density over each surface.

The situation is somewhat more complex under importance: importance measures the contribution at the view due to radiance across a given interaction; recall that I_{jk} expresses the importance arriving at area A_j from area A_k. If area A_j is subdivided, its descendent subareas make lesser contributions to the radiance at A_k, and, eventually, the view (for example, in a direct view, descendents occupy/illuminate less area on the image plane), and thus should inherit only proportionally the importance at A_j in the direction of A_k. The dual to proportional inheritance is summation across children: regrouping the children of a given A_j resums their importance.

Subdivision of area A_k, however, is averaged/inherited — subdivision over A_k does not lessen the contribution of A_j in the direction of the descendent subregions A_{k_d}, rather it lessens the importance at each A_{k_d} in other directions within the environment.

5.5 Results

Preliminary results of our application over glossy reflection are presented in Section 4.5.2: Figure 47 and Table 4. These results show a significant reduction in transport computation: improvement to 3 hundred thousandths of one percent over a naive triples-based approach, and indicate that an importance and radiance driven hierarchical approach possesses great promise for the development of efficient global illumination algorithms for general reflection. Note that the environment shown in Figure 47 is very simple; the use of importance and hierarchical techniques allows the feasible employment of fine meshing to capture significant reflection of relatively high specularity ($\kappa = 500$), but the initial $O(k^3)$ examination of polygon triples is still prohibitively expensive over complex environments.

Chapter 6

Summary and Discussion

6.1 Summary

This dissertation is a discussion and development of hierarchical algorithms for illumination. These algorithms operate through recursive, adaptive refinement of the environment into hierarchical meshes and associated transport links, as governed by user specified error bounds. Rather than computing light transport only between elements at the finest level of refinement, the algorithms allow computation of transport between higher level subpatches, provided such transport satisfies given bounds. Note that the existence of both an acceptable nonzero error bound and a means of estimating error over computed transport is crucial to application of these algorithms. As discussed in this dissertation, employment of hierarchical methods yields significant savings in computation.

The initial work in fully hierarchical methods was that of Hanrahan and Salzman [HS90] for the computation of radiosity over unoccluded environments. As discussed in Chapter 3, this algorithm has been extended to occluded environments, incorporating visibility heuristics and acceleration via brightness weighting and multigrid techniques. Given an environment consisting of k polygonal patches and n elements at the finest level of allowed refinement, the algorithm requires at most $O(n + k^2)$ transport interactions; traditional methods would require $O(n^2)$. The hierarchical transport representation computed by the algorithm thus corresponds to a decomposition of the $O(n^2)$ form factor matrix into at most $O(n + k^2)$ blocks. In implementation, the algorithm computes within minutes solutions over highly refined meshing.

In Chapter 4 we discuss application of hierarchical transport to nondiffuse reflection. The directional distribution of light under such reflection is captured via explicit solution for interpatch radiances within the system. We have thus developed a radiance formulation for discrete three point transport, incorporating a new measure and description of reflectance: *area reflectance*. This formulation and associated reflectance allow an estimate of error in the computation of radiance across triples of surface elements, and lead directly to a hierarchical refinement algorithm for global illumination.

As discussed in this dissertation, we have implemented and analyzed the algorithm over surfaces exhibiting glossy specular and diffuse reflection. The glossy reflection function employed is a highly simplified version of the Torrance-Sparrow reflection function; error in transport is estimated by interval width computed through the use of bounding cones. Growth in transport computation is shown to be $O(n + k^3)$ for sufficient refinement — this growth is exhibited in experimental trials. Naive application of three point transport would require computation over $O(n^3)$ element-triple interactions. The employment of hierarchical methods thus corresponds to a decomposition, within user specified error bounds, of an $O(n^3)$ area reflectance matrix into at most $O(n + k^3)$ subblocks.

The hierarchical radiosity techniques of Chapter 3 have been further improved and accelerated by the work of Smits et al. [SAS92] introducing importance driven refinement. Under their algorithm, the dual systems governing transport of radiosity from light sources and importance from the eye are solved simultaneously — hierarchical refinement is controlled by error computed as transport error multiplied by transport radiosity and importance, resulting in an extremely significant reduction in computation of viewed radiosity within complex environments.

Global illumination within nondiffuse environments is ideally suited for computation under importance and radiance driven hierarchical refinement: a transport interaction is significant to the view from the eye only if it lies within paths of directional reflection of both radiance originating at a light source, and importance originating at the eye. As discussed in Chapter 5, we may therefore expect importance weighting to be even more significant for three point transport over glossy reflection than for diffuse reflection under radiosity. In Chapter 5 we thus derive the adjoint to our original transport formulation, and discuss the preliminary results of application of this adjoint in the form of an importance driven version of our glossy implementation. These results show significant reduction in transport computation, and indicate that an importance and radiance driven hierarchical technique possesses great promise for the development of efficient global illumination algorithms for general reflection.

6.2 Discussion

Hierarchical algorithms incorporating importance and radiance driven refinement possess great potential for the efficient computation of true global illumination. The key to these algorithms is the capability to concentrate computation where it is needed, while avoiding unnecessary work: a radiance and importance driven hierarchical refinement algorithm would in theory require the minimum computation necessary to capture all transport of energy significant to the view from the eye, essentially employing lazy evaluation over the minimum expression required for transport. Crucial to the effectiveness of these algorithms is the need to push all computation up as high as possible within the hierarchies, while still capturing, and evaluating efficiently, all necessary computation at the proper level of refinement.

Three interrelated aspects determine the degree to which these goals may be satisfied: the employed hierarchical representation itself, the error estimate under which the hierarchy is constructed, and the solution mechanics employed for actual computation.

As discussed in Chapter 4, current hierarchical methods are *one-sided*: allowing subdivision and refinement over individual patches, but not clustering to represent higher level aggregates. Without such capability, the efficiency and feasibility of these methods are severely degraded: consider scattering pushpins across the surface of a desk in a typical radiosity interior — applying similar detailing and texturing to all objects and surfaces in an environment yields realism and complexity over which no current algorithm may efficiently compute. Feasible computation of illumination requires a means of capturing and representing higher level clusters for transport.

Hierarchical structures must allow and satisfy integration of varying meshing/refinement criteria. Of particular importance is the incorporation of discontinuity meshing [H91]:

meshing to capture discontinuities in illumination may provide large gains in both efficiency and accuracy of computation. Recent work by Lischinski et al. [LTG93] demonstrates the effectiveness of integrating hierarchical radiosity and discontinuity criteria.

As the environment increases in complexity hierarchical methods require more accurate, general, and efficient estimators for error in transport computation. In theory, the error estimation strategy employed for glossy reflection could be applied to any reflection function composed of relatively well behaved monotonic components. However, with increased complexity and size, a less conservative and more accurate estimator would result in large savings in computation, eliminating unnecessary refinement and calculation. Furthermore, naive employment of geometric and interval techniques over more complex reflection functions may result in relatively arcane error expressions. As discussed in Chapter 4, the application of wavelet theory to hierarchical methods over general reflection provides a nearly automatic means of error estimation when approximating any smooth kernel [GSCH93].

This application of wavelets illustrates possible benefits of more sophisticated solution mechanisms given more general environments. For simple environments consisting of large, relatively well behaved planar polygons the application of very simple finite element techniques over piecewise constant basis functions, combined with quadtree hierarchies and visibility estimation via ray casting, is sufficient. However, the useful application of hierarchical methods for true global illumination requires computation over more realistic, general, and exact environments. Algorithms must thus efficiently allow general and accurate reflection functions, general, complex surfaces, participating media (solution over general volumes rather than simple surfaces), dynamic environments — that is, more physics, more objects, and less restriction. Computation over such environments requires further development and adaptation of numeric algorithms and methods from transport theory, finite element methods and wavelets (particularly over higher level bases), perturbation theory, and geometric modeling. The exploitation of such work for computer graphics remains an effective and fertile source of instruction, methods, and inspiration.

In the same fashion, increasingly complex environments require the development of visibility methods and algorithms capable of efficiently exploiting and satisfying hierarchical transport computation [TH93, Chapter 3]. With greater generality, the computation of transport visibility and occlusion over clustering and complex surfaces presents significant challenges.

In addition, there is the problem of adapting hierarchical methods to external requirements for more efficient computing, particularly parallel processing. Note that the gathering iteration employed by the hierarchical algorithms presented in this dissertation correspond to many read/exclusive write operations. As discussed in the work of Singh et al. [SHTGH92], hierarchical radiosity, in common with other n-body methods, seems well suited to coarse parallelism.

Finally, as discussed in Chapter 4, the estimation of overall error under hierarchical illumination methods remains a crucial open problem subsuming hierarchy, transport error estimation, and solution mechanics. Such an error analysis is necessary for the controlled and accurate application of hierarchical algorithms to illumination.

Bibliography

[A85] A. A. Appel. An efficient program for many-body simulation. *SIAM Journal on Scientific and Statistical Computing,* 6(1):85–103, 1985.

[A92] J. Amanatides. Algorithms for the detection and elimination of specular aliasing. *Proceedings of Graphics Interface '92,* 86–93, 1992.

[AH93a] L. Aupperle, and P. Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. *Computer Graphics,* 27:155–162, 1993.

[AH93b] L. Aupperle, and P. Hanrahan. Importance and discrete three point transport. *Proceedings of the Fourth EuroGraphics Workshop on Rendering,* 85–94, 1993.

[B77] J. F. Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics,* 11(2):192–198, 1977.

[B90] J. Blinn. Triage tables. *IEEE Computer Graphics and Applications,* 10(1):70-75, 1990.

[BF89] C. Buckalew, and D. Fussel. Illumination networks: fast realistic rendering with general reflectance functions. *Computer Graphics,* 23(3):89–98, 1989.

[BFGS88] L. Bergman, H. Fuchs, E. Grant, and S. Spach. Image rendering by adaptive refinement. *Computer Graphics,* 20(4);29–38, 1988.

[BH86] J. Barnes, and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature,* 324:446–449, 1986.

[BRW89] D. R. Baum, H. E. Rushmeier, and J. M. Winget. Improving radiosity solutions through the use of analytically determined form factors. *Computer Graphics,* 23(3):325–334, 1989.

[C86] R. L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics,* 5(1):51–72, 1986.

[CCWG88] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics,* 22(4):75–84, 1988.

[CF90] A. T. Campbell, and D. S. Fussel. Adaptive mesh generation for global diffuse illumination. *Computer Graphics,* 24(4):155–164, 1990.

[CG85] M. F. Cohen, and D. P. Greenberg. The hemi-cube: a radiosity approach for complex environments. *Computer Graphics,* 19(3):31–40, 1985.

[CGIB86] M. F. Cohen, D. P. Greenberg, D. S. Immel, and P. J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications,* 6(2):26–30, 1986.

[CRMT91] S. E. Chen, H. E. Rushmeier, G. Miller, and D. Turner. A progressive multi-pass method for global illumination. *Computer Graphics,* 25(4):165–174, 1991.

[CSD93] P. H. Christensen, D. H. Salesin, and T. D. DeRose. A continuous adjoint formulation for radiance transport. *Proceedings of the Fourth EuroGraphics Workshop on Rendering,* to appear, 1993.

[CT82] R. L. Cook, and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics,* 5(1):51–72, 1982.

[E89] E. Esselink. About the order of Appel's algorithm. Computing Science Note KE5-1, Department of Computer Science, University of Groningen, 1987.

[G71] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers,* C-20(6):623–629, 1971.

[G88] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, Cambridge, MA, 1988.

[GSCH93] S. J. Gortler, P. Schröder, M. F. Cohen, and P. Hanrahan. Wavelet radiosity. *Computer Graphics,* 27:65–72, 1993.

[GTGB84] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. *Computer Graphics,* 18(3):213–222, 1984.

[H89] R. Hall. *Illumination and Color in Computer Generated Imagery.* Springer-Verlag, New York, 1989.

[H90] P. S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics,* 24(4):145–154, 1990.

[H91] P. S. Heckbert. *Simulating Global Illumination Using Adaptive Meshing.* Doctoral Dissertation, University of California, Berkeley, 1991.

[H92] P. Hanrahan. From radiometry to the rendering equation. *SIGGRAPH '92 Radiosity Course Notes,* 1992.

[HS67] H. C. Hottel, and A. F. Sarofim. *Radiative Transfer.* McGraw-Hill, New York, 1967.

[HS90] P. Hanrahan, and D. B. Salzman. A rapid hierarchical radiosity algorithm for unoccluded environments. In K. Bouatouch, editor, *Photosimulation, Realism and Physics in Computer Graphics.* Springer-Verlag, 1991. Also: Princeton University CS-TR-281-90, 1990.

[HSA91] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics,* 25(4):197–206, 1991.

[ICG86] D. S. Immel, M. F. Cohen, and D. P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics,* 20(4):133–142, 1986.

[K86] J. T. Kajiya. The rendering equation. *Computer Graphics,* 20(4):143–150, 1986.

[K90] J. T. Kajiya. Radiometry and photometry for computer graphics. *SIGGRAPH '90 Advanced Topics in Ray Tracing Course Notes,* 1990.

[LTG93] D. Lischinski, F. Tampieri, and D. P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. *Computer Graphics,* 27:199–308, 1993.

[M36] P. H. Moon. *The Scientific Basis of Illuminating Engineering.* McGraw-Hill, New York, 1936.

[M88] T. J. V. Malley. *A Shading Method for Computer Generated Images.* Master's Thesis, University of Utah, 1988.

[M92] D. Mitchell. Manuscript. 1992.

[N63] F. E. Nicodemus. Radiance. *American Journal of Physics,* 31(5):368–377, 1963.

[NN85] T. Nishita, and E. Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. *Computer Graphics,* 19(3):23–30, 1985.

[NRHGL77] F. E. Nicodemus, J. C. Richmond, I. W. Ginsberg, and T. Limperis. *Geometrical Considerations and Nomenclature for Reflectance.* National Bureau of Standards, MN-160, 1977.

[P75] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM,* 18(6):311–317, 1975.

[S89] F. Sillion. *Lighting Simulation for Image Synthesis: Realism and Interactivity.* Doctoral Dissertation, University of Paris, Centre d'Orsay, 1989.

[S90] P. Shirley. A ray tracing method for illumination calculation in diffuse-specular scenes. *Proceedings of Graphics Interface '90,* 205–212, 1990.

[S91] P. S. Shirley. *Physically Based Lighting Calculations for Computer Graphics.* Doctoral Dissertation, University of Illinois at Urbana-Champaign, 1991.

[SAS92] B. E. Smits, J. R. Arvo, and D. H. Salesin. An importance-driven radiosity algorithm. *Computer Graphics,* 26(2):273–282, 1992.

[SAWG91] F. X. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg. A global illumination algorithm for general reflectance distributions. *Computer Graphics,* 25(4):186–196, 1991.

[SC78] E. M. Sparrow, and R. D. Cess, *Radiative Heat Transfer.* Hemisphere Publishing, Washington, 1978.

[SH81] R. Seigel, and J. R. Howell. *Thermal Radiation Heat Transfer.* Hemisphere Publishing, Washington, 1981.

[SHTGH92] J. P. Singh, C. Holt, T. Totsuka, A. Gupta, and J. L. Hennessy. Load balancing and data locality in hierarchical n-body methods. Technical Report CSL-TR-92-505, Department of Computer Science, Stanford University, 1992.

[SP89] F. Sillion, and C. Puech. A general two-pass method for integrating specular and diffuse reflection. *Computer Graphics,* 23(3):335–344, 1989.

[SPL88] M.-Z. Shao, Q.-S. Peng, and Y.-D. Leng. A new radiosity approach by procedural refinements for realistic image synthesis. *Computer Graphics,* 22(4):93–101, 1988.

[TH93] S. J. Teller, and P. Hanrahan. Global visibility algorithms for illumination computations. *Computer Graphics,* 27:239–246, 1993.

[TN87] W. Thibault, and B. Naylor. Set operations on polyhedra using binary space partition trees. *Computer Graphics,* 21(4):153–162, 1987.

[TS67] K. E. Torrance, and E. M. Sparrow. Theory of off-specular reflection from roughened surfaces. *Journal of the Optical Society of America,* 57(9):1104–1114, 1967.

[W69] J. Warnock. A hidden surface algorithm for computer-generated half-tone pictures. Technical Report TR 4-15, Computer Science Department, University of Utah, 1969.

[W80] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM,* 23(6):343–349, 1980.

[W92] G. J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics,* 26(2):265–272, 1992.

[WEH89] J. R. Wallace, K. A. Elmquist, and E. A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics,* 23(3):315–324, 1989.

[WRC88] G. J. Ward, F. M. Rubinstein, and R. D. Clear. A ray tracing solution for diffuse environments. *Computer Graphics,* 22(3):85–92, 1988.

Appendix A

Error Estimation

A.1 Error Analysis

Recall the error expression derived in Section 4.2.3:

$$\pi(\Delta F_{ji}S_{kji} + \Delta S_{kji}F_{ji})$$

In implementation, we have divided ΔS_{kji} into separate components for each subfactor of f_g (Section 4.4.1). Our error measure is thus:

$$\frac{\kappa+2}{8}(\Delta F_{ji}\frac{\cos^{\kappa}\theta_m}{\cos\theta_i\cos\theta_r} + \Delta\cos^{\kappa}\theta_m F_{ji}\frac{1}{\cos\theta_i\cos\theta_r} + \Delta\sec\theta_i F_{ji}\frac{\cos^{\kappa}\theta_m}{\cos\theta_r} + \Delta\sec\theta_r F_{ji}\frac{\cos^{\kappa}\theta_m}{\cos\theta_i})$$

The refinement procedure of Section 4.3.2 takes in implementation an additional argument, `Ceps`, against which the two estimates of error in reciprocal cosine are tested.

We are left with the computation of $\Delta\sec\theta_i$, $\Delta\sec\theta_r$, and $\Delta\cos^{\kappa}\theta_m$. The variance (and associated error) in these cosine terms over given patches A_i, A_j, A_k is determined by the set of possible $\vec{\omega}_i$, $\vec{\omega}_r$ lying between the patches (we dispense with $'$ notation in this section).

Consider patches A_i and A_j (Figure 54): we enclose these patches in spheres S_i, S_j with centers c_i, c_j, and radii r_i, r_j, respectively. For the present we will assume that the interiors of S_i and S_j do not intersect, and thus there exists a tangent cone lying between the spheres.

Note that this cone is a right circular cone centered on the line joining c_i and c_j. Consider the nappe containing S_i: it may be regarded as a cone of direction vectors centered about

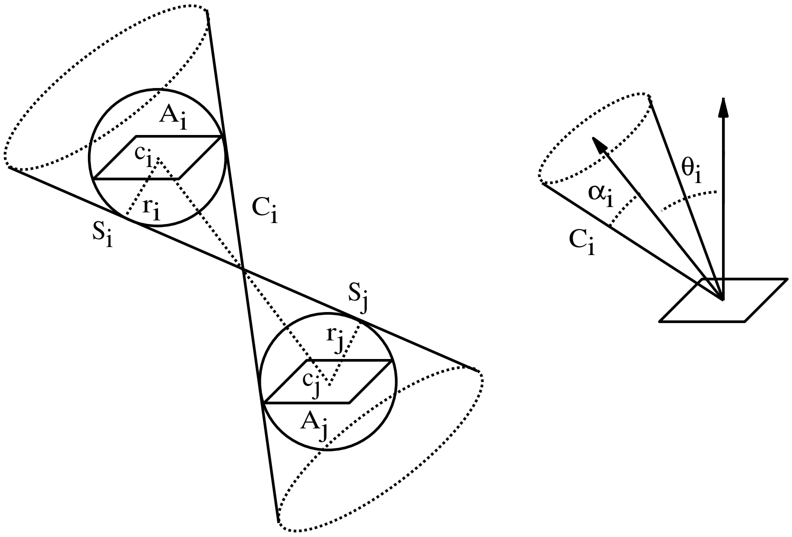


Figure 54: Estimating Cones

the vector $c_i - c_j$. We will call this vector cone C_i. If p_i and p_j are any two points on or in S_i, S_j, then the vector $p_i - p_j$ lies within C_i. C_i thus bounds the set of possible $\vec{\omega}_i$. We may characterize C_i by the angle α_i defined by its axis, $c_i - c_j$, and boundary, and computed as (Figure 55):

$$\sin \alpha_i = \frac{r_i}{D_i} = \frac{r_j}{D_j} = \frac{r_i + r_j}{D_i + D_j}$$

giving:

$$\alpha_i = \arcsin \frac{r_i + r_j}{D_{ij}}$$

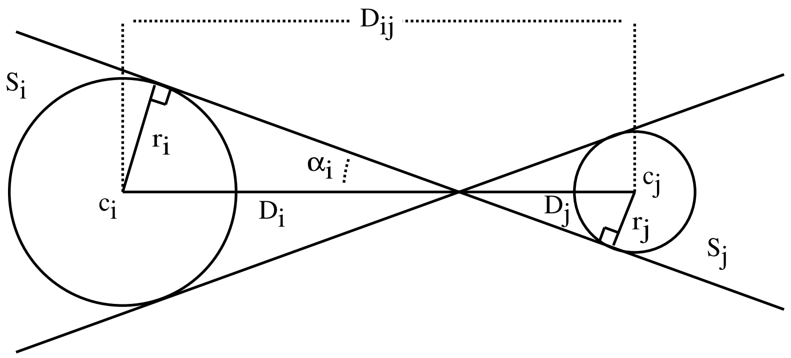


Figure 55: Computing α_i

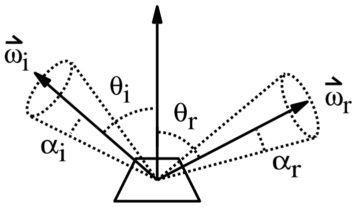


Figure 56: C_i and C_r

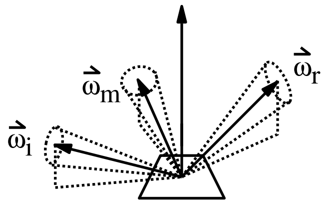


Figure 57: C_i, C_r, and C_m

where D_{ij} is the distance between c_i and c_j. Cone C_r and angle α_r may be similarly defined over A_j and A_k. If either pair of spheres intersect, we set the corresponding $\alpha = \pi$.

Given cones C_i and C_r (Figure 56) bounding variation in $\vec{\omega}_i$ and $\vec{\omega}_r$, we may easily compute maximum and minimum bounds for $\sec\theta_i$, $\sec\theta_r$:

$$\begin{aligned}
\max(\sec\theta) &= \begin{cases} \sec(\theta+\alpha) & \text{if } (\theta+\alpha) \leq \theta_c \\ \sec\theta_c & \text{otherwise} \end{cases} \\
\min(\sec\theta) &= \begin{cases} \sec(\theta-\alpha) & \text{if } \theta > \alpha \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$

We may then compute error in estimation as $(\max - \min)/2$.

The cones C_i and C_r centered about $\vec{\omega}_i$ and $\vec{\omega}_r$ induce a similar cone of variation about $\vec{\omega}_m$ (Figure 57). We will first examine the dependency of $\vec{\omega}_m$ on $\vec{\omega}_i$, assuming $\alpha_r = 0$.

Consider the perturbation of $\vec{\omega}_i$ by some $\Delta\vec{\omega}_i$ such that $\vec{\omega}_i + \Delta\vec{\omega}_i = \langle\vec{\omega}_i\rangle$ (Figure 58), where $\langle\vec{\omega}_i\rangle$ is a unit vector. We will call the resulting perturbed mirror vector $\langle\vec{\omega}_m\rangle$:

$$\langle\vec{\omega}_m\rangle = \frac{\langle\vec{\omega}_i\rangle + \vec{\omega}_r}{|\langle\vec{\omega}_i\rangle + \vec{\omega}_r|}$$

Let c and $\langle c\rangle$ be the points of intersection of $\vec{\omega}_m$ and $\langle\vec{\omega}_m\rangle$ with the line segments connecting $\vec{\omega}_i$ and $\vec{\omega}_r$, $\langle\vec{\omega}_i\rangle$ and $\vec{\omega}_r$, respectively. Then, by similarity of triangles, the vector $\langle c\rangle - c$ is

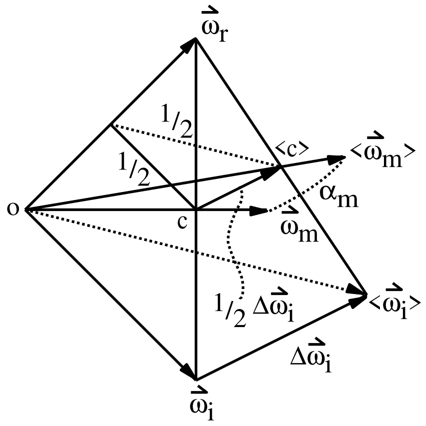


Figure 58: Perturbation of $\vec{\omega}_i$

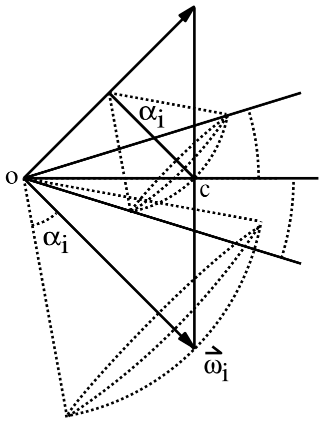


Figure 59: Generalized Perturbation

parallel to $\Delta\vec{\omega}_i$ and of length exactly half that of the perturbation. Similarly, the segments through c parallel to $\vec{\omega}_i$ and $\langle c\rangle$ parallel to $\langle\vec{\omega}_i\rangle$ intersecting $\vec{\omega}_r$ must be of length:

$$\frac{1}{2}|\vec{\omega}_i| = \frac{1}{2}|\langle\vec{\omega}_i\rangle| = \frac{1}{2}$$

Furthermore, the angle between these two segments is equal to that between $\vec{\omega}_i$ and $\langle\vec{\omega}_i\rangle$.

Let α_m be the angle between $\vec{\omega}_m$ and $\langle\vec{\omega}_m\rangle$; note that α_m is subtended by the segment $\overline{c\langle c\rangle}$. If we generalize the perturbation to arbitrary $\Delta\vec{\omega}_i$, we find that α_m is spanned by the base of a spherical cone of angle α_i and radius $\frac{1}{2}$, as shown in Figure 59. Let l be the length of the longest possible $\overline{c\langle c\rangle}$ within this cone; then (Figure 60):

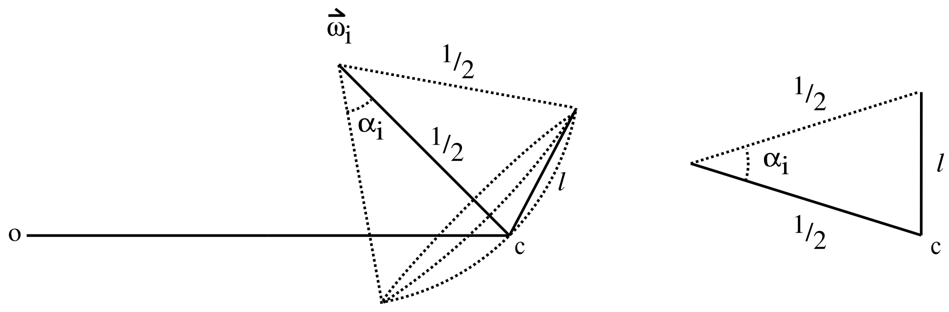


Figure 60: The Spherical Cone

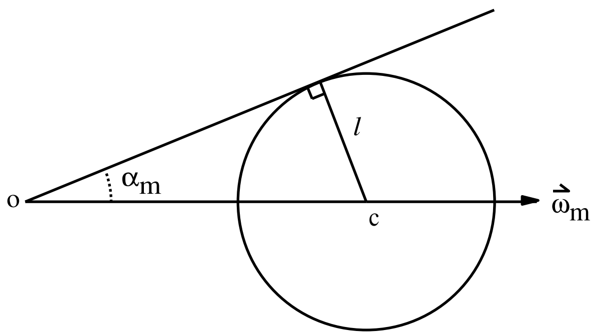


Figure 61: Estimating α_m

$$l = \sin \frac{\alpha_i}{2}$$

The base of the cone is contained within a sphere of radius l centered at c, and we may thus conservatively estimate α_m as the angle subtended by this sphere (Figure 61):

$$\alpha_m \leq \arcsin \frac{l}{|c|} = \arcsin \frac{\sin \frac{\alpha_i}{2}}{|c|}$$

where $|c| = \vec{\omega}_i \cdot \vec{\omega}_m = \vec{\omega}_r \cdot \vec{\omega}_m$, by definition. Note that if $|c|$ is nearly 1, then $\alpha_m \approx \alpha_i/2$, which is as expected for the nearly planar case.

Allowing both α_i and $\alpha_r > 0$ (Figure 62), we have:

$$\alpha_m \leq \arcsin \min(\frac{\sin \frac{\alpha_i}{2} + \sin \frac{\alpha_r}{2}}{\vec{\omega}_i \cdot \vec{\omega}_m}, 1.0)$$

Given α_m, determination of $\max(\cos^\kappa \theta_m)$, $\min(\cos^\kappa \theta_m)$, and thus $\Delta \cos^\kappa \theta_m$ immediately follows.

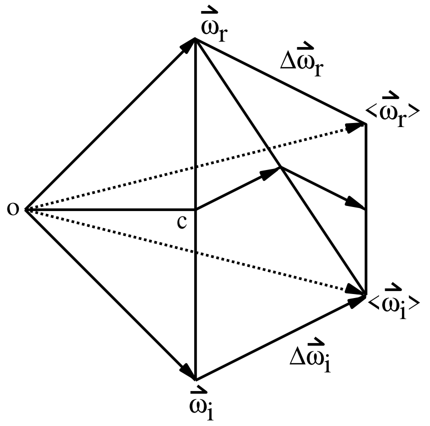


Figure 62: $\Delta\vec{\omega}_i$ and $\Delta\vec{\omega}_r$

Having computed the above estimates and maxima, and incorporating the estimates for form factor computation, we may bound and estimate error in transport as:

$$\begin{aligned}\tfrac{\kappa+2}{8}\quad &(F_{dji}^2 \max(\cos^\kappa \theta_m) \max(\sec \theta_i) \max(\sec \theta_r) + \Delta \cos^\kappa \theta_m F_{dji} \max(\sec \theta_i) \max(\sec \theta_r) \\ &+ \Delta \sec \theta_i F_{dji} \max(\cos^\kappa \theta_i) \max(\sec \theta_r) + \Delta \sec \theta_r F_{dji} \max(\cos^\kappa \theta_i) \max(\sec \theta_i))\end{aligned}$$

It is this error measure that we employ in our implementation.

A.2 Error Measures, Bounds, and Adaptive Refinement

The adaptive refinement procedures employed in Chapters 3 and 4 ensure that each subcomputation for light transport satisfies a given error bound, `Eps`. However, as is discussed in Chapter 4, it is not immediately clear how bounding the error over these individual subcomputations is related to the resulting error in the total computation. Some aspects of this problem, particularly the employment of absolute versus relative error bounds, may be better understood by examining the following abstraction:

Given an output device that may display integer values from 0–255, we wish to compute some quantity such that the result is within $\pm c$ of the correct value (provided the value lies within the display range). We assume that we have an adaptive subdivision scheme in which we may estimate the absolute error in each subcomputation.

Let $c = 1$, and assume the refinement procedure subdivides computation into two subcomputations such that:

1. The sum of the computed subvalues approximates the desired value.
2. The absolute error in each subcomputation is half that of the original.

Refinement may then be illustrated as binary tree, as shown in the following example:

```
                                    5/0.9375
                         10/1.875
                                    5/0.9375
               20/3.75
                                    5/0.9375
                         10/1.875
                                    5/0.9375
      40/7.5
               ...
80/15
      ...
```

where the pairs r/e indicate the correct result, and the maximum absolute error in the computed result. Note that, under these conditions, refinement will eventually yield 16 values within $\pm$0.9375 of the correct subvalue, 5, thus satisfying the error bound, c. However, assuming that all of these subcomputations exhibit maximum error and the same bias, the absolute error in their sum is 16 $*$ 0.9375 = 15, the original error in the root computation. Thus, subdivision does not improve the approximation, and does not yield a result within $\pm c$ of the correct value.

Note also that further reduction of the error bound, by any arbitrary amount, may have no effect on the accuracy of the final computation. The lower error bound will simply generate more levels of subdivision, and smaller final subvalues and associated errors, which then sum to the original, unacceptable, result.

The situation is more promising for a relative error measure and bound. Assume we specify an error bound of 1.25%, chosen because 80 $*$ 0.0125 = 1, thus satisfying c. Subdivision produces an identical tree, save that errors are now percentage relative error:

				5/0.9375%
			10/1.875%	
				5/0.9375%
		20/3.75%		
				5/0.9375%
			10/1.875%	
				5/0.9375%
	40/7.5%			
		...		
80/15%				
	...			

In this case, again assuming maximum error and identical bias, the relative error over the entire computation is (16 $*$ 0.9375) / 16 = 0.9375%, thus yielding a result within ± 1 of the true value. Note that reducing the error bound directly reduces the error in the total computation.

However, there are two difficulties in the use of relative error — both related to the dependence of the error measure on the magnitude of the computed value. If the computed result were 250, rather than 80, then a relative error of 1.25% would yield an uncertainty of $\pm$3.125, in excess of the given c. Similarly, if the result had been 5, computed uncertainty would be $\pm$0.0625, requiring unnecessary subdivision and computation.

Consider a different formulation for subdivision and computation: in this formulation, the refinement procedure subdivides computation into two subcomputations such that:

1. The *average*, not the sum, of the computed subvalues approximates the desired value.

2. The absolute error in each subcomputation is half that of the original.

```
                        70/0.9375
                71/1.875
                        72/0.9375
        72/3.75
                        72/0.9375
                73/1.875
                        74/0.9375
75/7.5
                ...
        78/3.75
                ...
80/15
        ...
85/7.5
        ...
```

Again we have a final error for individual subcomputations at a maximum of 0.9375. The maximum error in their average, however, is (16 $*$ 0.9375) / 16 = 0.9375, preserving the error bound for the root value. This ensures that, as for relative error above, reducing the error bound directly reduces the error in the total computation.

Note that in addition to preserving absolute error, this subdivision scheme also preserves relative error as (0.9375% $*$ (1280 / 16)) / 80 = 0.9375%. Relative error is, of course, still sensitive to the magnitude of the result.

The question becomes whether the operation of the actual adaptive refinement procedures may be modeled by the first abstraction or the second.

Consider a patch A_j: under Chapter 3 we wish to compute the radiosity (flux density) at A_j due to some patch A_i; under Chapter 4 we wish to compute the radiance (angular flux density) at A_j in the direction of a patch A_k due to a given patch A_i. Upon subdivision of A_j, we average the radiosity/radiance of the children to approximate that at A_j, and thus the refinement scheme is similar to that of the second abstraction.

Consider, however, three cases in which error in computation resembles that of the first abstraction:

1. Transport over the subdivision of A_i or A_k is additive rather than averaged. It is thus not guaranteed that the absolute error in the total computation is directly related to the error bound imposed on subcomputed values. Relative error is maintained, but, due to dependence on the magnitude of the result, may allow over or underaccurate computation.

   It should be noted that, even under an absolute error measure/bound, reducing the bound will (eventually) improve the accuracy of the computation, unlike the first abstraction. As the bound is lowered, subdivision over A_i and A_k will eventually alternate with (accuracy preserving/improving) subdivision over A_j. Thus, the user bound does provide a means of controlling the accuracy of the computation. Note also that if subcomputation error sums to less than original computation error, then even additive subdivision improves accuracy.

2. When solving for illumination, the radiosity/radiance at A_j must be computed over multiple A_i's and A_k's, a source of additional additive error. Consider, for example, the computation of radiosity over k polygons — the radiosity at A_j due to $k-1$ other A_i must be computed, and added, along with the absolute error in the $k-1$ computations. Note that multiple computations over the environment complicates relative error bounds as well: to ensure that the final radiosity at A_j is within ± 1 of the true value, relative error must be set to 1/(total radiosity from *all* patches) — prerequiring an accurate estimate of global illumination.

   Note, however, that there is a limit to the degree to which this variety of error may be compounded, even assuming it is strictly additive. The actual (absolute) error bound may be seen as no more than k times the set user bound, and accuracy does improve as the bound is lowered.

3. Error is dependent upon iteration. That is, the computation of the radiosity/radiance at A_j due to A_i is compounded with the computation of radiosity/radiance at A_i due to A_h, which is compounded with the computation of radiosity/radiance at A_g due to

A_f, etc. This multiplicative combination of estimates leads to additive relative error, and, assuming that the multiplied estimates, such as form factors, are less than or equal to 1, no worse than additive absolute error.

As in the second case above, there is a limit to the number of terms in the error sum — assuming that the estimated values are less than 1, as should occur in an environment modeling accurate reflection/transport (and conservation of energy), the multiplicative terms, and thus the requirement for iteration, will become negligible after a finite number of computations, thus preventing unbounded compounding of error.

We have found in practice that an absolute error measure/bound appears to be adequate for the refinement procedure employed for discrete three point transport as described in Chapter 4. The hierarchical radiosity algorithm presented in Chapter 3 was implemented under a relative error bound and estimate; however, an absolute error measure, combined with importance weighting, has been very successfully employed by Smits et al. [SAS92]. Absolute error is arguably the natural bound to employ when computing for display on a crt possessing a limited and discretized intensity range. Although the error bound may not directly relate to the error in total computation, it does control this error, and reduction in the bound provides improvement in computation accuracy.

Relative error appears to be more robustly maintained over refinement, but its dependence on the magnitude of the computed result, combined with the probable adequacy of absolute error, may render it less attractive for employment with an inherently absolute display device.

We note that the requirement of computation within $\pm c$ of the true result tends to bias the simple analysis presented above against relative error. Allowing greater absolute error at higher intensities, and less at lower intensities, does reflect the response of the human eye to variance in perceived brightness.

Finally, the simple arguments above do not discuss the multiplication of computed values by the energy (radiosity/radiance) at each patch — thus increasing or decreasing the absolute error in computation. Radiance and importance weighted subdivision criteria will account for this effect.

Appendix B

Conservation of Energy over Glossy Reflection

A reflectance distribution function that fully and accurately modeled the physics of reflection, for example at the level of geometric or wave optics, would require no normalization. The BRDF that we employ, however, and those employed in most illumination systems, only crudely model such reflection. These approximations should be normalized to ensure that they satisfy basic constraints of physical consistency — of particular importance in the computation of global illumination is conservation of energy.

It is not necessarily disastrous if conservation of energy is not maintained over a system that models only local illumination. Within such a system light is reflected from at most one surface before entering the eye — this view from the eye must be renormalized in any case, to account for the limited range of intensities provided by the display. It is perhaps for this reason that most reflection functions neglect conservation of energy, as discussed in, and with the exception of, the work presented in [W92].

In the case of global illumination systems, however, conservation of energy is of extreme importance. If energy is not conserved with each reflection, a sequence of multiple reflections may lead to an arbitrarily large increase in the amount of energy within a modeled environment. Normalization for conservation of energy is particularly crucial for the BRDF that we employ, as, unlike in many systems, it does not neglect reciprocal cosine terms, nor, as discussed in Section 2.3, normalization for projected facet area, and thus, correctly, may exceed one.

Conservation of energy may be expressed as the requirement that directional-hemispherical reflectance:

$$\rho(d\omega_i, 2\pi) = \int_{2\pi} f_r(\vec{\omega}_i, \vec{\omega}_r) \cos\theta_r d\omega_r$$

be of value no greater than 1 over all $\vec{\omega}_i$ (Section 2.1.4, [NRHGL77]). Unfortunately, as is discussed in the following section, this integral has proven to be extremely intractable over f_g.

Unable to normalize for conservation of energy directly, we must try to ensure that our function is sufficiently physically consistent to inherently preserve conservation. Such conservation over f_g is examined in Section B.2; as discussed in that section, it is necessary to take into account microfacet self shadowing and masking to maintain conservation of energy over our modeled BRDF.

B.1 Conservation of Energy and Reflectance

Recall glossy reflection function f_g (Section 4.4.1):

$$f_g(\vec{\omega}_i, \vec{\omega}_r) = \frac{\kappa + 2}{8\pi} \frac{\cos^\kappa \theta_m}{\cos\theta_i \cos\theta_r}$$

To ensure that energy is conserved over this BRDF we must compute:

$$\hat{\rho}(d\omega_i, 2\pi) = \int_{2\pi} f_g(\vec{\omega}_i, \vec{\omega}_r) \cos\theta_r d\omega_r$$

and normalize f_g by the inverse of this quantity.

We will present and discuss briefly different formulations for this integral. Unfortunately, direct evaluation of the integral for arbitrary $\vec{\omega}_i$ appears to be extremely difficult over f_g; in practice we have thus employed an alternative, relatively crude scheme for conservation of energy presented in Section B.2 below.

We have:

$$\begin{aligned} \hat{\rho}(d\omega_i, 2\pi) &= \int_{2\pi} \frac{\cos^\kappa \theta_m}{4\cos\theta_i \cos\theta_r} \cos\theta_r d\omega_r \\ &= \frac{1}{4\cos\theta_i} \int_{2\pi} \cos^\kappa \theta_m d\omega_r \end{aligned} \tag{14}$$

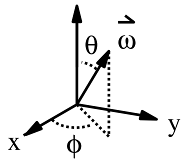


Figure 63: Coordinate Axes

The leading factor of $(\kappa + 2)/2\pi$ has been omitted — normalizing the above expression will introduce a function of κ that subsumes facet normalization. The 4 has been preserved in the divisor, however, as it will simplify a later derivation.

Expanding $d\omega_r$:

$$\hat{\rho}(d\omega_i, 2\pi) = \frac{1}{4\cos\theta_i}\int_0^{2\pi}\int_0^{\frac{\pi}{2}} \cos^{\kappa}\theta_m \sin\theta_r d\theta_r d\phi_r$$

Given the above equation, we must express $\cos\theta_m$ in terms of ϕ_r, θ_r, ϕ_i, θ_i.

Recall that $\cos\theta_m = N \cdot \vec{\omega}_m$, where $\vec{\omega}_m = (\vec{\omega}_i + \vec{\omega}_r)/|\vec{\omega}_i + \vec{\omega}_r|$. Note that:

$$\begin{aligned} |\vec{\omega}_i + \vec{\omega}_r| &= \sqrt{(\vec{\omega}_i + \vec{\omega}_r) \cdot (\vec{\omega}_i + \vec{\omega}_r)} \\ &= \sqrt{2(1 + \vec{\omega}_i \cdot \vec{\omega}_r)} \end{aligned}$$

By the symmetry of f_g, we may assume that $\phi_i = 0$, and thus, by the geometry of Figure 63, we have:

$$\begin{aligned} N &= (0, 0, 1) \\ \vec{\omega}_i &= (\sin\theta_i, 0, \cos\theta_i) \\ \vec{\omega}_r &= (\sin\theta_r \cos\phi_r, \sin\theta_r sin\phi_r, \cos\theta_r) \end{aligned}$$

giving:

$$\cos\theta_m = N \cdot \vec{\omega}_m = \frac{\cos\theta_i + \cos\theta_r}{\sqrt{2(1 + \sin\theta_i \sin\theta_r \cos\phi_r + \cos\theta_i \cos\theta_r)}}$$

The normalization integral is thus:

$$\hat{\rho}(d\omega_i, 2\pi) = \frac{1}{4\cos\theta_i}\int_0^{2\pi}\int_0^{\frac{\pi}{2}} \frac{\sin\theta_r(\cos\theta_i + \cos\theta_r)^{\kappa}}{2^{\frac{\kappa}{2}}(1 + \sin\theta_i \sin\theta_r \cos\phi_r + \cos\theta_i \cos\theta_r)^{\frac{\kappa}{2}}} d\theta_r d\phi_r \quad (15)$$

We have been unable to derive an analytic solution to this integral. In practice, it may be possible to employ numeric integration to tabulate values indexed by θ_i and κ, although there may potentially be problems with convergence for very large κ.

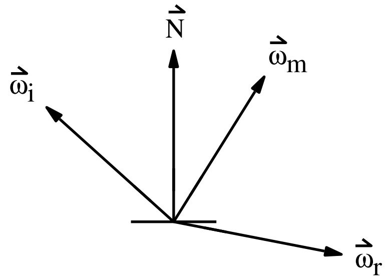


Figure 64: Limit to $\vec{\omega}_m$

We have examined possible simplification of $\hat{\rho}(d\omega_i, 2\pi)$ by expressing it in terms of $d\omega_m$. Recall equation (14):

$$\hat{\rho}(d\omega_i, 2\pi) = \frac{1}{4\cos\theta_i}\int_{2\pi}\cos^{\kappa}\theta_m d\omega_r$$

Let $\theta_{im} = \theta_{rm}$ be the angle between $\vec{\omega}_i$ and $\vec{\omega}_m$; then $d\omega_r = 4\cos\theta_{im} d\omega_m$ [TS67], giving:

$$\hat{\rho}(d\omega_i, 2\pi) = \frac{1}{\cos\theta_i}\int_0^{2\pi}\int_0^{\text{limit}_r(\theta_i,\phi_m)}\cos^{\kappa}\theta_m\cos\theta_{im}\sin\theta_m d\theta_m d\phi_m$$

The upper limit on the inner integral is $\text{limit}_r(\theta_i, \phi_m)$ rather than $\frac{\pi}{2}$, because, for a given $\vec{\omega}_i$ and ϕ_m, θ_m beyond some limit corresponds to $\vec{\omega}_r$ below the horizon (Figure 64). We will express limit_r as a limit on $\cos\theta_m$ rather than θ_m, as the inner integral may easily be rewritten with $d(\cos\theta_m)$ in place of $d\theta_m$.

The limit_r on $\cos\theta_m$ is achieved when $\vec{\omega}_r$ is at the horizon. For $\theta_r = 0$, we have:

$$\begin{aligned}\vec{\omega}_r &= (\cos\phi_r, \sin\phi_r, 0)\\ \vec{\omega}_i &= (\sin\theta_i, 0, \cos\theta_i)\end{aligned}$$

and thus:

$$\cos\theta_m = N\cdot\vec{\omega}_m = \frac{\cos\theta_i}{\sqrt{2(1+\sin\theta_i\cos\phi_r)}}$$

giving the desired limit. We must now express $\cos\phi_r$ in terms of θ_r, ϕ_m:

$$\begin{aligned}\cos\phi_m &= \frac{x_m}{y_m}\\ &= \frac{\sin\theta_i+\cos\phi_r}{\sin\phi_r}\\ &= \frac{\sin\theta_i+\cos\phi_r}{\sqrt{1-\cos^2\phi_r}}\end{aligned}$$

and thus:

$$(1+\cot^2\phi_m)\cos^2\phi_r + (2\sin\theta_i)\cos\phi_r + \sin^2\theta_i - \cot^2\phi_m = 0$$

Solving for $\cos\phi_r$ we have:

$$\cos\phi_r = -sin\theta_i\sin^2\phi_m + \cos\phi_m\sqrt{1-\sin^2\theta_i\sin^2\phi_m}$$

The limit as a function of θ_i and ϕ_m is thus:

$$\text{limit}_r(\theta_i,\phi_m) = \cos\theta_i(2(1+\sin\theta_i(\cos\phi_m\sqrt{1-\sin^2\theta_i\sin^2\phi_m} - \sin\theta_i\sin^2\phi_m)))^{-\frac{1}{2}}$$

We will avoid this limit for two reasons: first, it is rather daunting, and second, it may not actually be applicable.

It is true that:

$$\int_0^{2\pi}\int_0^{\frac{\pi}{2}} \frac{\cos^\kappa\theta_m}{4\cos\theta_i}\sin\theta_r d\theta_r d\phi_r = \frac{1}{\cos\theta_i}\int_0^{2\pi}\int_0^{\text{limit}_r(\theta_i,\phi_m)} \cos^\kappa\theta_m\cos\theta_{im}\sin\theta_m d\theta_m d\phi_m$$

and that these integrals express conservation of energy over the given BRDF. However, the limit_r appearing in the second integral allows insight into an oddity of this conservation.

We are ensuring that energy reflected out from a surface is no greater than that incident in: limit_r enforces the constraint that energy out appear only above the surface. However, as expressed by limit_r, and inherent also in the left hand integral, incident energy somehow avoids microfacets that would not reflect it above the horizon. That is, all incident energy accounted for is "caught" and reflected by facets for which $\vec{\omega}_r$ lies about the surface.

Realistically, incident energy and reflecting facets may not be disallowed in such a fashion, rather, incident energy that is reflected "into" the surface will undergo multiple reflections before finally escaping (Figure 65). Such energy, and other energy experiencing self shadowing after reflection, may be modeled as being scattered diffusely. It should still, however, be taken into account when normalizing for conservation of energy.

Energy incident along a given $\vec{\omega}_i$ may be reflected by any microfacet that is at least partially facing the incidence vector; that is any facet oriented such that $0 \leq \theta_{im} \leq \frac{\pi}{2}$. This constraint may be visualized as a disk perpendicular to $\vec{\omega}_i$ at the point of incidence (Figure 66). If energy is incident from directly above, that is $\theta_i = 0$, then θ_m may range freely from 0 to $\frac{\pi}{2}$, for all ϕ_m, $0 \leq \phi_m \leq 2\pi$. Otherwise, θ_m is unconstrained for

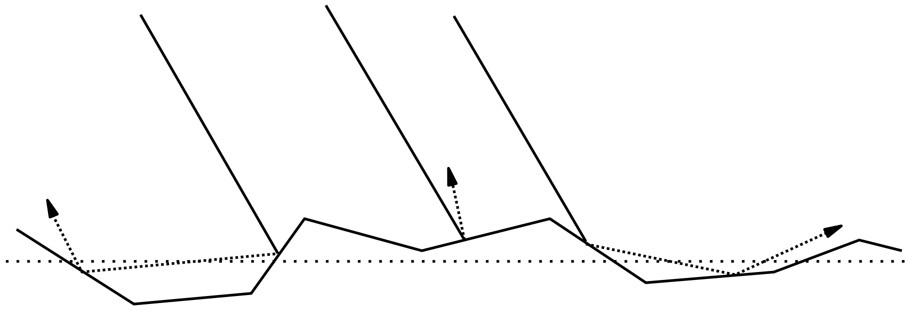


Figure 65: Multiple Reflection

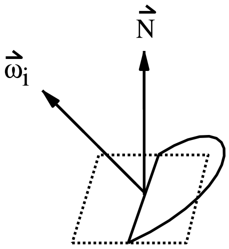


Figure 66: Limit$_m$

$0 \leq \phi_m \leq \frac{\pi}{2}$, $\frac{3\pi}{2} \leq \phi_m \leq 2\pi$, but is bounded by the disk for $\frac{\pi}{2} < \phi_m \leq \frac{3\pi}{2}$. For ϕ_m over the later range, the limit to θ_m is at $\cos\theta_{im} = 0$.

Recall that $\vec{\omega}_i = (\sin\theta_i, 0, \cos\theta_i)$, $\vec{\omega}_m = (\sin\theta_m \cos\phi_m, \sin\theta_m \sin\phi_m, \cos\theta_m)$, then:

$$\cos\theta_{im} = \vec{\omega}_i \cdot \vec{\omega}_m = \sin\theta_m \cos\phi_m \sin\theta_i + \cos\theta_m \cos\theta_i$$

Setting $\cos\theta_m = 0$ and solving yields:

$$\cos\theta_m = -\frac{\sin\theta_i \cos\phi_m}{\sqrt{\cos^2\theta_i + sin^2\theta_i \cos^2\phi_m}}$$

Given the symmetry of f_g, we will compute the normalization integral over $0 \leq \phi_m \leq \pi$, as doubling this quantity will give the desired value. We thus have:

$$\frac{1}{2}\hat{\rho}(d\omega_i, 2\pi) = \begin{array}{lll} \text{(a)} & \displaystyle\int_0^{\pi} \int_0^{\frac{\pi}{2}} \cos^{\kappa}\theta_m \cos\theta_{im} \sin\theta_m d\theta_m d\phi_m & \text{if } \theta_i = 0 \\ \text{(b)} & \begin{array}{l} \frac{1}{\cos\theta_i} \displaystyle\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \cos^{\kappa}\theta_m \cos\theta_{im} \sin\theta_m d\theta_m d\phi_m + \\ \frac{1}{\cos\theta_i} \displaystyle\int_{\frac{\pi}{2}}^{\pi} \int_0^{\text{limit}_m(\theta_i, \phi_m)} \cos^{\kappa}\theta_m \cos\theta_{im} \sin\theta_m d\theta_m d\phi_m \end{array} & \text{otherwise} \end{array}$$

where in the second term of (b) we may recast the integral over $d\cos\theta_m$ and employ $\text{limit}_m(\theta_i,\phi_m) = -\sin\theta_i\cos\phi_m/\sqrt{\cos^2\theta_i+\sin^2\theta_i\cos^2\phi_m}$.

We shall first examine formulation (a):

$$\begin{aligned}
&\int_0^{\pi}\int_0^{\frac{\pi}{2}} \cos^\kappa\theta_m\cos\theta_{im}\sin\theta_m d\theta_m d\phi_m\\
&= \int_0^{\pi}\int_0^{\frac{\pi}{2}} \cos^\kappa\theta_m\sin\theta_m(sin\theta_m\cos\phi_m\sin\theta_i+\cos\theta_m\cos\theta_i)d\theta_m d\phi_m\\
&= \int_0^{\pi}\int_0^{\frac{\pi}{2}} \cos^{\kappa+1}\theta_m\sin\theta_m d\theta_m d\phi_m\\
&= \frac{\pi}{\kappa+2}
\end{aligned}$$

Recall that this solution is for a half hemisphere, and thus the normalization coefficient is $(\kappa+2)/2\pi$. As will be discussed in the following section, it is not a coincidence that this coefficient agrees exactly with the normalization for projected facet area (Section 2.3).

It is interesting to note that, as may easily be shown, the value of the original conservation integral (equation (15)), for $\theta_i = 0$, is $\frac{2\pi}{\kappa+2}(1-\frac{1}{2^{\frac{\kappa}{2}+1}})$. The difference of $\frac{1}{2^{\frac{\kappa}{2}+1}}$ is accounted for by the neglect of energy that is initially reflected below the mean surface. This quantity rapidly approaches 0 as κ increases and fewer facets are distributed with orientation beyond $\theta_m = \frac{\pi}{4}$.

If $\theta_i \neq 0$, we must employ formulation (b). The interior integral for this formulation is:

$$\begin{aligned}
&\int_0^{\frac{\pi}{2}:\text{limit}_m(\theta_i,\phi_m)} \cos^\kappa\theta_m\cos\theta_{im}\sin\theta_m d\theta_m\\
&= \int_0^{\frac{\pi}{2}:\text{limit}_m(\theta_i,\phi_m)} \cos^\kappa\theta_m\sin\theta_m(sin\theta_m\cos\phi_m\sin\theta_i+\cos\theta_m\cos\theta_i)d\theta_m
\end{aligned}$$

The second term of this integral has immediate solution: $\frac{1}{\kappa+2}\cos^{\kappa+2}\theta_m|_0^{\frac{\pi}{2}:\text{limit}_m(\theta_i,\phi_m)}\cos\theta_i$. We are left with the first term, disregarding the leading $\cos\phi_m\sin\theta_i$:

$$\int_0^{\frac{\pi}{2}:\text{limit}_m(\theta_i,\phi_m)} \cos^\kappa\theta_m\sin^2\theta_m d\theta_m$$

This integral may be solved analytically for arbitrary κ, employing identities:

$$\begin{aligned}
\int\cos^\kappa\theta\sin^2\theta d\theta &= \frac{-sin\theta\cos^{\kappa+1}\theta}{\kappa+2}+\frac{1}{\kappa+2}\int\cos^\kappa\theta d\theta\\
\int\cos^\kappa\theta d\theta &= \frac{\cos^{\kappa-1}\theta\sin\theta}{\kappa}+\frac{\kappa-1}{\kappa}\int\cos^{\kappa-2}\theta d\theta
\end{aligned}$$

These may yield values that are fairly tractable for limits of integration $\theta_m = (0, \frac{\pi}{2})$, as all terms of the form $\frac{\cos^b \theta \sin \theta}{d} c$ will be eliminated as equal to zero. However, for large κ, an arbitrary upper limit will yield a long residue of terms of the form:

$$\frac{(c-1)(c-3)(c-5)\cdots}{c(c-2)(c-4)\cdots}(\text{limit})^d \sqrt{1-(\text{limit})^2}$$

Given $\text{limit}_m = -\sin\theta_i \cos\phi_m / \sqrt{\cos^2\theta_i + \sin^2\theta_i \cos^2\phi_m}$, the exterior integral faces considerable complexity. For example, for $\kappa = 35$, the first term of the interior integral yields:

$$\sqrt{1 - \frac{\cos^2\phi_m \sin^2\theta_i}{\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i}} \left(-\frac{4294967296}{755505013725} + \frac{\cos^{36}\phi_m \sin^{36}\theta_i}{37\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{18}} - \right.$$
$$\frac{\cos^{34}\phi_m \sin^{34}\theta_i}{1295\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{17}} - \frac{34\cos^{32}\phi_m \sin^{32}\theta_i}{42735\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{16}} -$$
$$\frac{1088\cos^{30}\phi_m \sin^{30}\theta_i}{1324785\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{15}} - \frac{2176\cos^{28}\phi_m \sin^{28}\theta_i}{2561251\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{14}} -$$
$$\frac{8704\cos^{26}\phi_m \sin^{26}\theta_i}{9879111\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{13}} - \frac{226304\cos^{24}\phi_m \sin^{24}\theta_i}{246977775\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{12}} -$$
$$\frac{1810432\cos^{22}\phi_m \sin^{22}\theta_i}{1893496275\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{11}} - \frac{3620864\cos^{20}\phi_m \sin^{20}\theta_i}{3614856525\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{10}} -$$
$$\frac{14483456\cos^{18}\phi_m \sin^{18}\theta_i}{13736454795\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{9}} - \frac{1703936\cos^{16}\phi_m \sin^{16}\theta_i}{1526272755\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{8}} -$$
$$\frac{27262976\cos^{14}\phi_m \sin^{14}\theta_i}{22894091325\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{7}} - \frac{4194304\cos^{12}\phi_m \sin^{12}\theta_i}{3270584475\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{6}} -$$
$$\frac{16777216\cos^{10}\phi_m \sin^{10}\theta_i}{11992143075\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{5}} - \frac{33554432\cos^{8}(\phi_m)\sin^{8}(\theta_i)}{21585857535\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{4}} -$$
$$\frac{268435456\cos^{6}(\phi_m)\sin^{6}(\theta_i)}{151101002745\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{3}} - \frac{536870912\cos^{4}(\phi_m)\sin^{4}(\theta_i)}{251835004575\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)^{2}} -$$
$$\left. \frac{2147483648\cos^{2}\phi_m \sin^{2}\theta_i}{755505013725\left(\cos^2\theta_i + \cos^2\phi_m \sin^2\theta_i\right)} \right) + \frac{4294967296}{755505013725}$$

It may be possible to make headway with these integrals by expressing them via some form of hypergeometric function. Numeric tabulation might also be employed for an approximate solution. The implemented system employs instead the scheme described in the following section.

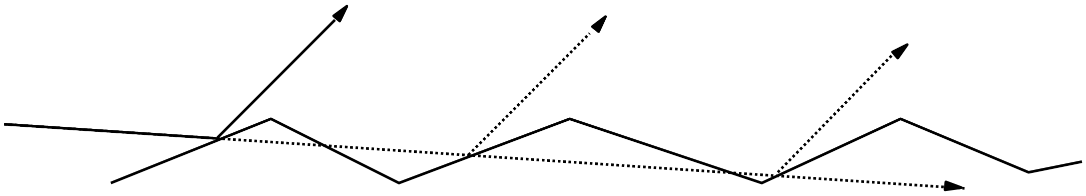


Figure 67: Self Shadowing

B.2 Self Shadowing and Conservation of Energy

Recall the development of glossy reflection function f_g given in Section 2.3; as discussed in that section, we normalized the microfacet distribution underlying f_g to account for projected facet area over the modeled surface. The resulting BRDF thus describes a physically realizable surface at the level of microfacets. Given a physically consistent reflecting surface, it may appear that conservation of energy over reflection will inherently follow, the argument being that the physics of reflection is fully modeled, at least at the microfacet level. Integrating reflectance over an area-normalized BRDF for an arbitrary, oblique θ_i may, however, yield a value that is greater than 1.

The factor that is missing is that of self shadowing. Consider energy incident to a surface at θ_i near $\frac{\pi}{2}$ (Figure 67). Multiple facets may be oriented toward the incidence vector; if self shadowing is not taken into account, energy will be incident to the surface at several points, and thus multiply reflected (and multiplied) before "penetrating" below the surface.

The degree to which masking occurs is dependent upon incident angle and facet orientation. Facets with relatively low θ_m are less conducive to masking than those of greater slope, as they cast a smaller shadow over the surface; similarly, low angle of incidence indicates few surface-vector intersections. Masking is thus dominant for highly oblique incidence over facets oriented near the horizon. Given the low proportion of obliquely oriented facets distributed over a near specular surface, geometric self shadowing should not become critical until relatively high θ_i.

This dependence is reflected in the shadow function developed by Sancer for surfaces described by Gaussian height distributions [H89]. This function employs both θ_i and θ_r as arguments, as it takes into account masking of the reflection vector, as well as self

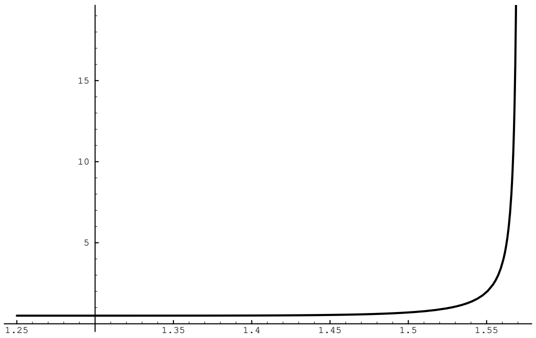


Figure 68: $C_1(\theta) \quad m = 0.1$

shadowing over incidence:

$$
\begin{aligned}
S_h(\theta_i, \theta_r) &= \frac{1}{C_1(\theta_i) + C_1(\theta_r)} \\
C_1(\theta) &= \tfrac{1}{2}((\sqrt{\pi C_2(\theta)}e^{C_2(\theta)}))^{-1} + \mathrm{erf}(\sqrt{C_2(\theta)}) \\
C_2(\theta) &= (2m^2 \tan^2 \theta)^{-1}
\end{aligned}
$$

where m is the average slope over microfacets.

$C_1(\theta)$ measures the degree of masking that occurs over a surface: as its magnitude increases, the value of S_h, and thus the BRDF, approaches 0. For a fairly specular surface $(m = 0.1)$, C_1 stays relatively constant at 0.5 over most angles of incidence, but rapidly approaches infinity for θ_i near $\frac{\pi}{2}$ (Figure 68). The graph of S_h against θ_i, for $\theta_r = 0.5$, shows a corresponding rapid decrease from 1 to 0 near the horizon (Figure 69).

In Figure 70 we have graphed S_h over both θ_i and θ_r for $m = 0.1$. The rapid fall off is preserved over the entirety of both ranges. The final graph (Figure 71) shows S_h as a factor of θ_i and m for $\theta_r = 0.5$. For $m = 1$, S_h experiences a relatively gentle decline; however value in the vicinity of 1 is maintained until near the horizon from about $m = 0.25$ on.

Given the sudden fall off that S_h exhibits near the horizon for near specular surfaces, the implemented system simply clamps S_h to 0 when θ_i or θ_r exceeds a preset θ_c. This scheme should preserve conservation of energy, and serves as a crude approximation to the shadowing function; however, in full implementation a better strategy, particularly in view of the well behaved nature of S_h, would be to employ a much fuller tabulation of

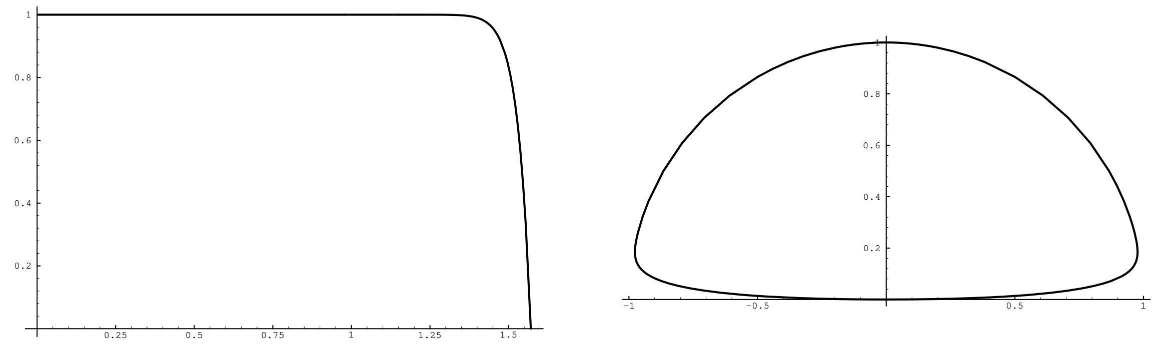


Figure 69: $S_h(\theta_i,\ \theta_r = 0.5) \quad m = 0.1$

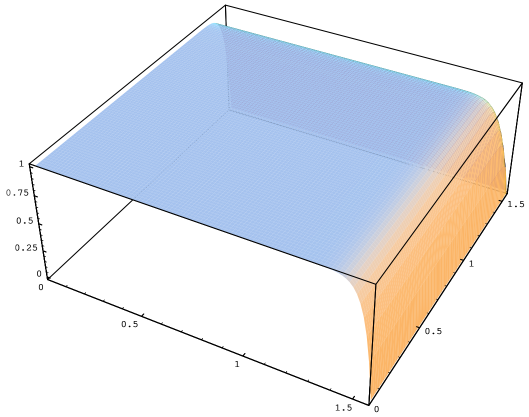


Figure 70: $S_h(\theta_i, \theta_r) \quad m = 0.1$

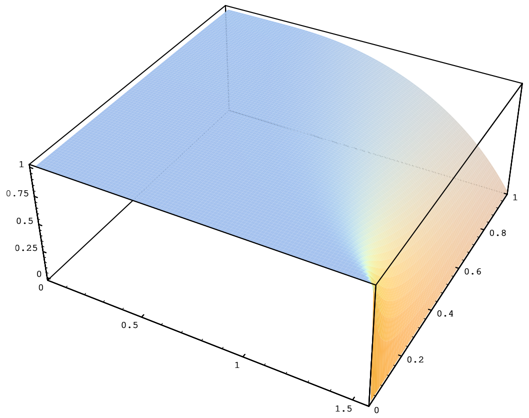


Figure 71: $S_h(\theta_i,\ \theta_r = 0.5) \quad m = 0 \ldots 1$

the function. The implemented shadowing scheme is equivalent to employing a table with only one entry. Given the monotonicity of S_h over θ, the shadowing function could be incorporated into the error analysis presented in Sections 4.4.2 and A.1.

Finally, we note that masking of incident energy is not a factor when $\theta_i = 0$. It is for this reason that reflectance equation (a) of Section B.1 yields $(\kappa + 2)/2\pi$ — the normalization coefficient for a physically consistent surface.