ACCURATE AND RELIABLE ALGORITHMS
FOR
GLOBAL ILLUMINATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Daniel Lischinski

August 1994

ACCURATE AND RELIABLE ALGORITHMS
FOR
GLOBAL ILLUMINATION

Daniel Lischinski, Ph.D.

Cornell University 1994

The simulation of global illumination is one of the most fundamental problems in computer graphics, with applications in a wide variety of areas, such as architecture and lighting design, computer-aided design, and virtual reality. This problem concerns the transport of light energy between reflective surfaces in an environment. During the past decade, radiosity has become the method of choice for simulating global illumination in diffuse environments.

Despite much recent progress in efficiency and applicability of radiosity methods, there are several very important open issues remaining: 1) Radiosity images suffer from many visual artifacts, resulting from lack of reliable automatic discretization algorithms; and 2) Current radiosity algorithms do not provide the user with guaranteed bounds or reliable estimates of the approximation errors. As a result, current radiosity systems require very careful and time-consuming user intervention in the discretization process, and the accuracy of the resulting solutions can only be assessed by visual appearance.

This thesis presents new radiosity algorithms for diffuse polyhedral environments that address the open problems mentioned above. First, we have improved and combined together two recently developed radiosity approaches: hierarchical radiosity and discontinuity meshing. An improved hierarchical ra-

diosity algorithm that is based on a discontinuity-driven subdivision strategy to achieve better numerical accuracy and faster convergence is used to compute the global distribution of light energy in an environment. Then, a new algorithm based on discontinuity meshing uses the hierarchical solution to reconstruct a visually accurate approximation to the radiance function. Thus, results of high visual quality can be obtained even from coarse global illumination simulations. The solution is performed entirely in object-space, which enables users to "walk" through high-fidelity shaded virtual environments in real time, using appropriate display hardware.

Second, we have developed algorithms that compute *a posteriori* error bounds and estimates for local and total errors in hierarchical radiosity solutions. A conservative algorithm computes guaranteed upper bounds on the errors. A non-conservative algorithm is capable of computing more realistic error estimates more efficiently. These error estimates are used in a new error-driven refinement strategy for hierarchical radiosity, resulting in faster convergence.

Biographical Sketch

Dani Lischinski was born in Moscow, former Soviet Union, on November 18th, 1962. In 1973 his family immigrated to Israel and settled in Jerusalem.

In 1980, Dani graduated from the May Boyer High School in Jerusalem and joined the ranks of the Israeli Defense Army for a period of five years. He is currently holding the rank of captain in the reserve.

Concurrently with his two last years of military service, Dani began his studies towards a BSc in Computer Science and Chemistry at the Hebrew University in Jerusalem. He graduated *summa cum laude* in 1987.

From 1987 to 1989, Dani was working on his MSc degree in Computer Science at the Hebrew University. During that time, he became exposed to computer graphics, particularly to the area of ray-tracing, which resulted in his Master's thesis: "Improved Techniques for Ray Tracing Parametric Surfaces."

In 1989, Dani entered the PhD program in Computer Science at Cornell University, where he spent the next five years researching accurate and reliable radiosity algorithms.

To Ruti and Maya

Acknowledgements

First, I would like to thank my family: my wife Ruti, and my little daughter Maya, for all their love, patience and support. I could not have done this without them. I would also like to thank my parents for their love and understanding: being so far away from their son and their first grand-daughter for so long could not have been easy for them.

I am deeply grateful to my advisor, Professor Donald P. Greenberg for his enthusiasm and support. Don has motivated me to perform the research described in this thesis, and along the way has provided many suggestions that greatly improved my papers and my presentations. Don has created a truly extraordinary environment at the Program of Computer Graphics, and I thank him for giving me the opportunity to work and study there.

Thanks also go to my other two committee members: Professors Charles Van Loan and Lars Wahlbin, for serving on my committee, for teaching some of the best courses I have taken at Cornell, and for always being willing to help.

Next, I would like to thank Filippo Tampieri — my friend, officemate, and co-author on several papers. Together with Filippo we have laid down the foundations on which the work described in this thesis was built. In particular, Section 2.4 and parts of Chapter 4 were written together by both of us for the papers that we have co-authored. I am also grateful to Filippo for letting me use several figures that appeared in his thesis.

I am also grateful to the other two global illuminators in the lab: Jim Arvo and Brian Smits. Jim has been a great resource not only for me, but for all the students in the lab. He was always willing to discuss global illumination, read my paper drafts, and help me with my presentations. Brian and I had many valuable discussions on hierarchical radiosity, and I thank him for coming up with the derivation of Equation (5.10) just a few days before the Siggraph submission deadline.

Thanks to Ben Trumbore, the lab's "cruise director", for all the parties and the picnics that he organized, for being my squash partner, and for his help on many occasions.

Many thanks go to the PCG staff: Ellen French, Fran Brown, Hurf Sheldon, Jim Ferwerda, and Jonathan Corson-Rikert for their support in too many aspects to mention.

Thanks to Richard Lobb for many discussions and for his invaluable comments on paper drafts and presentations.

I am also grateful to all the students in the lab. Each and every one of them has contributed in some way to making my five years here an enjoyable and a productive experience. In particular, I would like to thank Chris Schoeneman for the WM graphics library, Dan Kartch for LaTeXsupport and for his food seminars, Gene Greger for modeling the Mackintosh room and for implementing the walkthrough program, Bretton Wade for writing the Motif front-end for my radiosity program, and to Matt Hyatt, Albert Dicrutallo, and Quay Thompson for their wonderful models.

Thanks to David Salesin for collaborating with me on a paper, and for offering me a place to go to from here.

Thanks to Ari Rappoport for keeping in touch and for providing me access to a computer during my visit to Israel in 1992, enabling me to finish my Siggraph submission on time.

Thanks to Paul Chew for explaining me about constrained Delaunay triangulations, and for teaching me computational geometry.

This work was supported by the National Science Foundation grant "Interactive Input and Display Techniques" (CCR-8617880) and by the NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219). I would also like to acknowledge the generous equipment grant from Hewlett Packard Corporation on whose workstations this research was conducted.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Photorealism and Global Illumination

Photorealistic image synthesis is one of the most important fields of computer graphics. Given a geometrical and physical description of a three-dimensional scene (environment), the goal of photorealistic image synthesis is to produce images that would be indistinguishable from photographs taken of the actual scene. Applications of photorealistic image synthesis arise in the fields of architecture and lighting design, computer-aided design, flight simulators, virtual reality, entertainment (computer animation, special effects), and the creative arts.

There are three main tasks involved in the simulation process: (i) obtaining accurate input to the simulation; (ii) accurately and efficiently simulating the interaction of light with the environment; and (iii) mapping the results to output devices. This thesis is concerned with the second of these three tasks. This problem is known as the *global illumination problem* since, in general, the light emitted from the light sources can bounce off various surfaces several times before reaching a given point. Thus, the illumination at any given point is determined by many other surfaces in the environment.

Solving the global illumination problem is important not only for photorealistic image synthesis, but also in a number of other disciplines such as radiative heat transfer, remote sensing, machine vision ("shape from shading"), and infrared signature analysis. However, the simulation of global illumination in the context of photorealism is uniquely challenging, as the resulting images must appear convincing to the human visual system, an extremely sensitive instrument that is not yet sufficiently understood.

Existing methods for solving the global illumination problem can be classified into two broad categories:

View-dependent methods are geared towards solving the problem at a set of locations on the image plane (e.g. pixel centers). These methods typically use ray tracing and Monte-Carlo techniques. The concepts of solution and image are nearly synonymous in this context, and thus a new solution is necessary for each image.

View-independent methods compute an approximation to the radiance function across each surface in the environment. Most of these methods are derivatives of the radiosity method. In this approach the solution and the image are two separate entities: many different images can be produced by different projections of the same solution onto the image plane.

Of course, many forms of hybrids of these two approaches are possible. Each approach has its own advantages and disadvantages that make it more suitable for certain applications and less suitable for others.

This thesis deals with view-independent global illumination simulations.

Such simulations are particularly advantageous when many different views of the same static environment are to be displayed. The main applications in which the need for multiple views or "walk-through"s arises are: architectural design, interior and lighting design, illumination engineering, and virtual reality.

1.2 Radiosity Methods

To date, most view-independent methods have been derived from the radiosity method that was originally developed to solve radiative heat transfer problems [SPAR63b]. Computer graphics researchers adopted this method to compute the global illumination in diffuse polyhedral environments [GORA84, COHE85,NISH85]. Subsequent research has concentrated on improving the efficiency of the method [COHE86,COHE88,HANR91] as well as on extending the method to handle non-diffuse surfaces [IMME86,WALL87,SHAO88,SILL89, RUSH90,SILL91] and participating media [NISH87,RUSH87].

Despite this impressive progress in efficiency and applicability, there are still several important open problems remaining in radiosity methods:

Visual artifacts: Radiosity images suffer from many visual artifacts such as Mach bands, light and shadow leaks, jagged shadow boundaries, and missing shadows. There artifacts are the result of inadequate discretization. There are currently no automatic discretization algorithms for radiosity that are both reliable and efficient.

As the numerical accuracy of the solution increases, visual artifacts diminish. Unfortunately, experience has shown that the human visual system is extremely sensitive to small artifacts that are difficult to quantify. The simulated environments can be very complex and, therefore, the computation of ultra-accurate solutions is generally impractical.

Lack of error bounds: Radiosity methods are only capable of approximating the solution; thus, reliable error bounds estimates are crucial for assessing the acceptability of a particular solution, as well as for automatic adaptive refinement. Current radiosity algorithms do not provide the user with guaranteed bounds on the total approximation error, or reliable estimates of the total error. As a result, the accuracy of the solution can only be judged by its visual appearance. Various heuristics have been used to guide adaptive refinement, but these heuristics are not completely reliable.

As a result of these deficiencies, radiosity systems are seldom user-friendly and require massive user intervention: typically, a time consuming trial-and-error process is required to produce an image that looks right. Baum *et al.* [BAUM91] and Haines [HAIN91a] provide good discussions and more details regarding the various pitfalls of radiosity methods.

Because of the above problems, radiosity has not yet become a standard tool in the hands of its intended users, and almost all commercial systems that offer photorealistic rendering capabilities use only ray tracing.

1.3 Contributions

This thesis presents new radiosity algorithms that overcome the open problems described in the previous section. These algorithms were designed and implemented as a part of a radiosity system with the following properties:

- high visual accuracy through automatic and reliable meshing;

- guaranteed error bounds or reliable error estimates;
- faster convergence than previous methods;
- easy user control through a small number of intuitive parameters;

Specifically, progress has been achieved in two orthogonal directions:

First, we have improved and combined together two recently developed radiosity approaches: hierarchical radiosity [HANR91] and discontinuity meshing [HECK92,LISC92]. An improved hierarchical radiosity algorithm that uses a discontinuity-driven subdivision strategy to achieve better numerical accuracy and faster convergence is used to compute the global distribution of light energy in an environment. Then, a new algorithm based on discontinuity meshing uses the hierarchical solution to reconstruct a visually accurate approximation to the radiance function. Thus, results of high visual quality can be obtained even from coarse global illumination simulations. The solution is performed entirely in object-space, which enables users to walk through high-fidelity shaded virtual environments in real-time, using the proper display hardware.

Second, we have developed algorithms that compute *a posteriori* error bounds and estimates for local and total errors in hierarchical radiosity solutions. A conservative algorithm computes guaranteed upper bounds on the errors. A non-conservative algorithm is capable of computing more realistic error estimates. These error estimates are used in a new error-driven refinement strategy for hierarchical radiosity, resulting in faster convergence.

The common thread throughout the research presented in this thesis is that through better understanding of the sources of error in radiosity simulations and through improved error control, we are able not only to achieve more accurate results, but also to achieve them faster than previously possible. Together, our contributions constitute a significant step forward towards high-fidelity global illumination simulations and photorealistic image synthesis.

1.4 Organization

This thesis is structured as follows. Chapter 2 provides the necessary background by presenting the rendering equation and discussing the properties of radiance functions. A survey of previous work is given in Chapter 3. Chapter 4 describes the combination of hierarchical radiosity with discontinuity meshing. Chapter 5 presents an algorithm for computing guaranteed error bounds and realistic error estimates for radiosity solutions. Chapter 6 concludes the thesis with a summary of the current contribution and discusses direction for future research.

Chapter 2

Background

2.1 The Rendering Equation

The distribution of radiant energy in an environment is governed by a Fredholm integral equation of the second kind [HECK91a]. This equation was first presented to the computer graphics community by Jim Kajiya [KAJI86], who coined the name *rendering equation*. Here we present a slightly different formulation of the rendering equation using radiometric terms:

$$L(x,x'') = L^e(x,x'') + \int_{\mathcal{S}} f_r(x',x,x'')L(x',x)\frac{\cos\theta_i \cos\theta_j}{r^2}v(x,x')dA', \tag{2.1}$$

where

$L(x,x'')$ is the *radiance* (the radiant flux per unit solid angle per unit projected area) leaving a differential surface element centered at x in the direction of the point x''. The units of radiance are watts per steradian per square meter ($Wsr^{-1}m^{-2}$).

$L^e(x,x'')$ is the radiance emitted from x towards x''.

$\mathcal{S}$ is the union of all the surfaces in the environment.

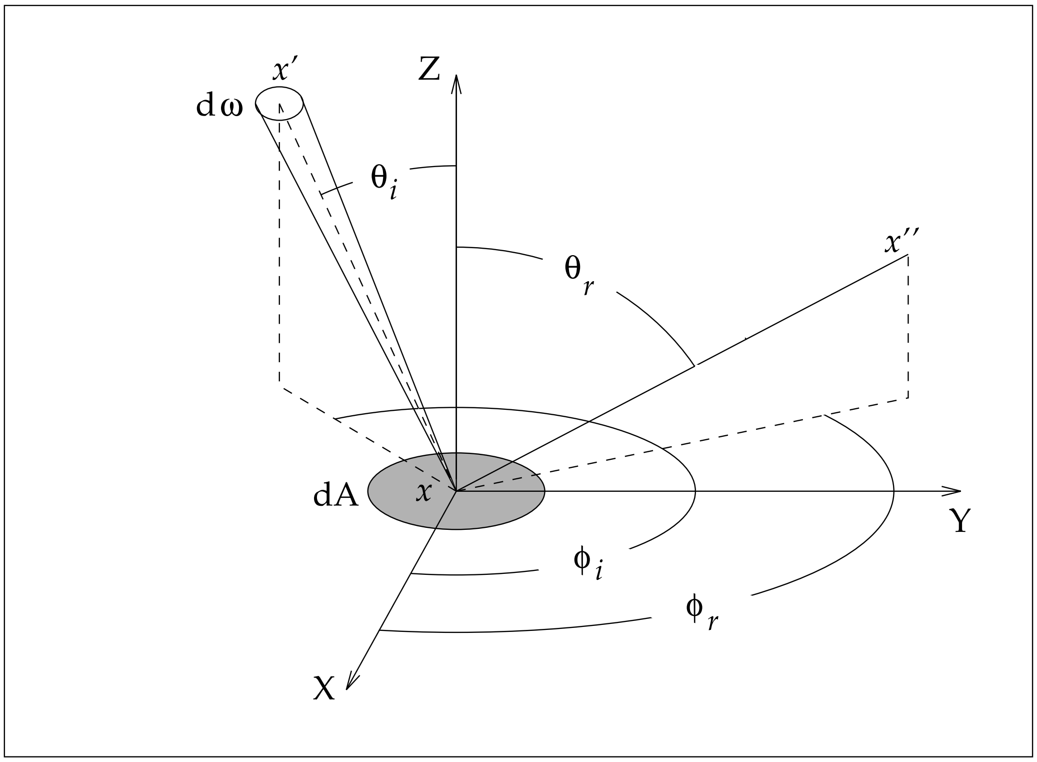


Figure 2.1: Geometry for BRDF's

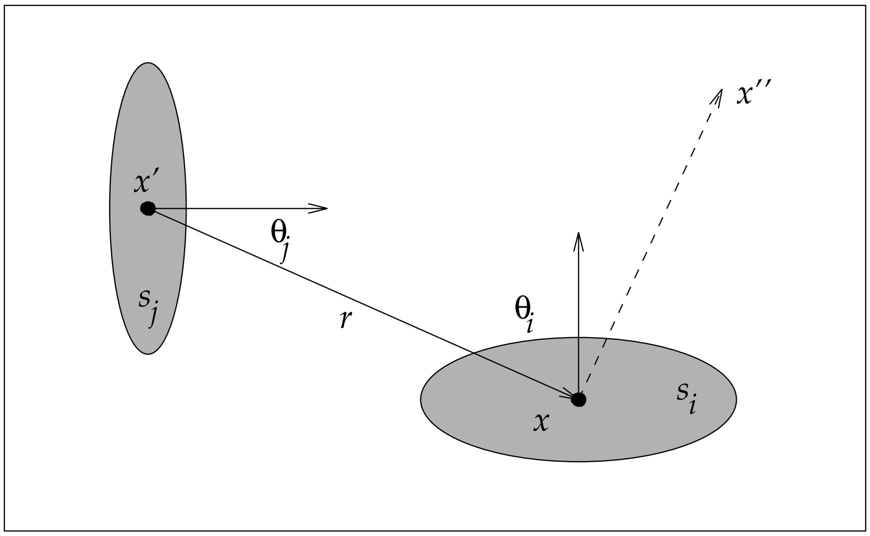


Figure 2.2: Geometry for Equation (2.1)

$f_r(x', x, x'')$ is the *bidirectional reflectance-distribution function (BRDF)* at point x, defined as the ratio of the radiance reflected in the direction of x'' to the radiant flux density incident from the direction of x' through a differential solid angle $d\omega$ (see Figure 2.1). The BRDF has units of inverse steradians.

θ_i and θ_j are the angles between the surface normals at x and x' and the line connecting the two points (see Figure 2.2).

r is the distance between x and x';

$v(x, x')$ is a visibility term; it is 1 if x and x' are visible to each other, and 0 otherwise;

dA' is a differential surface area at x'.

The radiances L, L^e, and the BRDF f_r all depend on the light wavelength λ. This dependency is omitted for simplicity of notation. This form of the rendering equation excludes participating media, transparency, and energy transfer between different wavelengths (for instance, by absorption and re-emission at surfaces). For a detailed derivation of the equation the reader is referred to Pat Hanrahan's chapter in the recent book by Cohen and Wallace [COHE93].

If all the surfaces are ideal diffuse (Lambertian) reflectors, the radiance L is equal in all the outgoing directions from a given point x, and Equation (2.1) can be simplified to:

$$L(x) = L^e(x) + f_r(x) \int_S L(x') \frac{\cos\theta_i \cos\theta_j}{r^2} v(x, x') dA' \qquad (2.2)$$

This equation will be referred to as the *diffuse rendering equation*.

In the computer graphics literature, the diffuse global illumination problem has traditionally been formulated in terms of radiosity [GORA84]. *Radiosity*,

$B(x)$, is the total radiant flux density leaving a differential surface element centered at x. For ideal diffuse surfaces it is easy to show that $B(x) = \pi L(x)$, which yields the (continuous) radiosity equation:

$$B(x) = E(x) + \rho(x) \int_S B(x') \frac{\cos\theta_i \cos\theta_j}{\pi r^2} v(x,x') dA' \qquad (2.3)$$

where $E(x)$ is the emitted radiosity at x, and $\rho(x) = \pi f_r(x)$ is the *reflectance*, i.e., the fraction of incident radiant energy that is reflected at x. Reflectance is a unitless scalar in the interval $[0,1]$.

2.2 Solution Methods

One basic approach towards solving the global illumination problem is the use of Monte-Carlo ray tracing to solve Equation (2.1) at locations and directions determined by the view specification [KAJI86,WARD88,SHIR90a]. These global illumination methods are typically referred to as *view-dependent*, and the concepts of *image* and *solution* are synonymous in this case.

This thesis is concerned with radiosity methods, which compute an approximation to the radiance function across the surfaces in the environment. Solutions produced by radiosity methods can be used to render images from any viewpoint, and therefore these methods are referred to as *view-independent*.

Heckbert and Winget [HECK91b] have shown that the radiosity method that originated in the field of radiative heat transfer [SPAR63b] is essentially a simple finite element method. The idea behind the finite element formulation is to look for an approximate solution $\widehat{B}(x)$ in some known finite-dimensional function space X_n, a subspace of the infinite-dimensional function space X that contains the exact solution $B(x)$.

The basic steps of the finite element approach are [COHE93]:

1. Subdivide the surfaces into a *mesh* of *elements*.

2. Select locations or *nodes* on the elements and associate a *basis function* $N_i(x)$ with each node. The sum of the basis functions (weighted by the corresponding nodal values B_i) yields an approximation $\widehat{B}(x)$ to the radiosity function:
$$\widehat{B}(x) = \sum_{i=1}^{n} B_i N_i(x)$$

3. Select a finite error metric to minimize by projecting the *residual function*
$$r(x) = \widehat{B}(x) - E(x) - \rho(x) \int_S \widehat{B}(x') \frac{\cos\theta_i \cos\theta_j}{\pi r^2} v(x,x') dA' \tag{2.4}$$
onto a set of *weighting functions* $\{W_i(x)\}$. Thus, the infinite dimensional problem is cast into a finite set of linear equations.

4. Compute the coefficients of the linear system.

5. Solve the resulting system of equations for the unknown nodal values.

The Galerkin formulation [DELV85,HECK91b,ZATZ93], for example, selects the same basis functions used to approximate the radiosity function as the weighting functions. Thus, a solution $\widehat{B}(x)$ is sought that satisfies
$$\int_S r(x) N_i(x)\, dA = 0, \quad \text{for } i \in [1,n].$$

Substituting the definitions of $r(x)$ and $\widehat{B}(x)$ and regrouping results in:
$$\begin{aligned}\left[\sum_{j=1}^{n} B_j \left[\int_S N_i(x)N_j(x)dA - \int_S N_i(x)\rho(x)\int_S N_j(x')\frac{\cos\theta_i\cos\theta_j}{\pi r^2} v(x,x')dA'dA\right]\right] \\ - \int_S E(x)N_i(x)dA = 0\end{aligned} \tag{2.5}$$

There are n such expressions, one for each $i \in [1, n]$, so we obtain a linear system of size n with the nodal values B_i as the unknowns.

Most radiosity algorithms use constant basis functions,

$$N_i(x) = \begin{cases} 1 & \text{if } x \text{ is inside element } i \\ 0 & \text{otherwise} \end{cases}$$

This simplifies Equation (2.5) to

$$\left[\sum_{j=1}^{n} B_j \left[\delta_{ij} A_i - \rho_i A_i F_{ij}\right]\right] - E_i A_i = 0, \tag{2.6}$$

where E_i is the area-averaged emission for element i, ρ_i is the average reflectance for element i, and F_{ij}, called the *form factor*, is given by

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} v(x, x') dA_j dA_i. \tag{2.7}$$

The form factor represents the fraction of energy that leaves element i and arrives directly at element j. Because of energy conservation it follows that

$$\sum_j F_{ij} \leq 1.$$

From definition (2.7) it is easy to see that form factors satisfy a reciprocity relationship:

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

Dividing Equation (2.6) through by A_i and using the reciprocity relationship we derive the simplified equation

$$\sum_{j=1}^{n} B_j \left[\delta_{ij} - \rho_i F_{ij}\right] = E_i,$$

which can be written in matrix notation as

$$MB = (I - K)B = E, \tag{2.8}$$

where $K_{ij} = \rho_i F_{ij}$, and I is the identity matrix.

2.3 Sources of Error

Sources of error in global illumination algorithms can be classified into three categories [ARVO94a]:

Perturbed Boundary Data: Boundary data consists of surface geometry, reflectance functions, and emission functions, all of which may be perturbed by errors in measurement, or by simplifications made for computational efficiency.

Discretization Error: Discretization error is introduced by replacing the continuous rendering equation with a finite-dimensional linear system. This error depends mostly on the distance between the actual solution B and the finite-dimensional subspace X_n that contains the approximate solution $\widehat{B}$.

Computational Errors: Computational errors perturb the finite-dimensional linear system through imprecise form factors, inner products, visibility, etc., as well as by halting iterative solvers after a finite number of steps.

In this thesis we will concentrate on the two latter sources of error, since boundary data is usually provided as input to radiosity algorithms, and thus, the algorithms have no control over errors of the first category.

2.4 Properties of Radiance Functions

Radiosity methods solve the global illumination problem by computing an approximation to the radiance function over the surfaces of an environment. In this section we examine various properties of these functions. This will enable us to better evaluate and criticize existing radiosity methods, as well as motivate the algorithms described in this thesis. We limit the analysis to the domain of polygonal environments where the polygons do not interpenetrate.

Consider the simple case of a receiving surface R illuminated by a single source surface S. From Equation (2.2) it follows that the radiance function on the receiver is given by:

$$L_R(x) = f_r(x) \int_{x' \in S} L_S(x') \frac{\cos\theta_i \cos\theta_j}{r^2} v(x, x') dA'. \tag{2.9}$$

Since $v(x, x')$ can only assume values of 0 or 1, we can take it out of the integrand by limiting the range of integration to the parts of S visible from x, which we denote by $V_S(x)$:

$$L_R(x) = f_r(x) \int_{x' \in V_S(x)} L_S(x') \frac{\cos\theta_i \cos\theta_j}{r^2} dA'. \tag{2.10}$$

Let us assume, for now, that the radiance function across S is smooth (C^∞), as is the case with typical primary sources. We will also assume that there is no contact between R and S, so that $r \neq 0$. Under these assumptions the integrand is a smooth function, and hence L_R will be smooth itself, so long as the visibility of the source is constant or varies smoothly. It has been conjectured that for a convex source L is unimodal, i.e., it has a single maximum, and it is monotonically decreasing as a function of the distance from that maximum [DRET93]. Abrupt changes in visibility, however, will introduce discontinuities of various orders into L_R. In the rest of this section we will enumerate the various possible types of discontinuities and explain their causes. A similar discussion first appeared in Heckbert's thesis [HECK91a], and we therefore adhere to the terminology and notation used there.

We will use the notation that a function L has a D^k discontinuity at x, if it is C^{k-1} but not C^k there. If the radiance over the light source is smooth, the radiance function on the receiver can have D^0, D^1, and D^2 discontinuities.

2.4.1 D^0 Discontinuities

By definition, D^0 discontinuities are value (or jump) discontinuities in the function itself. They are introduced by edges or vertices of occluders (or the light source itself) lying on the receiver. Thus, they can occur either along line segments, or in a pointwise fashion. An example is shown in Figure 2.3.

D^0 discontinuities also occur along shadow boundaries cast by point light sources. In this case no penumbra regions are present, and the transition from the umbra into the unoccluded area is a discontinuous one. For conciseness, only finite area light sources will be treated in this thesis. The correct treatment of point light sources is simpler and is a straightforward extension of the described techniques.

In an environment with m edges, no interpenetrating polygons, and no point light sources, there can be $O(m^2)$ D^0 discontinuities, as it is possible for each of $O(m)$ edges to lie on $O(m)$ faces. However, in typical environments without interpenetration the number of D^0 segments is roughly linear in m, because each edge typically lies on only a small number of input surfaces.

2.4.2 D^1 and D^2 Discontinuities

Discontinuities in the first and second derivatives of the function are referred to as D^1 and D^2 discontinuities, respectively. They are caused by objects that intervene between the source and the receiver, causing abrupt changes in the visibility of the former as seen from the latter. Such visibility changes (known as *visual events*) have been thoroughly studied in the computational geometry and computer vision literature [GIGU90,GIGU91]. In particular, it has been shown that all the visual events in a polyhedral environment can be classified into two types:

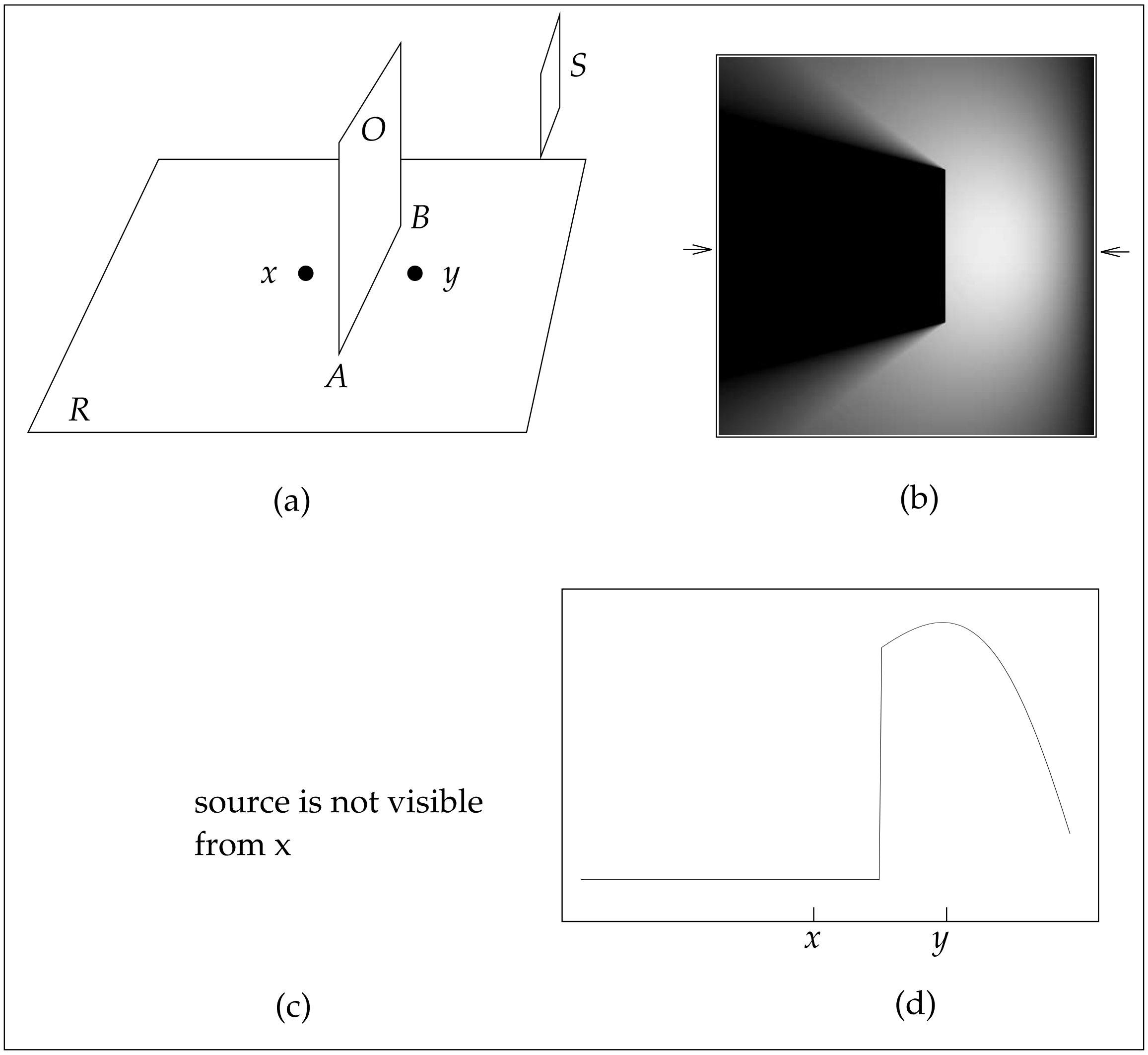


Figure 2.3: An example of a D^0 discontinuity. (a) Edge AB of the occluder O lies on the receiver R. (b) The radiance function over R. (d) The radiance function along the line through x and y.
Points on R immediately to the left of AB cannot see the source S and the radiance there is zero. Immediately to the right of AB, the entire source is visible, and the radiance there is non-zero. Thus, the radiance function is discontinuous along AB. The points A and B are points of singularity in the radiance function.

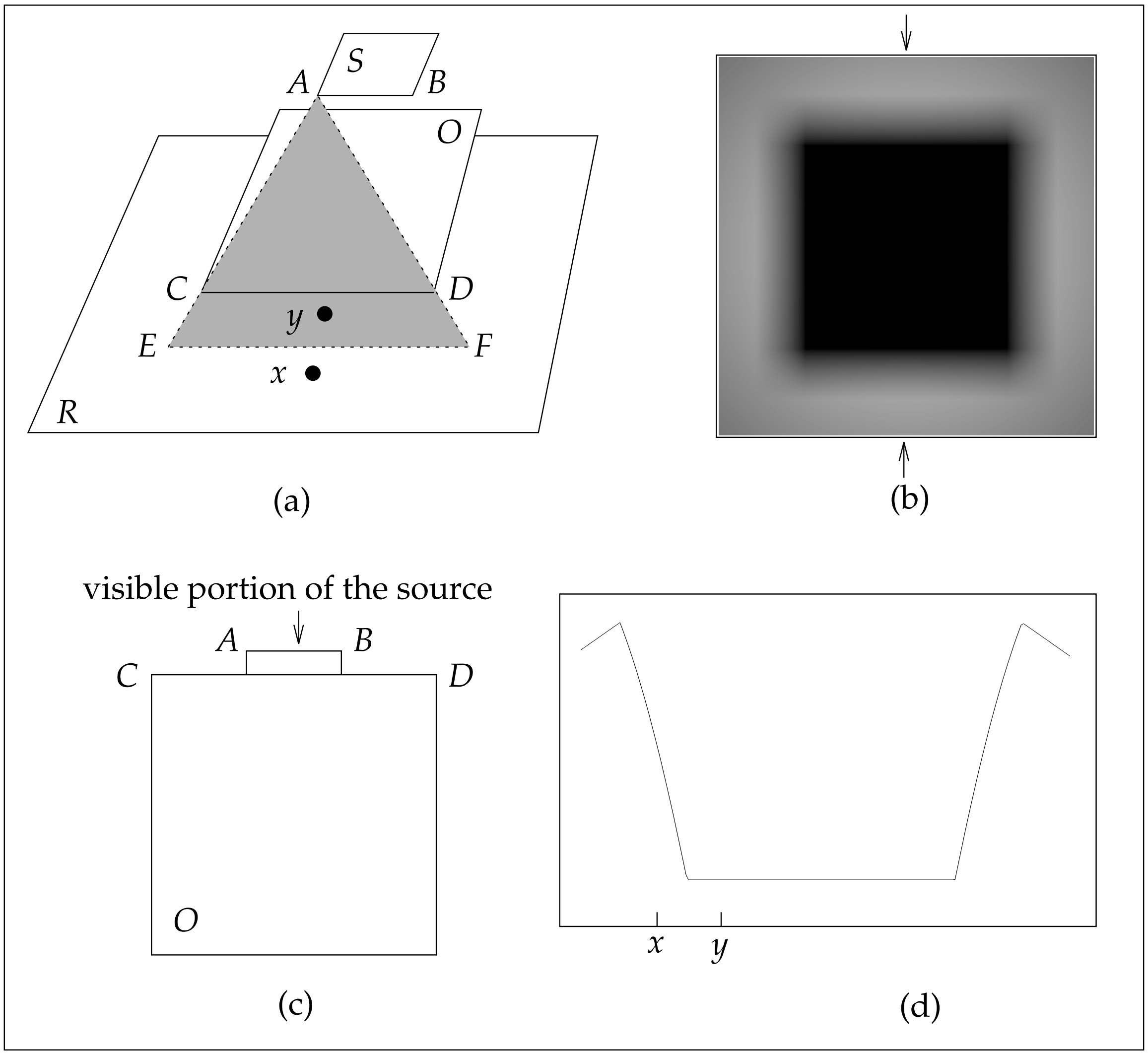


Figure 2.4: A D^1 discontinuity caused by edge-vertex (EV) events. (a) Edge AB of the light source S is coplanar to edge CD of the occluding polygon O. The critical surface (shaded area) defined by A and CD intersects the receiving plane along EF. (b) The radiance function over R. (c) The occluder O and the light source S as seen from x. (d) The radiance function along the line through x and y. From point y on R none of the source is visible, hence the radiance there is zero. As we move from y towards x, part of the source adjacent to AB becomes revealed. The visible area grows linearly in the displacement from EF towards x. Thus, along EF the radiance function has a D^1 discontinuity. In this example there are in fact two partially overlapping EV events, one involving vertex A and the other involving vertex B.

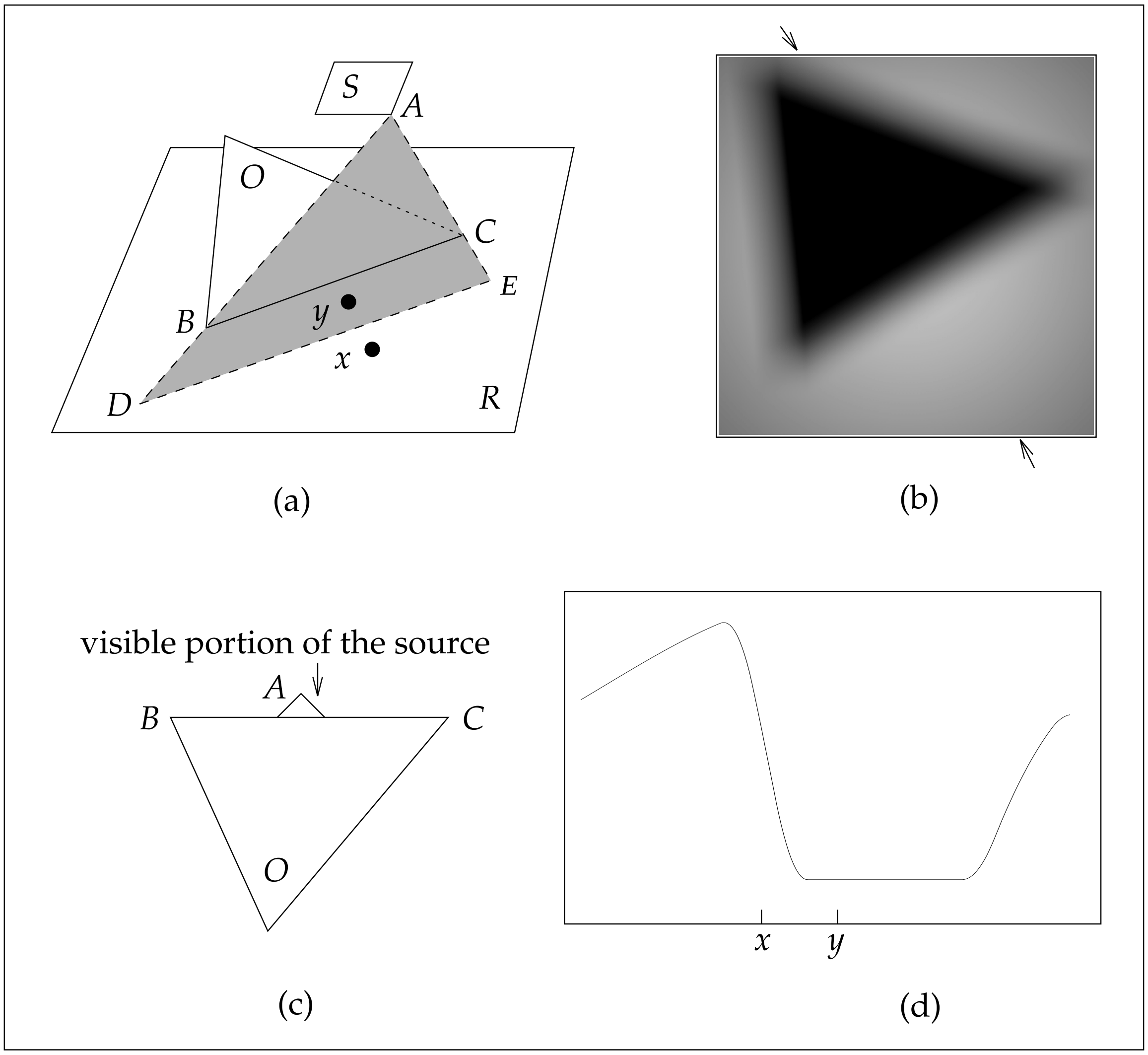


Figure 2.5: A D^2 discontinuity caused by an edge-vertex (EV) event. (a) The critical surface (shaded area) defined by vertex A and edge BC intersects the receiving plane along DE. (b) The radiance function over R. (c) The occluder O and the light source S as seen from x. (d) The radiance function along the line through x and y.

From point y on R none of the source is visible, hence the radiance there is zero. As we move from y towards x, part of the source adjacent to vertex A becomes revealed. The visible area grows quadratically in the displacement from DE towards x. Thus, along DE the radiance function has a D^2 discontinuity.

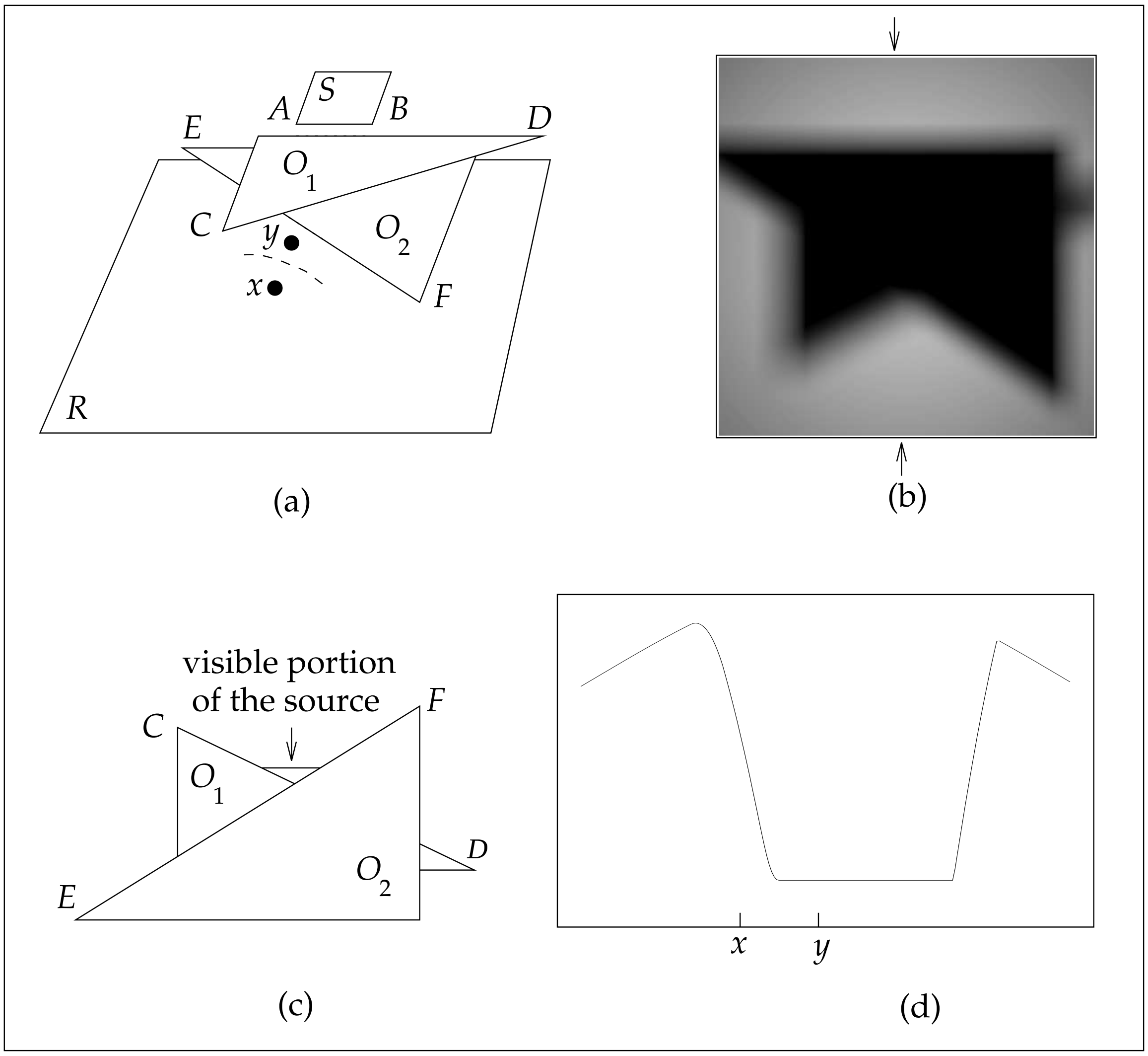


Figure 2.6: An edge-edge-edge (EEE) event. (a) The quadric critical surface defined by the three edges *AB*, *CD*, and *EF* intersects the receiver *R*, resulting in a conic critical curve (shown dashed). (b) The radiance function over *R*. (c) The occluders O_1 and O_2 and the light source *S* as seen from *x*. (d) The radiance function along the line through *x* and *y*.
As we move from *y* towards *x*, part of the source becomes revealed. A displacement from the critical curve towards *x* results in quadratic growth in the visible source area, hence the discontinuity along that curve is D^2. This example illustrates that when several occluding obstacles are involved, the boundaries between umbra and penumbra regions on a receiver may be curved.

edge-vertex (EV) edge-edge-edge (EEE) events [GIGU90].

EV events occur on a subset of the plane defined by an edge and a vertex fully or partially visible to each other. The set of points on that plane from which the vertex can be seen coinciding with the edge is called the *critical surface*. Visibility changes on a receiver occur along the curves of intersection between the receiving plane and the critical surfaces. These *critical curves* are line segments in the case of an EV event. They correspond to either D^1 or D^2 discontinuities, as demonstrated in Figures 2.4 and 2.5. D^1 discontinuities are perceived as Mach bands on the receiver. Strong D^2 discontinuities are noticeable as well.

EEE events occur on a subset of a ruled quadric surface defined by the family of lines that go through three skew edges, each visible to the others. The critical surface is the set of all the points from which the three participating edges can be seen intersecting at the same point. The critical curves are conics and generally correspond to D^2 discontinuities as demonstrated in Figure 2.6.

In the examples shown in the figures all the visual events involve either a vertex or an edge of the source. However, visual events can be defined by edges and vertices belonging to any object in the environment. Thus, in an environment with m edges, there can be $O(m^2)$ EV critical surfaces, and $O(m^3)$ EEE critical surfaces. These can give rise to $O(m^3)$ and $O(m^4)$ critical curves, respectively. In general, the event will cause a discontinuity in the radiance on the receiver if it is defined by edges and vertices located between the receiver and the source and if it is visible from both.

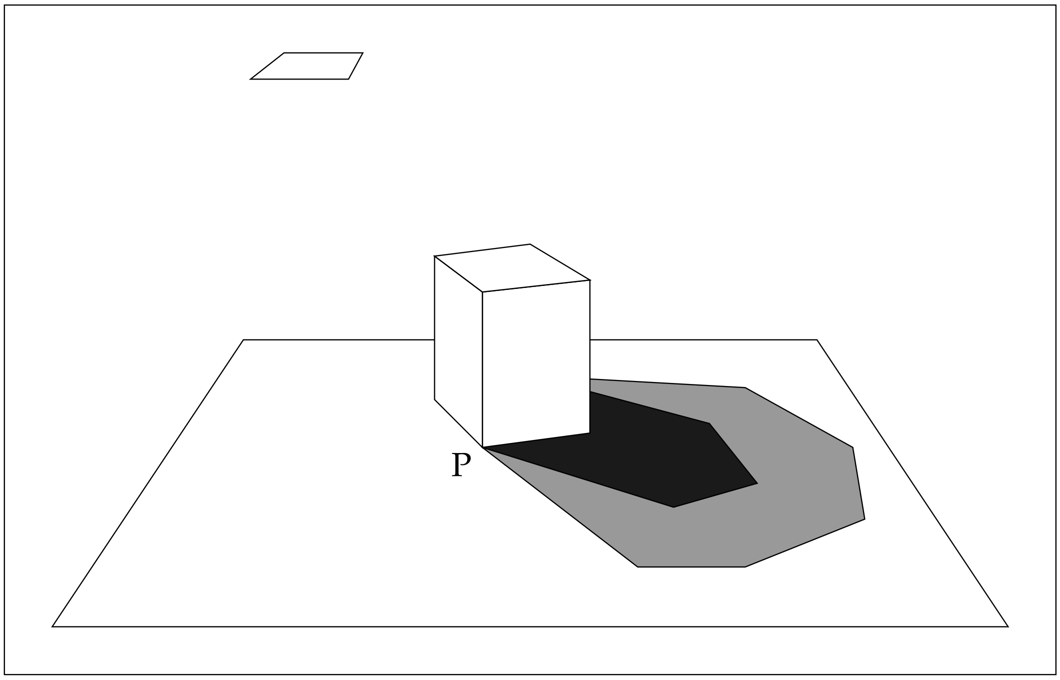


Figure 2.7: A singularity in the radiance function

2.4.3 Propagation of Discontinuities

The preceding discussion assumed that the radiance function on the source is smooth, as is usually the case for primary light sources. However, the radiance over secondary light sources can have discontinuities of various orders, as we have just seen. Heckbert [HECK91a] summarized the effect of these discontinuities as the "Discontinuity Propagation Law", which states that D^k discontinuities on the source can result in D^{k+1} and D^{k+2} discontinuities on the receivers. In general, discontinuities of higher orders are less noticeable than low order ones.

2.4.4 Singularities

Singularities in the radiance function may occur at the endpoints of D^0 discontinuity segments where the radiance gradient may become infinite.

As an example, consider the simple environment shown in Figure 2.7. In this environment a light source illuminates a cube resting on the floor. The umbra and the penumbra regions on the floor are shown shaded. The penumbra region provides a continuous transition between the unoccluded and the totally occluded (umbra) regions. As we move towards point P, at which a vertex of the cube touches the floor, the width of the penumbra gets smaller and the gradient of the radiance function becomes larger. In the limit, the transition from total visibility of the source to complete shadow becomes instantaneous, and the radiance gradient becomes infinite. Points A and B in Figure 2.3a are also examples of singularities.

Chapter 3

Previous Work

3.1 Early Radiosity Methods

Radiosity was introduced to computer graphics by Goral *et al.* ten years ago [GORA84]. Their method used the Galerkin formulation with constant elements, as described in Section 2.2. Form-factors were computed by transforming each of the two area integrals in Equation (2.7) into a contour integral using Stokes' theorem [SPAR63a,SPAR78]. These contour integrals were then evaluated using Gaussian quadrature. The linear system (2.8) was solved using Gaussian elimination with partial pivoting.

This first radiosity implementation was limited to quadrilaterals and did not account for occlusion. Each input surface was discretized using a uniform mesh. The number of elements in the mesh had to be specified to the program by the user. Obviously, this was a limited implementation, unable to handle general complex environments.

A more general and improved radiosity method was soon to follow: Cohen's hemi-cube algorithm [COHE85] enabled computation of form-factors in the presence of occlusion. Gaussian elimination was replaced by Gauss-Seidel iteration, utilizing the fact that the matrix in Equation (2.8) is strictly diagonally dominant. Cohen observed that the iteration typically converged very rapidly, finding a solution in a fraction of the time required for Gaussian elimination.

Subsequent research has extended radiosity's applicability from purely diffuse to partially specular environments [IMME86,WALL87,SILL89,RUSH90], to arbitrary bidirectional reflectance functions [SILL91]. Radiosity has also been extended to simulate participating media [NISH87,RUSH87]. Researchers have also investigated alternative solution methods [COHE88], and accurate algorithms for computing form-factors [BAUM89,WALL89]. A complete survey of the radiosity literature is beyond the scope of this thesis, and the reader is referred to the recent textbook by Cohen and Wallace [COHE93]. In the remainder of this chapter we shall survey only results that are directly relevant to our research.

3.2 Hierarchical Methods

At first glance, radiosity methods appear to inherently require at least $O(n^2)$ work, where n is the number of elements in the solution. This is so, because forming Equation (2.8) requires computing an interaction between every pair of elements. In this section we survey hierarchical radiosity algorithms that avoid computing all the possible interactions by grouping elements together.

3.2.1 Substructuring and Adaptive Subdivision

Cohen *et al.* [COHE86] developed the first hierarchical radiosity algorithm. This method uses a two-level hierarchy: input surfaces are subdivided coarsely into *patches* which are further subdivided into *elements*. The patches act as the sources and the elements as the receivers of illumination in the environment.

Let m be the number of patches and n the number of elements in the environment ($m \ll n$). The algorithm first computes the mn form-factors from all the elements to all the patches. By averaging the element form-factors of each patch the m^2 patch-to-patch form-factors are obtained. The $m \times m$ linear system (2.8) is then formed and solved using Gauss-Seidel iteration. The solution gives the patch radiosities. Element radiosities can now be computed by "gathering" light to each element from all the patches using the precomputed element-to-patch form-factors. Thus, a solution with n elements is computed using only mn form-factors, as opposed to the n^2 form-factors required by the previous methods.

An additional improvement introduced in this algorithm is *adaptive subdivision*. The solution is first computed with a relatively coarse element mesh. If the difference in radiosity between neighboring elements is higher than a user-specified threshold, the elements are further subdivided. The form-factors are then updated and the solution is recomputed. This step can be repeated as needed.

Note that even though adaptive subdivision can be triggered by shadow boundaries, the shape of the mesh depends on the original geometry of the surface (e.g., an axis aligned rectangle must have axis aligned rectangular elements). Thus, the shape of the elements is not affected by the geometry of the shadow boundary. As we shall see in Section 3.3 this may result in a variety of visual artifacts.

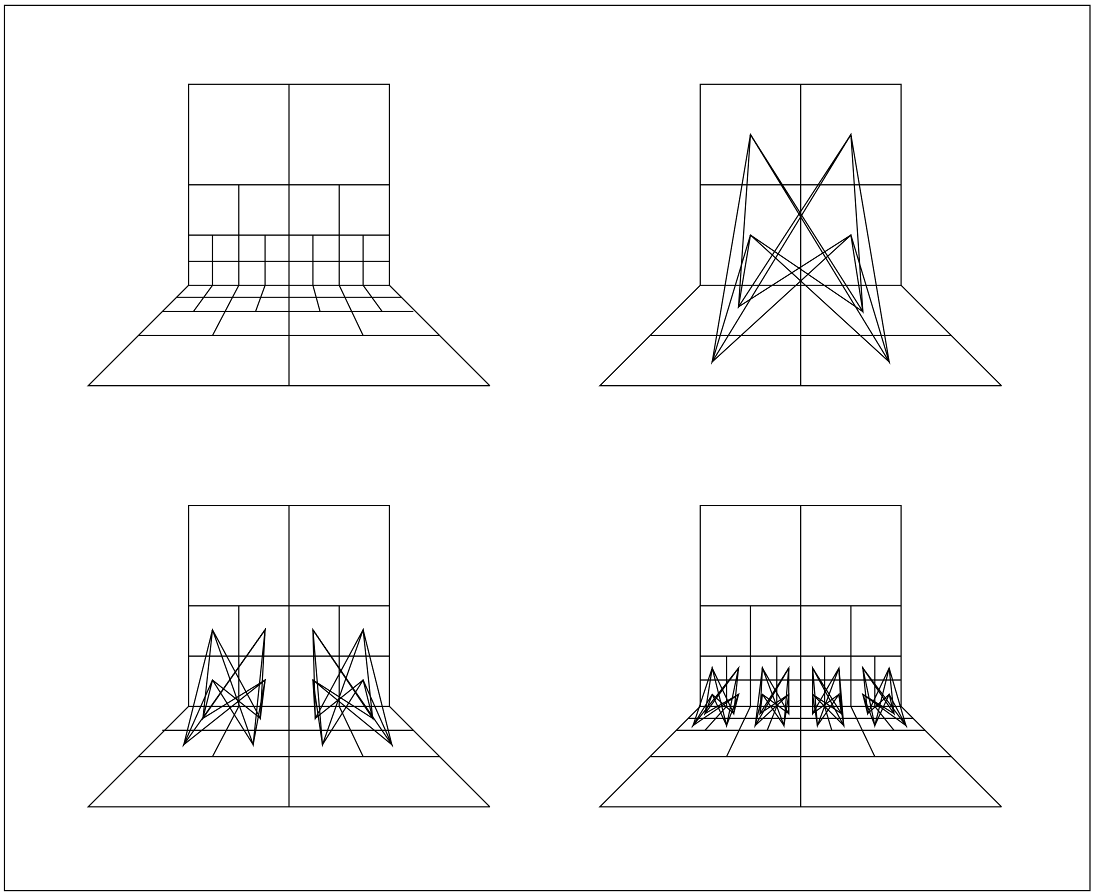


Figure 3.1: Hierarchical subdivision of a pair of perpendicular polygons. The line segments show the interactions between pairs of elements at different levels in the quadtree hierarchies (after Hanrahan [HANR91]).

3.2.2 Hierarchical Radiosity

Hanrahan *et al.* [HANR91] have generalized the substructuring idea to multiple levels. In their method, a patch at any level of the hierarchy can interact (i.e., exchange light energy) with any other patch at any other level (see Figure 3.1).

Given the accuracy at which a single interaction is to be represented, the $n \times n$ matrix of interactions is decomposed into $O(n)$ blocks. These blocks correspond to interactions of roughly equal magnitude, and the same computational effort is required for computing each block. This is achieved by constructing a hierar-

```
Void Refine(p, q, Feps)
Fpq ← FormFactorEstimate(p, q)
Fqp ← FormFactorEstimate(q, p)
if Fpq < Feps and Fqp < Feps
   Link(p, q)
else
   if Fpq ≥ Fqp and Subdivide(p)
      foreach r ∈ p.children do Refine(r, q, Feps)
   else if Subdivide(q)
      foreach r ∈ q.children do Refine(p, r, Feps)
   else
      Link(p, q)
   end if
end if
```

Figure 3.2: Pseudocode for the *Refine* routine

chical subdivision of each input surface. Each node in the hierarchy represents some area on the surface. Two nodes are linked together if the interaction between their corresponding areas can be adequately represented by a single form-factor; otherwise, the algorithm attempts to link their children with each other. Each link corresponds to a block in the interaction matrix. Pseudocode for the *Refine* routine that performs the linking is given in Figure 3.2.

Once all the links have been formed, the system is solved using Jacobi iteration. At each iteration, light is first gathered by each patch through all of its links; the radiosities of all the patches are then updated by pushing the gathered value down to the leaves of the hierarchies, and then pulling them up towards the roots. See pseudocode in Figure 3.3.

The number of links created by HR is $O(n+m^2)$ where n is the final number of

```
Void Gather(node)
node.tmp ← 0
foreach l ∈ node.links do
    node.tmp ← node.tmp + Contribution(l)
end for
if IsInterior(node) then
    foreach r ∈ node.children do Gather(r)
end if

Spectrum PushPull(node, rad)
node.tmp ← node.tmp + rad
if IsInterior(node) then
    node.rad ← 0
    foreach r ∈ node.children do
        w ← r.area/node.area
        node.rad ← node.rad + w * PushPull(r, node.tmp)
    end for
else
    node.rad ← rad
end if
return node.rad
```

Figure 3.3: Pseudocode for *Gather* and *PushPull* routines

nodes and m is the number of input surfaces. As the complexity of the environment increases, the m^2 term eventually becomes dominant, drastically reducing the efficiency of the algorithm. This problem can be solved by grouping the input surfaces into higher level clusters, as described by Smits *et al.* [SMIT94].

HR is the most promising method for us to build upon: it is the most efficient radiosity algorithm to date, the error in each interaction created by the algorithm is bounded, and it is easy to control: the user is able to trade-off speed for accuracy by specifying only two parameters (the error tolerance for each interaction and the minimum element area).

However, HR still suffers from visual artifacts such as shadow leaks and jagged shadow boundaries. This occurs because surfaces are subdivided regularly, not taking into account the geometry of the shadows, as explained in the next section. HR uses point sampling to classify the inter-visibility between two surfaces, so it is prone to missing small shadows altogether. Of course, as the user-specified error tolerance becomes smaller, the solution becomes more accurate, and the visual artifacts decrease. Nevertheless, images of high visual quality would require solutions of prohibitively high accuracy.

3.3 Discontinuity and Shadow Meshing

Most radiosity methods attempt to approximate the radiance function with constant elements and use linear interpolation to display the result. However, as was shown in Section 2.4, the actual radiance function is neither piecewise constant nor piecewise linear. It is usually smooth, except along certain curves across which discontinuities in value or in derivatives may occur.

These discontinuities are very important both numerically and visually: all the boundaries separating unoccluded, penumbra, and umbra regions correspond to various discontinuities. When a discontinuity curve crosses a mesh element, the approximation to the radiance function over that element becomes less accurate. The resulting errors usually correspond to the most visually distracting artifacts in radiosity images, such as missing shadows, jagged or otherwise distorted shadow boundaries, shadow and light “leaks”, and “floating” objects. These problems are surveyed by Baum *et al.* [BAUM91] and by Haines [HAIN91a].

Radiosity algorithms typically use adaptive subdivision [COHE86] to reduce these errors; however, there are several problems with this approach. First, the user must specify an initial mesh that is sufficiently dense, or features will be lost. Second, the shape of the mesh is determined by the geometry of the surface being meshed, and the discontinuities are not resolved exactly. As a result, many small elements are created as the method attempts to converge to shadow boundaries.

In order to circumvent these problems and produce high-quality images, radiosity systems frequently require intensive user intervention. The density of the mesh elements is often specified on a surface-by-surface basis and repeatedly adjusted through a tedious and time consuming trial-and-error process. Despite these manual adjustments, the resulting radiosity solutions are generally accurate for only a small range of views and thus, are not truly view-independent. For example, zooming in on a shadow is likely to reveal a jagged boundary.

Discontinuity meshing (DM) algorithms compute the location of certain discontinuities and represent them explicitly, as boundaries, in the mesh. This leads to solutions which are both numerically and visually more accurate. This idea has been used to various extents in several different recent algorithms that we now survey.

3.3.1 D^0 Preprocessing

Baum *et al.* [BAUM91] describe a series of clever techniques designed to preprocess an input model into a mesh that meets certain geometrical and topological requirements needed for computing accurate solutions. In particular, they have realized that the D^0 discontinuities at surface intersections are responsible for severe errors in radiosity solutions, which manifest themselves visually as shadow

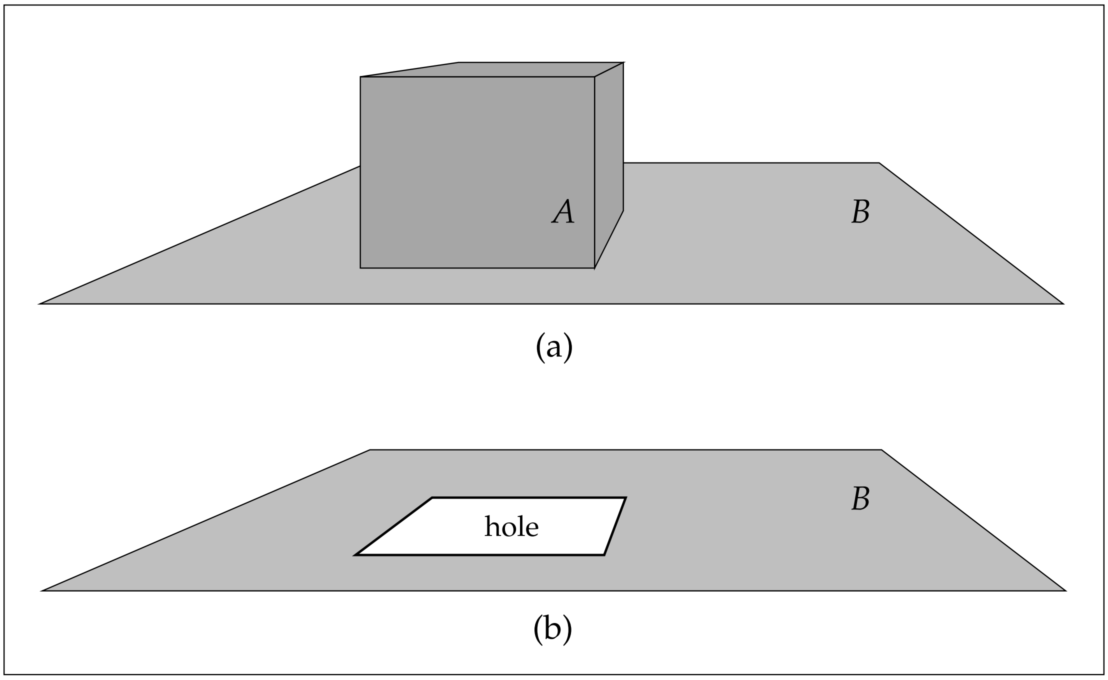


Figure 3.4: Computing D^0 discontinuities prior to meshing. (a) The initial environment has object A resting on surface B. (b) The area covered by object A is removed from surface B; the resulting D^0 edges are shown as thicker lines (object A is not shown).

and light "leaks". To solve this problem, all pairs of intersecting polygons in the model are identified, and cut along their lines of intersection. Thus, no elements in the mesh are allowed to be crossed by a D^0 discontinuity. See Figure 3.4.

However, errors and visual artifacts due to higher order discontinuities are not eliminated. Mesh boundaries are not aligned with shadow boundaries, so shadows may still appear jagged or distorted. The method uses numerical algorithms for determining visibility, so small shadows or light spots may be missed entirely. Also, the meshing is still not completely automatic: it is up to the user to specify an adequate initial patch density.

3.3.2 Shadow Meshing

Shadow meshing for radiosity originates from earlier object-precision shadow algorithms, such as the algorithm by Atherton *et al.* [ATHE78], and by Chin and Feiner [CHIN89]. These algorithms split the input polygons along the D^0 discontinuities corresponding to shadow boundaries from point light sources. In order to make these algorithms applicable to radiosity they must be extended to handle area light sources.

Campbell and Fussell [CAMP90] approximate each area light source by a set of point light sources. Polygons in the environment are meshed along their intersections with the shadow volumes from each of the points sources, as illustrated in Figure 3.5(a,c). This is equivalent to computing discontinuities with respect to the approximating point light sources, rather than the actual area sources. This method works well for small light sources, which can be adequately represented by a small number of points. However, as the number of point sources needed for an adequate approximation grows, subdivision within regions of penumbra becomes too fine.

Campbell's approach was later improved by treating area light sources in a more analytical fashion [CAMP91]. Umbra and penumbra volumes are computed for each area light source. Using these volumes, the environment is classified into unoccluded, umbra, and penumbra regions, as shown in Figure 3.5(b,d). Each region is further split into elements, and the illumination at the vertices of each element is computed analytically. To our knowledge, this is the first object-space algorithm to accurately compute shadows cast by area light sources. A weakness of this method seems to be its reliance on numerical optimization for determining the correct element density inside penumbra regions. We suspect

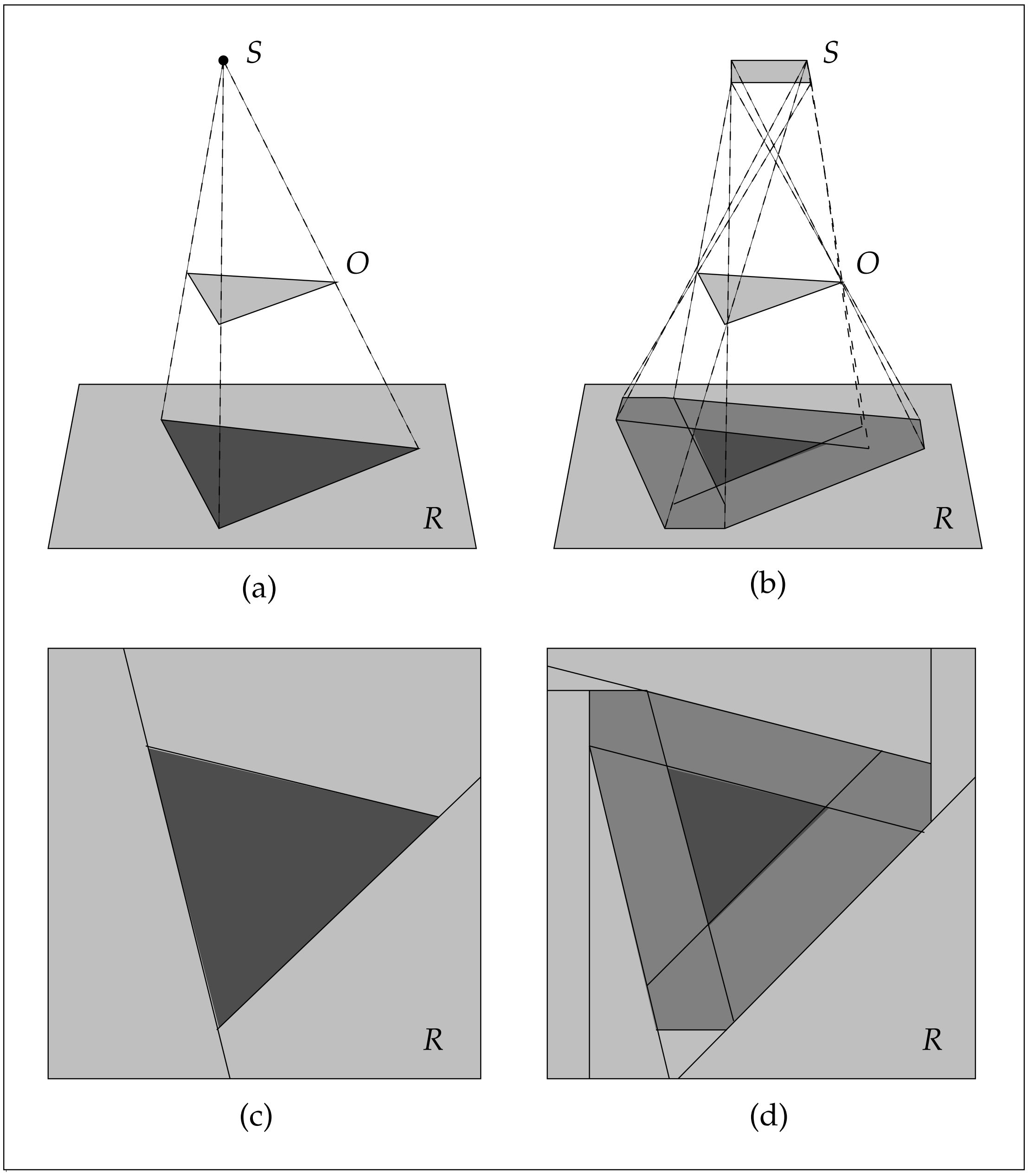


Figure 3.5: Shadow volumes (a) and penumbra volumes (b) constructed from source S and object O are used to classify surface R into regions of umbra, penumbra, and total visibility. The resulting subdivisions (c-d) are represented as two-dimensional BSP trees.

that this might become expensive and/or inaccurate for large penumbrae. This algorithm is also liable to produce poorly shaped elements.

Chin and Feiner [CHIN92] describe a very similar algorithm, but they provide no automatic means for determining the element density inside penumbra regions. Instead, a regular grid is superimposed over each (non-umbra) region.The density of the grid is user-specified and can be different for unoccluded and penumbra regions.

3.3.3 Discontinuity Meshing

The idea of explicitly representing discontinuities in radiance functions and their derivatives as boundaries between mesh elements was first developed by Heckbert [HECK91a]. Similar ideas were pursued independently by Lischinski *et al.* [LISC91]. In both cases, the ideas were implemented and tested in 2D, showing promising results. In particular, Heckbert has shown that the potential of higher order elements for radiosity could fully be realized only in conjunction with discontinuity meshing.

Both Heckbert and Lischinski *et al.* have independently extended their ideas to 3D environments. Heckbert [HECK92] constructs a discontinuity mesh that contains all the discontinuity lines arising from EV events in which primary light sources directly participate. First, the critical lines are computed on all the surfaces in the environment, and then Delaunay triangulation and mesh relaxation are used to produce the final discontinuity mesh on each surface. This implementation forms and solves the full $n \times n$ radiosity matrix; thus, it is impractical for all but the simplest environments.

Lischinski *et al.* [LISC92] describe a progressive DM radiosity algorithm for

3D polyhedral environments. At each iteration of the algorithm one input polygon is chosen as the current light source S, and a discontinuity mesh is computed with respect to S. This mesh is used to compute and represent the contribution of S to the other surfaces in the environment. Adaptive subdivision is used to increase the mesh density automatically where necessary. The contributions due to different light sources are combined into a cumulative radiance function on each surface, also represented as a discontinuity mesh.

This algorithm has been able to produce radiosity solutions of impressive visual accuracy. Experimental evidence indicates that this algorithm is also very accurate numerically [TAMP93]. The meshing in this algorithm is completely automatic.

However, this method is too expensive for computing converged solutions of complex environments. The main reason for this is that all energy transfers are computed very accurately, regardless of their magnitude. This may be appropriate for capturing the direct component of the illumination (due to the primary light sources), but quickly becomes an overkill as the amount of light energy shot from a source becomes smaller. Also, this algorithm only offers limited user control in trading off speed for accuracy.

3.4 Solution Post-Processing

The simplest form of solution post-processing was described by Goral *et al.* [GORA84]. Observing that smoothing the piecewise-constant approximation enhances the quality of the image, they perform linear interpolation of the element radiosities in image space.

Cohen *et al.* [COHE85] suggested performing the smoothing in object space:

they obtain radiosity values for element corners by interpolating and extrapolating from element centers; then, the radiosity at each point is bilinearly interpolated from the corners of the containing element.

Several researchers have realized that one can compute a reasonable approximation to the indirect component of global illumination with a relatively coarse regular mesh. This is so, because in typical environments the indirect illumination varies rather slowly across the surfaces. Most of the high frequency illumination features that we see come from direct illumination (or from non-diffuse effects such as caustics). This observation has motivated several two-pass approaches by Nishita and Nakamae [NISH85], Shirley [SHIR90b], Chen *et al.* [CHEN91], Kok and Jensen [KOK91], and Reichert [REIC92]. The first pass typically solves for the indirect component of the illumination using a coarse mesh. The second pass computes an accurate value for each pixel in the image by computing the direct illumination there and combining it with the indirect component.

Note that the second pass in the above approaches operates in image space, after the view and the resolution have been specified. While this may be appropriate when the objective is to obtain a single image or a small number of images, this form of solution post-processing is impractical for applications such as architectural walkthroughs.

3.5 Error Estimation

Cohen's adaptive subdivision algorithm [COHE86] can be viewed as the first attempt at *a posteriori* error estimation. In this algorithm, elements are subdivided if their radiosity differs from that of their neighbors by more than a certain threshold. Thus, the difference between radiosities of adjacent elements is treated as an indicator of the local error. Note that this heuristic may fail to identify elements that should be subdivided.

A more conservative approach is described by Campbell [CAMP91], who uses numerical optimization to find the minimum and the maximum radiosity over each element. The difference between these extrema is used as a criterion for adaptive subdivision.

Among the various radiosity methods, hierarchical radiosity comes closest to bounding the errors in the solution: a bound is computed on the error in each interaction (link) between patches. Hanrahan *et al.* [HANR91] compute an approximate upper bound on the form-factor corresponding to the link, and use it to obtain an approximate upper bound on the transferred energy.

Smits *et al.* [SMIT92] estimate the error in each interaction by point sampling the kernel function over the areas of the two interacting patches. The difference between the maximum and the minimum values thus obtained is taken to be the error in the form-factor. This error estimate is less conservative than Hanrahan's, but it is generally more accurate.

Given error estimates for all the links in an environment, one can approximate the total error over each patch by summing up the errors associated with all its links. However, this does not take into account propagation of errors, which parallels the propagation of light.

Several other error estimation techniques are surveyed by Cohen and Wallace [COHE93]. These techniques can be classified into two groups: (i) comparing the approximation to a higher order approximation, and (ii) computing the residual given by Equation (2.4). Both of the above are computationally expensive and do not provide foolproof subdivision criteria.

To summarize, radiosity methods to date can provide the user with neither guaranteed bounds on the total error, nor reliable error estimates. Some heuristics that are used to guide adaptive refinement are not always reliable and may miss some elements with large errors; other heuristics may cause unnecessary refinement.

Chapter 4

Combining Hierarchical Radiosity and Discontinuity Meshing

4.1 Overview

Hierarchical radiosity (HR) and discontinuity meshing (DM) seem to complement each other in their strengths and weaknesses: HR is fast, but the visual appearance of the results can be disappointing; DM, on the other hand, has produced visually accurate results, but so far it has been too expensive for simulation of complex environments. This observation motivated us to look for ways of merging the two methods. Our investigation resulted in the following two-pass approach:

The global pass uses a modified HR algorithm to compute a radiosity solution within a prespecified tolerance. Instead of regular quadtree subdivision, the modified algorithm subdivides surfaces along discontinuity segments. This improves the numerical accuracy and results in faster convergence.

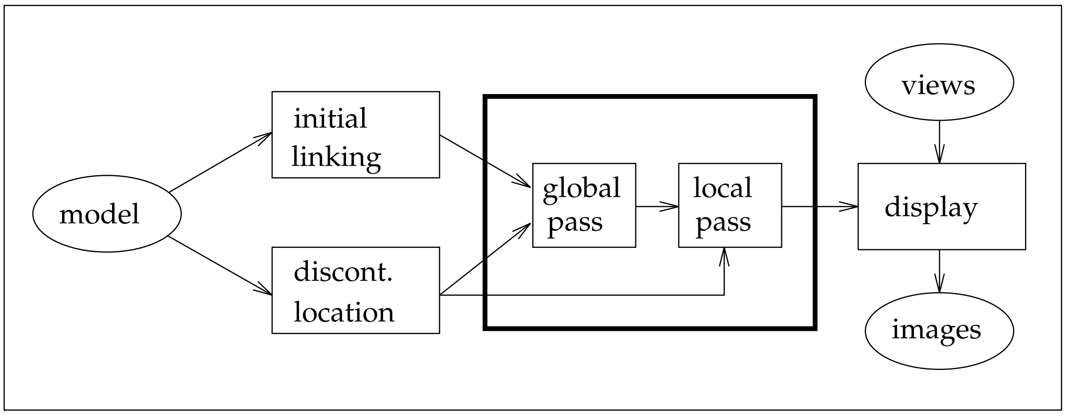


Figure 4.1: The structure of the new radiosity system

The local pass uses DM and quadratic interpolation to refine the approximation to the radiance function locally on each surface in the environment. Thus, the piecewise-constant approximation computed by the global pass is transformed into a more visually accurate form.

When the computation is arranged in this way the simulation becomes more efficient. The global pass need not be concerned with visual accuracy. This eliminates the need to maintain a topologically connected mesh, to prevent T-vertices, or to use extremely fine subdivision around shadow boundaries, since this has little effect on the global distribution of light in the environment. The local pass, on the other hand, can create as many elements as necessary for a high quality reconstruction of the radiance function, without overburdening the global illumination simulation. As a result, it is possible to produce images of high visual accuracy even from quick simulations.

To test our approach we have implemented a new radiosity system whose overall structure is shown in Figure 4.1. The global and the local passes are discussed in detail in the next two sections. In the rest of this section we briefly describe the preprocessing stages of initial linking and discontinuity location.

```
Void ILink(p, q)
if CanCullLink(p, q)
    return
case ClassifyVisibility(p, q, blockers) in
    VISIBLE:    vis ← 1
    PARTIAL:    vis ← EstimateVisibility(p, q, blockers)
    OCCLUDED:   return
end case
Link(p, q, vis), Link(q, p, vis)
```

Figure 4.2: Pseudocode for the *ILink* routine

4.1.1 Initial Linking

The initial linking stage creates for each input polygon a list of links to all the other polygons that are visible from it. For each link it is determined whether the two polygons are completely or partially visible to each other. This creates a starting point for the global pass, which proceeds to refine these links as needed. Pseudocode for initial linking is given in Figure 4.2.

We test visibility between two polygons p and q using a combination of shaft-culling [HAIN91b] and the visibility algorithms described by Teller and Hanrahan [TELL93]. The routine *ClassifyVisibility* first constructs the *shaft* defined by p and q, which is the convex hull of the two polygons' vertices. Then, we compute a list of *blockers*: all the polygons in the environment that intersect the shaft. This is done efficiently by using a hierarchy of bounding boxes [GOLD87]. If the blocker list is empty, p and q are entirely visible to each other and *ClassifyVisibility* returns VISIBLE. Otherwise, p and q are at least partially occluded. In this case, rays between all the vertices of p and q are tested against the blockers in the list. If a single blocker is found that intersects all the rays, there is complete occlusion and no links need be created. Otherwise, *ClassifyVisibility* returns PARTIAL.

For partially visible links *EstimateVisibility* computes an approximate intervisibility factor by shooting a constant number (typically sixteen) rays, which randomly sample the areas of p and q, as described by Hanrahan *et al.* [HANR91].

Unfortunately, the initial linking stage can produce $O(n^2)$ links, where n is the number of input polygons. Thus, in complex environments there can be a very large number of initial links, most of which represent interactions of very small magnitude. A rigorous solution to this problem was recently proposed by Smits *et al.* [SMIT94]. In our current implementation we use *link culling*, a simple heuristic to reduce the number of initial links. Before classification of visibility between two polygons p and q takes place, the predicate *CanCullLink* computes the unoccluded form factor between the two. If neither p nor q is a primary source, and the form factor between them is less than a user-specified threshold, *CanCullLink* returns TRUE, and no links are created. This *ad hoc* solution was found to work well in practice: the number of initial links can be significantly reduced without severe effects on the resulting images.

4.1.2 Discontinuity Location

The discontinuity location stage first computes the location of all the D^0 discontinuities. These discontinuities are responsible for the most severe errors in global illumination simulations, both numerically and visually. To locate the D^0 discontinuities, which arise due to contact between surfaces, we use the algorithm described by Tampieri [TAMP93]. This algorithm utilizes the same hierarchy of bounding boxes that is used for visibility computations. Given an edge of an input polygon, the algorithm traverses the hierarchy visiting only the nodes whose bounding boxes contain at least part of the edge. If a leaf node is reached, the contained polygons are tested for intersection with the edge, yielding D^0 discontinuity segments or points of singularity.

In order to produce accurate solutions we need to take into account not only D^0 discontinuities, but also strong higher order (D^1 and D^2) discontinuities. In general, it is impossible to establish *a priori* which discontinuities are going to be the strongest. In our current implementation we assume that these discontinuities are all caused by the primary light sources (emitters). Therefore, we compute all of the D^1 and D^2 discontinuities caused by EV events involving the primary light sources. The computed discontinuities are henceforth collectively referred to as *primary* discontinuities.

As explained in Chapter 2, D^1 and D^2 discontinuities are also caused by EEE events. EEE events are more difficult to handle because their corresponding critical surfaces are curved, rather than planar. However, the resulting discontinuities always lie within penumbra regions, and never define the outer boundaries of a shadow. Thus, it is unclear how important it is to take them into account. For these reasons, we have excluded EEE events from our current implementation.

The discontinuity location algorithm for EV events is described by Lischinski *et al.* [LISC92] and in more detail by Tampieri [TAMP93]. This algorithm uses a binary space partitioning (BSP) tree [FUCH80] to trace the wedge-shaped visual events through the environment. Heckbert [HECK92] and Teller [TELL92] describe alternative algorithms for locating these discontinuities. It should be noted that Teller's algorithm is also capable of handling EEE events.

4.2 The Global Pass

In this section we describe a modified HR algorithm that computes the global distribution of light energy in an environment using a discontinuity-driven strategy, capable of achieving better accuracy and faster convergence.

In order to understand how the accuracy of HR can be improved, we must examine the sources of error in this algorithm. Consider two nodes s and r linked together directly by a single link. Let $B_{rs}(x)$ denote the actual radiosity function due to node s at point x on node r. The algorithm approximates this function by a constant

$$B_{rs}(x) \approx \widehat{B}_{rs} = \rho_r B_s F_{rs} V_{rs},$$

where ρ_r is the reflectivity of node r; B_s is the average radiosity of node s; F_{rs} is the unoccluded form factor from r to s; and V_{rs} is the inter-visibility factor between r and s (the visible fraction of the area of s, averaged over r).

We are interested in bounding the error between the computed and the actual radiosities

$$E_{rs} = \sup_{x \in r} \left| B_{rs}(x) - \widehat{B}_{rs} \right|. \tag{4.1}$$

To that end, we define the following upper and lower bounds:

$$\begin{array}{ll}
\underline{B}_s = \inf_{x \in s} B_s(x) & \overline{B}_s = \sup_{x \in s} B_s(x) \\
\underline{\rho}_r = \inf_{x \in r} \rho_r(x) & \overline{\rho}_r = \sup_{x \in r} \rho_r(x) \\
\underline{F}_{rs} = \inf_{x \in r} F_{xs} & \overline{F}_{rs} = \sup_{x \in r} F_{xs} \\
\underline{V}_{rs} = \inf_{x \in r} V_{xs} & \overline{V}_{rs} = \sup_{x \in r} V_{xs}
\end{array}$$

where $B_s(x)$ is the radiosity at point x on s; $\rho_r(x)$ is the reflectivity at point x on r; F_{xs} is the unoccluded form factor from point x to s; and V_{xs} is the fraction of the area of s visible from x. Clearly, both $B_{rs}(x)$ and $\widehat{B}_{rs}$ lie in the interval

$$\left[\underline{\rho}_r \underline{B}_s \underline{F}_{rs} \underline{V}_{rs}, \overline{\rho}_r \overline{B}_s \overline{F}_{rs} \overline{V}_{rs}\right].$$

Therefore, the error E_{rs} is bounded by the width of the above interval

$$E_{rs} \leq \left(\overline{\rho}_r \overline{B}_s \overline{F}_{rs} \overline{V}_{rs} - \underline{\rho}_r \underline{B}_s \underline{F}_{rs} \underline{V}_{rs}\right). \tag{4.2}$$

Assuming that the reflectivity across r is constant, i.e., $\underline{\rho}_r = \overline{\rho}_r = \rho_r$, three main factors remain that affect the magnitude of the error:

1. the variation in the actual radiosity B_s across the source
2. the variation in the point-to-area form factor F_{xs} across the receiver
3. the variation in the point-to-source visibility V_{xs} across the receiver

Therefore, if we find the potential error in the transfer of light energy from s to r too large, we can try to reduce the error by reducing any of these factors. For instance, subdividing the receiving node will reduce the variation in the form factor. Subdividing the source will reduce the variation of the radiosity on the source. Subdividing either of the two may reduce the variation in the visibility.

Unfortunately, errors due to visibility are more difficult to handle than errors of the other two types. If the two nodes are completely visible to each other, the error usually decreases rapidly as the nodes are subdivided. When the two nodes are completely occluded from each other no light energy transfer occurs, and the error is zero. Partial visibility, on the other hand, often results in very fine subdivisions, primarily because of loose bounds on the variation in visibility between two finite areas. In HR, visibility is estimated by casting a number of rays between the two nodes. Thus, if partial visibility is detected, all we know is that the actual visibility is in the interval $[0, 1]$.

Clearly, it would be to our advantage to use a subdivision strategy that would result in as many totally visible or totally occluded pairs, as quickly as possible. Since discontinuity lines on the receiver correspond to abrupt changes in the visibility of the sources, subdividing the receiver along these lines should quickly resolve partial occlusion conditions.

We have modified the HR algorithm to perform discontinuity-driven subdivision instead of regular subdivision. There are two main changes in the data structures used by the new algorithm: first, we store with each node a list of all the discontinuity segments on the corresponding polygon; second, we use a 2D binary space partitioning (BSP) tree, instead of a quadtree to represent the hierarchical subdivision of each initial polygon, since BSP-trees allow for subdivision of polygons along arbitrarily oriented lines. Two-dimensional BSP-trees were also used by Campbell [CAMP91] in his shadow-meshing algorithm.

Pseudocode for subdividing a node is given in Figure 4.3. When a node is subdivided we choose one of its discontinuity segments and split the node using the corresponding line equation. *ChooseBestSegment* chooses a segment such that the split is as balanced as possible. Priority is given to D^0 segments over higher order ones, since the former typically bound areas totally occluded from the rest of the environment. The subdivision is completed by splitting the list of discontinuity segments into two new lists, one for each child. If no segments are stored with the node, *SplitEqual* splits the node by connecting the midpoint of the longest edge to a vertex or another midpoint chosen so that the resulting children have roughly equal areas.

```
Boolean Subdivide(node)
if not IsLeaf(node) then
    return TRUE
end if
if node.area < minNodeArea then
    return FALSE
end if
if node.DSegments ≠ NIL then
    DSegment s ← ChooseBestSegment(node)
    (left,right) ← SplitNode(node,s)
    (leftList,rightList) ← SplitSegmentList(node,s)
else
    (left,right) ← SplitEqual(node)
    (leftList,rightList) ← (NIL, NIL)
end if
node.left ← CreateNode(left,leftList)
node.right ← CreateNode(right,rightList)
return TRUE
```

Figure 4.3: Pseudocode for the *Subdivide* routine

4.2.1 Results

Figure 4.4 demonstrates the improved hierarchical algorithm using a simple environment illuminated by two small triangular light sources. A 3D view of the environment is shown in image a1. All the other images show a top view of the floor. The radiance function on the floor is shown in image a2. Image a3 shows the discontinuity segments on the floor. D^0 discontinuities are drawn in red; D^1 and D^2 discontinuities in yellow. In rows b and c, we compare the subdivision produced by the discontinuity-driven algorithm to the one produced by regular subdivision. The level of subdivision shown increases from left to right: the

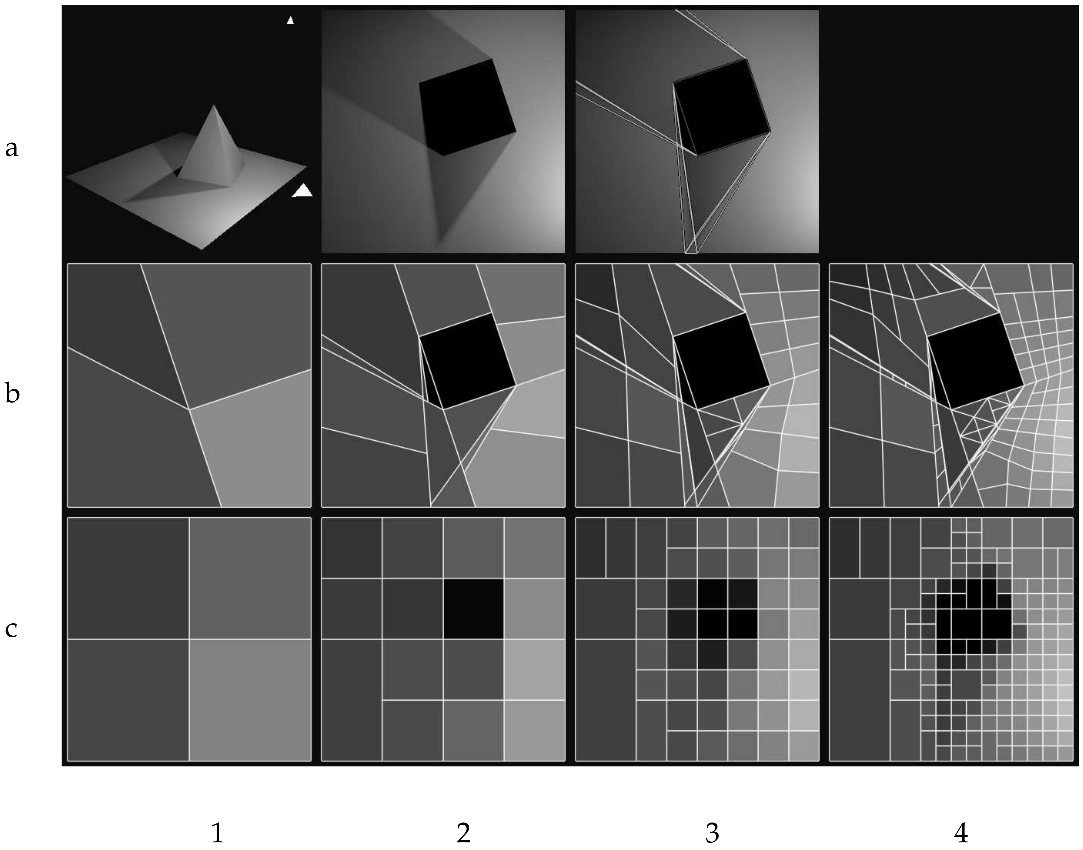


Figure 4.4: Discontinuity-driven vs. regular subdivision

leftmost pair shows the subdivision at level 2, then level 4, 6, and 8.

The new algorithm is much quicker to correctly separate regions corresponding to complete occlusion, partial visibility, and complete visibility. Already at subdivision level 4 (image b2), most of the nodes can be classified as either totally visible or totally occluded with respect to each of the two light sources. For these areas there are no more visibility errors. At subdivision level 6 (image b3) all of the discontinuities have been used, and the partially visible nodes are now confined exactly to the areas of penumbra.

In order to compare the rates of convergence of the two strategies we computed a set of approximations to the direct illumination on the floor using a suc-

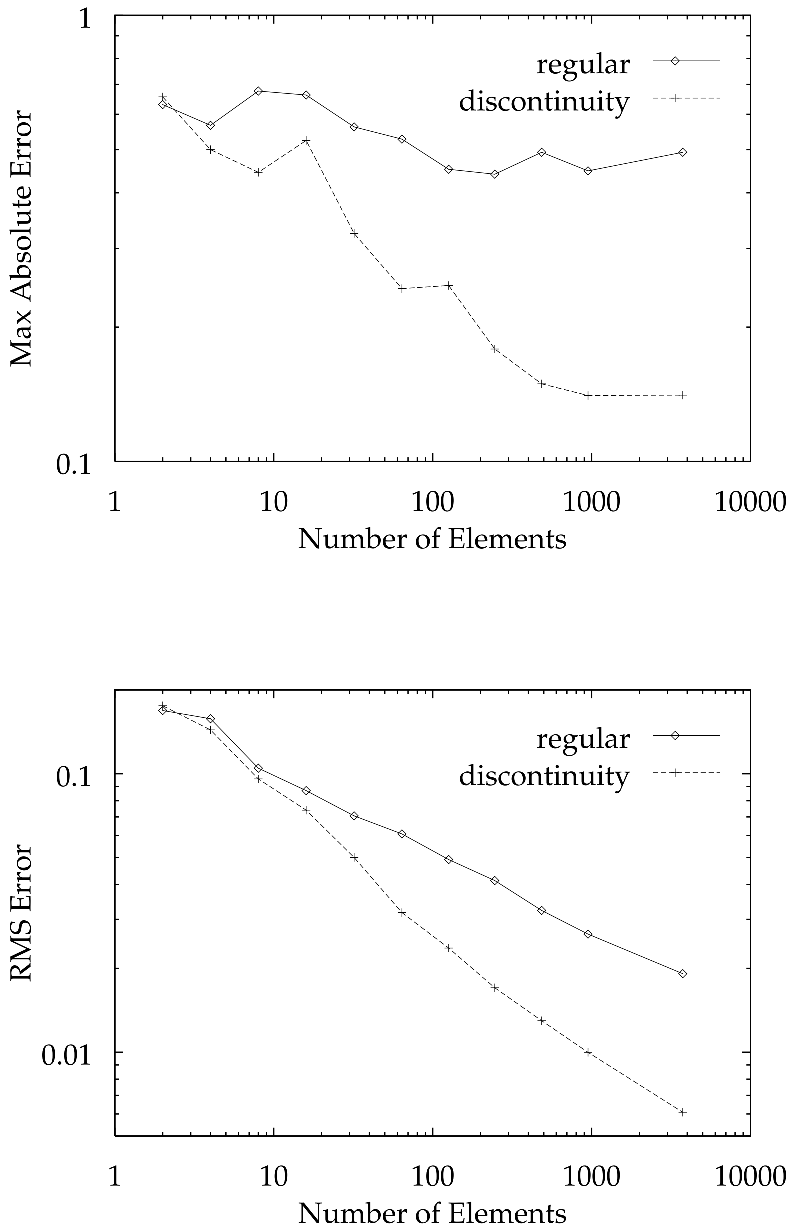


Figure 4.5: A comparison of errors between the two subdivision strategies

cessively larger number of elements. Figure 4.5 shows the RMS and the maximum absolute (L_∞) errors as a function of the number of elements for the two strategies. These errors were computed with respect to an analytical solution at the vertices of a 400 by 400 grid on the floor. All the values were scaled to set the maximum brightness on the floor to 1.

The regular subdivision strategy fails to converge in the L_∞ error metric. The reason is that there are D^0 segments on the floor that are not aligned with the subdivision axes. Thus, there are always elements that are partially covered by the pyramid while the remaining part is brightly illuminated by the light sources. The algorithm assigns a single constant value to each such element, and this results in a large error there. This error cannot be diminished by making the floor elements smaller. Our algorithm, on the other hand, resolves D^0 discontinuities and therefore does not suffer from this problem.

In the RMS error metric regular subdivision does converge, because the elements that contain the errors become progressively smaller, and this is accounted for by this area-weighted metric; however, the convergence is substantially slower than that of the discontinuity-driven method.

4.3 The Local Pass

The global pass results in a hierarchical solution that is essentially a piecewise-constant approximation to the radiance function on each polygon in the environment. Often, this approximation is quite coarse. Now our goal is to convert this solution into a form more suited for producing visually accurate images. To that end, we need to locally refine the radiance approximation on each polygon.

Previous DM algorithms [LISC92,TAMP93] have demonstrated that reproducing the discontinuities in the radiance function, while maintaining a smooth approximation elsewhere is key to achieving visual accuracy, especially when multiple views of the same solution are to be rendered. Therefore, in this pass we construct a discontinuity mesh containing all the precomputed primary discontinuities for each polygon. The element nodes in this mesh are assigned radiance values by gathering light through the links computed during the previous pass. Thus, the local pass essentially performs an additional gathering iteration over the environment. However, instead of gathering to the nodes in the hierarchy, we gather to the discontinuity mesh element nodes. The shaded mesh is then used for the display of the environment from any given viewpoint.

4.3.1 Constrained Delaunay Triangulation

The discontinuity mesh is constructed using constrained Delaunay triangulation (CDT) [CHEW89]. The Delaunay triangulation (DT) of a point set maximizes the minimum angle over all possible triangulations of that set and has a number of other desirable properties [BERN92]. These properties are important because they result in well-shaped elements that yield more accurate approximations and reduce visual artifacts during display [BAUM91]. CDT takes as input a point set and a set of edges connecting some of the points, and creates a triangulation of the points that is constrained to include all the input edges. CDT preserves the properties of DT over all the constrained triangulations. We have implemented an incremental CDT algorithm that is a simple extension of the incremental DT algorithm described by Guibas and Stolfi [GUIB85]. An alternative easy-to-implement algorithm is described in the excellent survey by Bern and Eppstein [BERN92].

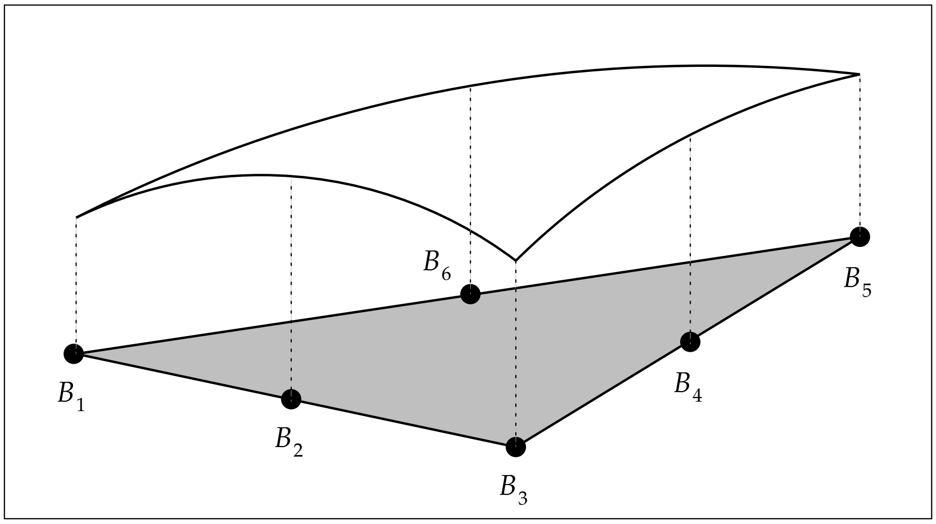


Figure 4.6: A quadratic triangular element

For each input polygon we provide the CDT routine with all of its boundary edges and discontinuity segments. The corners of all the leaf nodes in the corresponding hierarchy are given as well. Thus, the resulting mesh is dense enough to adequately sample the solution computed by the global pass. As a result of the properties of the CDT, most of the triangles are well shaped unless the hierarchy is very coarse.

4.3.2 Quadratic Elements

The radiance function across each triangle is approximated using a standard quadratic element commonly used in finite element methods [ZIEN89]. Six radiance values $B_1, \ldots, B_6$ are computed at the vertices and the edge midpoints of each element (see Figure 4.6). Except for D^0 edges, these values are shared between adjacent faces (the quad-edge data structure [GUIB85] used by our CDT algorithm is suitable such information sharing). The six values are then interpolated by a quadratic bivariate polynomial

$$B(x) = \sum_{i=1}^{6} B_i N_i(u,v,w)$$

where (u,v,w) are the barycentric coordinates of x with respect to the triangle, and the basis functions N_i are defined as follows

$$\begin{aligned} N_1(u,v,w) &= w(2w-1) & \qquad N_2(u,v,w) &= 4uw \\ N_3(u,v,w) &= u(2u-1) & \qquad N_4(u,v,w) &= 4uv \\ N_5(u,v,w) &= v(2v-1) & \qquad N_6(u,v,w) &= 4vw \end{aligned}$$

This scheme yields a C^0 piecewise quadratic interpolant to the radiance on each polygon. This interpolant was found to provide approximations that look smoother and are less prone to Mach bands than the traditional piecewise linear interpolation [LISC92]. Salesin *et al.* [SALE92] describe a piecewise cubic interpolant that can be used instead, if C^1 interpolation is desired.

4.3.3 Computing the Radiance at a Point

To obtain a radiance value at a point x we use the information available to us from the hierarchical solution. Below we describe four different methods that we have experimented with. Pseudocode for the last three methods is given in Figure 4.7.

Method A. The simplest approach is to use the radiance value stored in the hierarchy leaf that contains x. If x is on the boundary between two or more leaves, their values are averaged to yield the radiance at x. This method has no overhead other than locating the containing leaves.

The accuracy of the resulting value depends on the accuracy of the global pass solution. Consider the path from the root of the hierarchy to the leaf

```
Spectrum Shade(node,x)
rad ← 0
foreach l ∈ node.links do
    ff ← FormFactor(x, l.source)
    v ← Visibility(x, l)
    rad ← rad + ff * v * l.source.radiosity
end for
if IsInterior(node) then
    if Contains(node.left, x) then
        rad ← rad + Shade(node.left, x)
    elseif Contains(node.right, x) then
        rad ← rad + Shade(node.right, x)
    else
        rad ← rad + (Shade(node.left, x) + Shade(node.right, x))/2
    end if
end if
return rad

Real Visibility(x, link)
case ShadingMethod in
    B: v ← link.visibility
    C: v ← RecomputeVisibility(x,link.source)
    D: if IsPrimary(link.source) then
            v ← RecomputeVisibility(x, link.source)
        else
            v ← link.visibility
        end if
end case
return v
```

Figure 4.7: Pseudocode for the *Shade* routine

containing the point x. Every node along this path has zero or more links to other nodes, representing areas on primary or secondary sources that illuminate x. The error at x due to one such link between a containing node r and an illuminating node s is bounded by equation (4.2). The total error at x is the sum of the errors over all the contributing links.

Method B. Each contributing link stores the unoccluded form factor from the center of its node to the corresponding source, as well as the visibility factor. To obtain a more accurate radiance value for x we can recompute the unoccluded form factor to each source at point x. Each form factor is multiplied by the visibility stored with the link and by the radiosity of the source. This results in a smaller bound on the error due to a link between r and s

$$E_{rs}(x) \leq \rho_r F_{xs} \left(\overline{B}_s \overline{V}_{rs} - \underline{B}_s \underline{V}_{rs} \right). \tag{4.3}$$

Method C. The next logical step is to recompute both the form factor and the visibility of each source as seen from x. In order to obtain an accurate visibility value the visible parts of the source are computed analytically [LISC92]. As a result, the error bound shrinks further:

$$E_{rs}(x) \leq \rho_r F_{xs} V_{xs} \left(\overline{B}_s - \underline{B}_s \right). \tag{4.4}$$

However, the computation becomes more expensive.

Method D. To reduce the cost, we can recompute the visibility for links to primary light sources only. This is justified by the fact that primary sources are typically responsible for the most noticeable shadows. Moreover, these are precisely the sources for which discontinuities have been computed and inserted into the mesh. Thus, we obtain the same accuracy as in method C

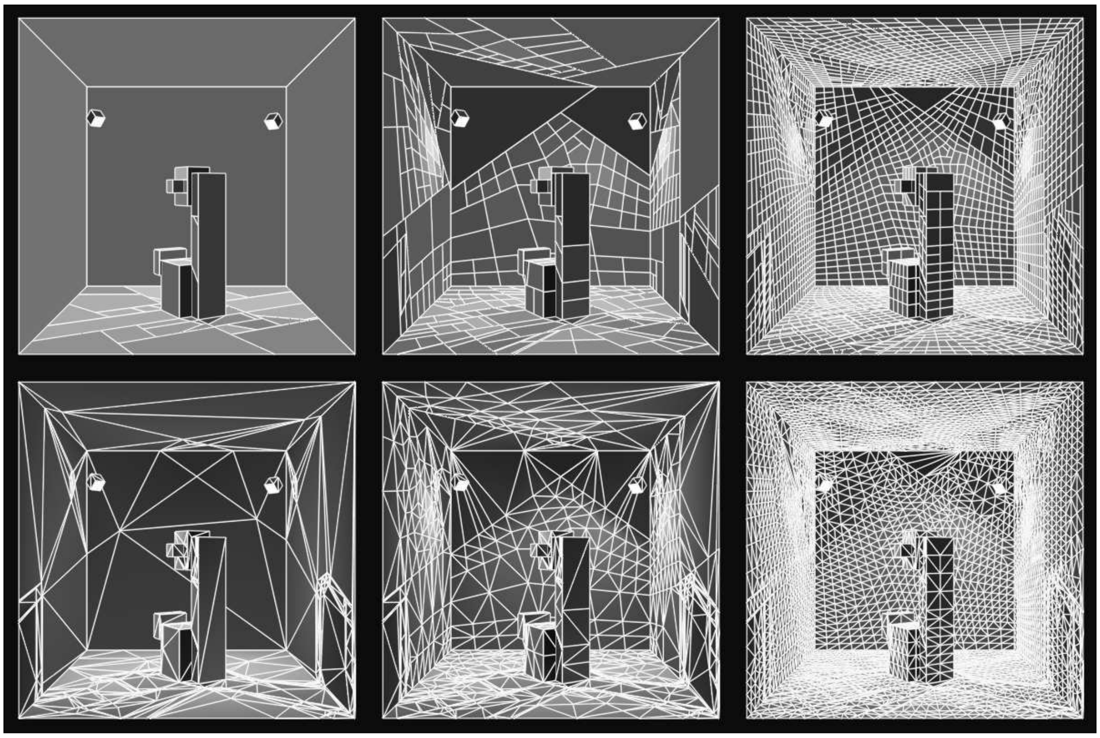


Figure 4.8: Exhibit Room. Global pass solutions (top row) and the corresponding local pass meshes (bottom row). The accuracy of the solutions increases from left to right.

for links to primary sources, while the error due to other links remains the same as in method B.

4.3.4 Results

We demonstrate the local pass on a simple model of a square exhibit room displaying a modern sculpture illuminated by two small square light sources.

Three global pass solutions of the exhibit room are shown at the top row of Figure 4.8, in order of increasing accuracy starting from the left. For each solution, the elements (leaf nodes) of the hierarchical subdivision are shown as flat shaded, outlined polygons. The bottom row of the same figure shows the corre-

Table 4.1: Statistics for images in Figures 4.8 and 4.9. All times are in seconds for execution on an HP 9000/755 workstation.

	Solution Accuracy		
	low	medium	high
input polygons	47	47	47
disc. segments	662	662	662
initial links	674	674	674
total links	749	1376	22466
total nodes	161	855	6095
total leaf nodes	104	451	3071
CDT elements	1691	2446	7884
shading calls	4130	5841	17455
initial linking	0.85	0.85	0.85
discontinuity comp.	0.45	0.45	0.45
hierarchical sol.	0.26	1.00	16.90
triangulation	0.16	0.21	0.65
method A	0.32	0.58	3.32
method B	1.94	3.30	25.24
method C	64.80	92.54	387.20
method D	3.62	4.56	26.20

sponding local pass meshes. Table 4.1 reports statistics for both passes.

The results of the global pass were fed to the local pass four times, once for each of the methods A, B, C, and D, yielding a total of twelve radiosity solutions shown in Figure 4.9. Columns 1, 2, and 3 were computed respectively from the low, medium, and high accuracy global pass solutions shown in Figure 4.8. Each row corresponds to a different shading strategy starting with method A for the top row.

As demonstrated in the top row, method A is prone to visual artifacts: the shading on walls is flat or not sufficiently smooth; some shadows are entirely

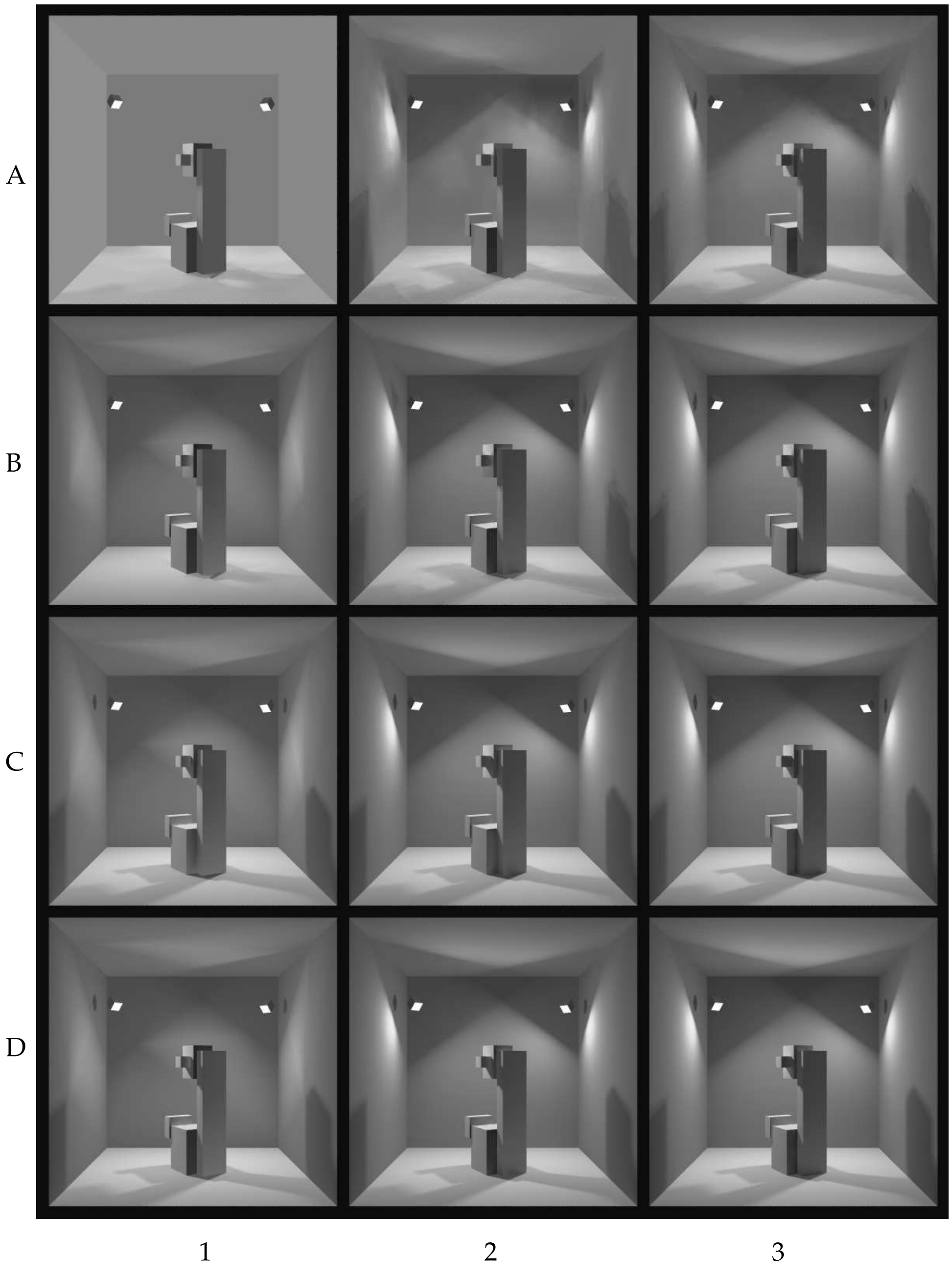


Figure 4.9: Exhibit Room. A comparison of shading strategies. Columns 1, 2, and 3 were computed respectively from the low, medium, and high accuracy global pass solutions shown in Figure 4.8. Each row corresponds to a different shading strategy; starting from the top: method A, method B, method C, and method D.

missing (image A1), while others have incorrect boundaries. These artifacts are the result of interpolating radiance values obtained by sampling the piecewise constant global pass solution.

Method B reduces some of these artifacts. The appearance of unoccluded areas is greatly improved, since accurate form factor are recomputed at every interpolated point in the mesh. However, the penumbra regions of the shadows cast by the sculpture are still incorrect and shadows are still missing from the coarse solution (image B1). The reason is that method B still uses node-to-node visibility factors to approximate node-to-point visibility.

As shown in row C, method C correctly reconstructs all of the shadows. In particular, note the appearance of the shadows in the coarse solution (image C1). This method results in the best visual accuracy we were able to obtain, given a global solution.

Method D yields results that are almost indistinguishable from those given by method C. However, as can be seen from the timings reported in Table 4.1, method D takes only a fraction of the time required by method C. In fact, it is not much more expensive than method B.

When using methods C or D, little difference can be seen between the medium and high accuracy solutions (columns 2 and 3). Although the latter solution is objectively more accurate, from a visual standpoint, the former solution is almost as good. If fact, it is apparent that even very low accuracy global pass solutions can yield results of reasonable visual quality when followed by a local pass using method D (image D1).

When comparing the computation times reported in Table 4.1, it can be seen that the local pass is in most cases costlier than the global pass. It may be argued that the time used by the local pass could be better spent in further refinement of the subdivision hierarchy in the global pass. One might expect that if the hierarchy were sufficiently refined, even a very simple shading strategy would have sufficed for visually accurate results. Figure 4.9, however, demonstrates that this is not the case. Image D2, computed from the medium accuracy global pass followed by method D for the local pass, is visually more accurate than images A3 and B3; yet, it took considerably less time to compute (7 versus 22 and 44 seconds, respectively).

A Quantitative Comparison

To obtain a more quantitative comparison between the four shading methods another set of experiments was performed. First, an extremely fine global solution was computed; a reference image of the environment was then computed by gathering light from every element of that solution for each pixel in the image. Following that, we computed seven increasingly accurate global solutions. The local pass was performed four times on each global solution using the four shading methods, yielding a total of 28 different solutions. From each solution an image was rendered, and the relative L_1 error was computed for each image with respect to the reference image. These errors are plotted in Figure 4.10.

The top plot simply shows the errors corresponding to the four shading methods. As expected, method C is the most accurate of the four, with method D as a close second. The bottom plot shows the errors as a function of the computation time. Here, the advantage of method D over method C becomes apparent: although the two methods result in similar accuracy, the computation times required by method D are by an order of magnitude smaller than these re-

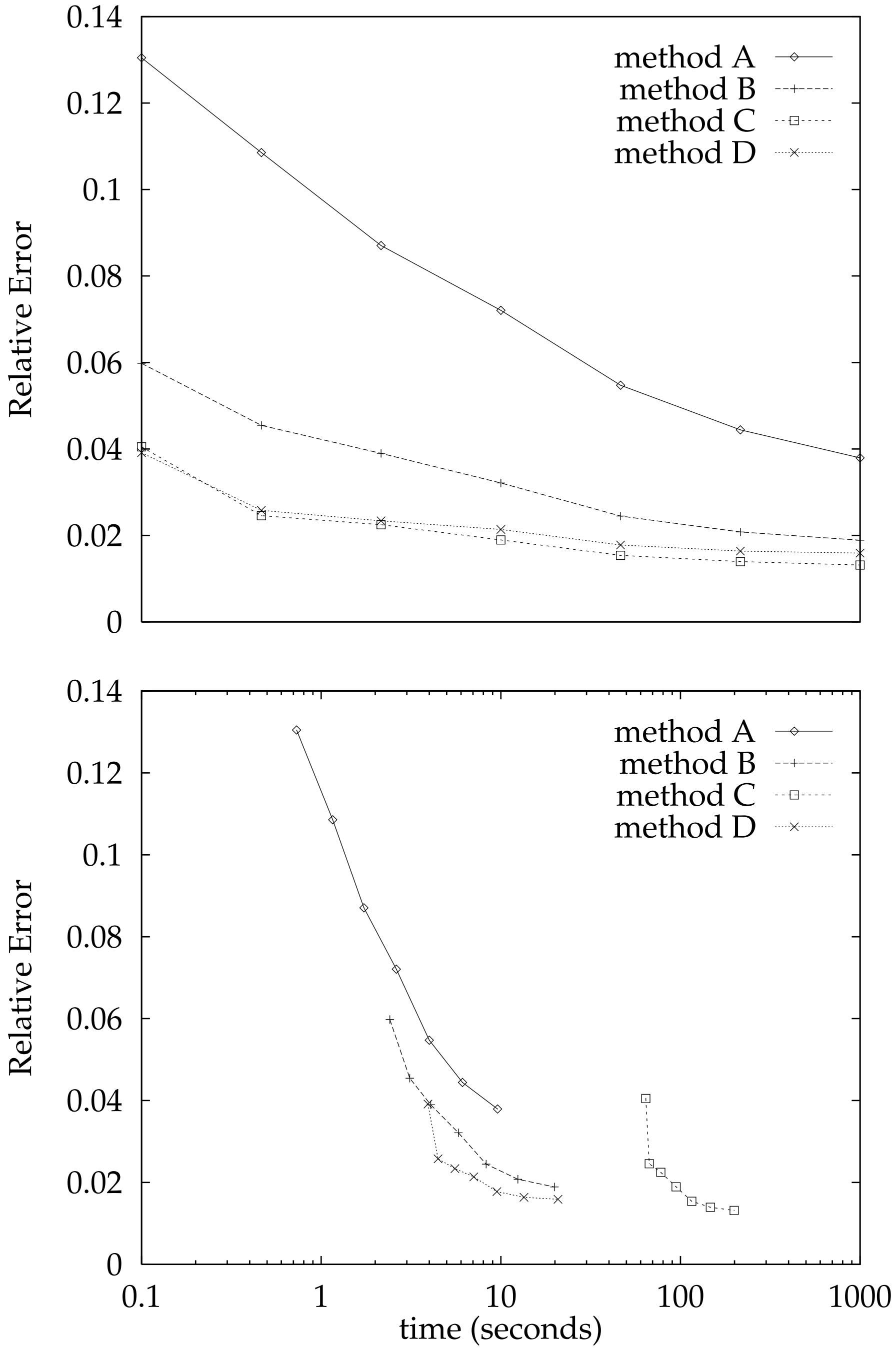


Figure 4.10: A comparison of errors between the four shading methods.

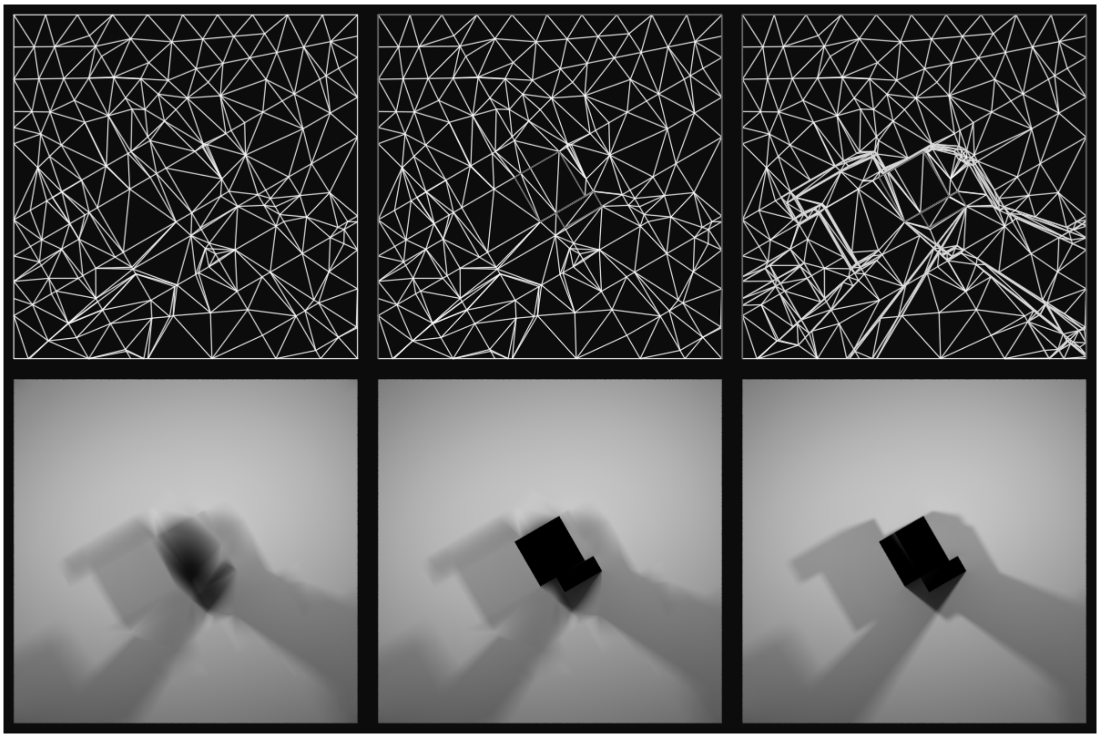


Figure 4.11: Exhibit Room Floor. A comparison of meshing strategies. Mesh (top row) and computed radiance (bottom row) on the floor using simple CDT (left), CDT with D^0 discontinuity segments (middle), and CDT with D^0, D^1, and D^2 discontinuity segments (right).

quired by method C. Note that as the accuracy of the global solutions increases, the difference between methods B and D becomes smaller, so it is possible that for extremely accurate global solutions there would be no advantage in choosing method D over method B.

Discontinuity Meshing Strategies

Another set of comparisons was made to illustrate the importance of including discontinuity segments in the mesh for the local pass. Figure 4.11 shows a view of the floor of the exhibit room. The top row shows the mesh in wireframe with D^0 discontinuities in red and D^1 and D^2 discontinuities in yellow. The bottom row shows the shaded floor as reconstructed by the local pass. All images were

Table 4.2: Statistics for the comparison of meshing strategies shown in Figure 4.11. All times are in seconds for execution on an HP 9000/755 workstation.

	Discontinuities in the Mesh		
	none	D^0	$D^0\ D^1\ D^2$
triangulation	0.09	0.10	0.21
shading	1.84	2.06	4.56
disc. segments	0	48	614
CDT elements	1174	1200	2446
shading calls	2753	3088	5841

computed from the medium accuracy global pass solution shown Figure 4.8 and all of them used method D in the local pass. As can be seen from the top row of Figure 4.11, no discontinuity segments were included in the left mesh, only D^0 discontinuities were included in the middle mesh, and all the discontinuity segments were included in the right mesh.

When comparing the corresponding images in the bottom row, the higher quality of the right image stands out. Image b1 presents many of the visual artifacts typical of conventional radiosity methods: shadow and light leaks, fuzzy shadow boundaries, and incorrectly shaped shadows. Image b2 shows how including D^0 discontinuities greatly reduces shadow and light leaks, but still has problems reproducing shadow boundaries and penumbra areas. Finally, image b3, correctly captures all shadow boundaries. We conclude, therefore, that it is necessary to represent discontinuities explicitly in the local pass mesh, even though some or all of them may have been resolved by the subdivision in the global pass.

As the statistics reported in Table 4.2 show, building a mesh that incorporates discontinuity segments takes longer than building one without discontinuities.

Furthermore, including the discontinuities generally results in a larger number of elements and, consequently, shading the mesh takes longer. We believe, however, that the increased computation time is well justified in most cases.

4.4 Display of Solutions

The radiosity solution computed by our system is represented and output as a collection of triangular meshes: one for each input surface. Each triangle in such a mesh is endowed with six radiosity values: three at the vertices and three at the edge midpoints.

Since the solution is represented in object-space, once it is computed it can be displayed from any viewpoint. We have experimented with two methods for rendering images from our solutions: ray tracing and hardware rendering.

4.4.1 Ray Tracing

To ray-trace an image from our solution we only need to trace primary rays (rays from the viewpoint through the image plane). There is no need to trace shadow rays to light sources, since the all the shading information is already represented by the mesh on each surface. In our implementation, all the input surfaces are culled and clipped against the viewing frustum. The surfaces inside the frustum are projected onto the image plane, and a bounding two-dimensional extent is computed for each projection. A hierarchy of such extents is then constructed and used to accelerate the ray-surface intersection search.

Once the first intersection along the ray has been found, we need to determine the color associated with this ray. The triangular mesh data structure of the intersected surface is searched to determine which face contains the intersection point. Once the containing face is found, the barycentric coordinates of the intersection point are computed. The quadratic polynomial describing the shading over the face is then evaluated at these coordinates, as described in Section 4.3.2.

With ray tracing it is possible to apply texturing to surfaces in the environment, as described by Cohen *et al.* [COHE86]. It is also possible to perform anti-aliasing by using stochastic sampling of the image plane [COOK86].

The images of the sunlit room and the gallery in Figures 4.14 and 4.15 were generated using ray tracing. On an HP 9000/755 workstation it can take from a few minutes to an hour to ray trace a full screen (1284×1024) image, depending on the amount of anti-aliasing that needs to be performed.

4.4.2 Hardware Rendering

In order to render our solutions using today's graphics engines, we need to convert the local pass meshes into a collection of linearly shaded triangles. Our current implementation accomplishes that by simply rendering each quadratic triangle as four linear triangles. For fine meshes this is sometimes unnecessary, and it is sufficient to render each quadratic triangle as a linear one. Figures 4.12 and 4.13 show a sequence of eight images, all hardware rendered from the same radiosity solution.

Obviously, the rendering speed and quality greatly depends on the graphics hardware used. On the HP 9000/755 graphics workstation with a CRX48Z frame buffer we have been able to render solutions consisting of between 60,000 and 167,000 quadratic elements, which were displayed as 240,000 and 668,000 Gouraud shaded triangles, at a rate of between 1.70 and 3.33 seconds per frame.

Figure 4.12: Four hardware rendered images from a radiosity simulation of a museum.

Although this rate is too slow for real-time walkthroughs, it still enables the user to interactively explore the simulated environment. When the user moves, the environment is displayed coarsely, e.g., in wireframe, or by displaying the input polygons only with constant shading (using the average color of each polygon's elements). When the user stops moving, the solution is displayed in full detail.

We have also experimented with displaying our solutions on the Pixel-Planes 5: an experimental heterogeneous multiprocessor graphics system that was de-

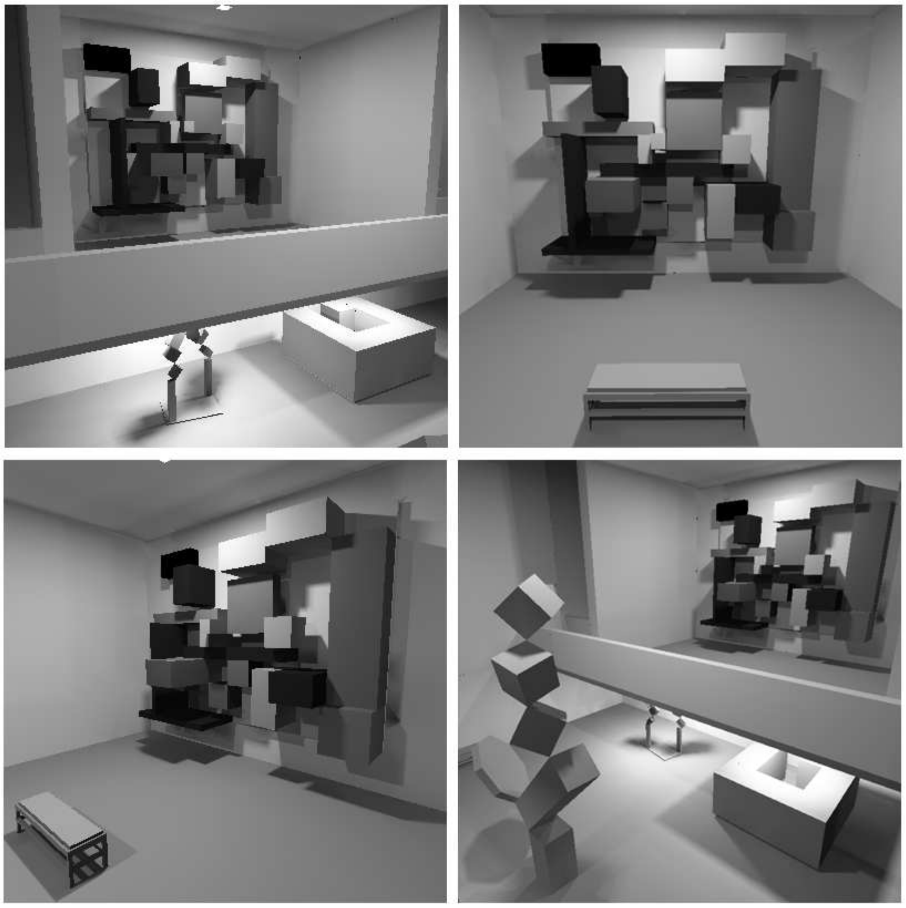


Figure 4.13: Four hardware rendered images from a radiosity simulation of a museum.

veloped at the University of North Carolina in Chapel Hill [FUCH89]. To display our solutions on the Pixel-Planes 5 we have converted each quadratic triangle to a single Gouraud shaded triangle. On this machine, we have been able to display our solutions at rates between 10 and 30 frames per second! Thus, users can walk through the full-detail global illumination rendering of these environments in real time.

Figure 4.14: A comparison of Hierarchical Discontinuity Meshing Radiosity (left) vs. Hierarchical Radiosity (right)

4.5 Results

4.5.1 A Comparison with HR

In this section we demonstrate the performance of our combined approach on an environment of moderate complexity (1,688 input polygons.) Figure 4.14 shows a rendered view of the scene. There are two primary light sources: a small distant polygonal source outside the room simulates sunlight, and another polygonal source close to the ceiling provides the artificial illumination.

The figure shows two images of the same environment. The left image (HDMR) was generated using D^0, D^1, and D^2 primary discontinuity segments in both passes with shading method D in the local pass. To generate the right image (HR) we modified our algorithm to essentially emulate regular HR: discontinuities were not used in either pass, the vertices of the triangles were shaded using

Table 4.3: Statistics for the comparison of hierarchical discontinuity meshing radiosity (HDMR) vs. hierarchical radiosity (HR) shown in Figure 4.14. All times are for execution on an HP 9000/755 workstation.

	Radiosity Algorithm	
	HDMR	HR
initial linking	09:00	09:00
discontinuity computations	02:40	00:00
hierarchical solution	05:02	36:26
triangulation	00:07	00:08
shading computations	05:48	00:37
total time (min:sec)	22:37	46:11
input polygons	1,688	1,688
discontinuity segments	19,340	0
initial links w/o culling	301,636	301,636
initial links with culling	30,781	30,781
total links	62,411	566,532
total nodes	14,730	74,460
total leaf nodes	8,209	38,074
avg. depth of hierarchy	1.91	2.95
CDT elements	61,392	84,072
shading calls	160,306	57,382
recomputed form factors	11,720,352	0
recomputed visibility terms	185,230	0

method A, and linear interpolation was used for display.

As can be expected in a complex environment, the initial linking stage would have resulted in a very large number of initial links, most of which represent interactions of very small magnitude. Using link culling (Section 4.1.1) this number was significantly reduced, without significant changes to the appearance of the solution.

Table 4.3 reports various statistics for the two solutions from which the images in Figure 4.14 were rendered. The total time for the HR solution was roughly twice the total time to compute the HDMR solution. The HR solution produced almost ten times as many links as the HDMR solution, almost five times as many elements in the global solution, and 1.4 times as many triangles in the final mesh. Despite all this, the HDMR solution looks dramatically better than the HR solution. While the latter exhibits many of typical problems of radiosity images, HDMR produces sharp shadow boundaries and correct penumbrae, eliminates shadow and light leaks, and captures some small features that are entirely missed by HR.

The solution produced by HDMR does not only look visually more accurate than the HR solution, it is also more accurate numerically. A reference image was computed for this model, as described in Section 4.3.4 (page 60. Errors were computed with respect to the reference image for each of the two images in Figure 4.14. The maximum relative error over a single pixel for the HDMR image was 16 percent, while the same error for the HR image was 38 percent. The RMS error for the entire image was 0.095 for HDMR and 0.22 for HR.

We attempted to perform a similar comparison with the progressive DM algorithm [LISC92,TAMP93]. However, we were not able to obtain a converged solution for this environment in a reasonable amount of time; after several hours of computation the progressive DM algorithm had only completed a few iterations.

4.5.2 Complexity of Discontinuity Meshing

A legitimate concern regarding discontinuity meshing is that, in theory, l light source edges and m polygon edges can result in $O(lm)$ distinct EV visual events.

In the worst case, each event intersects $O(m)$ polygons, resulting in a total number of $O(lm^2)$ discontinuity segments. In such a case each polygon has $O(lm)$ discontinuity segments, which can result in as many as $O(l^2m^2)$ elements in the discontinuity mesh for that polygon.

We have found that this worst case analysis is too pessimistic in practice. Table 4.4 reports discontinuity meshing statistics for three different environments. The first environment is the sunlit room shown in Figure 4.14. The second environment is a gallery with six large ceiling light sources (see Figure 4.15). The third environment is a room modeled after an interior design by Charles Rennie Mackintosh (see Figure 4.16). Note that in all of these environments the actual number of discontinuity segments is nowhere near the worst-case predicted number.

Table 4.4: A comparison of worst case complexity versus observed complexity of discontinuity meshes for three different environment.

	Environment		
	sunlit room	gallery	Mackintosh
light source edges l	8	24	26
polygon edges m	6,744	2,112	11,544
worst case (per polygon)	107,904	101,376	369,408
actual segments (total)	19,340	22,737	64,238
most segments per polygon	2,653	5,058	8,072
most triangles per polygon	12,724	21,494	27,138

Figure 4.15: Gallery with sculptures

Figure 4.16: Mackintosh room

Chapter 5

A Posteriori Error Bounds and Estimates for Radiosity

In Section 2.1 it was shown that global illumination in diffuse environments is governed by a Fredholm integral equation of the second kind:

$$B(x) = E(x) + \rho(x) \int k(x,y) B(y)\, dy, \tag{5.1}$$

where the kernel of the integral operator

$$k(x,y) = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} v(x,y)$$

describes the point-to-point transfer from y to x.

Radiosity algorithms compute an approximate solution $\widehat{B}(x)$ to equation (5.1) by projecting the continuous integral equation into a finite dimensional function space. The projection of the continuous equation results in a discrete system of n linear equations, where n is the dimension of the finite dimensional space:

$$(I - K)B = E. \tag{5.2}$$

Since radiosity algorithms produce only approximate results, reliable error bounds and realistic error estimates are crucial for assessing the acceptability of a particular solution, as well as for automatic adaptive refinement.

The theory of integral equations [ANSE65,DELV85] provides an *a priori* error analysis of computational methods for Fredholm equations of the second kind. This general framework is valuable for analyzing convergence rates [DELV85], and for bounding various components of the error in terms of the corresponding operator norms [ARVO94b]. However, *a priori* error bounds are typically pessimistic and often difficult to evaluate, requiring information about the operators or the exact solution, which may not be available. In this chapter we discuss *a posteriori* methods that bound the combined effect of the various error sources for a particular instance of the problem. Because the analysis takes place after or during the computation of the solution, such methods can provide tighter bounds more easily.

During the 1980's the issues of *a posteriori* error estimation and automatic adaptive refinement received considerable attention in the finite elements literature [BABU86,SZAB91]. Most of the work has related to partial differential equations. More recently, results have become available also for boundary integral equations and boundary element methods [BREB93].

In the radiosity literature, however, very little has been written regarding *a posteriori* error analysis. In particular, current radiosity algorithms can provide the user with neither guaranteed bounds on the total error $\|B(x) - \widehat{B}(x)\|$, for any function norm $\|\cdot\|$, nor reliable estimates of the total error. Our goal in this chapter is to address these deficiencies.

In Section 5.1 we describe an algorithm that computes two piecewise constant functions that are guaranteed to bound the exact solution $B(x)$ from below and from above. This can be done simultaneously with the computation of the approximate solution $\widehat{B}(x)$. These bounds pertain to both $B(x)$ and $\widehat{B}(x)$, so they can be used to compute a conservative upper bound on the error

$$\|B(x) - \widehat{B}(x)\|.$$

Our algorithm resembles the general bracketing technique suggested by Brown [BROW65], but we utilize special properties of equation (5.1) to produce tighter bounds.

In many instances, a conservative error bound is not required, or is too pessimistic, and a good error estimate is preferable. In Section 5.2 we describe a non-conservative bounding algorithm that computes tighter "bounds", which are no longer guaranteed to contain the exact solution $B(x)$ everywhere. These non-conservative bounds are much cheaper to compute, and result in more realistic error estimates, i.e., estimates which are closer to the actual errors.

Our final contribution is the derivation of an expression that describes the effect that a particular interaction has on the total error bound. This expression uses bounds on importance and on radiosity that can be computed using our algorithms. This gives rise to a new error-driven refinement strategy for hierarchical radiosity, which is shown to be superior to brightness-weighted refinement.

5.1 A Radiosity-Bounding Algorithm

For simplicity, we start with a radiosity-bounding algorithm for full matrix radiosity. Later in this section, we extend this algorithm to work within a more efficient radiosity algorithm, namely hierarchical radiosity.

The goal of the bounding algorithm is to compute two piecewise constant functions $\underline{B}(x)$ and $\overline{B}(x)$ that bound the exact radiosity $B(x)$ from below and from above, respectively. In other words, given a discretization of the environment into a set of n elements $S_1, \ldots, S_n$, we compute $\underline{B}_1, \ldots, \underline{B}_n$ and $\overline{B}_1, \ldots, \overline{B}_n$, such that

$$\underline{B}_i \leq B(x) \leq \overline{B}_i \text{ for all } x \in S_i.$$

Roughly, this is achieved by replacing the coefficients of the discrete linear system (5.2) by infima and suprema bounding the corresponding continuous functions over the areas of the elements $S_1, \ldots, S_n$.

5.1.1 Computing Lower Bounds

Let $\underline{\rho}_i$ and $\underline{E}_i$ denote the infima on the reflectivity and on the emission, respectively, over S_i. Also, let $\underline{F}_{ij}$ denote the infimum on the (point-to-area) form-factor from a point on S_i to S_j:

$$\underline{F}_{ij} = \inf_{x \in S_i} \int_{S_j} k(x,y)\, dy,$$

and let $\underline{K}_{ij} = \underline{\rho}_i \underline{F}_{ij}$. Then the lower bounds vector $\underline{B} = (\underline{B}_1, \ldots, \underline{B}_n)^{\mathrm{T}}$ can be obtained by solving a linear system of n equations

$$(I - \underline{K})\, \underline{B} = \underline{E},$$

using a standard method such as Jacobi or Gauss-Seidel iteration.

Proof. It is easy to see that the matrix $I - \underline{K}$ is strictly diagonally dominant, from which it follows [GOLU89] that the spectral radius of $\underline{K}$ is strictly less than 1. Thus, the inverse matrix $(I - \underline{K})^{-1}$ exists, and can be expressed as a Neumann series

$$(I - \underline{K})^{-1} = I + \sum_{i=1}^{\infty} \underline{K}^i.$$

The solution vector $\underline{B}$ can then be written as

$$\underline{B} = (I - \underline{K})^{-1}\underline{E} = \underline{E} + \sum_{i=1}^{\infty} \underline{K}^i \underline{E}.$$

Alternatively, $\underline{B}$ can be expressed as the limit of the series of vectors $\underline{B}^{(k)}$ that are defined by the recurrence relation

$$\begin{aligned} \underline{B}^{(0)} &= \underline{E} \\ \underline{B}^{(k+1)} &= \underline{E} + \underline{K}\underline{B}^{(k)}. \end{aligned}$$

Note that the above corresponds to Jacobi iteration in matrix notation. It is easy to see by induction that $\underline{B}^{(k)}$ is a lower bound on the radiosity after k light bounces. Therefore, the limit $\underline{B}$ is a lower bound on the total radiosity function $B(x)$. □

5.1.2 Computing Upper Bounds

Unfortunately, we cannot use the same straightforward approach for computing the upper bounds $\overline{B} = (\overline{B}_1, \ldots, \overline{B}_n)^{\mathrm{T}}$. Let $\overline{K}$ be defined in the same way as $\underline{K}$, except that suprema are taken instead of infima. Then the matrix $I - \overline{K}$ is not necessarily diagonally dominant: the entries of $\overline{K}$ are upper bounds on the form-factors and thus the sum of the absolute values of the entries in a row of $I - \overline{K}$ can exceed 1. In such a case, iterative solution methods such as Jacobi or Gauss-Seidel could diverge. Furthermore, it is possible for $I - \overline{K}$ to be singular.

We now describe an iterative algorithm that transforms $I - \overline{K}$ into a strictly diagonally dominant matrix $I - \overline{K}^*$, such that the solution to the linear system of equations

$$\left(I - \overline{K}^*\right)\overline{B} = \overline{E}$$

is a vector of upper bounds. The solution $\overline{B}$ is computed simultaneously with the transformation of $\overline{K}$ into $\overline{K}^*$.

The algorithm starts by forming the matrix $\overline{K}$. We then perform Jacobi iterations using a modified matrix. Because the sum of the form-factors from a given element cannot exceed 1, we can zero out elements in each row of $\overline{K}$ until the form-factors corresponding to the remaining non-zero entries sum to at most 1. To ensure that the modified matrix still yields an upper bound on the current light bounce, we zero out those entries corresponding to the dimmest elements. Since the brightness ordering of the elements may change from one iteration to another, different entries on the matrix may be zeroed out in each iteration. However, as the iterates converge to the solution $\overline{B}$, the order of the elements becomes fixed, at which point we obtain $\overline{K}^*$.

More precisely, at each iteration we sort the current solution vector $\overline{B}^{(k)}$ in order of decreasing brightness, and permute the columns of the matrix accordingly. Each entry in the solution vector is updated by:

$$\overline{B}_i^{(k+1)} = \overline{E}_i + \overline{\rho}_i \left[\sum_{j=1}^{l} \overline{F}_{ij} \overline{B}_j^{(k)} + \left(1 - \sum_{j=1}^{l} \overline{F}_{ij} \right) \overline{B}_{l+1}^{(k)} \right], \tag{5.3}$$

where l is the largest index in row i such that

$$\sum_{j=1}^{l} \overline{F}_{ij} \leq 1.$$

It can be shown by induction that, if the iterates are sorted as described above, $\overline{B}^{(k)}$ is an upper bound on the radiosity after k interreflections. Thus, if the iterates converge, they converge to a total upper bound on the radiosity function $B(x)$.

Proof. To prove convergence, we must examine the behavior of each entry in the solution vector as the algorithm progresses. Because all the entries in the matrix $\overline{K}$ are non-negative, if $\overline{B}^{(0)}$ is set to $\overline{E}$, each entry in the solution vector increases at each iteration, i.e., for all i, $\overline{B}_i^{(k)} \leq \overline{B}_i^{(k+1)}$. Because we do not allow the form-factors in a row to exceed 1, each entry of $\overline{B}^{(k)}$ is bounded from above:

$$\overline{B}_i^{(k)} \leq E_{\max}\left(1+\rho_{\max}+\cdots+\rho_{\max}^k\right) < \frac{E_{\max}}{1-\rho_{\max}},$$

where $E_{\max}$ is the maximum emission and $\rho_{\max}$ the maximum reflectivity in the environment. From elementary analysis, we know that a monotonically increasing series that is bounded from above must converge. Since the vectors $\overline{B}^{(k)}$ are of a finite dimension, it follows that the vector sequence converges. □

5.1.3 A Radiosity-Bounding Algorithm for HR

The algorithms in Sections 5.1.1 and 5.1.2 are presented in the context of the full matrix radiosity algorithm. They involve $O(n^2)$ work to compute the bounds on the form-factors, and therefore are not practical. However, they can be modified to work more efficiently within the HR framework. This requires a few changes to the standard HR algorithm. We need to compute and store upper and lower bounds *Fmin* and *Fmax* on the form-factor associated with each link. We can then gather, push, and pull radiosity bounds, in a way similar to that of the patch radiosities themselves. See Figures 5.1 and 5.2.

Note that gathering, pushing, and pulling are all performed in a single sweep of the hierarchy. Thus, the bounds are updated in place, which makes this a Gauss-Seidel iteration. This iteration converges faster than the Jacobi iteration used in previous HR algorithms. The speedup applies not only to the computation of bounds, but to the computation of the radiosities as well.

Gathering upper bounds is a bit more involved than gathering lower bounds, as can be seen from the pseudocode in Figure 5.2. *GatherUpperBounds* essentially pushes links down to the leaves, where they are sorted according to the

```
GatherLowerBounds(node, lower)
foreach link ∈ node.links do
    lower += node.rho * link.Fmin * link.source.lower
end for
if IsLeaf(node) then
    node.lower = lower + node.emission
else
    node.lower = ∞
    foreach child ∈ node.children do
        GatherLowerBounds(child, lower)
        node.lower = min(node.lower, child.lower)
    end for
end if
```

Figure 5.1: Pseudocode for *GatherLowerBounds*

brightnesses of the corresponding source nodes. The upper bound of the leaf node is then updated using equation (5.3).

One undesirable feature of this algorithm is the sorting of the links, which implies that $O(k \log k)$ work must be done to update the upper bound at each leaf node, where k is the number of the contributing links. However, complete sorting can be avoided in many cases. Sorting is only necessary when the sum of the contributing form-factors exceeds 1. Even then, only a few of the dimmest sources must be removed from the list. This can be done efficiently using a heap data structure with a *DeleteMin* operation. Tarjan [TARJ83] provides a detailed description of such data structures.

```
GatherUpperBounds(node, contribList)
foreach link ∈ node.links do
    add the pair (link.Fmax, link.source.upper) to contribList
end for
if IsLeaf(node) then
    ffSum = 0
    node.upper = node.emission
    Sort(contribList)
    foreach pair (ff, upper) ∈ contribList do
        if ffSum + ff < 1 then
            ffSum += ff
            node.upper += node.rho * ff * upper
        else
            node.upper += node.rho * (1 − ffSum) * upper
            break
        end if
    end for
else
    node.upper = 0
    foreach child ∈ node.children do
        GatherUpperBounds(child, contribList)
        node.upper = max(node.upper, child.upper)
    end for
end if
```

Figure 5.2: Pseudocode for *GatherUpperBounds*

5.1.4 Bounding Form-Factors

Thus far we have assumed that the bounds $\underline{F}_{ij}$ and $\overline{F}_{ij}$ are available to us. How do we obtain them? We know of no analytical way of computing such bounds. Thus, we must resort to numerical optimization to find the minimum and the maximum values of the point-to-polygon form-factor from points on S_i to S_j. Such numerical methods are discussed in detail by Campbell [CAMP91]. These methods generally require the computation of form-factor gradients in the presence of occluders, which can be done either using finite differences or analytically, as described by Arvo [ARVO94a].

In our current implementation we estimate form-factor bounds by evaluating the analytical point-to-polygon form-factor [SPAR63a] at the center and at the vertices of each receiving patch. In the partially occluded case, the analytical formula is applied only to the visible parts of the source [NISH83]. The upper and lower bounds on the form-factor are then set to the maximum and the minimum of these values. While these bounds are not conservative, as elements decrease in size the point-to-area form-factor function becomes monotonic over most elements, and the heuristic yields accurate bounds. This is particularly so when the discontinuity-driven subdivision strategy described in Section 4.2 is employed.

5.1.5 Results

The plots in Figures 5.4, 5.5, and 5.6 show the piecewise-constant bounding functions computed by the hierarchical bounding algorithm for three simple environments shown in Figure 5.3. The exact radiosity function is also plotted for each of the three cases. The functions are plotted along the dotted lines across the floor

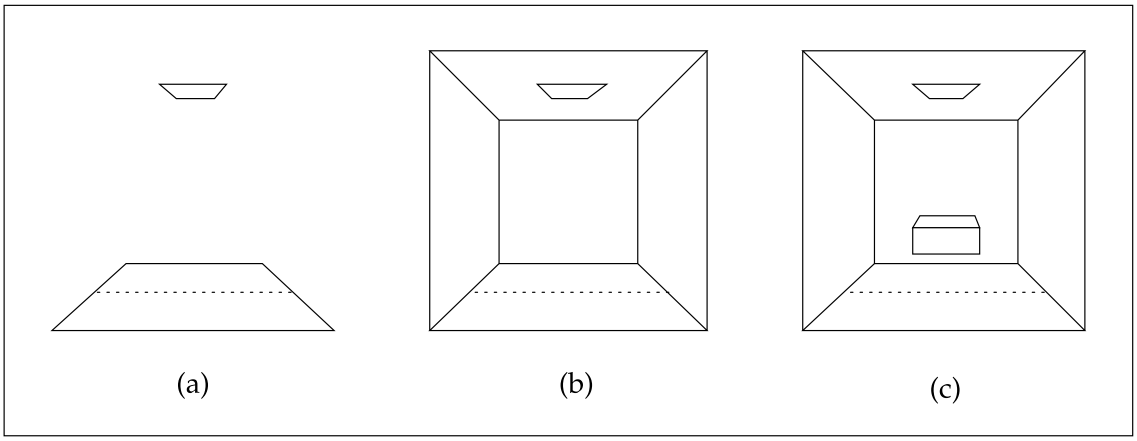


Figure 5.3: Three simple test cases: (a) direct illumination only; (b) inter-reflections without occlusion; (c) inter-reflections with occlusion. All the surfaces are grey with reflectivities between 0.3 and 0.7.

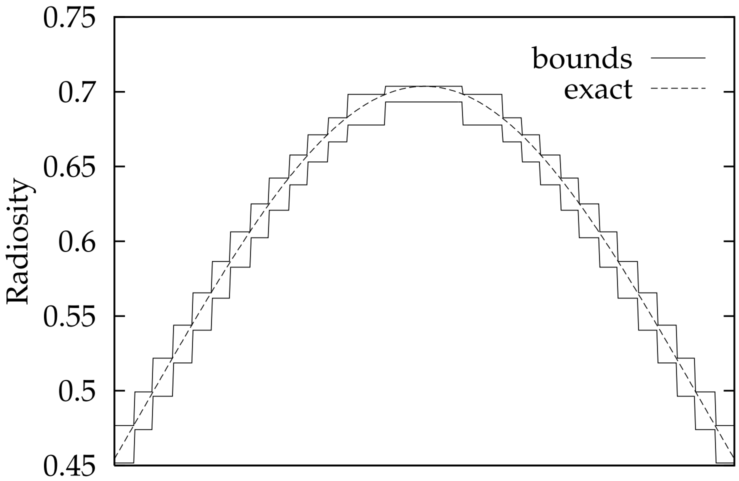


Figure 5.4: Upper and lower bounds for the configuration shown in Figure 5.3a. The smooth curve in the middle is the exact radiosity function.

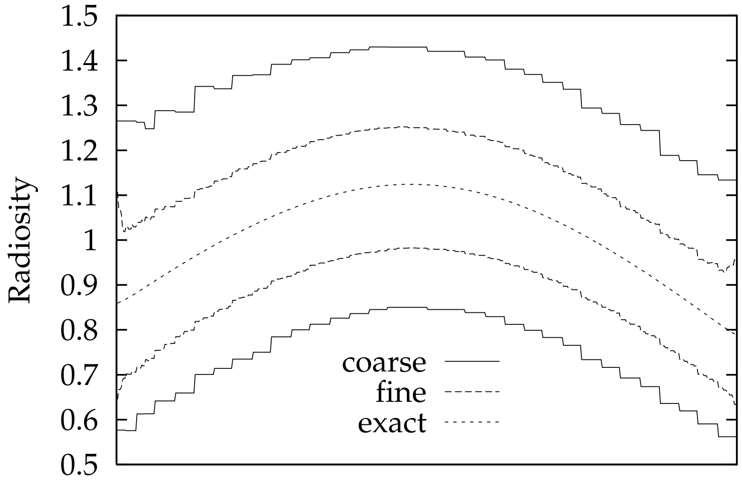


Figure 5.5: Nested lower and upper bounds corresponding to two HR solutions for the configuration in Figure 5.3b. One solution is coarse and the other is fine. The smooth curve in the middle is the exact radiosity function.

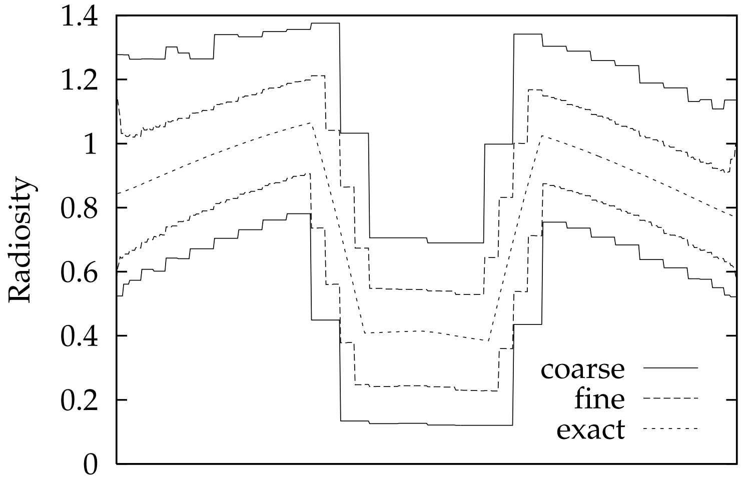


Figure 5.6: Nested lower and upper bounds corresponding to two HR solutions for the configuration in Figure 5.3c. One solution is coarse and the other is fine. The smooth curve in the middle is the exact radiosity function.

of each environment.

The bounds in Figure 5.4 were found in a single iteration, since there are no interreflections in this environment, and errors do not propagate. Because of this, the bounds are very tight: in fact, tighter piecewise constant bounds are only possible if more elements are used.

Figures 5.5 and 5.6 show two cases with interreflections, which result in looser bounds due to error propagation. As illustrated in the figures, the bounds computed by the algorithm become tighter as the accuracy of the solution increases. Note that the radiosity function and the bounds are not symmetric, because in the corresponding environments the reflectivity of the left wall is higher than that of the right wall.

5.2 Computing Realistic Error Estimates

Having obtained bounds on the radiosity of each element, we can obtain an upper bound on the local error there. Assuming that the approximate radiosity B_i lies halfway between $\underline{B}_i$ and $\overline{B}_i$, the errors over S_i in the $L_\infty, L_1,$ and L_2 norms are:

$$\|B(x) - B_i\|_\infty \equiv \max_{S_i} |B(x) - B_i| \leq (\overline{B}_i - \underline{B}_i)/2, \tag{5.4}$$

$$\|B(x) - B_i\|_1 \equiv \int_{S_i} |B(x) - B_i|\, dx \leq A_i(\overline{B}_i - \underline{B}_i)/2, \tag{5.5}$$

$$\|B(x) - B_i\|_2^2 \equiv \int_{S_i} |B(x) - B_i|^2 dx \leq A_i(\overline{B}_i - \underline{B}_i)^2/4, \tag{5.6}$$

where A_i is the area of S_i. However, these error bounds represent a worst case scenario, and do not give an accurate error estimate even when the bounds on radiosity are as tight as in Figure 5.4. More realistic estimates are obtained if we assume that the values of $B(x)$ over S_i are uniformly distributed between $\underline{B}_i$ and

$\overline{B}_i$. Under this assumption the L_1 and L_2 norm errors are estimated as:

$$||B(x) - B_i||_1 \approx A_i(\overline{B}_i - \underline{B}_i)/4, \tag{5.7}$$

$$||B(x) - B_i||_2^2 \approx A_i(\overline{B}_i - \underline{B}_i)^2/12. \tag{5.8}$$

We compute local error estimates at all the leaves of the hierarchy. These estimates are then pulled up to the roots of the hierarchies, and combined to obtain the estimate for the total error. The error at a parent node is the sum of the children's errors in the L_1 norm, and the maximum over the children's errors in L_∞ norm. For the L_2 norm, the squares of the children's errors are summed and the square root of the sum is taken.

5.2.1 Obtaining Tighter Non-Conservative Bounds

The conservative algorithm produces bounds which are quite loose even for simple environments, resulting in pessimistic error estimates. For most practical purposes good non-conservative error estimates could prove more useful than conservative error bounds. To quote Delves and Mohamed [DELV85]:

> Given two error estimates of equal realism and equal cost and given that one is also a bound, we would clearly prefer the bound. In practice bounds usually prove expensive and pessimistic. Most users will then usually prefer a cheaper and more realistic error estimate, recognizing and accepting that estimates can sometimes err on the side of optimism.

We now present a way for computing tighter, although not conservative, bounds on radiosity. Error estimates computed from these radiosity bounds are not guaranteed error bounds, but they are typically much closer to the true errors. To compute these tighter bounds we make the assumption that the error over an element results primarily from the errors in its links, and not from errors on the other elements. Under this assumption, we approximate $\underline{B}$ and $\overline{B}$ as follows:

$$\begin{aligned}\underline{B} &\approx \underline{E} + \underline{K}\widehat{B} \\ \overline{B} &\approx \overline{E} + \overline{K}\widehat{B}.\end{aligned}$$

Thus, we are using the approximate radiosities $\widehat{B}$ in order to compute the upper and lower bounds $\underline{B}$ and $\overline{B}$. This is done by performing a single gathering iteration over the environment using the same routines as before for gathering lower and upper bounds. Because only one iteration is performed, error propagation is no longer accounted for, but the computation becomes significantly cheaper. More conservative estimates can be obtained by using two or more iterations, yielding a trade-off between speed and conservativity.

An additional change to the algorithm has been used to produce even tighter bounds. Instead of bounding each form-factor separately and summing the bounds, we directly bound the radiosity contributed to each node through all its links: the contribution is evaluated at the center and at the vertices of the patch, and the extrema values are used as bounds.

5.2.2 Results

To test the quality of our error estimates we computed an extremely fine radiosity solution for each of the three environments in Figure 5.3. The floor polygon in each environment was sampled on a 400 by 400 grid to serve as a reference. We then computed a series of radiosity solutions of various degrees of accuracy for each environment. The floor polygon was sampled as before, and the difference between each solution and the reference solution was computed. This difference is referred to as the *measured* error. During each solution, two error esti-

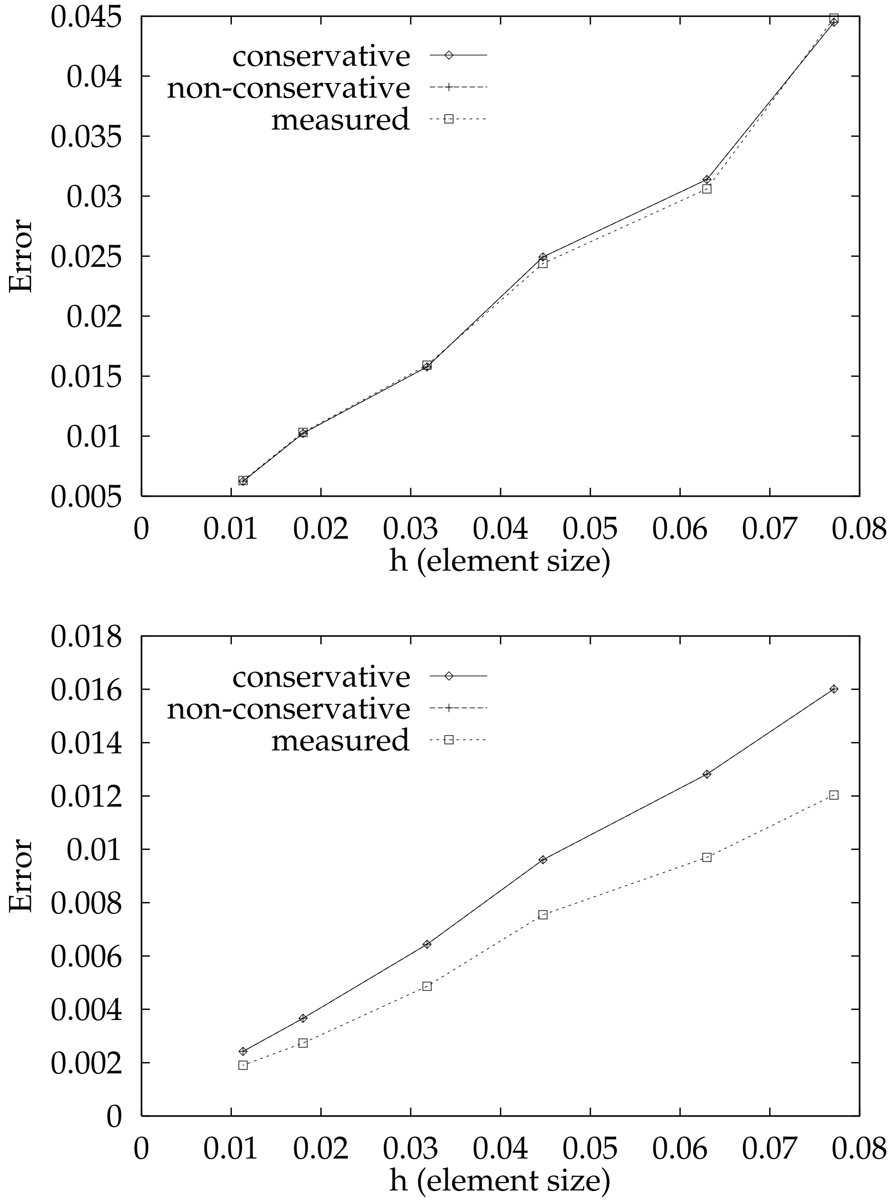


Figure 5.7: Estimated and measured errors as a function of the element size for the environments shown in Figure 5.3a. L_∞ errors are shown in the top and L_1 errors in the bottom.

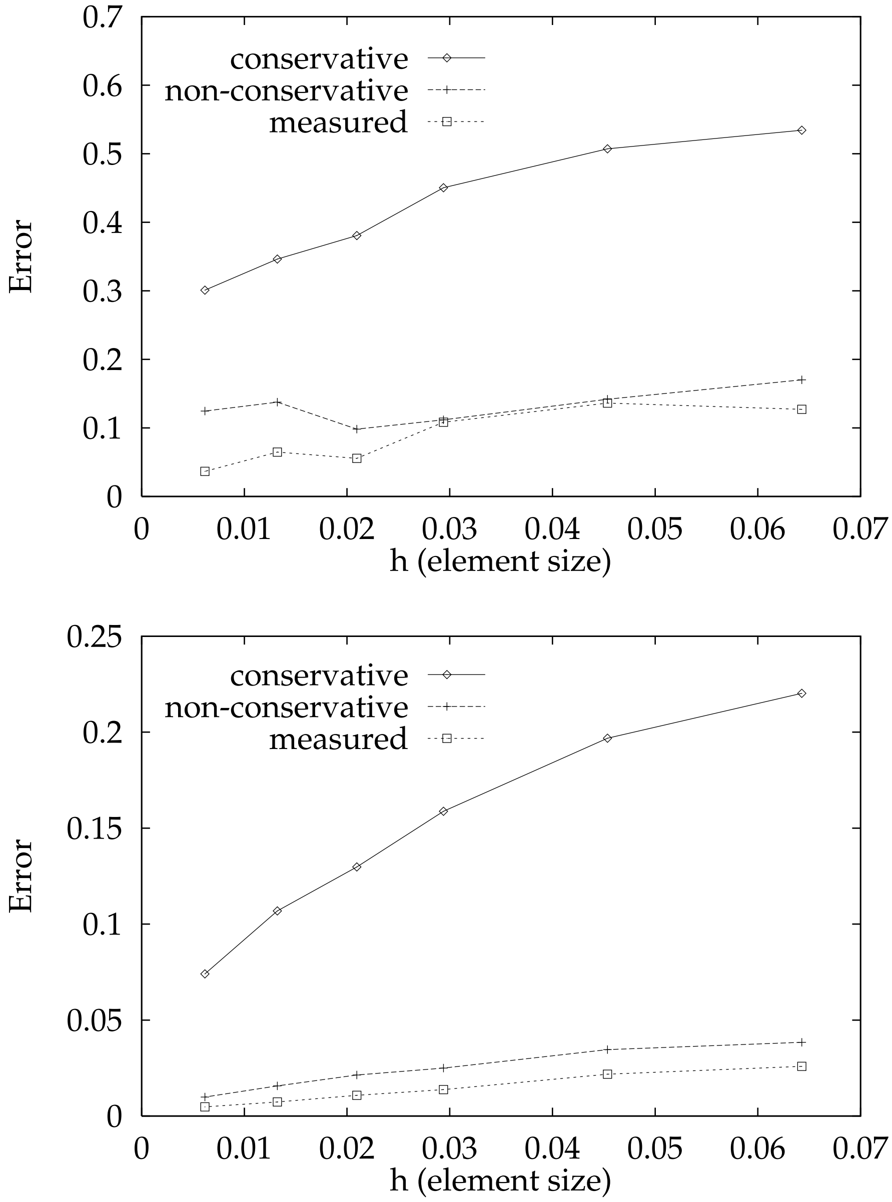
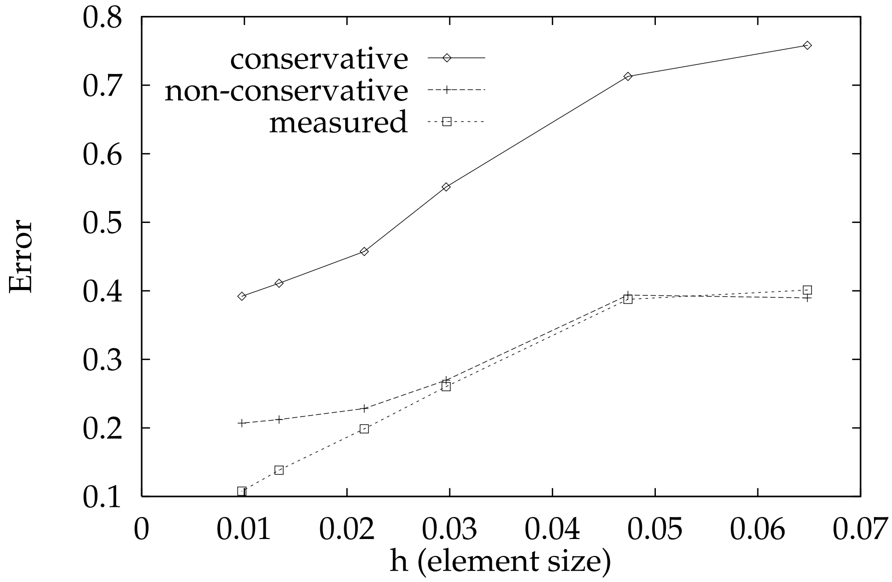


Figure 5.8: Estimated and measured errors as a function of the element size for the environments shown in Figure 5.3b. L_∞ errors are shown in the top and L_1 errors in the bottom.



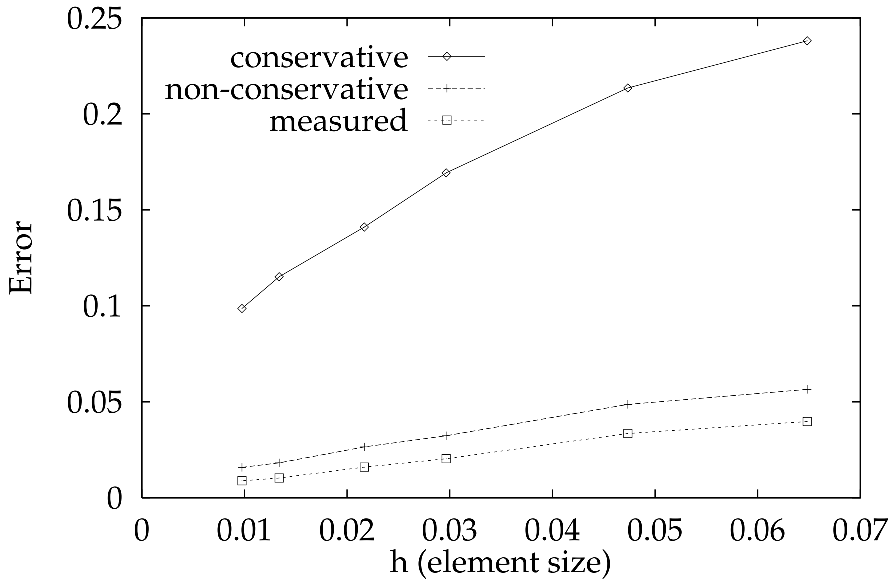


Figure 5.9: Estimated and measured errors as a function of the element size for the environments shown in Figure 5.3c. L_∞ errors are shown in the top and L_1 errors in the bottom.

mates were computed: one from the conservative bounds, and the other from the tighter non-conservative bounds. In both cases equations (5.4), (5.7), and (5.8) were used to estimate the errors.

Figures 5.7 through 5.9 show plots of the error estimates together with the measured error for each of the three test cases. Error plots for L_2 norm are not given, since they look almost exactly the same as the L_1 plots for these test cases.

In the first pair of plots (corresponding to Figure 5.3a) the conservative and the non-conservative bounding algorithms yield the same bounds, and therefore the corresponding error estimates are always the same. For the L_∞ norm these estimates follow tightly the measured error, and for the L_1 norm they give realistic upper bounds.

In the remaining two cases there is a significant difference between the two types of error estimates. In the L_∞ norm, the non-conservative estimates do not always yield a bound on the error (see top plot in Figure 5.9). Also, these estimates do not always decrease as the accuracy of the solution increases.

In the L_1 norm, both types of estimates bound the measured error from above, but the conservative error estimates are much more pessimistic than their non-conservative counterparts. Both types of estimates become closer to the measured error as the accuracy of the solutions increases. Both the estimated and the measured errors vary linearly in the element size h, as expected for piecewise-constant approximations.

5.3 Error-Driven Refinement

Hanrahan's hierarchical radiosity algorithm [HANR91] adaptively improves the accuracy of a solution by estimating the error associated with each link. All links for which this error estimate exceeds a given threshold ε are refined and the solution is recomputed. More specifically, Hanrahan et al. suggest refining all those links for which the estimated upper bound on the transferred energy exceeds ε. A link is refined by subdividing the node which has the greater form-factor, as seen from the center of the other node. This strategy is referred to as *brightness-weighted* refinement, as it gives priority to links transferring energy from bright sources.

We make two observations with respect to brightness-weighted refinement: (i) The links carrying the largest amount of energy are not necessarily the ones with the greatest errors in the transferred energy. (ii) Due to propagation of errors, refining the link between nodes i and j reduces not only the error over patch i, but may also reduce errors on other patches receiving light from it. Thus, to make the most benefit of each link refinement, priority should be given to links with the greatest effect on the total solution error.

The first observation suggests using $\Delta K_{ij} = \overline{K}_{ij} - \underline{K}_{ij}$ as the estimate of the link form-factor error, rather than using an upper bound on the form-factor. The error in the transferred energy is then given by $\Delta K_{ij}\overline{B}_j$.

To address the second observation, we need an expression that relates ΔK_{ij} to the overall value of $\|\overline{B} - \underline{B}\|_1$, which is an upper bound on the overall error in L_1 norm. This expression is derived in the next section.

5.3.1 Linear Functionals and Importance

Any linear functional $f(B)$ of the radiosities can be expressed as an inner product $R^{\mathrm{T}}B$, where R_i gives the contribution of B_i to the value of $f(B)$. In particular, since all the entries in the vector $\overline{B} - \underline{B}$ are non-negative, the value of its L_1 norm can be expressed as a linear functional:

$$\left\|\overline{B}-\underline{B}\right\|_1 = R^{\mathrm{T}}\left(\overline{B}-\underline{B}\right),$$

where $R = [A_1, \ldots, A_n]^{\mathrm{T}}$ is the vector of patch areas.

Let M denote the transport matrix $I - K$. Since M is a real matrix, its adjoint is given simply by M^{T}. *Importance*, which we shall denote by Z, is defined as the solution to the adjoint of the transport equation [LEWI65,SMIT92]:

$$M^{\mathrm{T}}Z = R, \tag{5.9}$$

where R is determined by some linear functional $f(B)$. It follows that

$$f(B) = R^{\mathrm{T}}B = \left(M^{\mathrm{T}}Z\right)^{\mathrm{T}}B = Z^{\mathrm{T}}MB = Z^{\mathrm{T}}E.$$

Thus, the i-th element of the importance vector gives the contribution made by a unit of emission at patch i to the value of the scalar function $f(B)$.

Using the above, we can now derive the following expression:

$$\left\|\overline{B}-\underline{B}\right\|_1 = \underline{Z}^{\mathrm{T}}\Delta K\overline{B} + \underline{Z}^{\mathrm{T}}\left(\overline{E}-\underline{E}\right). \tag{5.10}$$

This expression relates $\|\overline{B}-\underline{B}\|_1$ to the link errors ΔK and to errors in the emission. The term $\underline{Z}^{\mathrm{T}}\Delta K\overline{B}$ is a sum over all links. The values summed are the link form-factor error ΔK_{ij} weighted by the lower bound on the importance of the receiver $\underline{Z}_i$ and the upper bound on the radiosity of the source $\overline{B}_j$. Intuitively, $\Delta K_{ij}\overline{B}_j$ is an upper bound on the incorrect radiosity that patch i receives from patch j, while $\underline{Z}_i$ gives the contribution of this incorrect radiosity on the total error bound. The $\underline{Z}^{\mathrm{T}}(\overline{E}-\underline{E})$ term gives the effect non-constant emission has on the total error bound.

Proof. Let $\overline{M} = I - \overline{K}^*$ and $\underline{M} = I - \underline{K}$. The system for lower bounds on the radiosities and its adjoint equation can now be written as

$$\underline{MB} = \underline{E} \quad \text{and} \quad \underline{M}^{\mathrm{T}}\underline{Z} = R,$$

and the upper bounds system can be written as $\overline{MB} = \overline{E}$. Noting that

$$\Delta K = \overline{K} - \underline{K} = \underline{M} - \overline{M},$$

we obtain

$$\overline{MB} = (\underline{M} - \Delta K)\overline{B} = \overline{E},$$

from which it follows that

$$\underline{M}\overline{B} = \overline{E} + \Delta K\overline{B}. \tag{5.11}$$

Since $R^{\mathrm{T}}\underline{B} = \underline{Z}^{\mathrm{T}}\underline{E}$, we can express $R^{\mathrm{T}}(\overline{B} - \underline{B})$ in terms of ΔK as follows:

$$\begin{aligned}
R^{\mathrm{T}}(\overline{B} - \underline{B}) &= R^{\mathrm{T}}\overline{B} - R^{\mathrm{T}}\underline{B} \\
&= R^{\mathrm{T}}\overline{B} - \underline{Z}^{\mathrm{T}}\underline{E} \\
&= (\underline{M}^{\mathrm{T}}\underline{Z})^{\mathrm{T}}\overline{B} - \underline{Z}^{\mathrm{T}}\underline{E} \\
&= \underline{Z}^{\mathrm{T}}\underline{M}\overline{B} - \underline{Z}^{\mathrm{T}}\underline{E}.
\end{aligned}$$

Substituting equation (5.11) into the above expression we have

$$\begin{aligned}
R^{\mathrm{T}}(\overline{B} - \underline{B}) &= \underline{Z}^{\mathrm{T}}(\overline{E} + \Delta K\overline{B}) - \underline{Z}^{\mathrm{T}}\underline{E} \\
&= \underline{Z}^{\mathrm{T}}\Delta K\overline{B} + \underline{Z}^{\mathrm{T}}(\overline{E} - \underline{E}).
\end{aligned}$$

□

5.3.2 A New Refinement Strategy

Equation (5.10) suggests a new error-driven refinement strategy for HR. In this strategy we refine all links for which $\underline{Z}_i\Delta K_{ij}\overline{B}_j > \varepsilon$, since these links have the greatest effect on the total error bound. In order to evaluate this expression for each

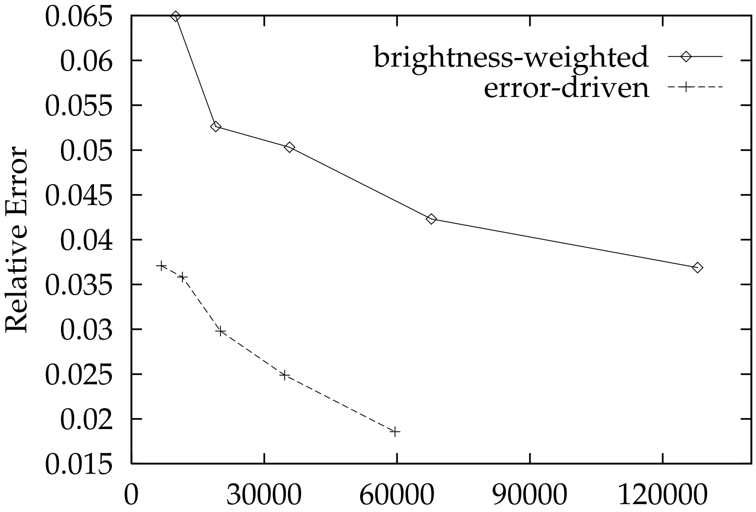


Figure 5.10: Measured relative L_1 error as a function of the number of links: brightness-weighted vs. error-driven refinement.

link, the routine *GatherLowerBounds* must be modified to solve for the lower bounds on importance in addition to lower bounds on radiosity. These changes are straightforward, and they do not significantly increase the amount of work done in the routine.

When we refine a link between a receiver node i and a source node j, we need to decide which node to subdivide to obtain the greatest reduction in the error. We use the following heuristic: node i is subdivided if

$$\left(\overline{K}_{ij} - \underline{K}_{ij}\right) \overline{B}_j \geq \overline{K}_{ij} \left(\overline{B}_j - \underline{B}_j\right),$$

and node j is subdivided otherwise. Intuitively, this means that we subdivide the receiver if the error due to the non-constant form-factor is greater than the error due to non-constant radiosity on the source.

5.3.3 Results

We implemented both brightness-weighted and error-driven refinement strategies and tested them on the simple environment shown in Figure 5.3c. The plot in

Figure 5.10 shows how the measured relative L_1 error on the floor decreases as a function of the number of links. This plot demonstrates clearly that error-driven refinement is much more efficient: it is capable of achieving much smaller errors using only a fraction of the links required by brightness-weighted refinement. Similar results were observed for the environment in Figure 5.3b.

Chapter 6

Conclusion

6.1 Summary

We have presented new radiosity algorithms that address several important open problems with current radiosity methods.

First, we have improved and combined together two recently developed radiosity approaches: hierarchical radiosity (HR) and discontinuity meshing (DM). The resulting radiosity method is superior to both of its ancestors: it is more accurate than the HR algorithm, both numerically and visually, and it is faster and more flexible than DM algorithms. The new method is capable of producing images of high visual quality even from coarse global illumination simulations.

In contrast to previous techniques for post-processing radiosity solutions, our algorithms operate entirely in object-precision, which enables users to walk through high-fidelity shaded virtual environments in real time, using high-end graphics hardware.

We have also described a hierarchical radiosity algorithm that computes conservative bounds on the exact radiosity function simultaneously with the computation of the approximate solution. In contrast to previous attempts to estimate the errors in radiosity, our algorithm properly accounts for the propagation of errors due to interreflections, and provides conservative upper bounds on the error. A non-conservative version of this algorithm has been presented that provides tighter bounds and more realistic error estimates. Finally, we have derived a new error-driven adaptive refinement strategy for hierarchical radiosity that significantly outperforms the brightness-weighted refinement strategy.

We hope that the results described in this thesis will prove useful not only for high-fidelity image synthesis, but also for other fields in which similar integral equations arise. Examples include radiative heat transfer, illumination engineering, and remote sensing. In all of these fields the numerical accuracy of the results is even more crucial than in image synthesis.

6.2 Directions for Further Research

6.2.1 Extensions for Complex Environments

Using our system we have been able to compute radiosity solutions for a variety of environments consisting of up to approximately 5,000 polygons. It is difficult to compute solutions for more complex environments because of two reasons:

1. As was mentioned earlier, the complexity of the initial linking stage in HR is $O(n^2)$, where n is the number of input polygons. For complex environments, the initial linking time becomes dominant, even when link culling heuristics are used.

2. Representing the radiance function on thousands of surfaces, many of which have complex shadows, results in many elements in the local pass mesh. Once the number of elements becomes too large to fit in the physical RAM memory, paging starts to drastically slow down the computation.

However, these problems are not inherent in our approach. Several techniques, described below, have been recently presented that improve the handling of very complex environment, and we believe that our method can (and should) be combined with these techniques.

Clustering

Smits *et al.* [SMIT94] present a clustering algorithm, which significantly speeds up HR in complex environments. This algorithm works by clustering the input surfaces together and allowing energy transfer to occur between the clusters, while maintaining reliable error bounds on each transfer. In contrast with the $O(n^2)$ links that may be created by HR, the clustering approach generates only $O(n)$ links.

It should not be difficult to incorporate clustering into our system. In the global pass, inter-cluster energy transfers should be handled as described by Smits *et al.* , while interactions between individual patches should be handled as described in Chapter 4. For the local pass, we need to develop methods for gathering energy from a cluster of surfaces to a mesh node.

Importance

Importance-driven radiosity [SMIT92] is another improvement to HR that is very useful when only a part of the environment is of visual interest to the user. This method drastically reduces the amount of computation for regions that contribute little to the surfaces of interest. For instance, suppose we have a model of a building that is too complex to simulate entirely on a current workstation. Using importance, we can simulate the global illumination one room at a time, by making each room important in turn.

Again, it is easy to extend our system to solve for importance, simultaneously with the radiosity solution in the global pass. The local pass will then need to reconstruct the radiance function only on surfaces that were specified as important.

Visibility Preprocessing

Complex environments are often densely occluded: any given surfaces in such an environment can only see a small fraction of the environment. Teller and Hanrahan [TELL93] describe a global visibility preprocessing algorithm, which removes totally invisible pairs of surfaces from further consideration, and generates lists of blockers for the partially occluded pairs. This can result in dramatical acceleration for global illumination algorithms such as HR. Global visibility preprocessing would be even more useful for our system, since it can speed up not only the HR algorithm, but also the process of locating the discontinuities.

6.2.2 Improving Discontinuity Meshing

Adaptive Discontinuity Meshing

Our algorithm is particularly effective for environments with a limited number of primary sources that are responsible for the most noticeable shadows. This is a result of computing and using only discontinuities due to primary light sources, both in the global and in the local pass. Additionally, in the local pass, we recompute the exact visibility only for primary sources. In general, however, primary light sources do not dominate the illumination on all the surfaces in an environment. Our algorithm should be extended to determine the most dominant sources, primary or secondary, with respect to each receiving surface. This set should then be used both for computing the discontinuities on that surface and for determining when visibility should be recomputed in the local pass.

Even if the source is important for a particular receiver, not all the discontinuities corresponding to the source are equally significant. In the global pass, for example, we should choose discontinuities that would resolve partial visibility most effectively and reduce the approximation error the most, rather than ones that split the node most evenly. In the local pass we need to identify the discontinuities that are visually significant and insert only these discontinuities into the mesh.

Thus, we should develop techniques for quantifying the effect of visual events and discontinuities on the approximation. Instead of precomputing all the discontinuities, we should compute on the fly only as many discontinuities as are needed to achieve the desired accuracy. This should result in considerable savings.

Isolux Contour Meshing

Discontinuity meshing is effective because it splits surfaces into regions over which the radiance function is smooth. However, it does not tell us how the interior of such regions should be meshed. In our local pass we insert the corners of the hierarchy leaves into the mesh, but it is not clear whether this is optimal.

We believe that isolux contour meshing [ARVO94a] should be very effective for meshing such smooth regions, particularly for large unoccluded regions or for large penumbrae regions due to large sources like windows and skylights. Thus, combining discontinuity meshing with isolux contour meshing within large smooth regions appears promising.

6.2.3 Real-Time Walkthroughs for Complex Environments

Although we have been able to walk through our radiosity solutions in real time on the Pixel-Planes 5 (an experimental parallel machine at University of North Carolina, Chapel Hill), solutions for more complex environments may contain too many elements to allow interactive frame rates with existing technology. Below, we briefly discuss some recent work on speeding up the hardware display of complex environments and offer two new ideas for further research.

Adaptive Detail Control

Funkhouser and Séquin [FUNK93] present an algorithm for interactive frame rates during display of complex environments. They observe that viewing frustum culling and visibility processing [TELL91] alone are not sufficient to guarantee an interactive frame rate, since even the visible complexity can be too large. A further reduction in the number of rendered polygons can be obtained by representing each object at multiple levels of detail and rendering the lowest acceptable level. Funkhouser and Séquin perform constrained optimization that selects a level of detail for each object to be rendered in order to produce the best results while still meeting the target frame rate.

Their algorithm has been used to display flat shaded building models and it should be extended to display radiosity solutions, where most of the complexity comes from the shading on the surfaces of the objects, rather than from the geometry of the object. Thus, we should develop algorithms for constructing multi-resolution representations for shading on surfaces, and for smooth blending between adjacent levels of detail.

Quadratic Element Display

The solutions produced by our system represent the shading on each surface by a collection of quadratic triangles. In our current implementation each of these triangles is rendered as four Gouraud shaded triangles. However, this approximation is quite coarse, and for large elements more Gouraud triangles are needed for accurate representation. Research is needed to determine whether it would be faster to scan convert quadratically shaded triangles directly, particularly if quadratic expressions can be evaluated in hardware.

Quadratic interpolation is most effective for large triangular elements. Small triangles in densely sampled regions can be adequately rendered with a single Gouraud shaded triangle. We need to determine the order of each element adaptively, while taking care to preserve continuity between adjacent elements of different order. This should speed up not only the display of the solution, but also the local pass, by reducing the number of mesh nodes at which radiosity must be computed.

6.2.4 Improving and Extending the Error Analysis

To improve the efficiency and the reliability of our *a posteriori* error bounds and estimates, we need techniques for efficiently computing guaranteed tight bounds on form-factors, perhaps using tools such as Arvo's irradiance Jacobians [ARVO94a].

Also, we would like to extend our error analysis to higher order basis functions. While we are able to bound the exact radiosity function, our bounds will not necessarily contain higher order approximations, as they may exhibit oscillations [ZATZ93]. Even if the approximation is contained within our bounds, equations (5.7) and (5.8) will probably produce pessimistic error estimates, since they assume that the approximation is constant over each element.

Prior to rendering an image, the radiosity solution is usually projected onto a new set of basis functions, typically piecewise-linear or piecewise-quadratic (as is the case in our system). We need to analyze the effect of such projections on the error in order to extend our error estimates from solutions to images.

Finally, we would like to emphasize the need for the development of perceptual error metrics for image synthesis. Fast convergence in a quantitative error metric, such as the L_1 norm, does not necessarily imply fast convergence of the resulting images, as perceived by a human observer. Thus, we expect that a perceptual error-driven refinement strategy would be more appropriate for image synthesis.

6.2.5 Parallelization

We have begun experimenting with a parallel implementation of our system. This implementation runs on a cluster of HP workstations, using the PVM message passing subroutine library [GEIS94]. Currently, only the local pass has been parallelized. A single *master* process reads in the description of the environment and computes the global pass solutions. Then, patches are distributed to *workers*, which perform the local pass. This is done using a dynamic load balancing strategy.

We chose to parallelize the local pass because it is currently the most expensive stage in our system, and because it lends itself well to parallelization: the information stored with the patch and with its links is sufficient to perform the local pass, so each worker can perform the computations independently of the others. Preliminary results indicate that this simple scheme is capable of significantly speeding up the local pass.

One problem that we have encountered in our experiments is that certain patches tend to dominate the local pass computation time; thus, good load balancing is hard to achieve. Usually, these are large patches with many elements and partially occluded links, such as floors, walls, and ceilings. One solution to this problem is to preprocess the environment by splitting large surfaces. This seems to improve the load balancing significantly, but sometimes results in the seams between the pieces of the same original surface being visible in the final solution. Therefore, we need to develop techniques for "seamless splitting".

Another direction for future research is the parallelization of the global pass. This is a much more challenging problem: it is not clear how to break down the work into pieces that could be done in parallel without excessive communication overhead.

Bibliography

[ANSE65] Anselone, P. M. "Convergence and Error Bounds for Approximate Solutions of Integral and Operator Equations," in Rall, Louis B., editor, *Error in Digital Computation*, John Wiley & Sons, New York, 1965, pages 231–252.

[ARVO94a] Arvo, James. "The Irradiance Jacobian for Partially Occluded Polyhedral Sources," Proceedings of SIGGRAPH'94 (Orlando, Florida, July 24–29, 1994), in *Computer Graphics* Proceedings, Annual Conference Series, July 1994.

[ARVO94b] Arvo, James, Kenneth Torrance, and Brian Smits. "A Framework for the Analysis of Error in Global Illumination Algorithms," Proceedings of SIGGRAPH'94 (Orlando, Florida, July 24–29, 1994), in *Computer Graphics* Proceedings, Annual Conference Series, July 1994.

[ATHE78] Atherton, Peter R., Kevin Weiler, and Donald P. Greenberg. "Polygon Shadow Generation," Proceedings of SIGGRAPH'78 (Atlanta, Georgia, August 23–25, 1978), in *Computer Graphics*, 12(3), August 1978, pages 275–281.

[BABU86] Babuška, I., O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliveira, editors. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, John Wiley & Sons, Chichester, 1986.

[BAUM89] Baum, Daniel R., Holly E. Rushmeier, and James M. Winget. "Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors," Proceedings of SIGGRAPH'89 (Boston, Massachusetts, July 31–August 4, 1989), in *Computer Graphics*, 23(3), July 1989, pages 325–334.

[BAUM91] Baum, Daniel R., Stephen Mann, Kevin P. Smith, and James M. Winget. "Making Radiosity Usable: Automatic Preprocessing and

Meshing Techniques for the Generation of Accurate Radiosity Solutions," Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 28–August 2, 1991), in *Computer Graphics*, 25(4), July 1991, pages 51–60.

[BERN92] Bern, Marshall and David Eppstein. "Mesh Generation and Optimal Triangulation," in Hwang, F. K. and D. Z. Du, editors, *Computing in Euclidian Geometry*, World Scientific, Singapore, 1992.

[BREB93] Brebbia, C. A. and M. H. Aliabadi, editors. *Adaptive Finite and Boundary Element Methods*, Computational Mechanics Publications and Elsevier Applied Science Publishers, Southampton and London, 1993.

[BROW65] Brown, R. W. "Upper and Lower Bounds for Solutions of Integral Equations," in Rall, Louis B., editor, *Error in Digital Computation*, John Wiley & Sons, New York, 1965, pages 219–230.

[CAMP90] Campbell, III, A. T. and Donald S. Fussell. "Adaptive Mesh Generation for Global Diffuse Illumination," Proceedings of SIGGRAPH'90 (Dallas, Texas, August 6–10, 1990), in *Computer Graphics*, 24(4), August 1990, pages 155–164.

[CAMP91] Campbell, III, A. T. *Modeling Global Diffuse Illumination for Image Synthesis*, PhD dissertation, University of Texas at Austin, Austin, Texas, December 1991.

[CHEN91] Chen, Shenchang Eric, Holly E. Rushmeier, Gavin Miller, and Douglass Turner. "A Progressive Multi-Pass Method for Global Illumination," Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 28–August 2, 1991), in *Computer Graphics*, 25(4), July 1991, pages 165–174.

[CHEW89] Chew, L. Paul. "Constrained Delaunay Triangulations," *Algorithmica*, 4, 1989, pages 97–108.

[CHIN89] Chin, Norman and Steven Feiner. "Near Real-Time Shadow Generation Using BSP Trees," Proceedings of SIGGRAPH'89 (Boston, Massachusetts, July 31–August 4, 1989), in *Computer Graphics*, 23(3), July 1989, pages 99–106.

[CHIN92] Chin, Norman and Steven Feiner. "Fast Object-Precision Shadow Generation for Area Light Sources Using BSP Trees," in Proceedings of 1992 Symposium on Interactive 3D Graphics (Cambridge, Massachusetts, March 29–April 1, 1992), March 1992.

[COHE85] Cohen, Michael F. and Donald P. Greenberg. "The Hemi-Cube: A Radiosity Solution for Complex Environments," Proceedings of SIGGRAPH'85 (San Francisco, California, July 22–26, 1985), in *Computer Graphics*, 19(3), July 1985, pages 31–40.

[COHE86] Cohen, Michael F., Donald P. Greenberg, David S. Immel, and Philip J. Brock. "An Efficient Radiosity Approach for Realistic Image Synthesis," *IEEE Computer Graphics and Applications*, 6(2), March 1986, pages 26–35.

[COHE88] Cohen, Michael F., Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. "A Progressive Refinement Approach to Fast Radiosity Image Generation," Proceedings of SIGGRAPH'88 (Atlanta, Georgia, August 1–5, 1988), in *Computer Graphics*, 22(4), August 1988, pages 75–84.

[COHE93] Cohen, Michael F. and John R. Wallace. *Radiosity and Realistic Image Synthesis*, Academic Press Professional, Cambridge, Massachusets, 1993.

[COOK86] Cook, Robert L. "Stochastic Sampling in Computer Graphics," *ACM Transactions on Graphics*, 5(1), January 1986, pages 51–72.

[DELV85] Delves, L. M. and J. L. Mohamed. *Computational Methods for Integral Equations*, Cambridge University Press, Cambridge, Great Britain, 1985.

[DRET93] Drettakis, George and Eugene Fiume. "Accurate and Consistent Reconstruction of Illumination Functions Using Structured Sampling," in Proceedings of Eurographics'93 (Barcelona, Spain, September 6–10, 1993, September 1993, pages 273–284.

[FUCH80] Fuchs, Henry, Zvi M. Kedem, and Bruce Naylor. "On Visible Surface Generation by a Priori Tree Structures," Proceedings of SIGGRAPH'80 (Seattle, Washington, July 14–18, 1980), in *Computer Graphics*, 14(3), June 1980, pages 175–181.

[FUCH89] Fuchs, Henry, John Poulton, John Eyles, Trey Greer, Jack Goldfeather, David Ellsworth, Steve Molnar, Greg Turk, Brice Tebbs, and Laura Israel. "Pixel-Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories," Proceedings of SIGGRAPH'89 (Boston, Massachusetts, July 31–August 4, 1989), in *Computer Graphics*, 23(3), July 1989, pages 79–88.

[FUNK93] Funkhouser, Thomas A. and Carlo H. Séquin. "Adaptive Display Algorithm for Interactive Frame Rares During Visualization of Complex Virtual Environments," Proceedings of SIGGRAPH'93 (Anaheim, California, August 1–6, 1993), in *Computer Graphics* Proceedings, Annual Conference Series, August 1993, pages 247–254.

[GEIS94] Geist, Al, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Mancheck, and Vaidy Sunderam. *PVM 3 User's Guide and Reference Manual*, Technical Report ORNL/TM-12187, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, May 1994.

[GIGU90] Gigus, Ziv and Jitendra Malik. "Computing the Aspect Graph for Line Drawings of Polyhedral Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2), February 1990, pages 113–122.

[GIGU91] Gigus, Ziv, John Canny, and Raimund Seidel. "Efficiently Computing and Representing Aspect Graphs of Polyhedral Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), June 1991, pages 542–551.

[GOLD87] Goldsmith, Jeffrey and John Salmon. "Automatic Creation of Object Hierarchies for Ray Tracing," *IEEE Computer Graphics and Applications*, 7(5), May 1987, pages 14–20.

[GOLU89] Golub, Gene H. and Charles F. Van Loan. *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, second edition, 1989.

[GORA84] Goral, Cindy M., Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. "Modeling the Interaction of Light Between Diffuse Surfaces," Proceedings of SIGGRAPH'84 (Minneapolis, Minnesota, July 23–27, 1984), in *Computer Graphics*, 18(3), July 1984, pages 213–222.

[GUIB85] Guibas, Leonidas and Jorge Stolfi. "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," *ACM Transactions on Graphics*, 4(2), April 1985, pages 74–123.

[HAIN91a] Haines, Eric A. "Ronchamp: A Case Study for Radiosity," SIGGRAPH'91 Frontiers in Rendering Course Notes, July 1991.

[HAIN91b] Haines, Eric A. and John R. Wallace. "Shaft Culling for Efficient Ray-Traced Radiosity," in Proceedings of the Second Eurographics

Workshop on Rendering (Barcelona, Spain, May 13–15, 1991), May 1991.

[HANR91] Hanrahan, Pat, David Salzman, and Larry Aupperle. "A Rapid Hierarchical Radiosity Algorithm," Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 28–August 2, 1991), in *Computer Graphics*, 25(4), July 1991, pages 197–206.

[HECK91a] Heckbert, Paul S. *Simulating Global Illumination Using Adaptive Meshing*, PhD dissertation, Department of EECS, UC Berkeley, California, June 1991.

[HECK91b] Heckbert, Paul S. and James M. Winget. *Finite Element Methods for Global Illumination*, Technical Report UCB/CSD 91/643, CS Division, University of California at Berkeley, Berkeley, California, July 1991.

[HECK92] Heckbert, Paul S. "Discontinuity Meshing for Radiosity," in Proceedings of the Third Eurographics Workshop on Rendering (Bristol, UK, May 18–20, 1992), May 1992, pages 203–216.

[IMME86] Immel, David S., Michael F. Cohen, and Donald P. Greenberg. "A Radiosity Method for Non-Diffuse Environments," Proceedings of SIGGRAPH'86 (Dallas, Texas, August 18–22, 1986), in *Computer Graphics*, 20(4), August 1986, pages 133–142.

[KAJI86] Kajiya, James T. "The Rendering Equation," Proceedings of SIGGRAPH'86 (Dallas, Texas, August 18–22, 1986), in *Computer Graphics*, 20(4), August 1986, pages 143–150.

[KOK91] Kok, Arjan J. F. and Frederik Jansen. "Source Selection for the Direct Lighting Computation in Global Illumination," in Proceedings of the Second Eurographics Workshop on Rendering (Barcelona, Spain, May 13–15, 1991), May 1991.

[LEWI65] Lewins, Jeffery. *Importance, The Adjoint Function: The Physical Basis of Variational and Perturbation Theory in Transport and Diffusion Problems*, Pergamon Press, New York, 1965.

[LISC91] Lischinski, Dani, Filippo Tampieri, and Donald P. Greenberg. *Improving Sampling and Reconstruction Techniques for Radiosity*, Technical Report 91-1202, Department of Computer Science, Cornell University, Ithaca, New York, August 1991.

[LISC92] Lischinski, Dani, Filippo Tampieri, and Donald P. Greenberg. "Discontinuity Meshing for Accurate Radiosity," *IEEE Computer Graphics and Applications*, 12(6), November 1992, pages 25–39.

[NISH83] Nishita, Tomoyuki and Eihachiro Nakamae. "Half-Tone Representation of 3-D Objects Illuminated by Area Sources or Polyhedron Sources," in Proceedings of the IEEE Computer Society's International Computer Software and Applications Conference (COMPSAC83) (Chicago, Illinois, November 1983), November 1983, pages 237–241.

[NISH85] Nishita, Tomoyuki and Eihachiro Nakamae. "Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflections," Proceedings of SIGGRAPH'85 (San Francisco, California, July 22–26, 1985), in *Computer Graphics*, 19(3), July 1985, pages 23–30.

[NISH87] Nishita, Tomoyuki, Yasuhiro Miyawaki, and Eihachiro Nakamae. "A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources," Proceedings of SIGGRAPH'87 (Anaheim, California, July 27–31, 1987), in *Computer Graphics*, 21(4), July 1987, pages 303–310.

[REIC92] Reichert, Mark C. *A Two-Pass Radiosity Method Driven by Lights and Viewer Position*, Master's thesis, Program of Computer Graphics, Cornell University, Ithaca, New York, January 1992.

[RUSH87] Rushmeier, Holly E. and Kenneth E. Torrance. "The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium," Proceedings of SIGGRAPH'87 (Anaheim, California, July 27–31, 1987), in *Computer Graphics*, 21(4), July 1987, pages 293–302.

[RUSH90] Rushmeier, Holly E. and Kenneth E. Torrance. "Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials," *ACM Transactions on Graphics*, 9(1), January 1990, pages 1–27.

[SALE92] Salesin, David, Dani Lischinski, and Tony DeRose. "Reconstructing Illumination Functions with Selected Discontinuities," in Proceedings of the Third Eurographics Workshop on Rendering (Bristol, UK, May 18–20, 1992), May 1992, pages 99–112.

[SHAO88] Shao, Min-Zhi, Qun-Sheng Peng, and You-Dong Liang. "A New Radiosity Approach by Procedural Refinements for Realistic Image

Synthesis," Proceedings of SIGGRAPH'88 (Atlanta, Georgia, August 1–5, 1988), in *Computer Graphics*, 22(4), August 1988, pages 93–102.

[SHIR90a] Shirley, Peter. *Physically Based Lighting Calculations for Computer Graphics*, PhD dissertation, Department of Computer Science, University of Illinois, Urbana-Champaign, Illinois, 1990.

[SHIR90b] Shirley, Peter. "A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes," in Proceedings of Graphics Interface'90 (Halifax, Nova Scotia, May 14–18, 1990), May 1990, pages 205–212.

[SILL89] Sillion, François and Claude Puech. "A General Two-Pass Method Integrating Specular and Diffuse Reflection," Proceedings of SIGGRAPH'89 (Boston, Massachusetts, July 31–August 4, 1989), in *Computer Graphics*, 23(3), July 1989, pages 335–344.

[SILL91] Sillion, François X, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. "A Global Illumination Solution for General Reflectance Distributions," Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 28–August 2, 1991), in *Computer Graphics*, 25(4), July 1991, pages 187–196.

[SMIT92] Smits, Brian E., James R. Arvo, and David H. Salesin. "An Importance-Driven Radiosity Algorithm," Proceedings of SIGGRAPH'92 (Chicago, Illinois, July 26–31, 1992), in *Computer Graphics*, 26(4), July 1992, pages 273–282.

[SMIT94] Smits, Brian, James Arvo, and Donald Greenberg. "A Clustering Algorithm for Radiosity in Complex Environments," Proceedings of SIGGRAPH'94 (Orlando, Florida, July 24–29, 1994), in *Computer Graphics* Proceedings, Annual Conference Series, July 1994.

[SPAR63a] Sparrow, E. M. "A New and Simpler Formulation for Radiative Angle Factors," *ASME Journal of Heat Transfer*, 85(2), May 1963, pages 81–88.

[SPAR63b] Sparrow, Ephraim M. "On the Calculation of Radiant Interchange between Surfaces," in Ibele, Warren E., editor, *Modern Developments in Heat Transfer*, Academic Press, New York, New York, 1963.

[SPAR78] Sparrow, E. M. and R. D. Cess. *Radiation Heat Transfer*, Hemisphere Publishing Corp., Washington D.C., 1978.

[SZAB91] Szabó, Barna and Ivo Babuška. *Finite Element Analysis*, John Wiley & Sons, New York, 1991.

[TAMP93] Tampieri, Filippo. *Discontinuity Meshing for Radiosity Image Synthesis*, PhD dissertation, Program of Computer Graphics, Cornell University, Ithaca, New York, May 1993.

[TARJ83] Tarjan, Robert Endre. *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1983.

[TELL91] Teller, Seth J. and Carlo H. Séquin. "Visibility Preprocessing for Interactive Walkthroughs," Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 28–August 2, 1991), in *Computer Graphics*, 25(4), July 1991, pages 61–70.

[TELL92] Teller, Seth J. "Computing the Antipenumbra of an Area Light Source," Proceedings of SIGGRAPH'92 (Chicago, Illinois, July 26–31, 1992), in *Computer Graphics*, 26(4), July 1992, pages 139–148.

[TELL93] Teller, Seth and Pat Hanrahan. "Global Visibility Algorithms for Illumination Computations," Proceedings of SIGGRAPH'93 (Anaheim, California, August 1–6, 1993), in *Computer Graphics* Proceedings, Annual Conference Series, August 1993, pages 239–246.

[WALL87] Wallace, John R., Michael F. Cohen, and Donald P. Greenberg. "A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods," Proceedings of SIGGRAPH'87 (Anaheim, California, July 27–31, 1987), in *Computer Graphics*, 21(4), July 1987, pages 311–320.

[WALL89] Wallace, John R., Kells A. Elmquist, and Eric A. Haines. "A Ray Tracing Algorithm for Progressive Radiosity," Proceedings of SIGGRAPH'89 (Boston, Massachusetts, July 31–August 4, 1989), in *Computer Graphics*, 23(3), July 1989, pages 315–324.

[WARD88] Ward, Gregory J., Francis M. Rubinstein, and Robert D. Clear. "A Ray Tracing Solution for Diffuse Interreflection," Proceedings of SIGGRAPH'88 (Atlanta, Georgia, August 1–5, 1988), in *Computer Graphics*, 22(4), August 1988, pages 85–92.

[ZATZ93] Zatz, Harold R. "Galerkin Radiosity: A Higher Order Solution Method for Global Illumination," Proceedings of SIGGRAPH'93 (Anaheim, California, August 1–6, 1993), in *Computer Graphics* Proceedings, Annual Conference Series, August 1993, pages 213–220.

[ZIEN89] Zienkiewicz, O. C. and R. L. Taylor. *The Finite Element Method*, pages 128–132, Volume 1, McGraw-Hill, London, UK, 4th edition, 1989.