GALERKIN RADIOSITY:
A HIGHER ORDER SOLUTION METHOD FOR
GLOBAL ILLUMINATION

SIGGRAPH '93 edition

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Harold R. Zatz

August 1992

Abstract

This thesis presents an alternative radiosity formulation based on piecewise smooth illumination functions that incorporates curved surfaces directly. Using the Galerkin integral equation technique as a mathematical foundation, approximate surface illumination functions are computed in terms of linear, quadratic, and higher-order descriptions across each surface. Since these illumination functions are smooth, they can be rendered directly, without interpolation. Nonplanar, parametric surfaces can also be used directly, without tiling. Convergence problems due to singularities in the radiosity kernel are analyzed and rectified. An analysis of the approximation error of this method is included. By using an approximating shadow masking technique, meshing is only required in particular extreme cases, as justified by the error analysis. The technique is demonstrated on traditional radiosity scenes, as well as environments with simple curved surfaces.

Biographical Sketch

Harold Robert Feldman Zatz was born on January 9, 1969 in Baltimore, Maryland. After a few years, his family moved to the Orlando area, where he spent the remainder of his formative years. At his first freshman physics lecture at Caltech, computer animations by Jim Blinn from *The Mechanical Universe* were shown, leading Harold to exclaim, "I want to do *that*." To that end, he spent two-and-a-half of his four years at Caltech working in the graphics lab, before receiving his B.S. with honors double majoring in Chemistry and Computer Engineering. Following a short summer in the computer animation industry at Rhythm & Hues, he came to Cornell's Program of Computer Graphics, hoping to complete his Master's degree in two years.

After graduation, Harold will be returning to Rhythm & Hues with his year-old dog, Einstein, to pursue a career in the computer animation industry.

To Al Barr, for his support and encouragement.

To Mom, Dad, Karen, and Einstein,

who were always there when I needed them.

Acknowledgements

The author is extremely grateful to the Hewlett Packard Corporation for donating the equipment used in this work. This research was partially conducted under a National Science Foundation grant entitled "Interactive Input and Display Techniques." The Program of Computer Graphics at Cornell is one of five sites of the new NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (Grant # ASC-8920219). The author was funded by a National Science Foundation Graduate Student Fellowship, and by a Cornell University Sage Fellowship. I would like to thank:

- Don Greenberg for providing insightful comments as to what was actually needed to write up this research.

- Al Barr for encouraging my initial interest, teaching me the basics of computer graphics and academic research, and providing the inspiration to excel.

- Dani Lischinski and Filippo Tampieri for providing the reference solution of Chapter 7, as well as for providing a lot of useful information about shadows and radiosity.

- Dave Salesin and Jim Arvo for helpful discussions along the way, and reading a lot of early material.
- Ken Torrance, Roy Hall, Brian Smits, David Baraff, and Dan Kartch for reviewing early drafts of the SIGGRAPH '92 paper submission, which was later expanded into this thesis.
- Hurf Sheldon, Jim Ferwerda, and Mitch Collingsworth for keeping the equipment running.
- Steve Westin, Jim Arvo, Erin Shaw, and Chris Schoeneman for putting up with me as an officemate.
- Peter Pruyn for providing a different perspective on the world.
- Len Wanger and Mark "woof" Reichert for showing that it *can* be done.
- Fran Brown and Ellen French for their eternal patience.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The behavior of light interacting with a macroscopic environment is extremely complex. Simulating this behavior using computers—the global illumination problem—has proven equally complex. Consider the scope of the problem: in a room with a single 100 Watt light bulb, there are on the order of 10^{20} photons[1] flying about the room in the brief moment taken by a CRT to refresh its screen. Today's fastest computers have nowhere near the amount of processing power needed to simulate this lighting behavior directly, and it is unlikely that computers with such capacity will be built any time in the foreseeable future. To produce computer-generated pictures in a reasonable amount of time, approximations must be used.

The radiosity approximation to the global illumination problem was first introduced to computer graphics by Goral *et al.* [GORA 84]. Goral's work was an application of the theory of radiative heat transport within an enclosure,

[1] $100\text{Watt} \cdot 20\%\text{efficiency} / (30\frac{\text{frame}}{\text{s}} \cdot E_{\text{photon}}) \approx 2 \times 10^{20}$

dating back at least to [HOTT 54]. Within the theory of heat transport, *radiosity* refers to the total radiative energy emitted from a surface per unit area per unit time [SPAR 78]. Radiosity simulations have been used for many years to simulate the interiors of ovens and combustion engines, as well as in the design of spacecraft heat radiators and other systems dominated by thermal black-body emissions. In computer graphics, only visible light energy is considered, generating radiosity values directly proportional to observed intensity.

Conventional radiosity methods simplify the global illumination problem with two fundamental assumptions. First, surfaces are assumed to have purely Lambertian, or diffuse reflectance distributions; equal numbers of photons are scattered from surfaces in all directions, regardless of the incoming energy distribution. Finite regions on these surfaces are also assumed to have locally constant radiosity values; a portion of a wall, for example, might be considered as having the same visible brightness throughout. To allow for variations in intensity across a surface, a surface might be *meshed*, or divided into a large number of smaller pieces. In the original computer graphics implementation [GORA 84], radiosities were computed for an empty box with colored walls, with each wall meshed into up to 49 parts.

Subsequent developments have enhanced the power and speed of the radiosity method. Using a progressive refinement technique [COHE 88], the effective speed of the original algorithm was vastly increased, adding the ability to observe pictures improving slowly over time. Sillion [SILL 91] showed how the Lambertian assumption could be removed from the radiosity model, allowing for general reflectance functions. Hanrahan [HANR 91] applied multigridding techniques to the problem, restructuring radiosity's mathematical model and thereby producing a significantly faster algorithm. The radiosity technique has also been applied to significantly more complex environments than the original box, as can be seen in [DORS 91]'s simulation of opera set lighting effects.

1.1 Limitations of Conventional Radiosity

Despite all of these advances, existing radiosity implementations still suffer from serious limitations.

- The mathematical techniques used to evaluate radiosity systems always require that objects be flat or approximated by a large number of small polygonal surfaces. Both approximate [GORA 84,COHE 85] and analytic [BAUM 89] approaches use planar surfaces, even though such a restriction is not a fundamental part of the radiosity model. Wallace's approach [WALL 89] allows the surface normal to vary across planar surfaces, reducing the error caused by tessellating a curved surface. However, no radiosity method to date has incorporated curved surfaces directly.

- Outside regions of total shadow, the radiosity of surfaces is not generally constant [TAMP 91]. Different points on surfaces are different distances from light sources, causing subtle variations in intensity. Partial occlusions or curvature variations across surfaces can lead to more dramatic changes in illumination. The assumption that radiosity values are locally constant is accurate over an extremely limited domain, and has significant implications for the accuracy of conventional radiosity solution methods.

- Generating images with accurately placed shadows is difficult using existing radiosity techniques. The original occlusion approach of Nishita and Nakamae [NISH 85], although restricted to convex polyhedra, used the explicit, geometrically determined shadow boundaries to guide their simple subdivision scheme. Campbell [CAMP 90] improved upon this technique by subdividing polygons along their linear shadow boundaries after each radiosity iteration. Hanrahan's technique [HANR 91] combined adaptive subdivision techniques with partial occlusion testing within its multigrid model. Lischinski and Tampieri's approach [LISC 92] adds discontinuity order information to Campbell's method, improving the ability to reconstruct the shadows into a reasonable radiosity picture. All these subdivision techniques can be enhanced by taking multiple occlusion samples across the emitter and receiver to provide intermediate levels of radiosity in partially occluded regions [WALL 89]. The difficulties of accurate shadow generation in radiosity have not yet been resolved.

Although all these methods work to varying degrees, at their highest level of accuracy they all must produce a relatively large mesh consisting of many small elements. These large meshes significantly increase the computation time required to perform a global illumination simulation, and place restrictions on the complexity of environments that can use radiosity methods. In finite element analysis, it is sometimes possible to trade off a large number of lower-order elements for a smaller number of higher-order elements. In thermal radiosity, variational methods have been used to create single element, higher-order solutions for some special cases [USIS 60,SPAR 60]. Heckbert experimented with using a large number of linear [HECK 91a] and quadratic [HECK 91b] elements to solve computer graphics radiosity problems in the plane.

1.2 Thesis Objective

In this thesis, the core of the radiosity method is rebuilt in a mathematically consistent manner that eliminates many of the difficulties present with current techniques. Beginning in Chapter 2, the inconsistencies caused by the constant radiosity assumption and conventional meshing techniques are examined, leading to a parametric reformulation of the radiosity integral equation. The ideal way to solve such an integral equation would be with analytic methods. Chapter 3 explores some of the more common, potentially applicable approaches to solving integral equations, and shows how they fail for the radiosity equation. The next section of the thesis provides mathematical background for the *Galerkin* method of approximating integral equations, and applies it to the radiosity equation. Difficulties caused by radiosity's edge singularities are analyzed, and a new mathematical approach is presented to compensate for these singularities. Chapter 6 suggests two possible approaches to dealing with shadow discontinuities in radiosity environments. The next chapter presents an error analysis of the technique, including comparisons with conventional methods and empirical error estimates. Further implementation details, such as rendering, the use of curved surfaces, and parallelization appear in Chapter 8. Finally, Chapter 9 states conclusions and suggests some potential directions for future research.

Chapter 2

Background

The radiosity model of global illumination is based on the principle of energy conservation. All emitted light energy must be received somewhere by the surrounding environment. This received light is either reflected back into the environment, or absorbed into some other energy mode. Certain regions in the environment known as light sources are emitters of energy, initiating the light transport process.

The radiosity model is often summarized in two equations, written in terms of differential areas:

$$
\begin{aligned}
B_{dA_i} dA_i &= E_{dA_i} dA_i + \rho_{dA_i} \int_j B_{dA_j} F_{dA_j - dA_i} dA_j, &(2.1)\\
F_{dA_j - dA_i} &= \mathrm{VIS}(dA_i, dA_j) \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_i. &(2.2)
\end{aligned}
$$

Only the Lambertian assumption has been used to develop these equations. Since any non-trivial environment would still have an infinite number of differential area patches dA_i, some further simplifications must be made. Conven-

Table 2.1: Conventional radiosity symbols, as used in Equations (2.1) and (2.2).

dA_i, dA_j	A differential area indexed by i or j. i is the receiving surface, and j is the emitting surface.
B_{dA_i}	Radiosity of differential area i
E_{dA_i}	Inherent emittance of differential area i
ρ_{dA_i}	Lambertian reflectivity of differential area i
$F_{dA_j-dA_i}$	Double-differential Form Factor from differential area j to differential area i
$\mathrm{VIS}(dA_i, dA_j)$	Mutual visibility of differential areas i and j
r	Distance between differential areas i and j
θ_i	Angle between the surface normal of differential area i and a line between differential areas i and j

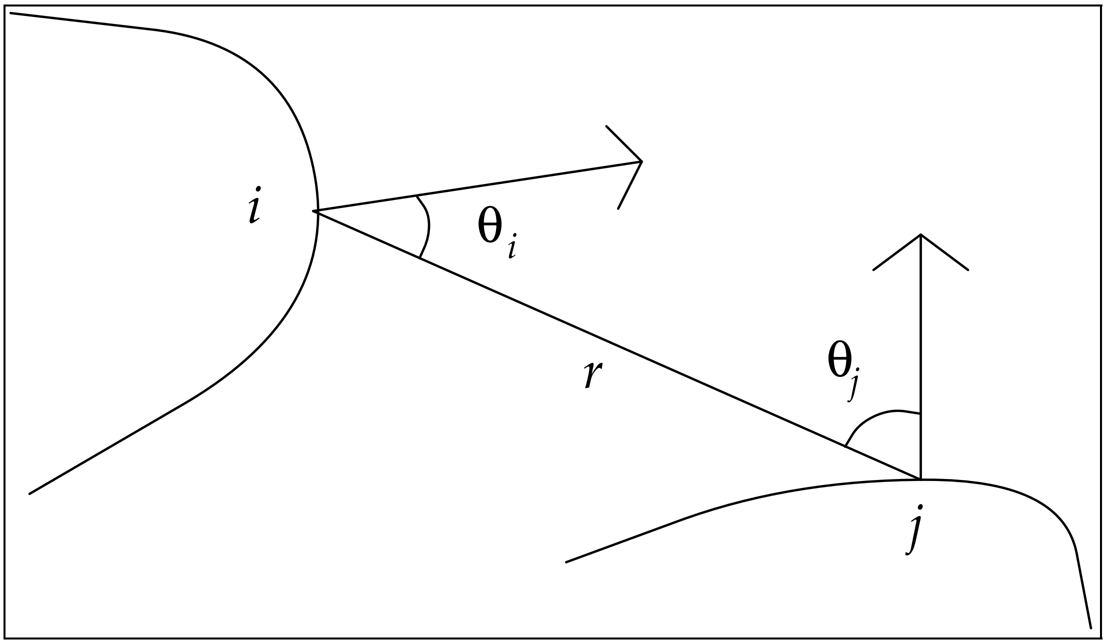


Figure 2.1: Geometry for conventional radiosity exchange between surfaces *i* and *j*.

tional radiosity techniques reduce the problem to finite dimensions by replacing differential areas with finite areas.

2.1 Conventional Radiosity

Conventional radiosity methods [BAUM 89] [CAMP 90] [COHE 85] [COHE 88] [GORA 84] [HANR 91] [NISH 85] [WALL 89] all depend upon the assumption that radiosity values are constant over finite regions. This has been referred to as the *Constant Radiosity Assumption*, or CRA [TAMP 91]. With the subsidiary assumptions that emittance and reflectivity are also constant over finite regions, the radiosity equation (2.1) can be rewritten

$$B_i = E_i + \rho_i \sum_j B_j \int F_{dA_j - dA_i} dA_j. \tag{2.3}$$

The remaining form factor integral has been approximated in several ways. Goral [GORA 84] originally followed the approach of [SPAR 78], reducing the environment to simple polygons and using an analytic expression for the form factor between two polygons. Although analytical, this approach does not take into account the mutual visibility of the two polygons. The hemicube method [COHE 85] projects the environment onto a surface, and then samples the area-to-point form factor between a particular point on the receiving surface and separate regions of the transmitting environment defined by the hemicube grid. The ray-traced form factor method [WALL 89] samples the double-differential form factor between a number of points on each transmitting and receiving surface. For polygonal surfaces, the CRA allows the use of an analytic formulation for the polygon-to-point form factor [BAUM 89]. This reduces the problem of computing an accurate form factor between two polygons to a sampling problem across one of the polygons. Because of the relative ease of calculating polygon-to-polygon form factors, any curved surfaces in an environment are generally tessellated into polygons before application of the radiosity method.

The complete set of form factors between surfaces in the scene form the coefficients of a linear system of equations whose solution is the radiosity across each surface. These equations form the standard representation of radiosity systems:

$$B_i = E_i + \rho_i \sum_j B_j F_{ji} \tag{2.4}$$

2.2 The Constant Radiosity Syndrome

Unfortunately, this constant, polygonal approach to the radiosity problem limits the accuracy of generated pictures. As explained by Tampieri and Lischinski [TAMP 91], true radiosity values are rarely constant outside regions of complete shadow. Because conventional radiosity techniques use an iterative scheme that does not compensate for CRA-induced error, their results may not converge to a physically accurate solution. Similarly, tessellating curved surfaces into polygons can only produce results as accurate as the tessellation.

Conventional radiosity methods attempt to compensate for these problems by increasing the amount of meshing; the algorithms depend upon the assumption that the world can be accurately approximated by constant radiosity polygons if enough of them are used. Unlike finite-element methods, whose meshes generally depend on solution estimates, computer graphic tessellations are usually based only on surface geometry. The same generalized meshing procedure is used, regardless of the problem it is intended to solve. Even adaptive meshing methods make no distinction between meshing techniques used to simulate any lighting effect, from shadow boundaries to variations in surface curvature. For even moderately complex scenes, standard meshing methods also easily produce far more intermediate polygons than can reasonably be solved by existing matrix methods.

The Constant Radiosity Assumption also leads to inconsistency and error between its illumination and rendering phases. During its energy transfer phase, radiosity is assumed constant across each polygon. However, conventional radiosity renderings are made by sampling each polygon at a few points and then *interpolating* brightness values between these points. A consistent CRA rendering would use constant radiosity values for both illumination and rendering.

Historically, this inconsistency has been ignored because a Gouraud shaded image is interpreted by the eye as more accurate than a consistent, flat shaded image; due to the retina's sensitivity to abrupt changes in intensity, images with such edges appear non-realistic. However, basic signal processing tells us that simple post-processing steps cannot improve the accuracy of a solution. Interpolating a solution may make an image *look* more accurate, but all such interpolation really does is mask error by blurring the image. A better scheme would model illumination using smooth functions more representative of actual behavior, and then use these functions directly in the rendering process.

2.3 The Radiosity Integral Equation

In order to apply the appropriate mathematical tools to the solution of radiosity problems, it is more convenient to express the radiosity equation (2.1) in parametric form. Parametrically, the key radiosity variables (radiosity, emittance,

Table 2.2: Symbols used for parametric radiosity in this thesis.

i, j	A particular surface indexed by i or j. i is the receiving surface, and j is the transmitting surface.
s, t	Parametric variables over surface i. Each value of s and t refers to a different location on surface i.
u, v	Parametric variables over surface j.
$B_i(s, t)$	Radiosity of surface i at parametric point (s, t).
$E_i(s, t)$	Inherent emittance of surface i at parametric point (s, t).
$\rho_i(s, t)$	Lambertian reflectivity of surface i at parametric point (s, t).
$F_{i-j}(s, t, u, v)$	Double-differential (point-to-point) form factor from parametric point (u, v) on surface j to parametric point (s, t) on surface i.
$\mathrm{VIS}_{ij}(s, t, u, v)$	Mutual visibility of parametric point (s, t) on surface i and parametric point (u, v) on surface j.
$dA_j(u, v)$	Differential area of surface j at parametric point (u, v).
$\vec{x}_i(s, t)$	Geometry of surface i at parametric point (s, t).
$\hat{n}_i(s, t)$	Unit surface normal of surface i at parametric point (s, t).

reflectivity, *etc.*) are represented as functions of two variables over each surface:

$$B_i(s, t) = E_i(s, t) + \rho_i(s, t) \sum_j \iint F_{i-j}(s, t, u, v) \mathrm{VIS}_{ij}(s, t, u, v) dA_j(u, v) B_j(u, v) du\, dv. \tag{2.5}$$

The form factor and differential area functions can be expanded in terms of the parametric functions describing both surfaces' geometry and surface normals:

$$
\begin{aligned}
F_{i-j}(s,t,u,v) &= \frac{\left(\hat{n}_i(s,t)\cdot\left(\vec{x}_j(u,v)-\vec{x}_i(s,t)\right)\right)\left(\hat{n}_j(u,v)\cdot\left(\vec{x}_i(s,t)-\vec{x}_j(u,v)\right)\right)}{\pi\|\vec{x}_i(s,t)-\vec{x}_j(u,v)\|^4} \\
dA_j(u,v) &= \left\|\frac{\partial\vec{x}_j(u,v)}{\partial u}\times\frac{\partial\vec{x}_j(u,v)}{\partial v}\right\| \qquad (2.6)
\end{aligned}
$$

By abstracting all the complexity of the double-differential form factor, reflectivity, area, and visibility terms into a single *kernel* function $K_{ij}(s,t,u,v)$, the radiosity equation can also be written

$$B_i(s,t) = E_i(s,t) + \sum_j \int\!\!\int K_{ij}(s,t,u,v)B_j(u,v)du\,dv. \qquad (2.7)$$

This is an example of an *integral equation*; the unknown function $B_i(s,t)$ appears both outside and inside the integral. Integral equations are analogous to differential equations, with integration replacing differentiation, and are usually only a little more difficult to solve [KANW 71]. The next chapter explores some of the difficulties of solving the radiosity integral equation with standard methods.

Chapter 3

Difficulties with Analytic Methods

The ideal solution to any mathematical problem is one in closed form. Outside of a few special cases, no such solution has yet been found for non-constant radiosity systems. This does not prove whether or not such a solution could be found, just that one has not yet been formulated. However, an examination of some of the simpler non-constant radiosity systems illustrates the difficulties involved in finding an analytic solution. Much of the general information on integral equations in this chapter comes from either [KANW 71] or [DELV 85].

3.1 Characterization of the Radiosity Equation

Integral equations can be classified into different types depending on their form. When expressed as in (2.7),

$$B_i(s,t) = E_i(s,t) + \sum_j \iint K_{ij}(s,t,u,v) B_j(u,v)\, du\, dv, \tag{3.1}$$

the radiosity equation is classified as a *linear system of two-dimensional singular linear Fredholm equations of the second kind*. Each of the mathematical terms in this description refers to a particular feature of the radiosity equation, as described below:

Linear System of equations

The radiosity equation is expressed not as a single equation, but a set of interrelated equations over different surfaces. Since the unknowns are connected by summation—the $\sum_j$—their relation to each other is linear. The equations could conceivably be separated by finding the eigenvalues and eigenvectors of the full system of equations, but since such operations are significantly more difficult than simply solving a system of equations, it's not worth the effort.

Two-dimensional equations

Radiosity is defined in terms of surfaces, and surfaces are two-dimensional objects. Therefore, the unknown radiosities $B_i(s,t)$ and the pre-defined emittances $E_i(s,t)$ have two parametric dimensions, the integral equation has a double integral (one for each dimension), and the kernel $K_{ij}(s,t,u,v)$ has four parametric dimensions. Heckbert performed a detailed study of radiosity [HECK 91a] in flatland—the one-dimensional analogy to conventional radiosity. Most of the existing one-dimensional integral equation theory can be easily extended to two-dimensional radiosity equations.

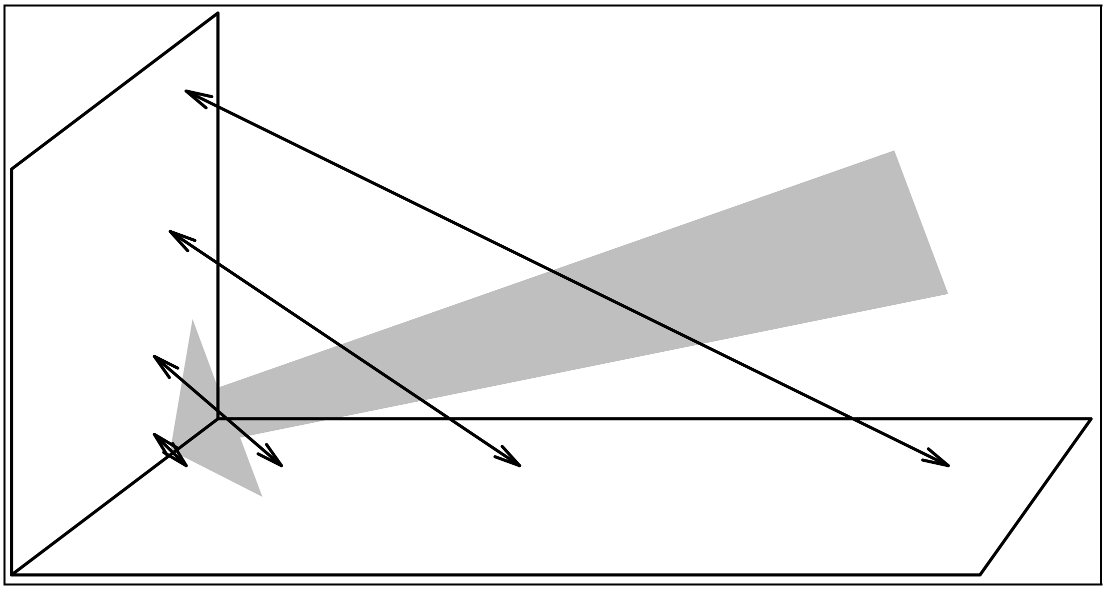


Figure 3.1: Near the corner of two surfaces, the distance between them (r) approaches zero, and the point-to-point form factor approaches infinity.

Singular equations

The most significant mathematical feature of the radiosity equation is its singularity. As can be seen in Figure 3.1, near the common edge of two adjacent surfaces, the distance r between points on these surfaces approaches zero. However, the point-to-point form factor (2.2) has a factor of r^2 in its denominator, implying that this form factor goes to infinity at the corner. Although the integration process prevents any such singularities from appearing in the solution, the presence of these singularities in the kernel requires special attention when a more analytic approach is taken to the radiosity equation.

Heckbert's planar collocation technique [HECK 91a] produces its greatest errors near so-called 'reflex corners'. The error appears because the method does not explicitly deal with the singularity at these points of intersection between two elements.

Linear Fredholm equations

Inside the linear system of equations, each individual integral equation is itself linear; the integrand contains nothing more complex than the kernel multiplied by the unknown function. Non-linear integral equations also exist, where the integrand includes powers or other more complex operations to be applied to the unknown function. However, non-linear integral equations are much more difficult to solve than linear ones.

Because of the dual linearity of the radiosity system—in both individual equations and in their combination—the principle of superposition operates. Any linear combination of two complete solutions of the radiosity equation is also a solution of the radiosity equation. In more intuitive terms, the radiosity in a room with two lights is simply the sum of the radiosity that would be visible from each light separately.

Fredholm equations

Equations with definite limits of integration are called Fredholm integral equations. Since the geometry of radiosity environments is generally fully known before the system is solved, the limits of integration are fixed.

Fredholm equations of the **second kind**

> In integral equations of the second kind, the unknown function appears both inside and outside of the integrand.

Since the radiosity equation has now been characterized in mathematical terms, it becomes possible to check the mathematical literature for references on the subject. Unfortunately, there is no general procedure for finding closed-form solutions to Fredholm equations of the second kind. Just as with differential equations, there are a few special cases where closed-form solutions are known, but most integral equations have no such solution.

However, there is a general theory about the existence and uniqueness of solutions to one-dimensional linear Fredholm equations. Using operator arithmetic, the radiosity equation can be expressed as

$$B = E + KB. \tag{3.2}$$

This can be factored and solved, as follows:

$$\begin{aligned} (I - K)B &= E \\ B &= (I - K)^{-1}E. \end{aligned} \tag{3.3}$$

As long as $||K|| < 1$, the equation can be solved by expanding it into an infinite "Neumann" series which converges to the correct unique solution [DELV 85].

$$\begin{aligned} B &= \left(I + K + K^2 + \ldots\right) E \\ &= \left(I + \sum_{i=1}^{\infty} K^i\right) E. \end{aligned} \tag{3.4}$$

Although no formal analysis of the magnitude of $\|K\|$ is presented here, for the physical system described by the radiosity method it seems intuitively reasonable that $\|K\| < 1$. Applying the kernel operator to a vector of radiosities is equivalent to propagating the light around the environment once; all light energy is emitted, travels to other surfaces, and then is either absorbed or reflected. Since energy is conserved, such a transfer can at most have $\|K\| = 1$. If the environment's reflectivities are not identically one, then some light energy must be absorbed with each step, yielding $\|K\| < 1$.

Kajiya's analysis of the radiosity method in terms of the rendering equation [KAJI 86] notes the similarity between conventional full-matrix radiosity methods and a Neumann series approximation to the rendering equation. Each iteration of the matrix solution technique adds another term to the Neumann series expansion (3.4). However, such expansions in conventional radiosity are in terms of constant patch radiosities, not smooth radiosity functions. As section 3.2 will show, there are only a few known cases where the kernel can be applied analytically to radiosity functions once, let alone the multiple times needed for a reasonably converged solution.

If the radiosity kernel were *degenerate*—separable into products of functions of (u, v) and (s, t)—it could be solved by expanding it in terms of these functions. However, the kernel is primarily a function of r, the difference between functions of (u, v) and (s, t). Approximating the radiosity kernel accurately as a degenerate kernel would probably require a large number of terms, and therefore be a relatively ineffective solution method.

The author has also attempted to solve simple variants of the radiosity equation using the Laplace transform and other symbolic methods, as well as MATHEMATICA, all to no avail. Heckbert [HECK 91a] experimented with a planar analog to radiosity and encountered similar resistance. No general solution method is yet known, although some radiosity problems can be solved for simple cases.

3.2 A Simple Radiosity System

The simple case presented in this section illustrates the complexity of the radiosity problem. Consider the environment in Figure 3.2. The top square is non-reflective, but has a constant emittance of one, while the bottom square is gray, and has no emittance. This system is the physical analog of the first progressive shot [COHE 88] from the top square to the bottom. The visibility and area terms from (2.5) are both unity everywhere, so the equations describing this system are:

$$\begin{aligned} B_0(s,t) &= 0+ \; .5\int_0^1\int_0^1 F_{0-1}(s,t,u,v)B_1(u,v)du\,dv. \\ B_1(s,t) &= 1+ \; 0\int_0^1\int_0^1 F_{1-0}(s,t,u,v)B_0(u,v)du\,dv, \end{aligned} \tag{3.5}$$

which can be simplified to

$$B_0(s,t) = .5\int_0^1\int_0^1 F_{0-1}(s,t,u,v)du\,dv. \tag{3.6}$$

$$B_1(s,t) = 1. \tag{3.7}$$

Sparrow and Cess [SPAR 78] suggest that Stoke's theorem can be applied to

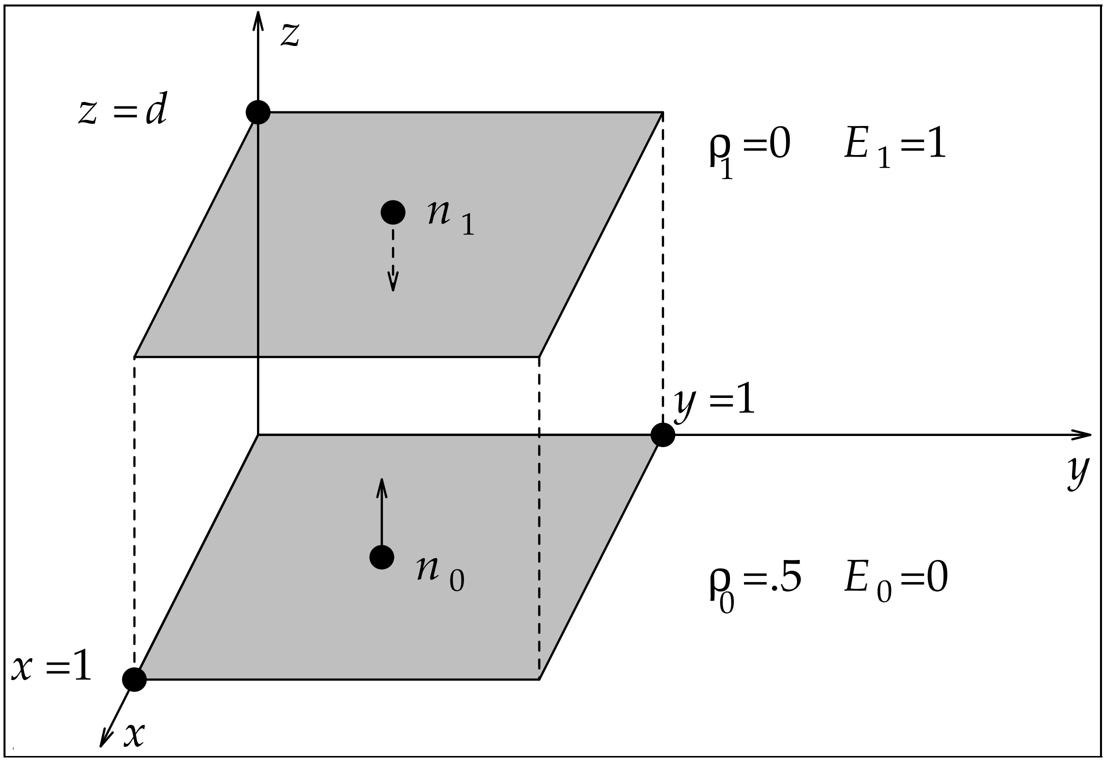


Figure 3.2: A simple radiosity environment. Two unit square patches are separated by a distance d.

integrals such as (3.6), transforming the surface integral into a contour integral. A derivation of this procedure can be found in [GORA 85], and is not repeated here. When applied to the simple case presented here, the radiosity function $B_0(s,t)$ is:

$$B_0(s,t) = \frac{1}{4\pi} \sum_{\text{surface 1's edges}} \left[\begin{array}{l} \left[\frac{v-t}{\sqrt{(t-v)^2+d^2}} \tan^{-1} \frac{u-s}{\sqrt{(t-v)^2+d^2}} \right]_u^{\phi} + \\ \left[\frac{s-u}{\sqrt{(s-u)^2+d^2}} \tan^{-1} \frac{v-t}{\sqrt{(s-u)^2+d^2}} \right]_v^{\phi} \end{array} \right] \tag{3.8}$$

$$= \frac{1}{4\pi} \left[\begin{array}{c} \dfrac{1-t}{\sqrt{(t-1)^2+d^2}} \left[\tan^{-1} \dfrac{1-s}{\sqrt{(t-1)^2+d^2}} - \tan^{-1} \dfrac{-s}{\sqrt{(t-1)^2+d^2}} \right] - \\ \dfrac{t}{\sqrt{t^2+d^2}} \left[\tan^{-1} \dfrac{-s}{\sqrt{t^2+d^2}} - \tan^{-1} \dfrac{1-s}{\sqrt{t^2+d^2}} \right] + \\ \dfrac{s-1}{\sqrt{(s-1)^2+d^2}} \left[\tan^{-1} \dfrac{-t}{\sqrt{(s-1)^2+d^2}} - \tan^{-1} \dfrac{1-t}{\sqrt{(s-1)^2+d^2}} \right] + \\ \dfrac{s}{\sqrt{s^2+d^2}} \left[\tan^{-1} \dfrac{1-t}{\sqrt{s^2+d^2}} - \tan^{-1} \dfrac{-t}{\sqrt{s^2+d^2}} \right] \end{array} \right] \tag{3.9}$$

Although this method does produce an analytic solution for the environment, this approach only works for extremely simple environments. In fact, the environment here is simple enough that direct integration, without recourse to contour integrals, also works (see Appendix A). However, attempts to extend the procedure to include even linear variations in the radiosity function have proven ineffective. If any other surface in the environment were available to reflect light emitted after this first shot, it would no longer be possible to use an analytic formulation; no technique is known for evaluating subsequent integrals analytically. Note, however, that while the original emitter B_1 must be planar for this sort of approach to work, the receiving surface B_0 can have any convex shape (if it were concave, self-reflection would preclude the use of analytic methods).

Another possible approach to solving radiosity systems is to use symbolic methods. By maintaining integrations as series of symbolic integrals, it would be possible to produce a closed-form radiosity approximation with a specific number of shots. Unfortunately, since the integrals can't be evaluated, such an approach would generate strings of mathematical symbols that would grow

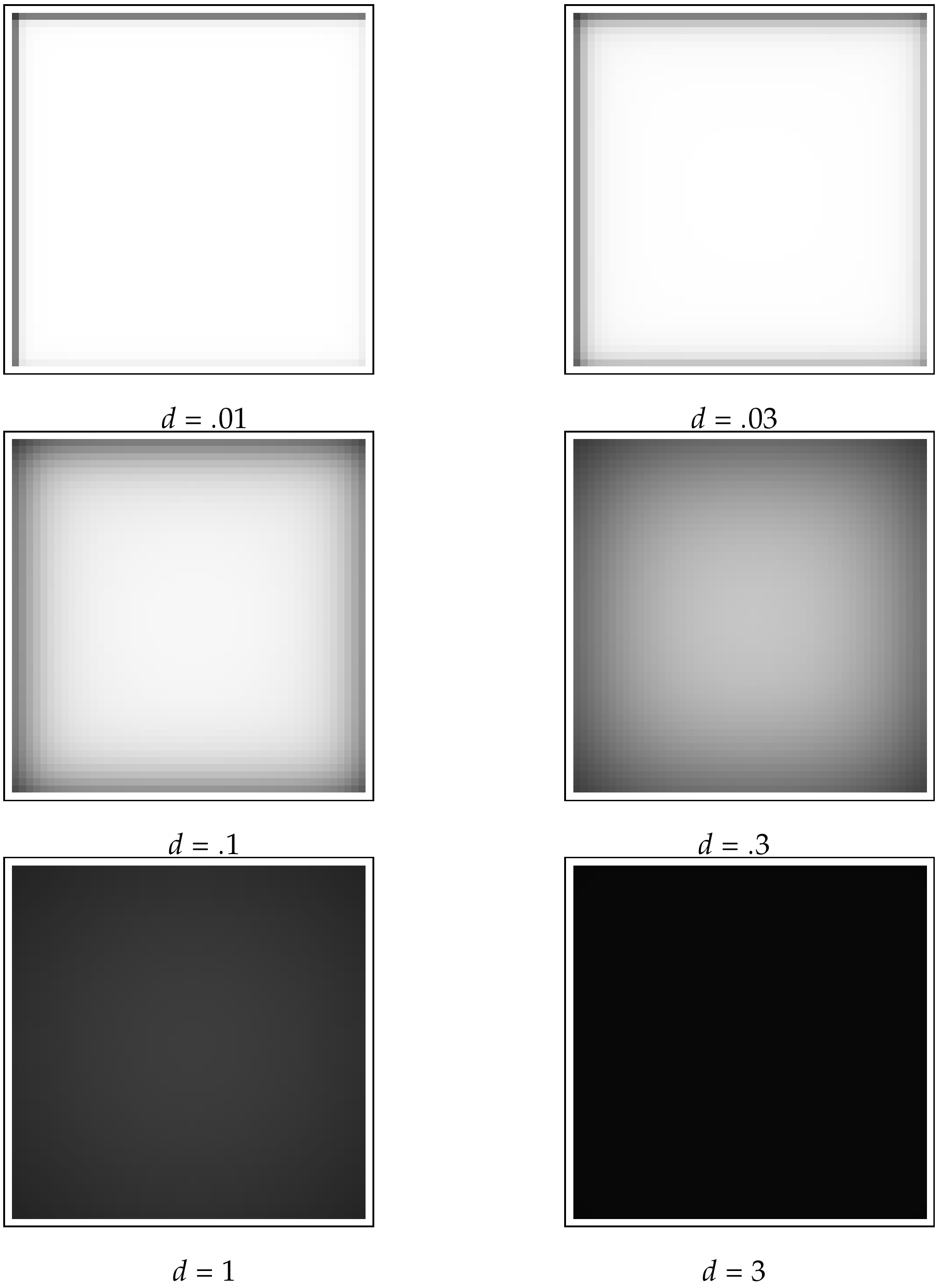


Figure 3.3: An analytic radiosity solution can be computed for the bottom square in Figure 3.2. The solutions shown here are for environments with different distances between emitting and receiving squares.

exponentially in size with each iteration. Furthermore, these integrals would still need to be numerically approximated to generate a useful solution.

In order to solve general radiosity systems, it will be necessary to use a non-closed form approximation. Integral equations are usually approached with two types of approximation techniques: collocation methods and Galerkin methods. Conventional radiosity uses a collocation technique; radiosity values are computed at a number of sample points, based upon assumptions about radiosity values elsewhere in the scene. The usual assumption is that radiosity values within some finite region are constant, allowing the use of an analytic form factor [BAUM 89] or other form factor approximation technique.

Galerkin techniques have not yet been applied to the radiosity equation, an absence noted by Heckbert [HECK 91a]. Galerkin methods have probably not been used before for two reasons—their use requires a higher degree of mathematical abstraction than collocation methods, and a simple application of the Galerkin technique fails because of radiosity's singularity, The next chapter provides the mathematical background needed for the method, and subsequent chapters outline its application to radiosity systems.

Chapter 4

Mathematical Background

Since the radiosity equation has resisted attempts at an analytical solution for all but trivial instances, traditional radiosity has focused on sample point-based methods. Instead of attempting to fully solve the equation, sample point methods produce approximate solutions by making assumptions about the equation's behavior at non-sample points. The Galerkin method provides an alternative method for solving integral equations in terms of a set of functions across each surface, instead of at a particular set of sample points. This chapter provides the mathematical background necessary to apply the function-based Galerkin method to the radiosity equation.

4.1 Linear Algebra

To approximate the radiosity distribution by a combination of functions, we need to develop a formalism for defining and manipulating appropriate functions. Since radiosity is a function defined over two-dimensional surfaces, its basis set should consist of two-dimensional functions. In this thesis, the basis set will be denoted by $\{\mathcal{T}_k(s,t)|k = 0, 1, \ldots\}$, where s and t are the parametric variables across a surface, and k specifies a particular function in the set. Basis set functions with different values of k refer to different combinations of s and t; $k = 0$ could be a constant function, $k = 2$ linear in s, $k = 6$ cubic in t, and so on.

Much of the linear algebra needed to use Galerkin methods can be understood by analogy to three-dimensional vectors. Just as geometric vectors have a dot product that projects one onto the other, functions have an *inner product* which projects one function onto another. We define the inner product of two functions $f(s,t)$ and $g(s,t)$ as

$$\langle f | g \rangle_{\mathcal{W}} = \int_{-1}^{1} \int_{-1}^{1} f(s,t) g(s,t) \mathcal{W}(s,t) ds\, dt. \tag{4.1}$$

$\mathcal{W}(s,t)$ is some weighting function that defines how important different spatial points are to the inner product. Each $\mathcal{W}(s,t)$ defines a distinct inner product and basis set. Only a few such weight functions are referred to in this thesis.

In a three-dimensional vector space, the unit vectors $\{\hat{\imath}, \hat{\jmath}, \hat{k}\}$ form an *orthonormal* basis set; they have unit length and are independent of each other. Because of their orthonormality, the unit vectors can easily be used to break up a vector into its parts. Given some vector $\vec{x}$, we can define coefficients x_k such that

$$x_0 = \vec{x} \cdot \hat{\imath}, \quad x_1 = \vec{x} \cdot \hat{\jmath}, \quad x_2 = \vec{x} \cdot \hat{k}. \tag{4.2}$$

These x_k values can be used to recreate the original vector $\vec{x}$ by the summation,

$$\vec{x} = x_0 \hat{\imath} + x_1 \hat{\jmath} + x_2 \hat{k}. \tag{4.3}$$

The analogous ability to break up and reassemble functions is needed to apply the Galerkin method to radiosity; we want to work with an orthonormal set of basis functions. Two basis functions $\mathcal{T}_k(s,t)$ and $\mathcal{T}_l(s,t)$ are said to be *orthonormal* with respect to the weight function $\mathcal{W}(s,t)$ if,

$$\forall_{k,l} \quad \langle \mathcal{T}_k | \mathcal{T}_l \rangle_{\mathcal{W}} = \begin{cases} 1, & \text{if} \quad k = l, \\ 0, & \text{if} \quad k \neq l \end{cases} \tag{4.4}$$

Finding the combination of orthonormal basis functions closest to some particular function is relatively simple. Given that the radiosity function over surface i is $B_i(s,t)$, we define the coefficients B_i^k such that

$$B_i^k = \langle B_i | \mathcal{T}_k \rangle_{\mathcal{W}} . \tag{4.5}$$

The original function's best approximation within the basis set $\{\mathcal{T}_k(s,t)\}$ is the weighted sum,

$$B_i(s,t) \approx \sum_k B_i^k \mathcal{T}_k(s,t). \tag{4.6}$$

Examples of orthonormal basis sets include spherical harmonics and certain classes of polynomials like Legendre and Jacobi polynomials.

4.2 Legendre and Jacobi Polynomials

The Galerkin method is usually solved using an orthonormal polynomial basis set. Orthonormal polynomials are special polynomials designed so that they are orthonormal with respect to a particular inner product. The integration limits of such polynomials are commonly -1 and 1. Legendre and Jacobi polynomials are one-dimensional, but they can be combined into a two-dimensional basis set by taking products of two polynomials in different variables.

For the inner product with weight function identically 1, i.e.

$$\langle f | g \rangle_{\text{Legendre}} = \int_{-1}^{1} f(x) g(x) dx, \tag{4.7}$$

the polynomials formed are the *Legendre* polynomials. The unnormalized Legendre polynomials can be generated by a recursion rule [GRAD 65],

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ (n+1)P_{n+1}(x) &= (2n+1)xP_n(x) - nP_{n-1}(x). \end{aligned} \tag{4.8}$$

The normalized Legendre polynomials are

$$\bar{P}_n(x) = \sqrt{n + \frac{1}{2}} P_n(x) \tag{4.9}$$

Polynomial sets can also be created with non-constant inner product weight functions $\mathcal{W}(x)$. Later in this thesis (section 5.3), a set of polynomials will be needed with a weight function that has a zero of multiplicity two at either $x = -1$ or $x = 1$. The *Jacobi* polynomials have such behavior. They are defined with the inner product:

$$\langle f | g \rangle_{\text{Jacobi}} = \int_{-1}^{1} f(x) g(x) (1-x)^{\alpha} (1+x)^{\beta}, \tag{4.10}$$

with weight function,

$$\mathcal{W}(x) = (1-x)^{\alpha}(1+x)^{\beta}. \tag{4.11}$$

For example, if $\alpha = 2$, the Jacobi polynomial weight function would have a zero of multiplicity two at $x = 1$. The Jacobi polynomials are conventionally written as $P_i^{(\alpha,\beta)}$.

The unnormalized Jacobi polynomials have a more complex recursion rule than the Legendre polynomials [DAVI 63]:

$$\begin{aligned} P_0^{(\alpha,\beta)}(x) &= 1 \\ P_1^{(\alpha,\beta)}(x) &= \frac{\alpha-\beta}{2} + \frac{2+\alpha+\beta}{2}x \\ P_{n+1}^{(\alpha,\beta)}(x) &= \frac{A_n^{(\alpha,\beta)}x - B_n^{(\alpha,\beta)}}{C_n^{(\alpha,\beta)}} \end{aligned} \tag{4.12}$$

where

$$\begin{aligned} A_n^{(\alpha,\beta)} &= (2n+\alpha+\beta+1)(\alpha^2-\beta^2+2n+\alpha+\beta+2) \\ &\quad \times(2n+\alpha+\beta)P_n^{(\alpha,\beta)}(x) \\ B_n^{(\alpha,\beta)} &= 2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)P_{n-1}^{(\alpha,\beta)}(x) \\ C_n^{(\alpha,\beta)} &= 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta) \end{aligned} \tag{4.13}$$

These polynomials can be normalized by multiplying them by [GRAD 65]:

$$\sqrt{\frac{\Gamma(n+1)\Gamma(\alpha+\beta+1+n)(\alpha+\beta+1+2n)}{\Gamma(\alpha+1+n)\Gamma(\beta+1+n)2^{\alpha+\beta+1}}} \tag{4.14}$$

Note that the Jacobi polynomial $P_i^{(0,0)}(x)$ is equivalent to the Legendre polynomial $P_i(x)$. Further, the two-dimensional polynomial bases used in this thesis are simple products of polynomials in different variables.

4.3 Quadrature Rules

A relatively readable explanation of one-dimensional quadrature rules can be found in Delves and Mohamed's book [DELV 85]. A condensed version is presented here.

A quadrature rule is a method for approximating the integral of the product of a class of functions $f(x)$ and a particular weight function $\mathcal{W}(x)$. Quadrature rules can be used to approximate inner product integrals, like that in (4.1). The simplest quadrature rules are weighted averages of samples of the function at particular points. Given two functions $f(x)$ and $\mathcal{W}(x)$, such a quadrature rule is a list of points ξ_i and weights w_i such that:

$$\int_a^b f(x)\mathcal{W}(x)dx \approx \sum_i w_i f(\xi_i) \tag{4.15}$$

The trapezoid rule

$$\int_0^h f(x)dx \approx \frac{h}{2}\left[f(0) + f(h)\right], \tag{4.16}$$

and Simpson's rule

$$\int_a^{a+2h} f(x)dx \approx \frac{h}{3}\left[f(a) + 4f(a+h) + f(a+2h)\right] \tag{4.17}$$

are two example quadrature rules from introductory calculus.

Quadrature rules can be designed to be exact for a certain class of functions. The trapezoid rule is exact for all linear polynomials $f(x) = Ax + B$, and Simpson's rule is exact for all quadratic polynomials $f(x) = Ax^2 + Bx + C$. These quadrature rules, with N evenly spaced sample points and variable weights w_i that are exact for polynomials up to order $N - 1$, are examples of *closed Newton-Cotes* rules.

However, it is possible to get more accuracy with the same number of sample points. The *Gaussian* quadrature rules, by computing optimal positions for the sample points ξ_i, achieve accuracy for polynomials up to order $2N-1$ for N sample points. The mathematics for deriving these quadrature rules is rather lengthy, but the resulting algorithm isn't too complex. The Gauss quadrature rule with weight function $\mathcal{W}(x)$ is closely tied to the set of orthogonal polynomials with the same orthogonalizing weight function.

To develop an N-point Gauss quadrature rule for the integral

$$\int_{-1}^{1} \mathcal{W}(x) f(x)\, dx \approx \sum_{i=1}^{N} w_i f(\xi_i), \tag{4.18}$$

start with a set of orthogonal polynomials $\mathcal{T}_i(x)$ with the same orthogonalizing weight function. This sequence of polynomials (to within a normalizing constant factor) can be expressed in terms of recursion rules [STOE 80]:

$$\begin{aligned}
\mathcal{T}_{-1}(x) &\equiv 0, \\
\mathcal{T}_{0}(x) &\equiv 1, \\
\mathcal{T}_{i+1}(x) &\equiv (x-\delta_{i+1})\mathcal{T}_i(x) - \gamma_{i+1}^2 \mathcal{T}_{i-1}(x).
\end{aligned} \tag{4.19}$$

Take these δ_i and γ_i coefficients, and construct a tridiagonal symmetric matrix:

$$\begin{bmatrix}
\delta_1 & \gamma_2 & & & 0 \\
\gamma_2 & \delta_2 & \gamma_3 & & \\
 & \gamma_3 & \cdot & \ddots & \\
 & & \ddots & \ddots & \gamma_N \\
0 & & & \gamma_N & \delta_N
\end{bmatrix} \tag{4.20}$$

The eigenvalues of this matrix, which are also the roots of the polynomial $\mathcal{T}_N(x)$, are the quadrature rule's positions ξ_i. The square of the first coefficient of the i^{th} eigenvector is the quadrature weight w_i. Finding eigenvectors and eigenvalues for tridiagonal symmetric matrices using the QR factorization is a fairly well-understood process, explained in [PRES 88], for example.

To create the Gauss-Legendre rule of order N, exact for integrating polynomials up to degree $2N-1$, the γ_i and δ_i coefficients are:

$$\delta_{i+1} = 0, \tag{4.21}$$

$$\gamma_{i+1} = \sqrt{\frac{i^2}{(2i+1)(2i-1)}}. \tag{4.22}$$

For general Jacobi polynomials $P_i^{(\alpha,\beta)}$, the necessary coefficients are:

$$\delta_{i+1} = \frac{(\alpha+\beta)(\beta-\alpha)}{(2i+\alpha+\beta+2)(2i+\alpha+\beta)}, \tag{4.23}$$

$$\gamma_{i+1} = \sqrt{\frac{4(i+\alpha)(i+\beta)i(\alpha+\beta+i)}{(\alpha+\beta+2i)^2(\alpha+\beta+2i+1)(\alpha+\beta+2i-1)}}. \tag{4.24}$$

When using these quadrature rules to project a function into a basis set using (4.5), it is important to use a quadrature rule with enough accuracy. If a one-dimensional polynomial basis set includes terms up to order n, the projection integral (4.5) must be accurate up to order $2n$—since the function is being represented as a polynomial of order n, the projection integrand will be a polynomial of order $2n$. Therefore, a one-dimensional Gaussian quadrature rule will have to have at least $N+1$ sample points to accurately perform the integration [DELV 85].

One-dimensional quadrature rules can be combined to create two and higher dimensional rules by applying the one-dimensional rules independently in each direction. For a discussion of this approach, as well as other higher-dimensional quadrature rules, see [STRO 71].

Chapter 5

Non-Constant Radiosity

Consider the effect of meshing a single surface into constant radiosity patches (Figure 5.1). Although the radiosity on each individual patch is smooth, when combined, the patches describe a stair-step radiosity function for the entire surface. Adjacent patches with different radiosity values cannot combine to form a smooth radiosity function for the entire surface. To solve radiosity without the constant radiosity assumption [TAMP 91], we need to formulate radiosity in terms of smooth functions across an entire surface, instead of disjoint patches on parts of a surface.

Figure 5.2 shows a projection of what a radiosity solution in terms of smooth functions might look like. In addition to the conventional constant part, one might use linear, quadratic, and higher-order parts. Combining these smooth functions in proper proportions might produce a better approximation to a real radiosity distribution than the same number of conventional stair-step functions. If the radiosity of every surface were represented by a combination of

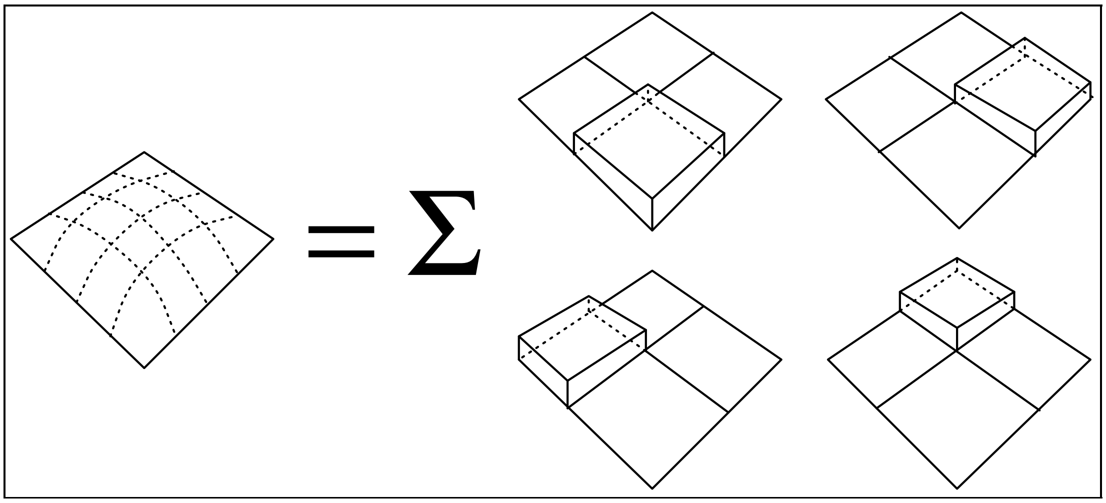


Figure 5.1: Conventional radiosity methods approximate a surface's radiosity by meshing it into a large number of constant intensity patches. Radiosity is represented by height above the surface.

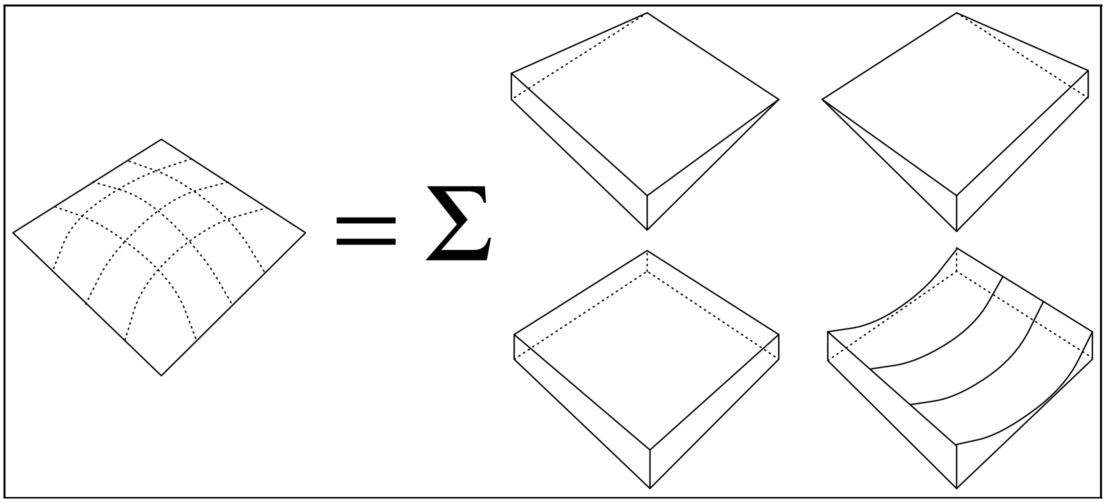


Figure 5.2: Higher-order radiosity approximates a surface's radiosity by dividing it into several different smooth functions. These smooth functions are scaled and combined to approximate the original radiosity distribution.

these functions, the radiosity equation would be solved by determining in what proportions these functions should be combined.

To properly compute these proportions, we will convert the radiosity equation (2.7) into a new matrix equation in terms of functions. This equation will replace the radiosity over patches with a linear combination of basis functions $\{\mathcal{T}_l(s,t)\}$. Instead of radiosity values, we define radiosity coefficients $\{B^l\}$, the weights used to define the relative contribution of each function $\mathcal{T}_l(s,t)$. The full radiosity distribution on a surface becomes the function

$$B_{\text{total}}(s,t) = \sum_l B^l \mathcal{T}_l(s,t). \tag{5.1}$$

Functions on different surfaces must interact in a manner analogous to the way conventional patches interact through form-factors. Just as conventional radiosity represents form factors by a form factor matrix, here the kernel function $K_{ij}(s,t,u,v)$ (see (2.7)) is represented by the *kernel matrix* K_{lm} which details how energy is transferred between functions on different surfaces. When two constant functions on different surfaces interact, the kernel matrix element describing that interaction is equivalent to a classical form factor. Other parts of the kernel matrix describe higher-order interactions, such as how a constant distribution on one surface leads to a linear variation on another surface. Just as the conventional form factor matrix relates constant radiosities on different elements, the kernel matrix relates radiosity functions across different surfaces (Figure 5.3).

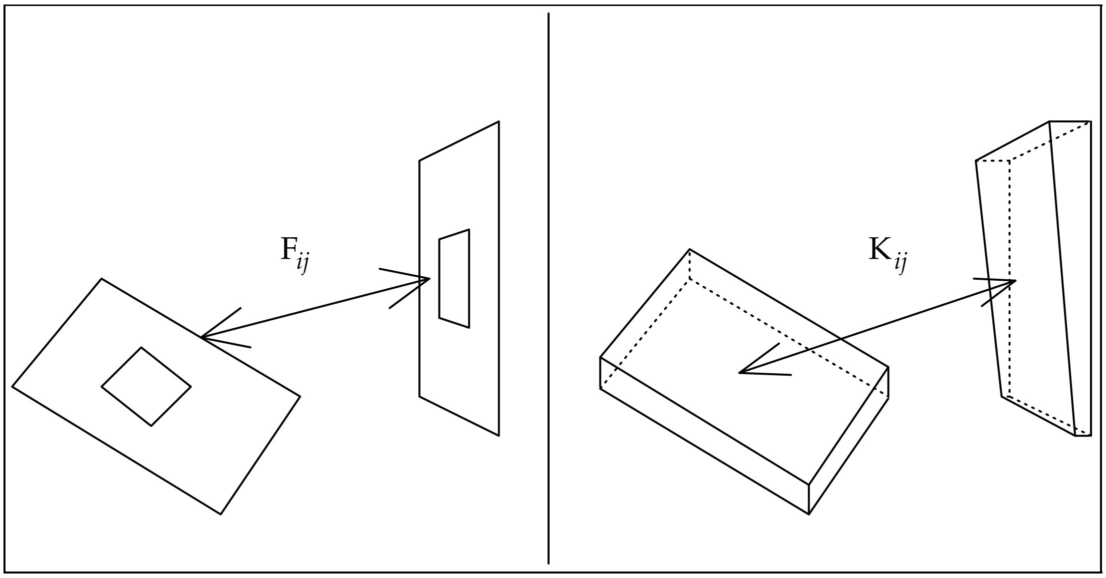


Figure 5.3: In traditional radiosity, the form factor matrix F_{ij} expresses the energy transfer between each pair of patches. In a function-based radiosity formulation, the kernel matrix K_{ij} expresses the energy transfer between basis functions on different surfaces.

5.1 The Galerkin Method

The Galerkin method approximates integral equations in terms of a set of basis functions [DELV 85]. Given an orthonormal basis set of easily integrable functions, the Galerkin technique finds the best possible fit to the integral equation's solution. Heckbert [HECK 91a,HECK 91b] suggested that the Galerkin method could be used to solve the radiosity integral equation in a plane. This and subsequent chapters explain how it can be applied to three-dimensional radiosity.

Starting with the short form of the radiosity equation from section 2.3[1],

$$B_i(s,t) = E_i(s,t) + \sum_j \iint K_{ij}(s,t,u,v) B_j(u,v)\,du\,dv. \tag{5.2}$$

Expand the $B_j(u,v)$ term inside the integral in terms of the basis set $\{\mathcal{T}_l(u,v)\}$ using (4.6). The B_j^l coefficient can be moved outside of the integral, and the summations over j and l can be combined to produce the equation

$$B_i(s,t) = E_i(s,t) + \sum_{j,l} B_j^l \iint K_{ij}(s,t,u,v)\mathcal{T}_l(u,v)\,du\,dv. \tag{5.3}$$

Now, take the inner product of both sides with some basis set function $\mathcal{T}_k(s,t)$. Using bilinearity and the relation described in (4.5),

$$B_i^k = E_i^k + \sum_{j,l} B_j^l \left\langle \iint K_{ij}(s,t,u,v)\mathcal{T}_l(u,v)\,du\,dv \middle| \mathcal{T}_k(s,t) \right\rangle_{\mathcal{W}}. \tag{5.4}$$

The inner product now depends only on known information; the kernel function K_{ij} is a function of the environment, and $\{\mathcal{T}_l(u,v)\}$ is a precomputed

[1]Remember, the kernel function K is only a notational shorthand:

$$K_{ij}(s,t,u,v) = \rho_i(s,t)F_{i-j}(s,t,u,v)\mathrm{VIS}_{ij}(s,t,u,v)dA_j(u,v)$$

basis set. The result of that inner product is denoted K_{ij}^{kl}, the *kernel matrix*. Evaluating this inner product is the most difficult part of a radiosity solution, since it effectively includes four integrations—two explicit, and two in the inner product. However, once it has been computed for each value of i, j, k, and l, the radiosity equation can be written as a matrix equation,

$$B_i^k - E_i^k = \sum_{j,l} B_j^l K_{ij}^{kl}. \tag{5.5}$$

Using M different functions $\mathcal{T}_k(s,t)$ to approximate the radiosity distribution, we can define new indices $\alpha = iM + k, \beta = jM + m$. Equation (5.5) can then be rewritten in standard matrix equation form,

$$B_\alpha - E_\alpha = \sum_\beta B_\beta K_{\alpha\beta}. \tag{5.6}$$

The $K_{\alpha\beta}$, B_α and E_α values are similar to classical form factors, patch radiosities, and emittances, respectively. However, each of these coefficients refers to some function representing part of the distribution of radiosity across a surface, as opposed to a constant value across a surface.

5.2 Progressive Refinement Methods

The matrix equation (5.6) can be solved using any standard matrix technique, such as Gaussian elimination, or even by using progressive refinement techniques [COHE 88]. If progressive methods are used, they must be slightly modified because the radiosity coefficients B_j^l may have negative values. These values are not negative energies; they indicate a radiosity distribution applied

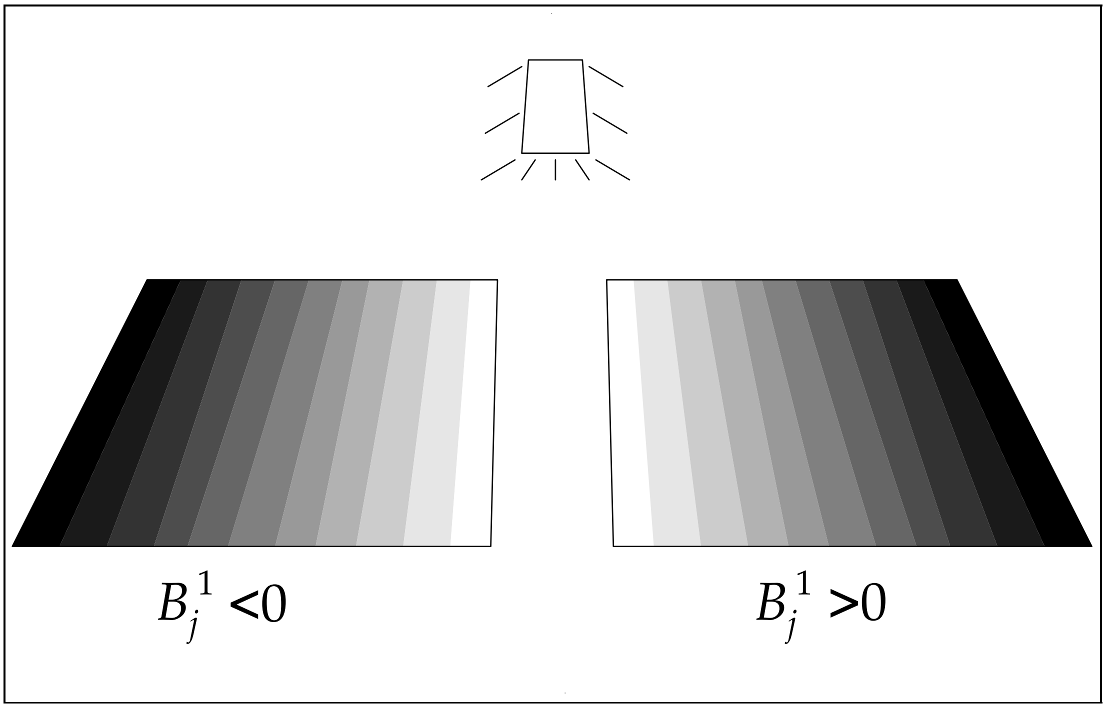


Figure 5.4: Although the radiosity of a surface is always positive, higher-order coefficients like the linear coefficient B_j^1 often take on negative values to describe the relative direction of a radiosity distribution.

opposite to the basis function (see Figure 5.4). Because of the possible presence of negative values, determining shooting order is somewhat more complex.

Traditional methods determine shooting order by computing the total unshot energy on each surface—the unshot radiosity $B_{j,\text{unshot}}$ multiplied by the area of the surface,

$$E_{\text{unshot}} = B_{j,\text{unshot}} \iint dA. \tag{5.7}$$

When using the Galerkin method, the analogous energy expression must also include a term for the particular basis element being used. Determining shooting order based on unshot energy,

$$E_{\text{unshot}} = B^{l}_{j,\text{unshot}} \iint \mathcal{T}_l(u,v) dA_j(u,v) du\, dv, \tag{5.8}$$

is not particularly useful because both $B_{j,\text{unshot}}$ and portions of higher-order basis functions $\mathcal{T}_l(u,v)$ can be negative. A much better sorting order can be found by using *unshot magnitude*, generated by taking the absolute values of the radiosity coefficient $B^{l}_{j,\text{unshot}}$ and the basis function $\mathcal{T}_l(u,v)$ in (5.8),

$$M_{j,\text{unshot}} = |B^{l}_{j,\text{unshot}}| \iint |\mathcal{T}_l(u,v)| dA_j(u,v) du\, dv. \tag{5.9}$$

This unshot magnitude expression provides a reasonable, sortable measure of the effect on the solution process of shooting from a particular basis element.

The refinement process usually continues until the total remaining unshot magnitude drops below some user-defined threshold. Because the basis functions are not strictly positive, the total unshot magnitude may occasionally increase as radiosity is transferred from low order elements on one surface to higher-order elements on another. However, the unshot magnitude does eventually decrease and converge to a solution once the magnitude has been shot from the appropriate lower-order basis functions. Figure 5.5 summarizes the progressive approach to Galerkin radiosity.

5.3 Edge Singularities

The only parameter left undefined in this solution method is the choice of basis set. Galerkin basis sets are usually chosen for some combination of speed and accuracy. If the radiosity equation were not singular, the best basis set would probably be the Chebyshev basis set, since transforming functions in and out of such a basis set can be achieved via a Fast Fourier Transform [DELV 85]. Unfortunately, the radiosity equation does not have a well-behaved kernel, so some other method must be used.

Equation (2.2) shows that near the common edge of two non-coplanar surfaces, the $\frac{1}{r^2}$ term in the double-differential form factor approaches infinity. Although this singularity disappears analytically upon double integration, in a finite basis set it can cause serious convergence problems. Ignoring the singularity is not a generally viable option; the system will converge extremely slowly (requiring many basis functions) for a mediocre basis set, and may even fail entirely for a bad basis set. Delves [DELV 81] shows that such a singularity will lead to complete loss of convergence for a Chebyshev basis set. Choosing an appropriate basis set is essential for accurate solution.

forall i, k:

  $B_i^k \leftarrow 0.$

  $B_{i,\text{unshot}}^k \leftarrow E_i^k.$

  $A_i^k \leftarrow \iint |\mathcal{T}_k(u,v)| dA_i(u,v) du\, dv.$ Precompute area/basis weights.

while forever

  $E_{\text{unshot}}^{\text{total}} \leftarrow 0, E_{\text{unshot}}^{\text{this shot}} \leftarrow 0.$

  forall i, k: Find the basis function to shoot from.

    if $E_{\text{unshot}}^{\text{this shot}} < B_{i,\text{unshot}}^k A_i^k$ **then**

      $E_{\text{unshot}}^{\text{this shot}} \leftarrow B_{i,\text{unshot}}^k A_i^k.$

      $BestI \leftarrow i, BestK \leftarrow k, BestB \leftarrow B_{i,\text{unshot}}^k.$

    $E_{\text{unshot}}^{\text{total}} \leftarrow E_{\text{unshot}}^{\text{total}} + B_{i,\text{unshot}}^k A_i^k.$

  if $E_{\text{unshot}}^{\text{total}} < E_{\text{user accuracy}}$ **return**

  forall j, l: Compute the shot.

    $K_{BestI,j}^{BestK,l} \leftarrow \left\langle \iint K_{BestI,j}(s,t,u,v)\mathcal{T}_l(u,v) du\, dv \middle| \mathcal{T}_{BestK}(s,t) \right\rangle_{\mathcal{W}}.$

    $B_j^l \leftarrow B_j^l + BestB\, K_{BestI,j}^{BestK,l}.$

    $B_{j,\text{unshot}}^l \leftarrow B_{j,\text{unshot}}^l + BestB\, K_{BestI,j}^{BestK,l}.$

  $B_{BestI,\text{unshot}}^{BestK} \leftarrow B_{BestI,\text{unshot}}^{BestK} - BestB.$

Figure 5.5: Pseudocode summary of the progressive refinement approach to Galerkin radiosity.

To insure reasonable convergence, the basis set will somehow have to compensate for the singularity. In (5.4), the singularity appears inside the quadruple integral that generates K_{ij}^{kl}. This integral also includes the as yet unspecified inner product weight function $\mathcal{W}(s,t)$. If the weight function $\mathcal{W}$ is chosen with zeroes of sufficiently high multiplicity where the kernel function K_{ij} has infinities, the two features can cancel and the integral will converge. Appendix B shows that the kernel singularity grows as a pole of order two as $K_{ij}(s,t,u,v)$ approaches a singularity, so the weight function should have zeroes of multiplicity two—double zeroes—at its edges. The Jacobi polynomial set $\mathcal{P}^{(2,2)}$ (see section 4.2), has the weight function

$$\mathcal{W}(s,t) = (1-s^2)^2(1-t^2)^2, \tag{5.10}$$

with double zeroes at $s = \pm 1$ and $t = \pm 1$. Using these polynomials as the basis set everywhere effectively cancels all edge singularities with the Jacobi weight function.

Although this basis set does yield solutions that converge, the convergence behavior is not uniform spatially. Pictures generated with a reasonably-sized $\mathcal{P}^{(2,2)}$ basis set tend to be dark near surface edges and bright near the center. These dark areas occur because of the assumed singularity; if there is no singularity, the light clumps towards the center. For light transfers across a singular edge, this behavior is appropriate, but for the vast majority of non-singular transfers, this Jacobi basis works badly.

These problems can be avoided by using a hybrid method. For non-singular

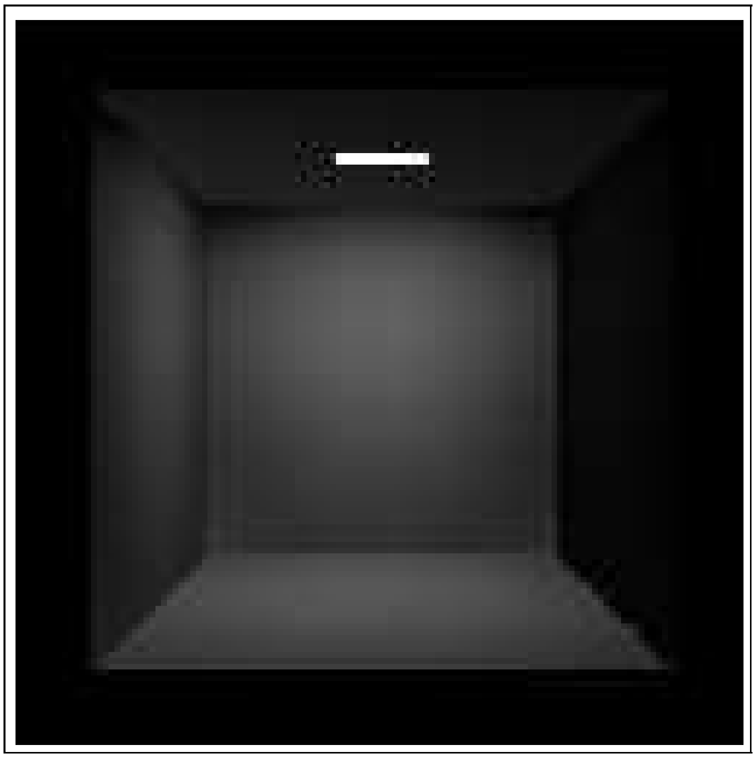


Figure 5.6: An empty box computed with up to fourth order polynomials, or 15 basis functions across each surface. On an HP 9000/720, the radiosity pass took 5.4 CPU seconds.

light transfers between surfaces that do not touch, a Legendre basis set is used. The choice of a Legendre basis set was somewhat arbitrary; it works much better than a Jacobi $\mathcal{P}^{(2,2)}$ basis set, and is relatively easy to use. It is possible that using a Chebyshev basis set, or something other formulation, might produce better convergence than a Legendre basis.

For the few transfers that are singular, a basis set of Jacobi polynomials is used, either $\mathcal{P}^{(0,2)}$ or $\mathcal{P}^{(2,0)}$ depending on the singularity's location. After the K_{ij}^{kl} coefficients and the associated amount of light transferred in a singular shot

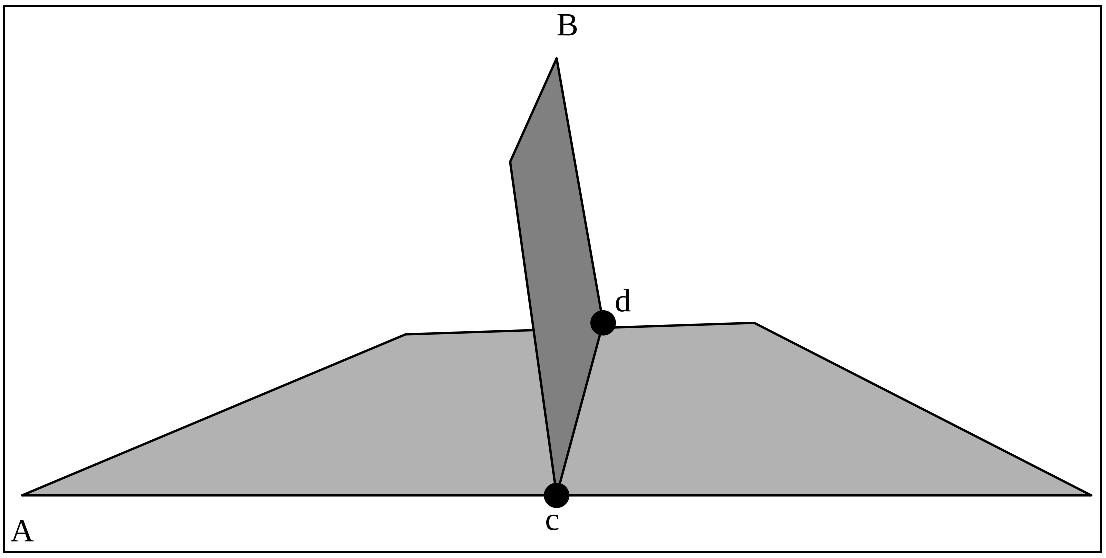


Figure 5.7: Surfaces A and B meet in a *T-intersection*; surface B divides surface A into two regions along the line $\overline{\mathrm{cd}}$.

has been computed, this polynomial function in s and t is projected back into a Legendre basis set for storage. This procedure is further explained in Section 5.4. A sample empty box computed with this hybrid method is shown in Figure 5.6.

Finding edge singularities in a scene with planar surfaces is fairly simple. Any collinear or near-collinear edges between two non-coplanar surfaces can form a singularity.

Because singularities are produced at all non-parallel intersections of distinct surfaces, geometries with T-intersections (the three-dimensional analog to Heckbert's T-corners [HECK 91b]) like those in Figure 5.7 significantly complicate the process of finding and dealing with singularities. Although such geometries could be handled by using a basis set with a two-dimensional weight function containing a double zero in the middle of the surface along the curve of intersection, constructing such basis sets would be relatively difficult even for polygonal surfaces. Instead, the method in this thesis suggests that either surfaces with T-intersections should be subdivided and avoided, or, if the contribution from the dividing surface is small enough, the singularity can simply be ignored.

5.4 Non-Singular Energy Transfers

A *non-singular transfer* is a radiosity transfer between two surfaces that do not meet in a singular edge. In order to generate radiosity solutions, the kernel matrix in (5.4) must be computed. For non-singular transfers, the weight function in that equation is the Legendre weight function,

$$\mathcal{W}_{\text{Legendre}}(s,t) = 1, \tag{5.11}$$

so the inner product of (5.4) can be rewritten,

$$K_{ij}^{kl} = \iint \iint K_{ij}(s,t,u,v)\mathcal{T}_k(s,t)\mathcal{T}_l(u,v)du\,dv\,ds\,dt. \tag{5.12}$$

This integral can be approximated to any degree of accuracy using an appropriate quadrature rule for each parametric variable (see section 4.3). The quadrature rule converts the integral to a weighted sum of kernel function samples, which is computed in the algorithm's inner loop. If the Gauss-Legendre quadrature points and weights are denoted p_α^L and w_α^L respectively, then each kernel matrix element is approximated by the summation,

$$K_{ij}^{kl} \approx \sum_{\alpha,\beta,\gamma,\delta} K_{ij}(p_\alpha^L, p_\beta^L, p_\gamma^L, p_\delta^L)\mathcal{T}_k^L(p_\alpha^L, p_\beta^L)\mathcal{T}_l^L(p_\gamma^L, p_\delta^L)w_\alpha^L w_\beta^L w_\gamma^L w_\delta^L. \tag{5.13}$$

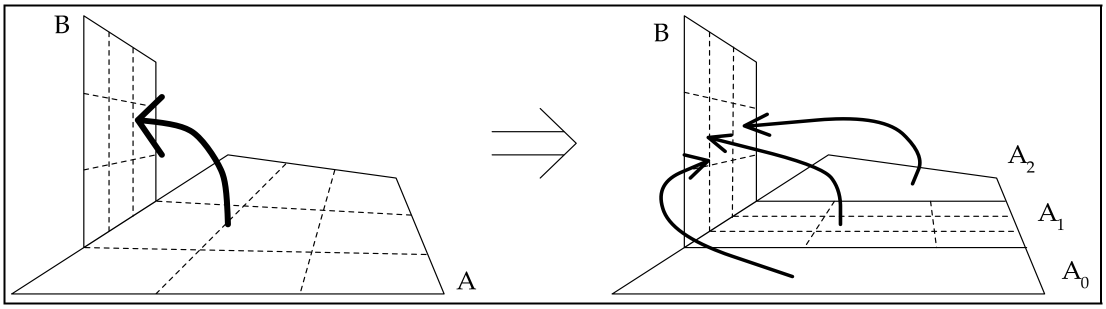


Figure 5.8: For accurate singular energy transfers, the parameterizations of surfaces sharing a singularity must be aligned along the singular edge.

When using a progressive solution method [COHE 88], this kernel matrix formulation allows a wide range of memory/CPU time tradeoffs. The approach requiring the least memory is to shoot from one basis function E_j^l at a time. Since only one particular value of l is used in the summation for any particular shot, only kernel matrix values with that l need be computed. Most of the higher-order basis functions never shoot during a progressive solution because of their small magnitude relative to lower-order terms. If memory or disk space is available, saving kernel function samples $K_{ij}(p_\alpha^L, p_\beta^L, p_\gamma^L, p_\delta^L)$ or results of the associated intersection tests can save significant CPU time, since each kernel sample requires a full intersection test with the environment.

5.5 Singular Energy Transfers

A *singular transfer* is a radiosity transfer between two surfaces that meet in a singular edge. Computing the singular transfer requires extra processing before evaluating the integral. To have a singularity that can be properly treated by a Gauss-Jacobi quadrature rule, the parametric functions for the shooting and receiving surface must match along the singular edge. For any other parametric representation of the two surfaces, such as that in the left of Figure 5.8, the derivation in Appendix B would not properly treat the singularity at the edge.

To convert the singularity into one recognizable by this method, the transmitting surface (A) is temporarily cut to insure that the the parametric functions match. After the singular part is cut out of the transmitting surface, the remainder of the transmitting surface transfers its energy to the receiving surface with a non-singular transfer. To improve the accuracy of the singular transfer, a higher-order quadrature rule can be used. Although this may take a relatively large amount of time for a single transfer, the total effect on the solution time is minimal because each surface only has a constant number of singular edges. Note that this subdivision is *temporary*; all non-singular interactions with surface A deal with the entire surface, not the three subsurfaces A_0, A_1, and A_2.

Although such a subdivision procedure could also be used when shooting from a smaller surface to a larger surface (such as from B to A in Figure 5.8), because of the relatively small amount of energy being transferred and the difficulty of projecting the resulting transfer back into A's basis set, the current implementation does not subdivide surfaces in this case.

Single point singularities, such as that between surfaces A_0 and B, do not require special treatment. Transfers between surfaces A_0 or A_2 and surface B are simple non-singular energy transfers.

Computing the kernel matrix elements for a singular transfer is more complex than for a non-singular transfer. Unlike the integral in (5.12), $\mathcal{W}_{kl}(s,t)$ is not identically 1, so a different quadrature rule with different weights and sample points must be used. Section 4.3 explains how to generate an appropriate quadrature rule for a general Jacobi polynomial basis set. Once the quadrature rule's points p^J_α and weights w^J_α have been computed, the kernel matrix elements can be computed by the summation,

$$K^{kl}_{ij} \approx \sum_{\alpha,\beta,\gamma,\delta} K_{ij}(p^J_\alpha, p^J_\beta, p^L_\gamma, p^L_\delta)\mathcal{T}^J_k(p^J_\alpha, p^J_\beta)\mathcal{T}^L_l(p^L_\gamma, p^L_\delta)w^J_\alpha w^J_\beta w^L_\gamma w^L_\delta. \tag{5.14}$$

When this weighted sum is evaluated, the resulting matrix entries K^{kl}_{ij} are in terms of a general Jacobi basis set, while the E^k_i and B^k_i values in storage are in terms of a Legendre basis set. The Jacobi matrix entries must be projected into the Legendre basis set before they can be combined with the other coefficients. Since the K^{kl}_{ij} coefficients are simply leading multipliers for polynomials, the easiest way to convert from Jacobi coefficients to Legendre coefficients is to expand the Jacobi coefficients into an ordinary polynomial in s and t, and then convert that polynomial back into a sum of Legendre polynomials.

Chapter 6

Shadow Discontinuities

As with any illumination algorithm, dealing with shadows caused by occlusions presents a special challenge. The easiest way to deal with shadows is to ignore them, and let the basis functions find a best fit to the shadows. Unfortunately, the detailed structure of most shadows violates the low frequency assumption of the Galerkin method. Shadow behaviors are not easily expressed in terms of a few polynomials.

It is generally not possible to accurately project a function with discontinuities into a low order basis set. The problem is a type of Gibbs behavior, such as occurs when anti-aliasing an image with a box filter. Attempting to model a partially-shadowed surface with a low order polynomial basis set produces a fuzzy shadow with ripples around it, as can be seen in Figure 6.1. Although these ripples diminish in intensity with a higher-order basis set, the ripples never vanish entirely.

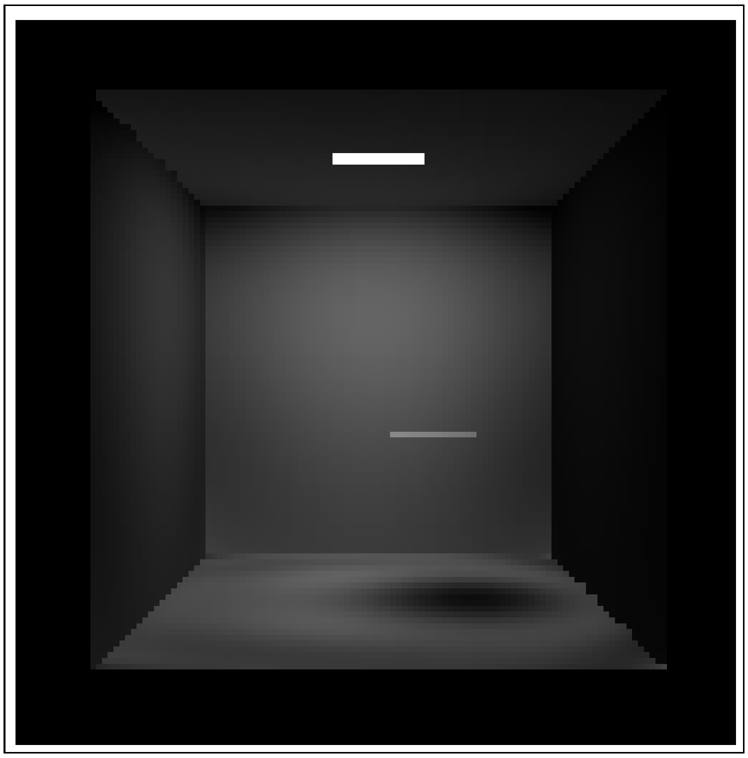


Figure 6.1: A non-shadow meshed box with an occluding rectangle computed with a fourth order basis set (15 basis functions) on all surfaces except the floor, which has an eighth order basis (55 basis functions). The ripples on the floor of the box appear because the sharp shadow edges in the radiosity equation cannot be accurately described by a low frequency Galerkin basis set.

Figure 6.2 illustrates how occlusions produce regions with different lighting behaviors. At the edges of penumbrae and umbrae, the radiosity function can be discontinuous in the first or second derivative [LISC 92,HECK 91a]. These discontinuities must somehow be dealt with explicitly. One way of solving these problems is by meshing the environment to remove most of the discontinuities—a procedure known as *shadow meshing*. Another approach is to apply a *shadow mask* approximation function to surfaces receiving shadows, dealing explicitly with the high-frequency behavior, while retaining the small size of the original environment. Both methods are explored in this chapter.

6.1 Shadow Meshing

Attempts to subdivide surfaces to find shadow boundaries have been investigated as part of radiosity solution schemes since radiosity's introduction to computer graphics. Nishita and Nakamae [NISH 85] accounted for shadow boundaries in polyhedral environments by finding the explicit shadow boundaries cast by a particular object onto a shadow receiver, meshing the shadow receiver independent of that information, and then using those boundaries to determine whether or not each element of the mesh was in the shadow or not. Lischinski and Tampieri [LISC 92] also begin by computing the exact shadow boundaries for each occluding object, but then use this information to explicitly mesh the shadow receiver along lines of low order discontinuities. Their method also retains the order of discontinuity across each edge, using this information

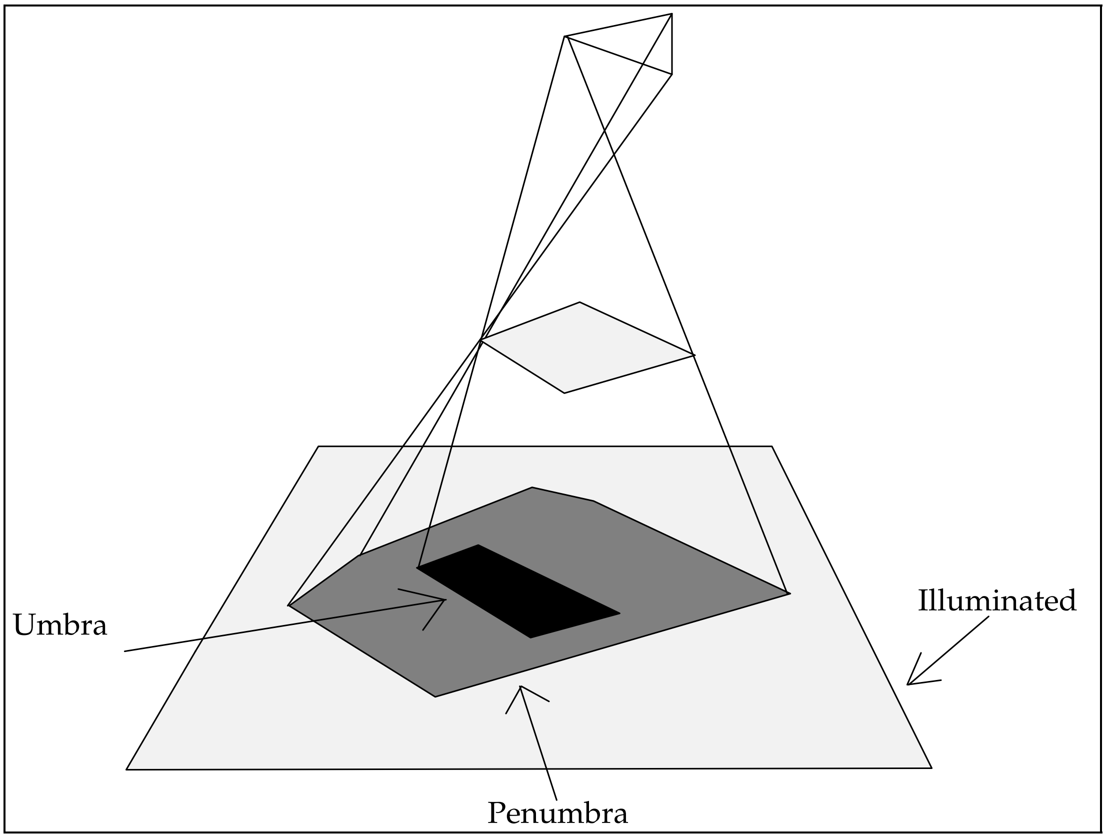


Figure 6.2: Every occlusion defines three types of shadow regions, with varying order of discontinuities between them. In simple environments, the boundaries of these regions can be found by casting rays from the corners of the light source past the corners of the occluding object to the receiving object.

to help generate the final picture. Unlike their discontinuity meshing procedure, *shadow meshing* as referred to in this thesis, involves only meshing along lines with discontinuities in the zeroth or first derivative. Although higher-order discontinuities can be important to the solution process, their subtleties can often be adequately fit by a relatively low order Galerkin method, without explicit meshing.

A fully shadow meshed environment has no shadow edges inside a meshed surface. Theoretically, one could shadow mesh an environment by iterating through all its light sources and occluding objects, determining which surfaces have shadow boundaries within them, and then dividing those surfaces along their shadow boundaries. Since in the radiosity model every surface is a light source, and each shadow casting adds more surfaces to the environment, this process could continue indefinitely. However, one can usually generate acceptable images by casting shadows only from primary light sources. Shadows cast by indirect light sources are relatively diffuse, and can be accurately modeled by polynomial functions. If there are enough lights in a region that illumination effects are fuzzy, shadow meshing is less important. Shadow meshing is only essential where the eventual solution may have a visually significant discontinuity.

Figure 6.3 shows a sample box with a single occluding square, properly shadow meshed. The shadow boundaries have been properly delineated, and shadow umbra and penumbra regions are easily visible. However, the CPU time to compute this picture has increased by a factor of three-and-a-half. The

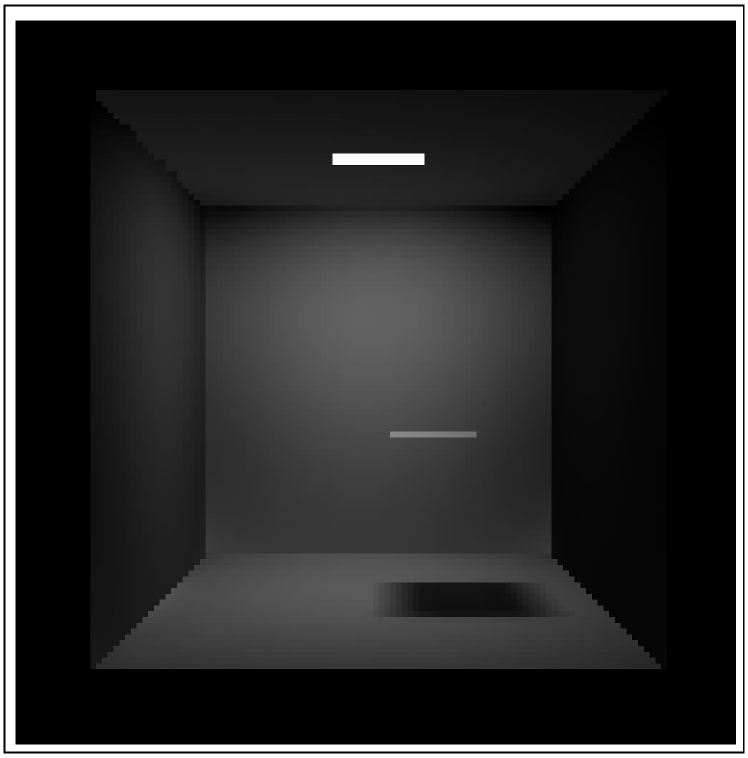


Figure 6.3: A box with a rectangular occluder. The floor is shadow meshed, and the scene is computed to fourth order on all surfaces except the floor and light source, which are computed to eighth order.

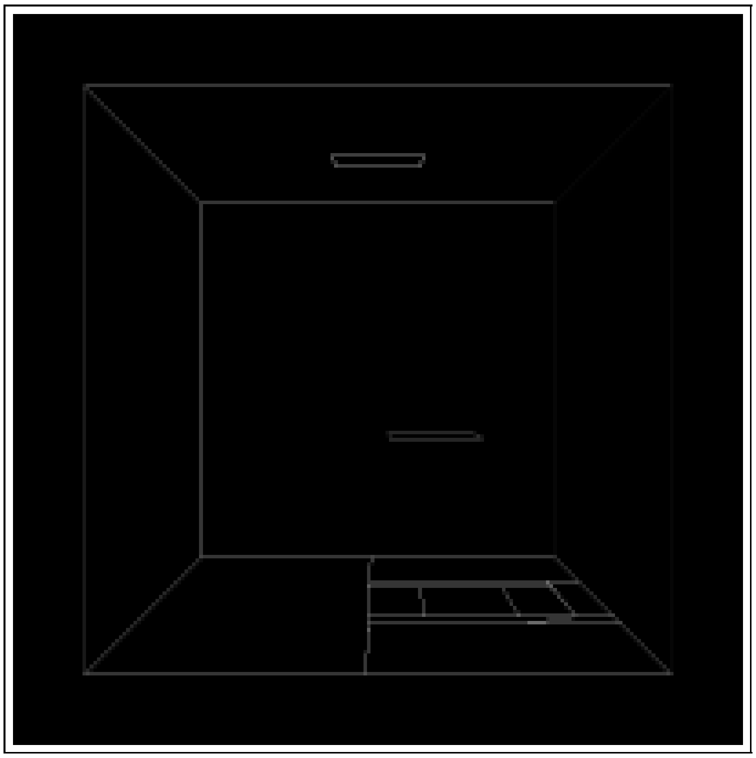


Figure 6.4: Mesh for figure 6.3.

reason for this is evident in Figure 6.4; the shadow meshing process increased the number of surfaces in the environment from seven to sixteen, and each radiosity shot takes at least $O(n \log n)$ time to compute. The problem with shadow meshing methods is that they increase the number of surfaces in the scene, vastly increasing computation time. Shadow masking methods avoid this difficulty.

6.2 Shadow Masking

Part of the difficulty in dealing with occlusions in radiosity is the relative scale of shadow features. Shadow features tend to contain local details, while radiosity is good at dealing with global illumination levels. By forcing radiosity methods to work on the local scale of shadows, a great deal of effort is spent on regions that have relatively little effect on the global solution. A better approach would be to somehow keep the details of shadow placement on a surface separate from that surface's general radiosity solution.

Returning to the parametric radiosity equation (2.5),

$$B_i(s,t) = E_i(s,t) + \rho_i(s,t) \sum_j \iint F_{i-j}(s,t,u,v)\mathrm{VIS}_{ij}(s,t,u,v)dA_j(u,v)B_j(u,v)du\,dv, \tag{6.1}$$

note that the visibility function $\mathrm{VIS}_{ij}(s,t,u,v)$ is dependent on both the shooting surface j and the receiving surface i. However, the occlusion's direct contribution to the solution will only be on surface i. Furthermore, the shadows caused by this occlusion will have a significant visual impact only where shadows are sharp, penumbras are small, or the radiosity difference across the shadow boundary is large. In such a case, it may be reasonable to approximate the visibility function with a *shadow mask* function $M_{i\leftarrow j}(u,v)$,

$$M_{i\leftarrow j}(s,t) = \frac{\iint \mathrm{VIS}_{ij}(s,t,u,v)du\,dv}{\iint du\,dv}, \tag{6.2}$$

where the denominator simply renormalizes $M_{i\leftarrow j}(s,t)$ to the range $[0,1]$. This approximation is valid if (with proper normalization)

$$\left[\iint \mathrm{VIS}_{ij}(s,t,u,v)F_{i-j}(s,t,u,v)dA_j(u,v)B_j(u,v)du\,dv\right] \approx$$
$$\left[\iint \mathrm{VIS}_{ij}(s,t,u,v)du\,dv\right]\left[\iint F_{i-j}(s,t,u,v)dA_j(u,v)B_j(u,v)du\,dv.\right] \qquad (6.3)$$

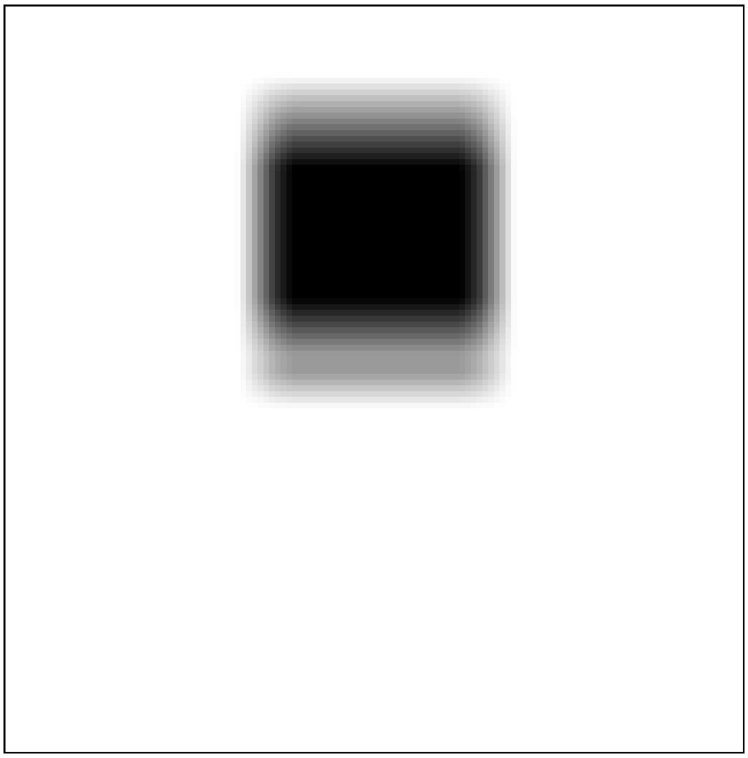


Figure 6.5: Shadow mask from the light source to the floor of Figure 6.6, computed on a regular 40 by 40 grid.

Since the shadow mask only depends on s and t, it can be moved outside of the integral,

$$B_i(s,t) = E_i(s,t) + \rho_i(s,t) \sum_j M_{i \leftarrow j}(s,t) \iint F_{i-j}(s,t,u,v) dA_j(u,v) B_j(u,v) du\, dv. \quad (6.4)$$

When expressed like this, the shadow mask is effectively a texture map describing the reflectivity across a surface, except that the reflectivity is different for every shooting surface j. Where the receiving surface is fully lit the shadow mask value is one, within the umbra region the shadow mask value is zero, and in the penumbra the shadow mask takes on intermediate values. Figure 6.5

shows a shadow mask from the light source to the floor of this chapter's test environment.

Shadow masks can be precomputed for light sources and receiving surfaces where shadows are expected to have a significant effect on the solution. For the majority of light-surface interactions, where shadows do not have a high-frequency effect on the solution, the traditional visibility calculations are used. In the current implementation, the user indicates explicitly between which surfaces to place shadow masks. Further research might develop a method to automatically detect where shadow masks are needed by looking for high-frequency Gibbs behavior in the radiosity coefficients B_i^k.

Although these shadow masks can be used as a reasonable approximation to an environment's occlusion behavior, they become far more useful when incorporated into the radiosity function. When expanding $B_i(s,t)$, instead of using a single set of coefficients to the basis functions $\{\mathcal{T}_l(s,t)\}$, add additional terms incorporating shadow masks cast from surfaces h.

$$B_i(s,t) = \sum_k B_i^k \mathcal{T}_k(s,t) + \sum_{h,k} M_{i\leftarrow h}(s,t) B_{i\leftarrow h}^k \mathcal{T}_k(s,t). \tag{6.5}$$

This separates high-frequency shadows $B_{i\leftarrow h}^k$ from low-frequency general radiosity behavior B_i^k. If radiosity in the shadow mask is maintained separately from radiosity caused by light coming from other light sources, when the final image is drawn, both radiosities can be combined to produce accurate shadow behavior, without the ringing Gibbs phenomena. Section 6.2.1 explains in detail how to add such computations to the progressive Galerkin radiosity algorithm.

Shadow masks can be implemented as a grid of sample points across s and t, an adaptive tree structure, or any other data structure which can take advantage of particular characteristics of the environment, such as actual shadow boundaries as described by Heckbert [HECK 91a] or Lischinski [LISC 92]. Shadow masks should produce approximately the same level of accuracy as meshing methods; however, they only increment the number of surfaces in the environment, and therefore add little to the radiosity solution time in environments with a limited number of important light sources.

In the meshable environments tested, even accounting for the time spent constructing shadow masks, the time required to compute a radiosity solution was significantly smaller than that for a reasonably accurate shadow mesh. For the simple environment in Figure 6.6, the shadow mask was a regular 40 by 40 grid of sample points on the floor. Since the shadow masked environment still has only seven surfaces, its radiosity pass was significantly faster than the radiosity pass for the sixteen-surface shadow meshed environment. In fact, since there was no Gibbs phenomena to transfer energy into higher order basis functions as in Figure 6.1, the radiosity pass actually required fewer shots and took less time to achieve the same fraction of unshot magnitude as the non-shadow masked version. Since shadow masking keeps the number of surfaces in the environment small, speed improvements over shadow meshing should be even more dramatic for complex environments.

Shadow masks have the further advantage that since they are defined in general (u, v) parametric space, a single set of shadow mask routines can cast

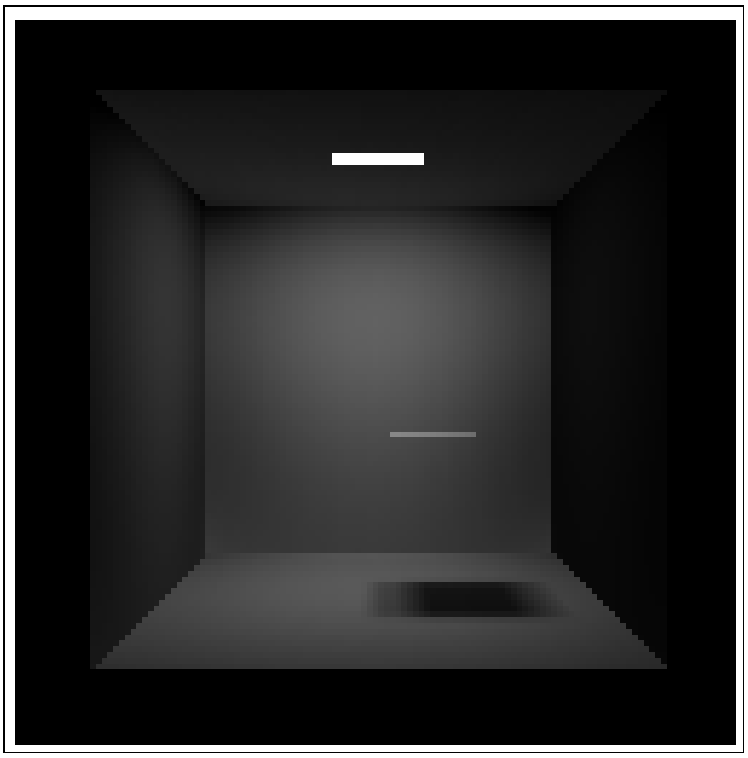


Figure 6.6: A box with the transfer from light source to floor shadow masked, computed to fourth order on all surfaces except the floor and light source, which are computed to eighth order.

shadows to and from any type of surface. In contrast, existing shadow meshing techniques require that shadows be cast onto a polygonal mesh. Shadow meshing non-polygonal surfaces without tessellation into hundreds of polygons would be mathematically challenging.

6.2.1 Implementation Details

During the radiosity pass, if surfaces i and j involved in a particular energy transfer have a shadow mask defined between them, the radiosity is transferred without visibility calculations directly to the $B^k_{i\leftarrow j}$ values. Because of the shadow masks, the visibility calculation has been effectively moved to the reflectivity of surface i, and should not be computed when transferring energy to surface i.

In terms of the pseudocode of Figure 5.5, lines 14–16, instead of computing a standard kernel matrix entry and distributing the energy to ordinary basis set coefficients,

```
14.
```
$$K^{BestK,l}_{BestI,j} \leftarrow \left\langle \rho_{BestI}(s,t) \iint F_{BestI-j}(s,t,u,v)\,dA_j(u,v)\mathcal{T}_l(u,v) \,\middle|\, \mathcal{T}_k(s,t) \right\rangle_{\mathcal{W}}.$$

```
15.
```
$$B^l_{j\leftarrow BestI} \leftarrow B^l_{j\leftarrow BestI} + BestB\, K^{BestK,l}_{BestI,j}.$$

```
16.
```
$$B^l_{j\leftarrow BestI,\mathrm{unshot}} \leftarrow B^l_{j\leftarrow BestI,\mathrm{unshot}} + BestB\, K^{BestK,l}_{BestI,j}.$$

The $B^k_{i\leftarrow j}$ coefficients accumulate energy just like the B^k_i coefficients, and may shoot energy if they become large enough following the standard progressive method. The main difference is that the appropriate shadow mask function must also be incorporated into the inner product. When energy is transferred from the high energy shadow B^k_{BestIJ}, the shadow mask function enters into the transfer integral of line 14,

```
14.
```
$$K^{BestK,l}_{BestIJ,j} \leftarrow \left\langle \iint K_{BestIJ,j}(s,t,u,v)M_{BestIJ}(u,v)\mathcal{T}_l(u,v) \,\middle|\, \mathcal{T}_{BestK}(s,t) \right\rangle_{\mathcal{W}}.$$

By manipulating the visibility term in this fashion, energy is properly transferred to and from the mask. When a surface has shadow masks on it, every other surface interacts either with a shadow, or with the standard surface description.

Chapter 7

Error Analysis

Because of the Galerkin method's formalism, the sources of error are relatively easy to describe. In fact, if all singularities were properly accounted for and the full infinite Jacobi/Legendre basis set were used, the Galerkin method would provide an exact solution to the radiosity equation. However, as in any practical implementation, some error is generated by the approximations required to implement the algorithm. Error in Galerkin radiosity generally comes from insufficient solution order and from inaccurate treatment of shadows.

The primary source of error in a Galerkin radiosity solution is the truncation of the basis set expansion after a finite number of terms. Just as the accuracy of conventional radiosity methods improves as more mesh elements are added, adding more functions to the Galerkin basis set generates a more accurate solution. As will be shown in later sections of this chapter, a solution order of six to eight across simple surfaces is generally accurate to within one part in two hundred, approximately that needed by computer graphics display algorithms.

Error caused by using a quadrature rule to approximate an integral is closely related to basis set truncation error. A Gaussian quadrature rule of a particular order is exact for polynomials up to that order; if the radiosity over a surface could be accurately described by a low order polynomial, the quadrature rule would accurately compute the integral. Since radiosity distributions are not exactly low order polynomials, quadrature rules generate some error.

Occlusions can also be a source of error. As explained at the start of Chapter 6, the few low order polynomials used in the Galerkin method cannot accurately model sharp shadow discontinuities. The resulting Gibbs ringing appears across the entire shadowed surface, producing not only unnecessarily fuzzy shadows, but error in places far distant from the occlusion. Although the ringing decreases when higher-order polynomials are used, the error often remains significant unless some explicit method is used to deal with shadows.

Regardless of the particular internal representation of a shadow mask, any shadow mask adds approximation error by moving the visibility calculation outside of the kernel integral (6.4). If a shadow mask is sampled on a regular or adaptive grid, sampling errors similar to those caused by conventional radiosity meshing algorithms can occur.

When a progressive refinement algorithm like that in section 5.2 is used to solve the radiosity system, the remaining unshot magnitude is another source of error. Although there may be less unshot energy after a particular number of shots than there would be with a conventional method, the energy may still be distributed to the wrong place until all surfaces have shot.

Finally, as with any computer algorithm, the floating point representation of the underlying real numbers always generates some error.

7.1 Energy Transfer Error

For the simple environment used by Sparrow's variational radiosity solution [SPAR 60], a fourth-order solution produced a relative error of less than one percent. Using the method of this thesis, error computations for a single energy transfer between parallel and perpendicular rectangles have produced similar levels of accuracy, except when the rectangle's centers are close relative to their size. All comparisons in this section are made against an analytic solution using the formulation of Sparrow and Cess [SPAR 78]. The relative error metric used is

$$E = \left\langle \frac{|B_{\text{Galerkin}}(s,t) - B_{\text{exact}}(s,t)|}{B_{\text{exact}}(s,t)} \right\rangle_{s,t}, \tag{7.1}$$

where the error is evaluated on a 500 by 500 grid of sample points on the receiving surface.

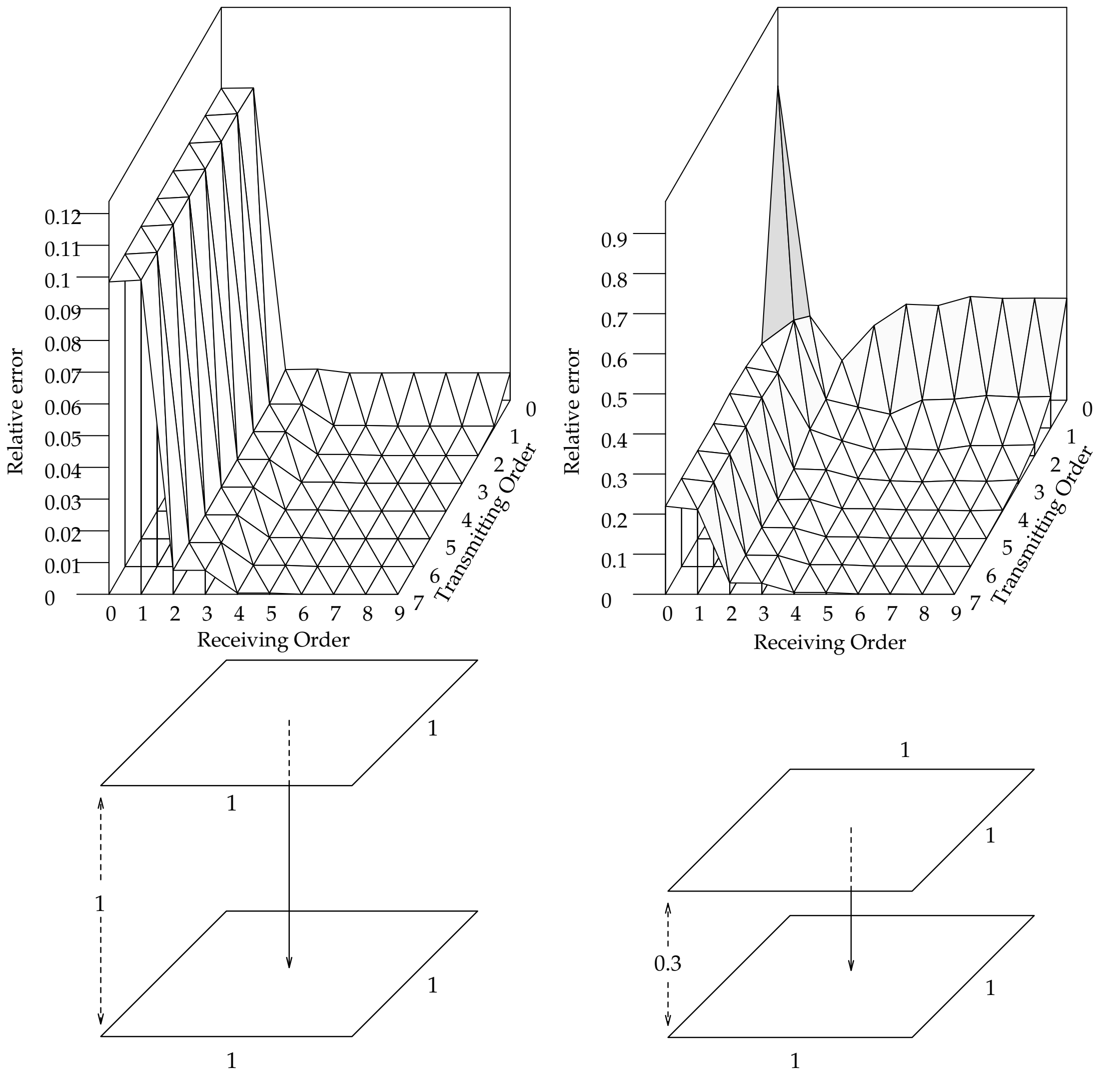
Error comparisons are graphed against the order of the transmitting and receiving surfaces. The order of the transmitting surface primarily affects transfer accuracy by changing the number of sample points taken by the quadrature rule on the transmitting surface. Increasing the order of the receiving surface increases the number of sample points taken on the receiving surface, as well as the number of functions used to represent the received radiosity distribution.

7.1.1 Parallel Squares

The simplest case is for a radiosity transfer between parallel squares with sides of length l that are some multiple of distance l apart. When the squares are $0.3l$ apart (Figure 7.1), a fourth-order transfer on both surfaces gives a relative error of 3.36%, while a fifth-order transfer gives a relative error of 0.61%. When the squares are even farther apart, the fourth-order transfer gives a relative error of 0.75%. Since the accuracy increases as the squares are placed farther away with respect to their size, a fourth or fifth order transfer should produce reasonable accuracy for computer graphics applications, where the final output device is accurate to only one part in 256.

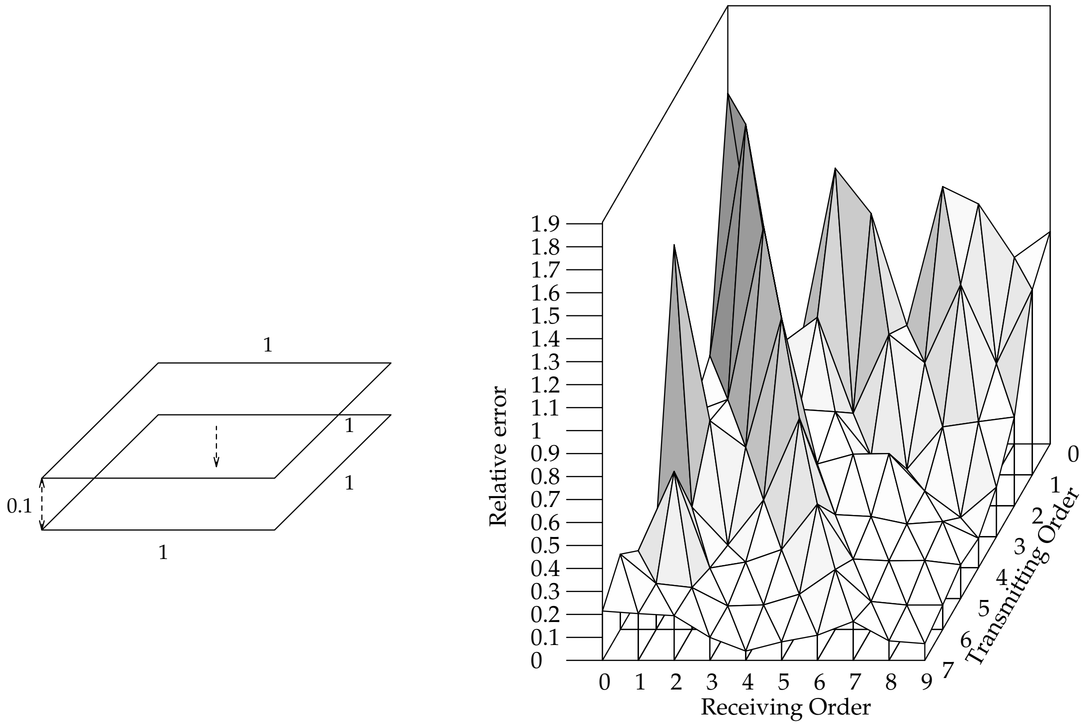
When the squares are relatively close to each other, the relative error becomes much higher, as can be seen in Figure 7.2. Even a transfer computed at eighth order on both transmitting and receiving surfaces produces an average relative error of 16.9%. Therefore when surfaces are extremely close relative to their size, to achieve reasonable accuracy, the surfaces must still be subdivided.

An interesting feature of Figure 7.2 is the error behavior along the graph's diagonal, where transmitting and receiving order are equivalent. These regions have significantly higher error than their off-diagonal neighbors. This poor on-diagonal behavior is a form of aliasing caused by the quadrature rule's positioning of sample points on both surfaces. On the diagonal, the kernel is sampled on both rectangles at points directly above each other. Since the kernel has a $\frac{1}{r^2}$ dependence, and the sample points are relatively close to each other, those samples dominate the computation, producing an inordinately high amount of



Distance between squares = l. Distance between squares = $0.3l$.

Figure 7.1: Average relative error for Galerkin radiosity transfers between two parallel squares of width l.



Distance between squares = 0.1l.

Figure 7.2: Average relative error for Galerkin radiosity transfers between two parallel squares of width l.

error. In such cases, a significantly more accurate result can be made by changing the solution order of the transmitting surface, thereby moving the sample points on both surfaces away from each other. Even *reducing* the transmitting order by one often avoids this aliasing problem.

7.1.2 Perpendicular Rectangles

Experiments with perpendicular rectangles illustrate the importance of direct treatment of radiosity's edge singularities. Using a non-singular Legendre basis set to compute the energy transfer produced large error even at relatively high order; a seventh order transfer produced a relative error of 33.2%, while even an off-diagonal transfer from order seven to order eight produced a relative error of 7.4%. As can be seen in Figure 7.3, the on-diagonal error behavior is again significantly worse than the off-diagonal behavior, because of a similar alignment of sample points on both surfaces.

Using a singular Jacobi basis set, results are much more accurate. Fourth and fifth order transfers both produce about 1.4% relative error. For the singular transfer, the receiving order appears to be much more important to producing an accurate computation than the transmitting order, as can be seen in Figure 7.3.

Figure 7.4 shows the effect of transmitting surface size on a singular energy transfer. In the left graph, where the transmitting surface is ten times the size of the receiving surface, the relative error stays low—about 1.2% for a fourth order transfer. However, when the transmitting surface is one-tenth the size of the receiving surface, the error becomes large again. The main reason for this is that

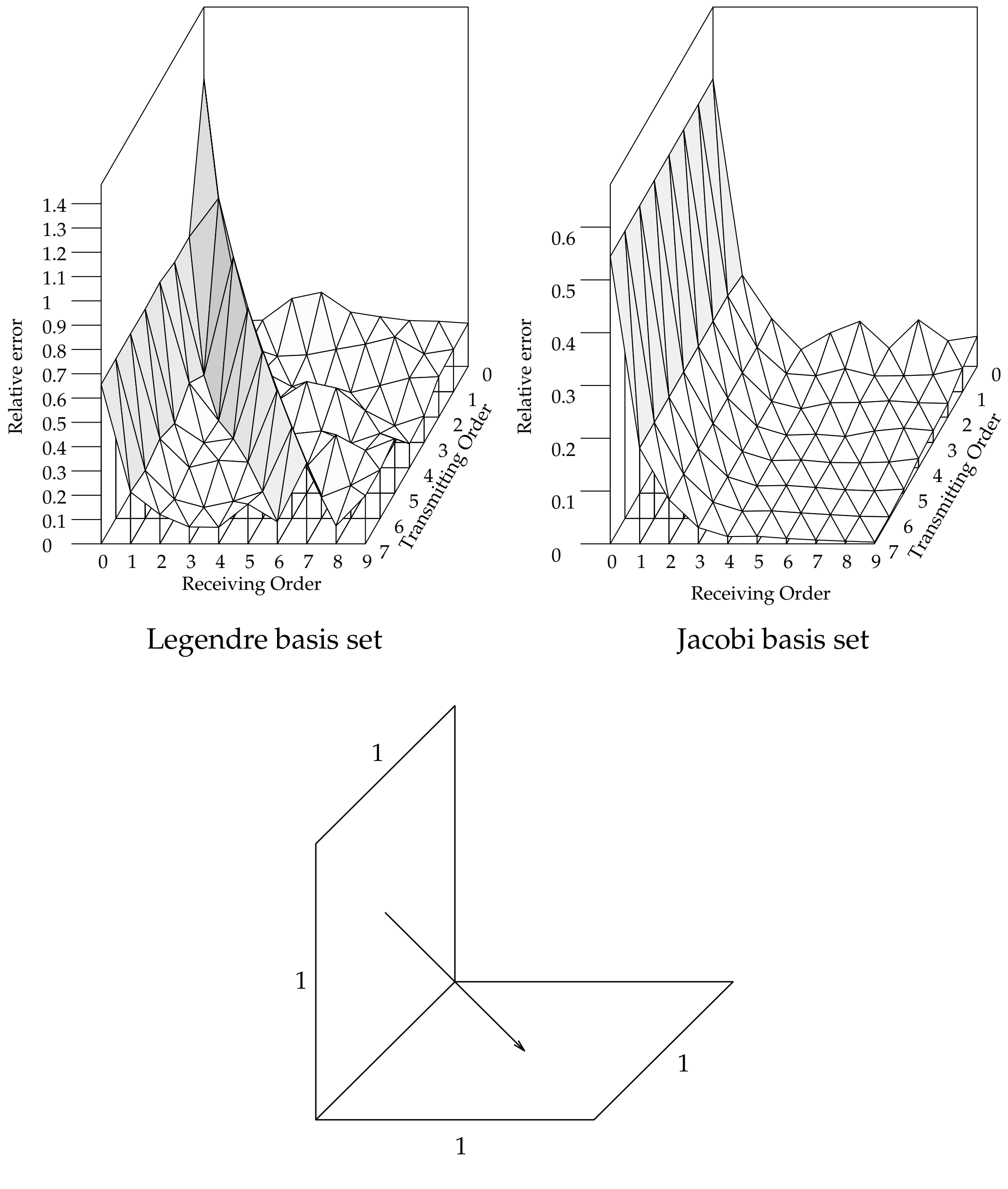


Figure 7.3: Average relative error for Galerkin radiosity transfers between two adjacent perpendicular squares. The Jacobi basis set computation produces significantly less error for this singular transfer than the Legendre basis set.

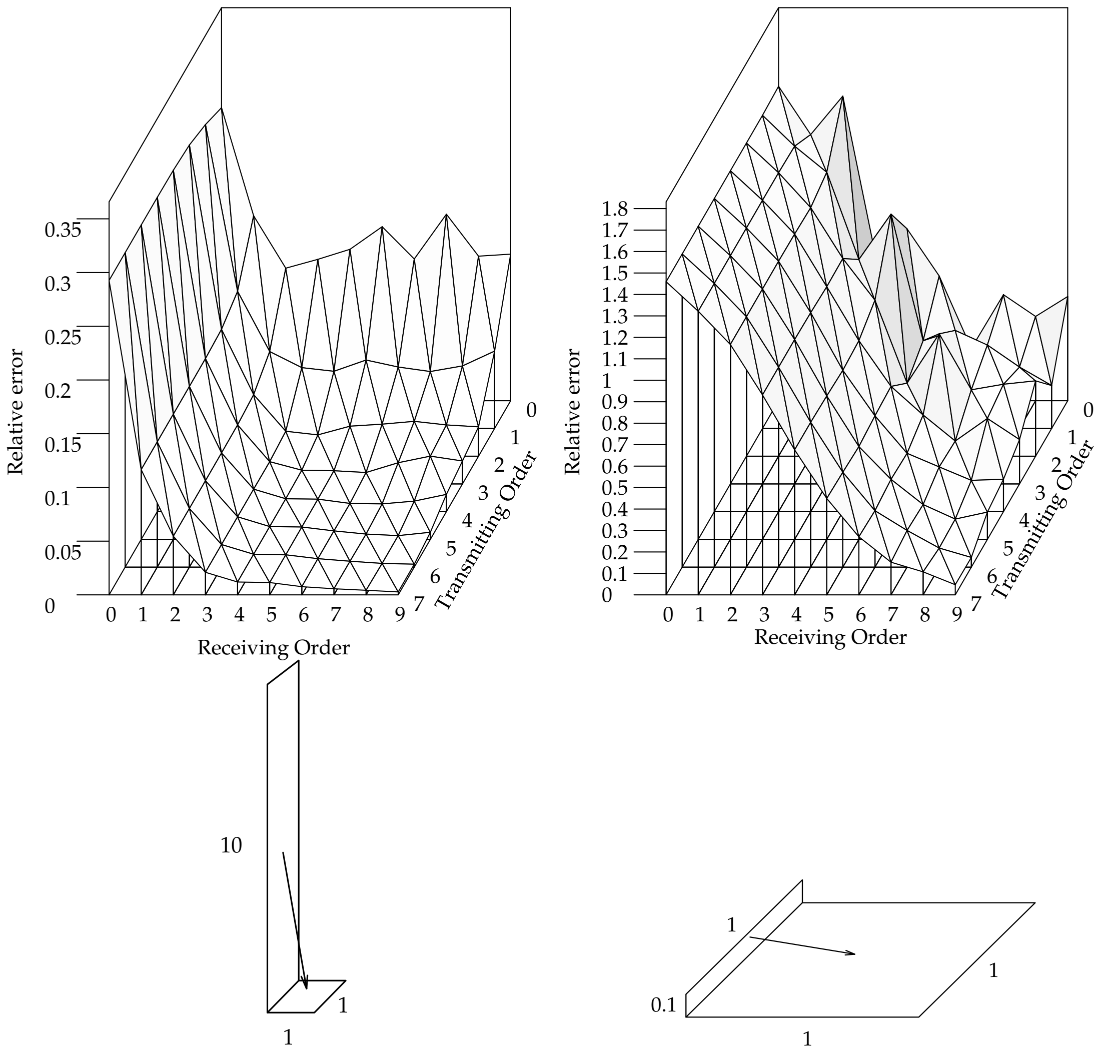


Figure 7.4: Average relative error for Galerkin radiosity transfers between rectangles of size ratio 10:1 and 1:10.

when light is transferred from a small area to a large one, most of the light ends up close to the transmitting surface. To represent such a radiosity distribution requires a lot of detail on the receiving surface, or a high order in Galerkin radiosity. Again, when such error-prone energy transfers are significant, the receiving surface may need to be subdivided.

7.2 Comparison with Conventional Radiosity

Lischinski and Tampieri provided the reference solution to the radiosity environment in Figure 7.5. This solution was computed using the discontinuity meshing techniques of [LISC 92], with adaptive integration using Wallace [WALL 89] point-to-point form factors. Individual triangles in the mesh were treated consistently as quadratic elements, reducing error caused by the Constant Radiosity Assumption [TAMP 91]. Error in the solution is primarily limited to a few meshing artifacts, visible near the corners of the top wall. This solution will be used as a baseline for comparison against images generated with Galerkin radiosity.

7.2.1 Effect of Order

The *order* of a solution is the highest total polynomial order used as a basis function for the solution. A zeroth-order solution would be equivalent to a conventional radiosity solution, with radiosity constant across a surface. A first-order solution would have linear radiosity variation, a second-order solution would have quadratic variation, and so on. Note that a basis function's order

Figure 7.5: A reference solution of a standard two-box radiosity test environment, computed using the method of [LISC 92]. This solution took 5.5 hours to compute, using 75 Megabytes of memory.

depends on the sum of the highest orders used in each dimension (st^4 is fifth order, for example). Different surfaces in a solution can be different orders; a high order basis could be used for large, visible areas, while a low order basis may be sufficient for shadowed regions.

Changing the order of a surface is like changing the mesh density on that surface. However, Galerkin methods differ from ordinary meshing schemes in that there is no mesh geometry to retain. The sample points on the surface are generated automatically, avoiding the memory explosion of adaptive meshing methods.

Figures 7.6 through 7.9 show pictures of a simple test environment solved with different solution orders. Shadows were created using a simple, 20 by 20 grid shadow mask. Notice how the floor appears smoother at higher order, even though no post-processing interpolation was used to smooth the meshing. Figure 7.10 shows the mesh used for all these solutions.

Figures 7.11 through 7.14 show difference images between the different order Galerkin solutions of the test environment and the reference solution of Figure 7.5. These images were created by first converting the Galerkin and conventional radiosity solution images to black and white, and then computing the absolute value of the intensity difference at each pixel. Dark regions of these images are where the two solutions agree; bright regions are where the two solutions differ. The Galerkin image was translated by half a pixel before comparison, so that outlines of the boxes and floors would be visible in the difference images. The difference image data was converted to film using the

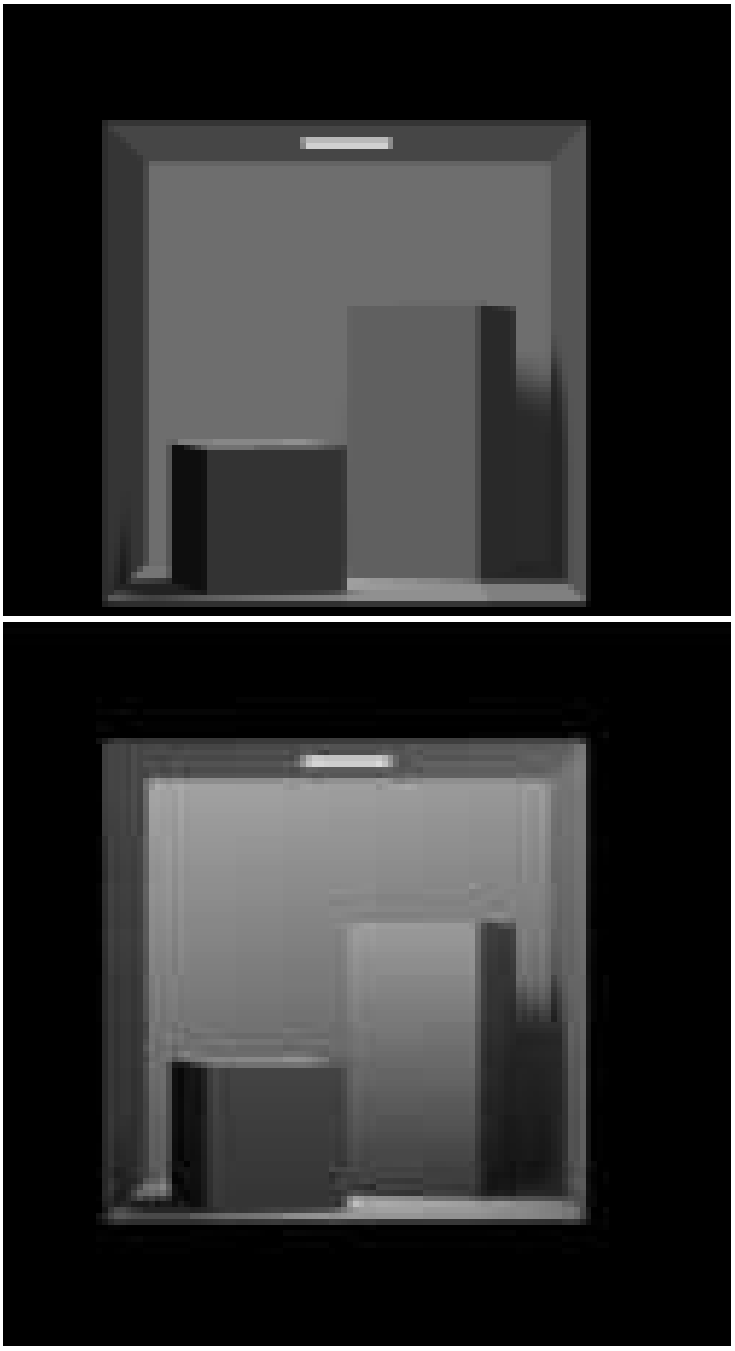


Figure 7.6: Solutions of the two box test environment, computed to less than 10 percent unshot magnitude, with solution orders zero (constant) and one (linear).

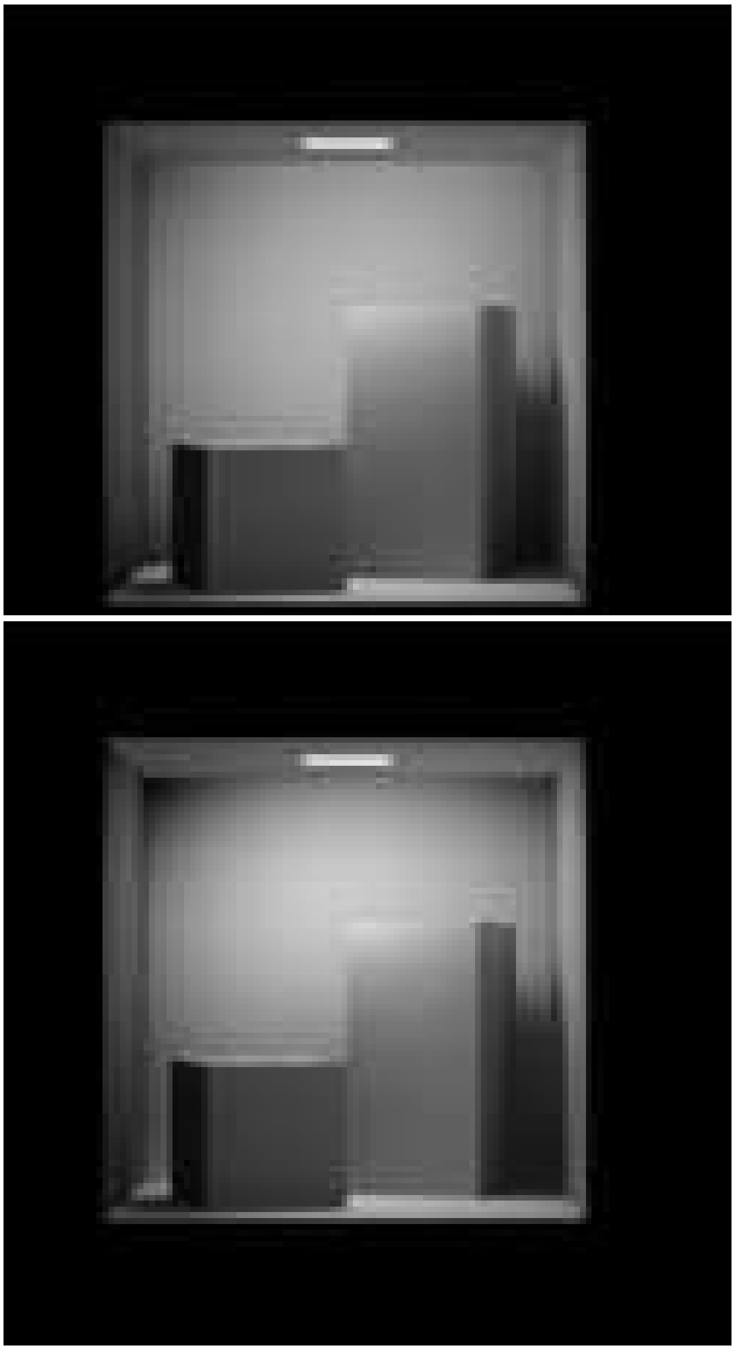


Figure 7.7: Solutions of the two box test environment, computed to less than 10 percent unshot magnitude, with solution orders two (quadratic) and three (cubic).

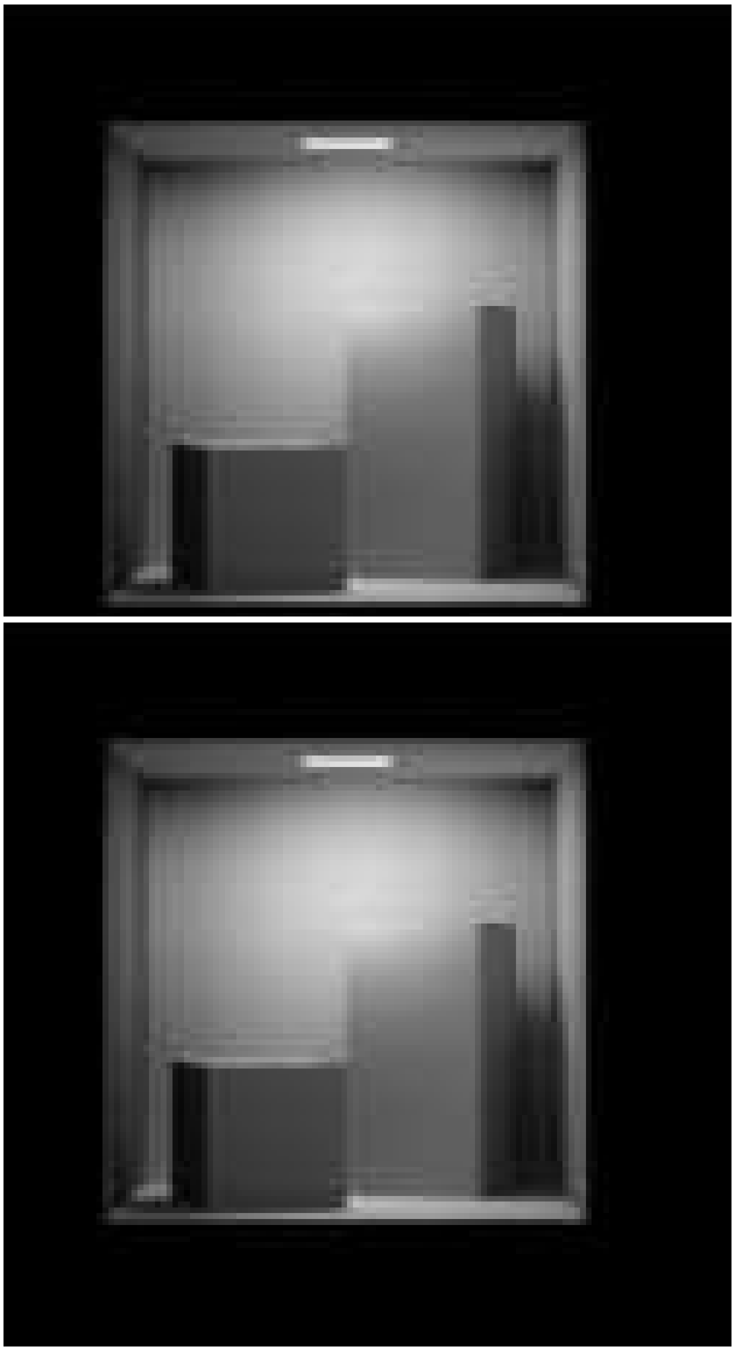


Figure 7.8: Solutions of the two box test environment, computed to less than 10 percent unshot magnitude, with solution orders four (quartic) and five (quintic).

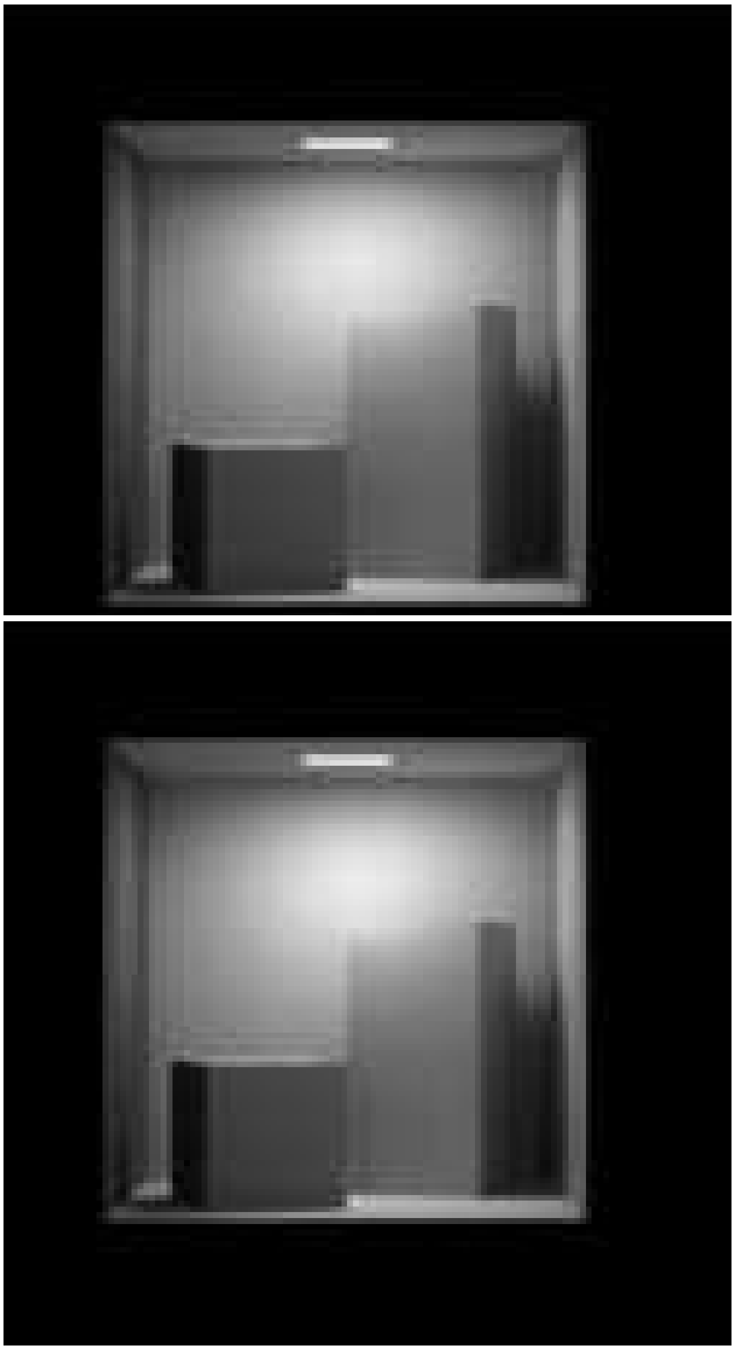


Figure 7.9: Solutions of the two box test environment, computed to less than 10 percent unshot magnitude, with solution orders six and seven. The seventh order solution took 4.2 hours to compute, using 6.5 megabytes of memory.

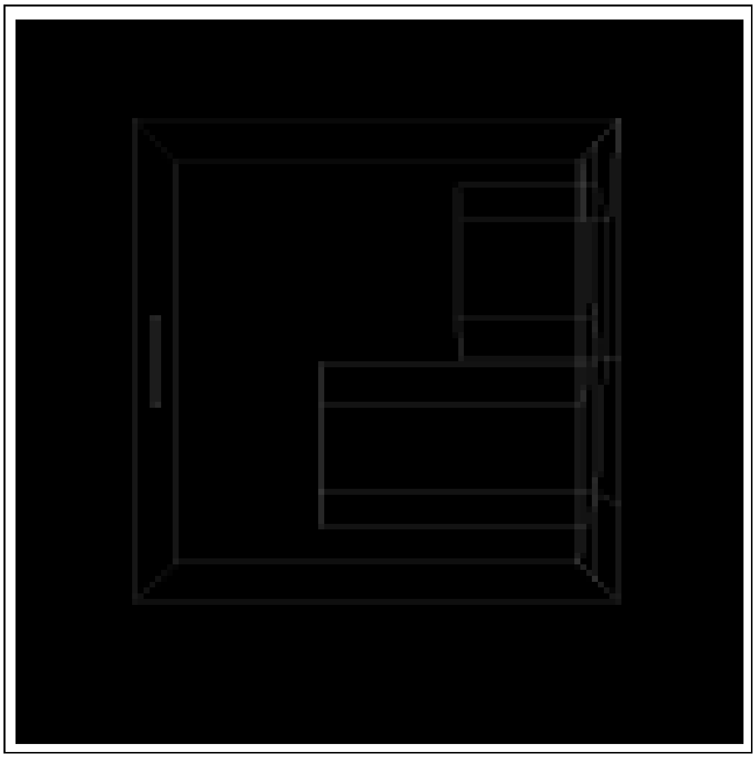


Figure 7.10: Mesh used for Figures 7.6–7.9

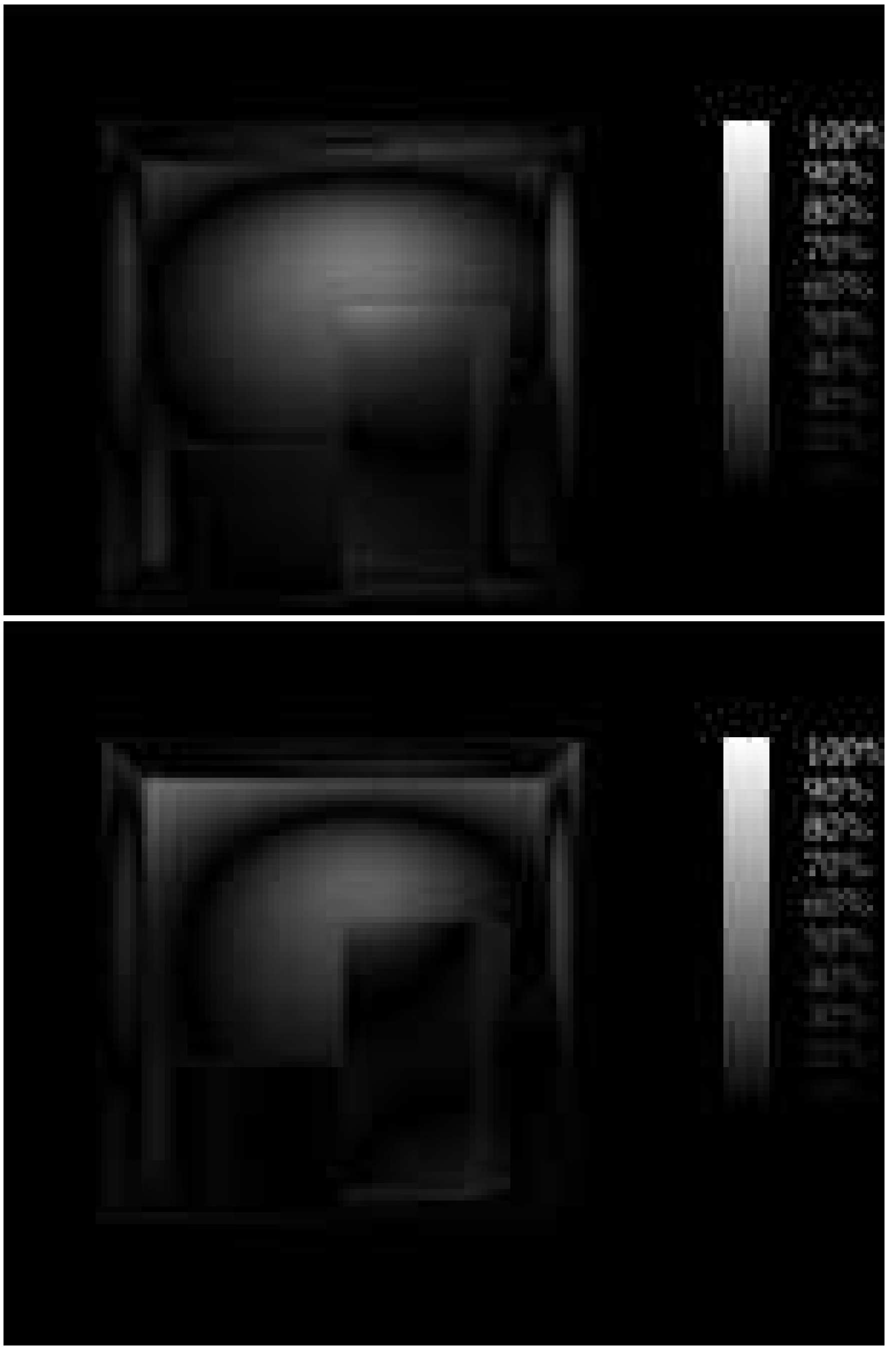


Figure 7.11: Difference images between the constant and linear Galerkin solutions of the two box test environment and the reference solution. The scale indicates the absolute intensity difference; brighter areas have more error.

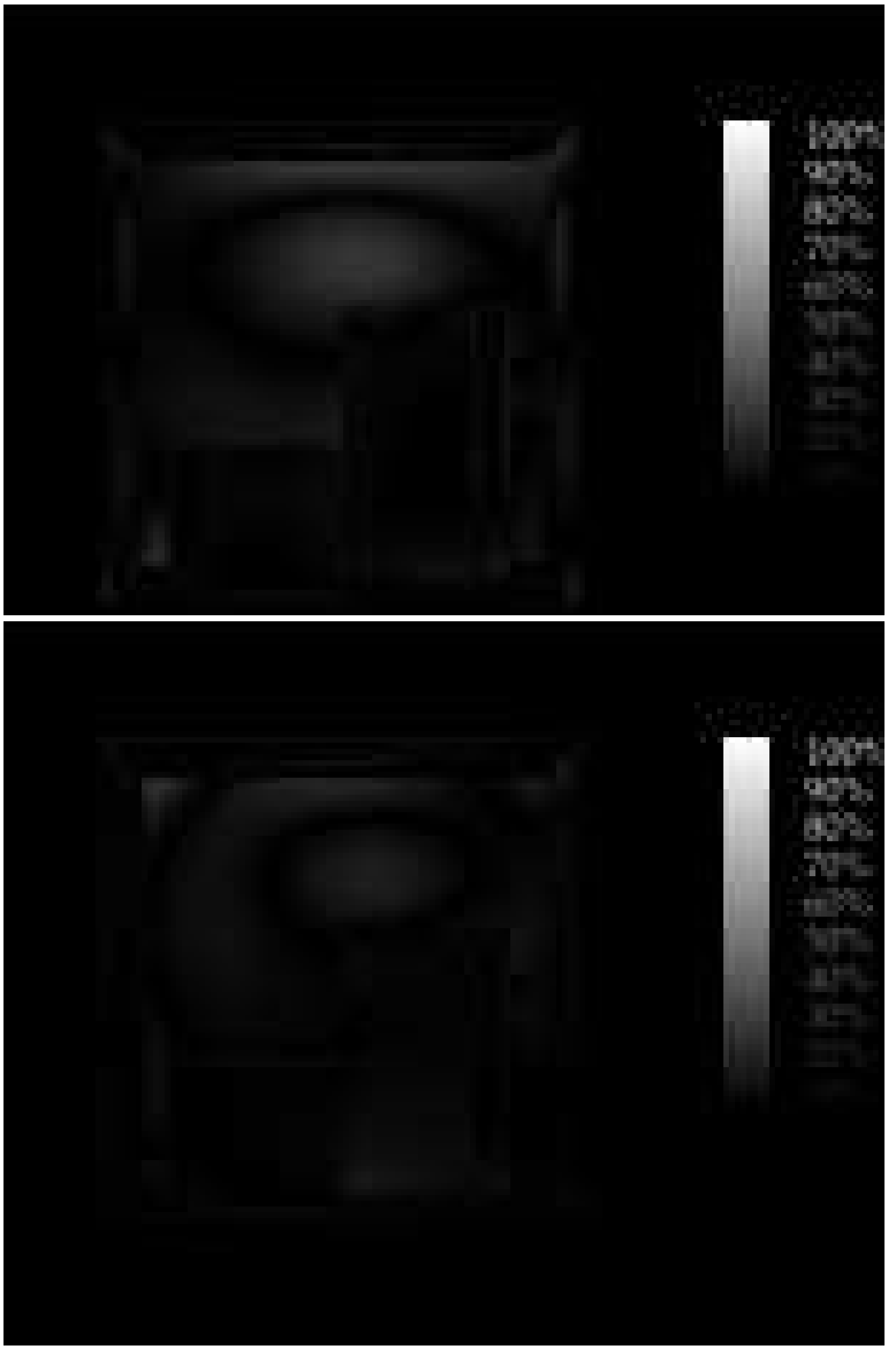


Figure 7.12: Difference images between the quadratic and cubic Galerkin solutions of the two box test environment and the reference solution. The scale indicates the absolute intensity difference; brighter areas have more error.

Figure 7.13: Difference images between the fourth and fifth order Galerkin solutions of the two box test environment and the reference solution. The scale indicates the absolute intensity difference; brighter areas have more error.

Figure 7.14: Difference images between the sixth and seventh order Galerkin solutions of the two box test environment and the reference solution. The scale indicates the absolute intensity difference; brighter areas have more error.

same procedure as the rest of the pictures, so comparisons of relative brightness are reasonable.

As would be expected, the difference images get progressively darker as the solution order increases, indicating that higher order solutions are more accurate. The regions where the solution is least accurate tend to be near singular edges. Just as conventional radiosity techniques often make these edges too dark, Galerkin techniques make these regions too bright at low order.

Another difference is that even though all Galerkin solutions were computed to the same level of unshot magnitude, the low order solutions are darker than the higher-order solutions. This behavior is indicative of the difference between unshot *energy* and unshot *magnitude*. In a constant radiosity solution, all of the unshot magnitude *is* unshot energy, so a solution can terminate with a lot of energy undistributed. In a higher-order solution, unshot energy is only part of the remaining unshot magnitude; much of the unshot magnitude is in higher-order basis elements. These basis elements generally don't store much energy; they simply indicate where the energy should be distributed. So, when solutions progress to the same level of unshot magnitude, but are computed at different order, the higher-order solution will distribute more energy, although some of it might not be in exactly the right place.

7.3 Discussion

In this particular test case, the method of [LISC 92] took about the same amount of time as the highest-order Galerkin solution. However, the Galerkin method only required 6.5 Megabytes of memory, compared to 75 Megabytes for a more conventional, meshing approach. For all environments tested in this thesis, Galerkin and conventional radiosity methods tend to take about the same amount of time to produce equivalent pictures. However, the Galerkin radiosity technique's lower memory usage is maintained in more complex environments, such as those in Chapter 8. Avoiding meshing saves a lot of memory.

Chapter 8

Results

8.1 Rendering

The radiosity solution computed by this method is a list of basis set expansion coefficients B_i^k for each surface i and basis function k. The actual illumination at a given point (s, t) on surface i can be recovered from these coefficients by computing

$$B_i(s,t) = \sum_k B_i^k \mathcal{T}_k(s,t). \tag{8.1}$$

If shadow masks were used, the additional coefficients B_{ih}^k can be incorporated with (6.5):

$$B_i(s,t) = \sum_k B_i^k \mathcal{T}_k(s,t) + \sum_{h,k} M_{i\leftarrow h}(s,t) B_{i\leftarrow h}^k \mathcal{T}_k(s,t). \tag{8.2}$$

In this implementation, environments are rendered by a simple ray-tracing technique. When a ray intersects a surface, that intersection point is projected back into the surface's parametric space. The resulting *st*-point is then used in (8.1) or (6.5) to compute a radiosity value for the appropriate pixel.

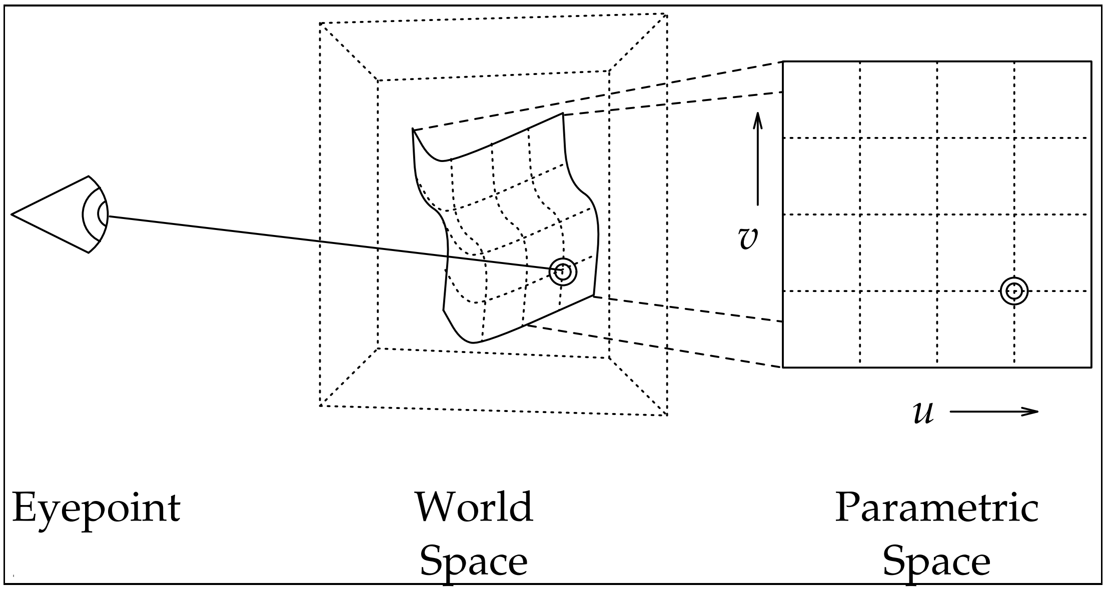


Figure 8.1: Images are rendered by ray-tracing. When an eye ray intersects an object in world space, that point on the object is mapped into the object's parametric space to find the actual radiosity value.

The functional solution $B_i(s, t)$ can also be interpreted as a shaded texture for a surface. Fast hardware texture mapping methods could conceivably be used to render radiosity solutions directly in a frame buffer.

8.2 Curved Surfaces

Curved surfaces can be easily incorporated into Galerkin radiosity; nothing in the solution approach excludes them. The kernel term's form factor as expressed in (2.6), like the ray-traced form factors in [WALL 89], includes surface normals explicitly. Curved surfaces are treated no differently from planar surfaces, except that their surface normals are a function varying over parametric space, instead

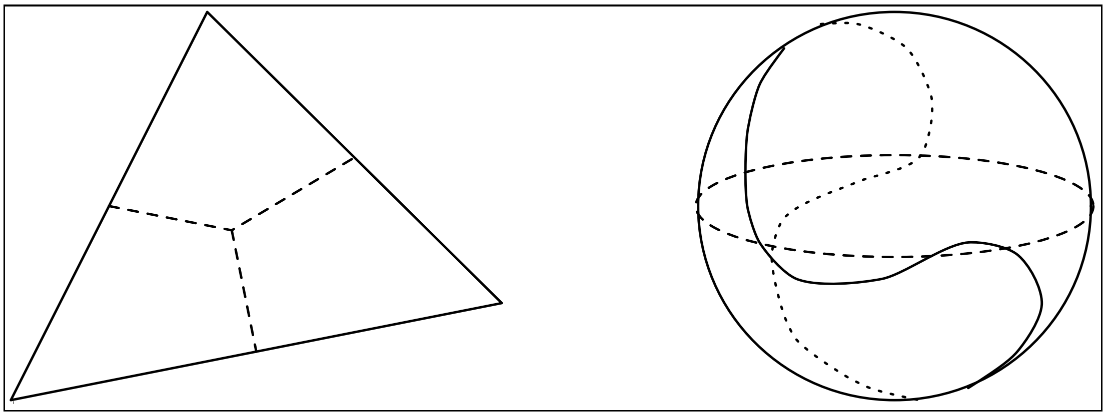


Figure 8.2: Some surfaces need to be subdivided to be expressed in terms of two parametric coordinates without any poles.

of a constant vector.

Because curved surfaces may have more complicated light distributions than planar surfaces, they may require a higher-order quadrature rule to achieve a similar absolute level of accuracy. No significant changes need to be made to the radiosity code to allow the use of curved surfaces. In an object-oriented programming domain, individual surface types interact with the radiosity core only through their parametric mapping and through a ray–surface intersection routine.

Some types of surfaces present special difficulties due to their parametric representation. Curved surfaces that touch themselves, such as cylinders and Mobius strips, are not inherently guaranteed by this thesis' method to have continuous radiosity values without seams. Surfaces with a point where one of the parametric dimensions collapses, such as at the poles of a sphere or one corner of a triangle, does not produce smooth solutions near those points.

These difficulties can be overcome in several ways. A relatively high order solution can be used for such a surface to compensate for the difficulty. Alternatively, most curved objects with these difficulties can be cut into two or three pieces without poles or self-contacting regions; a sphere can be cut like a baseball, or a triangle broken into three convex quadrilaterals (Figure 8.2). Another possibility might be to use a different quadrature rule and basis set for such surfaces, explicitly designed for the surface in question—such as using spherical harmonics to model the radiosity of a sphere.

Sample pictures are shown with cubic extrusions (Figures 8.3 and 8.4) and bicubic patches (Figure 8.5). Figure 8.6 shows an even more complex environment computed with Galerkin radiosity. The Galerkin radiosity method was applied directly to the surfaces in these environments; the curved surfaces were *not* tiled.

For comparison purposes, the teapot environment was also computed using a commercially-available radiosity package [STAR 90] (Figure 8.8). This package uses the Wallace point-sampling algorithm [WALL 89] to compute form factors, but only provides spatially uniform, non-adaptive meshing. Since this radiosity package cannot use bicubic patches directly, each of the teapot's patches were tessellated with a 20 by 20 grid—enough so that tessellation artifacts would be only barely visible. The radiosity solution took 6.2 hours, but over 54 megabytes of memory to compute; this simple forty-patch scene became a relatively complex, eight thousand polygon environment. In contrast, the computation of Figure 8.5 took 6.7 hours, but only required 3.9 megabytes of memory during

Figure 8.3: This picture is an example of a three minute, coarse solution to a radiosity environment containing curved surfaces. The shadow masks on floor sections, walls, and ceiling were a low-resolution 10 by 10 grid. The most obvious artifact of the coarseness of the solution is the jagged shadow edges throughout the environment.

Figure 8.4: This is a more complete, twenty-six minute solution to the same environment as Figure 8.3. Shadow masks were computed with up to an 100 by 100 grid on some surfaces.

Figure 8.5: The radiosity function across the clay teapot was solved directly, with a sixth-order basis set for each bicubic patch. The floor, walls, and portions of the teapot received shadow masks from the four lights.

Figure 8.6: This picture shows the interior of a temple containing over 600 surfaces, including polygons, bicubic patches, cylinders, cubic extrusions, and other parametrically defined surfaces.

Figure 8.7: This picture shows another view of the temple of Figure 8.6.

the radiosity pass. Most of this computation time was spent computing visibility samples; intersecting rays with bicubic patches is a notoriously difficult computation.

The two pictures appear remarkably similar, except for two notable differences. Both methods have some difficulties with the regions near the top corners of the box; standard radiosity computes values that are too dark, while Galerkin radiosity often places black artifacts in the top center of each wall and bright spots near the corners. The pictures also differ in their treatment of the teapot's shadow. Since standard radiosity produces shadow details by meshing, the shadow accuracy is limited to the user-defined mesh density. With Galerkin radiosity's shadow masks, the shadow on the floor could be sampled with a much denser (50 by 50) grid, without increasing the number of surfaces in the environment.

The teapot renderings are obviously not perfect. By paying more attention to either the details of standard radiosity meshing schemes, or to the Galerkin radiosity order and shadow mask density, the accuracy of either solution could have been improved. By running the program longer, both systems would have more closely approached their converged solution. The significant point of the comparison is that given approximately equivalent amounts of time to produce a solution, for these examples, conventional and Galerkin methods produce similar solutions. But since Galerkin methods don't have to maintain the detailed geometric structure of a mesh, they use significantly less memory than conventional methods.

Figure 8.8: The walls and clay teapot were subdivided using a 20 by 20 grid. The scalloping caused by the tessellation is faintly visible on the teapot.

8.3 Parallelization

Galerkin radiosity environments are not meshed into large, complicated data structures containing thousands of polygons. It is therefore relatively easy to transmit and maintain copies of the environment on multiple hosts. Since each individual light transfer between two surfaces depends only on the geometry and shadow masks, they can be computed on separate machines independently. For every shot, each light transfer from the shooting surface to receiving surfaces can be computed simultaneously, and then recombined before continuing with subsequent shots. Such a parallelization scheme was implemented, running concurrently on DECstations, HP 700's and 800's, and on multiple processors of an Apollo DN10000. The image of Figure 8.6 was computed in parallel on 20 workstations.

Table 8.1: Timings for shadow generation and radiosity pass combined for various pictures computed with this algorithm. All timings are for an HP 9000/720 workstation.

Figure	Description	CPU time	Shots	Unshot magnitude
5.6	Empty box	5.4s	7	40 percent
6.1	Non-shadow meshed box with single occluder	59.7 s	33	20 percent
6.3	Shadow meshed box	206 s	31	20 percent
6.6	Shadow masked box	42.8 s	22	20 percent
8.3	Coarse solution to room with sofa	207 s	10	50 percent
8.4	Detailed solution to room with sofa	25.9 m	78	20 percent
8.5	Clay teapot	6.71 h	53	50 percent

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Using the Galerkin method, this thesis has presented an alternative mathematical method for producing radiosity simulations. Through special treatment of the radiosity equation's singularities and discontinuities, the Galerkin technique's dependency on smooth kernels and smooth solutions can be overcome. Although the resulting pictures are similar to those produced by conventional radiosity methods, the method used to generate them is fundamentally different:

- The radiosity across a surface is allowed to be a smoothly varying function, limiting error in the energy transfer calculation to less significant, higher-frequency terms.

- Adequately sampled curved surfaces can be used directly. Since curved surfaces don't need to be tessellated, they can be incorporated into a scene without vastly increasing its complexity. Issues of approximating a surface's geometry and approximating a surface's radiosity are separated.

- Energy transfer error analysis shows that meshing is only needed when two surfaces are extremely close to each other relative to their size. Meshing is *not* needed to model variations in intensity across a surface. Furthermore, only one parameter with an explicit mathematical significance, the solution order of a surface, determines the level of accuracy of that surface.

- Pictures are rendered directly from the radiosity solution, without an intermediate interpolation step. The difficulties of choosing an appropriate interpolation scheme and the errors introduced by such schemes are both avoided.

- By using shadow masks, the local details of shadow edge generation are separated from the global issues of energy balance.

By focusing on a function-based solution, rather than on a point-based solution, this higher-order radiosity method eliminates much of the extra effort required to make conventional radiosity work. Many of the difficulties associated with meshing and computing vertex radiosities are avoided. Furthermore, since there is no form-factor computation dependent on objects being planar, and because the surface normal is actually part of the sampled kernel, computing radiosity for curved surfaces presents no special challenge.

This method is mathematically more consistent than most conventional radiosity methods. It avoids the inconsistency of computing radiosity values for a scene as if they were constant, and then rendering these constant values with interpolation. It is also orthogonal to other radiosity enhancements, such as the progressive refinement approach [COHE 88], or arbitrary reflectance functions [SILL 91].

Because of the high expense of raising the solution order, the order of every surface should be kept as low as possible. Using Gaussian quadrature, an energy transfer between two surfaces of order N requires $(N+1)^4$ kernel function samples. If a system has n surfaces with order N, then the computation time for each shot will be asymptotic to $N^4 O(n \log n)$ with proper intersection bounding volumes.

9.2 Deficiencies of the Method

As with most rendering algorithms, the Galerkin radiosity approach has its own particular disadvantages. Problems with the treatment of shadows are the most significant; if important shadows are missed, a solution will contain significant Gibbs ringing behavior. It may not always be easy to determine ahead of time where detailed shadow masking or meshing will be necessary, possibly requiring multiple solution attempts before all shadows are properly accounted for. Detailed shadow meshing can be very computationally expensive [LISC 92], especially when applied to curved surfaces. In the current implementation, the regular grid structure of shadow masks limits their spatial accuracy and requires extensive computation to produce detailed shadows.

Higher order methods also have the potential to be computationally expensive. Because of the $(N+1)^4$ samples required to transfer radiosity between surfaces of order N, radiosity calculations can become extremely expensive if too high a solution order is used. These difficulties become more significant when surfaces requiring a high order solution—surfaces that are self-intersecting or highly curved—are used.

The method does not mathematically insure radiosity continuity between adjacent coplanar surfaces. Such surfaces appear much less frequently in a shadow masked environment than in a meshed environment. However, the only way to generate the needed continuity is to approximate it by using a high enough order (usually above eighth for planar surfaces) that the error on each surface is reduced until their radiosity values match. This approach does not replace a continuity constraint; future research should develop a method for mathematically enforcing such a constraint.

Finding all the singularities in a system can also be difficult. Environments usually have a large number of T-intersections (see Figure 5.7), each of which could require a meshing step. Although T-intersections can often be ignored, there's always a risk that the ignored singularity will cause the solution to fail to converge, again requiring recomputing the environment.

9.3 Future Work

Shadow masks are currently implemented using bilinear interpolation on a simple grid of sample points. Many more efficient sampling schemes are possible, such as adaptive quadtrees, or some method that directly computes the location of shadow discontinuities. Additionally, some method should be developed for automatically determining where shadow masks are needed.

Some method for enforcing continuity between adjacent surfaces, possibly by using some sort of modified patch/element method could lower the required solution order, and significantly accelerate the algorithm when such surfaces are present. A method combining adaptive meshing and a low order Galerkin solution might produce reasonable images rapidly. Extending Hanrahan's hierarchical multigridding technique [HANR 91] to higher order functions could produce a means to do this.

The method of this thesis uses a Legendre basis set for non-singular energy transfers. Galerkin methods frequently use a Chebyshev basis; by examining the relative accuracy of different basis sets, it may be possible to find a better basis set for the radiosity problem.

This thesis is only a first attempt at applying higher order solution methods to the radiosity problem. Much work remains to fully integrate this approach into the general framework of global illumination and radiosity.

Appendix A

Direct Integration for Simple Environment

The radiosity of the lower patch in Figure 3.2, as expressed in (3.6), is

$$B_0(s,t) = .5 \int_0^1 \int_0^1 F_{0-1}(s,t,u,v)du\,dv. \tag{A.1}$$

The point-to-point form factor from (2.6) is:

$$\begin{aligned}
F_{i-j}(s,t,u,v) &= \frac{\left(\hat{n}_i(s,t)\cdot\left(\vec{x}_j(u,v)-\vec{x}_i(s,t)\right)\right)\left(\hat{n}_j(u,v)\cdot\left(\vec{x}_i(s,t)-\vec{x}_j(u,v)\right)\right)}{\pi\|\vec{x}_i(s,t)-\vec{x}_j(u,v)\|^4}, \\
dA_j(u,v) &= \left\|\frac{\partial\vec{x}_j(u,v)}{\partial u}\times\frac{\partial\vec{x}_j(u,v)}{\partial v}\right\|.
\end{aligned} \tag{A.2}$$

For the environment of Figure 3.2,

$$\begin{aligned}
&\vec{x}_0(s,t) = (s,t,0), \quad \vec{x}_1(u,v) = (u,v,d), \\
&\hat{n}_0(s,t) = (0,0,1), \quad \hat{n}_1(u,v) = (0,0,-1).
\end{aligned} \tag{A.3}$$

The area term $dA_1(u,v)$ reduces to

$$\begin{aligned}
dA_1(u,v) &= \left\|\frac{\partial(u,v,d)}{\partial u}\times\frac{\partial(u,v,d)}{\partial v}\right\| \\
&= \|(1,0,0)\times(0,1,0)\| \\
&= 1.
\end{aligned} \tag{A.4}$$

The entire form factor $F_{0-1}(s,t,u,v)$ reduces to

$$\begin{aligned} F_{0-1}(s,t,u,v) &= \frac{((0,0,1)\cdot((u,v,d)-(s,t,0)))\,((0,0,-1)\cdot((s,t,0)-(u,v,d)))}{\pi\|(s,t,0)-(u,v,d)\|^4} \\ &= \frac{(d)(d)}{\pi\left((s-u)^2+(t-v)^2+d^2\right)^2}. \end{aligned} \tag{A.5}$$

Substituting this back into (A.1) gives:

$$B_0(s,t) = \frac{d^2}{2\pi}\int_0^1\int_0^1 \frac{1}{\left((s-u)^2+(t-v)^2+d^2\right)^2}du\,dv. \tag{A.6}$$

Evaluating the inner integral over u produces

$$B_0(s,t) = \frac{d^2}{4\pi}\int_0^1 \left[\begin{array}{l} \frac{s}{(d^2+(t-v)^2)\,(d^2+s^2+(t-v)^2)}+ \\ \frac{1-s}{(d^2+(t-v)^2)\,(d^2+(1-s)^2+(t-v)^2)}+ \\ (d^2+(t-v)^2)^{-\frac{3}{2}}\left(\tan^{-1}\frac{1-s}{\sqrt{d^2+(t-v)^2}}+\tan^{-1}\frac{s}{\sqrt{d^2+(t-v)^2}}\right) \end{array}\right] dv, \tag{A.7}$$

and integrating the outer integral over v produces

$$\begin{aligned} 4\pi B_0(s,t) = {} & \frac{s}{\sqrt{d^2+s^2}}\left[\tan^{-1}\frac{1-t}{\sqrt{d^2+s^2}}+\tan^{-1}\frac{t}{\sqrt{d^2+s^2}}\right]+ \\ & \frac{1-s}{\sqrt{d^2+(1-s)^2}}\left[\tan^{-1}\frac{1-t}{\sqrt{d^2+(1-s)^2}}+\tan^{-1}\frac{t}{\sqrt{d^2+(1-s)^2}}\right]+ \\ & \frac{t}{\sqrt{d^2+t^2}}\left[\tan^{-1}\frac{1-s}{\sqrt{d^2+t^2}}+\tan^{-1}\frac{s}{\sqrt{d^2+t^2}}\right]+ \\ & \frac{1-t}{\sqrt{d^2+(1-t)^2}}\left[\tan^{-1}\frac{1-s}{\sqrt{d^2+(1-t)^2}}+\tan^{-1}\frac{s}{\sqrt{d^2+(1-t)^2}}\right]. \end{aligned} \tag{A.8}$$

This result is equivalent to the contour integral formulation of [SPAR 78] and [GORA 85]. Because of the complexity of this single-shot result in a nearly trivial geometry, it seems unlikely that analytic methods could be reasonably used for more complex environments.

Appendix B

Order of a Corner Singularity

This appendix examines the nature of radiosity's singularity by considering the simplest system affected by it: two square, perpendicular surfaces that meet in a corner, as shown in Figure B.1. Define these surfaces parametrically,

$$\begin{aligned} \vec{x}_0(s,t) &= s\hat{\imath} + t\hat{\jmath} + \hat{k}, & \vec{x}_1(u,v) &= \hat{\imath} + u\hat{\jmath} + v\hat{k}. \\ \hat{n}_0(s,t) &= -\hat{k} & \hat{n}_1(u,v) &= -\hat{\imath}. \end{aligned} \tag{B.1}$$

The kernel function will be:

$$K_{01}(s,t,u,v) = \frac{\rho(s,t)}{\pi} \frac{(\hat{n}_0 \cdot (\vec{x}_0 - \vec{x}_1))\,(\hat{n}_1 \cdot (\vec{x}_1 - \vec{x}_0))}{\|\vec{x}_0 - \vec{x}_1\|^4} \tag{B.2}$$

Substitute the values for this example, and take the limit along the common y-coordinate.

$$\begin{aligned} K_{01}(s,t,u,v) &= \frac{\rho(s,t)}{\pi} \frac{(v-1)\,(s-1)}{\left((s-1)^2 + (t-u)^2 + (1-v)^2\right)^2} \\ \lim_{u \to t} K_{01}(s,t,u,v) &= \frac{\rho(s,t)}{\pi} \frac{(v-1)\,(s-1)}{\left((s-1)^2 + (1-v)^2\right)^2} \end{aligned} \tag{B.3}$$

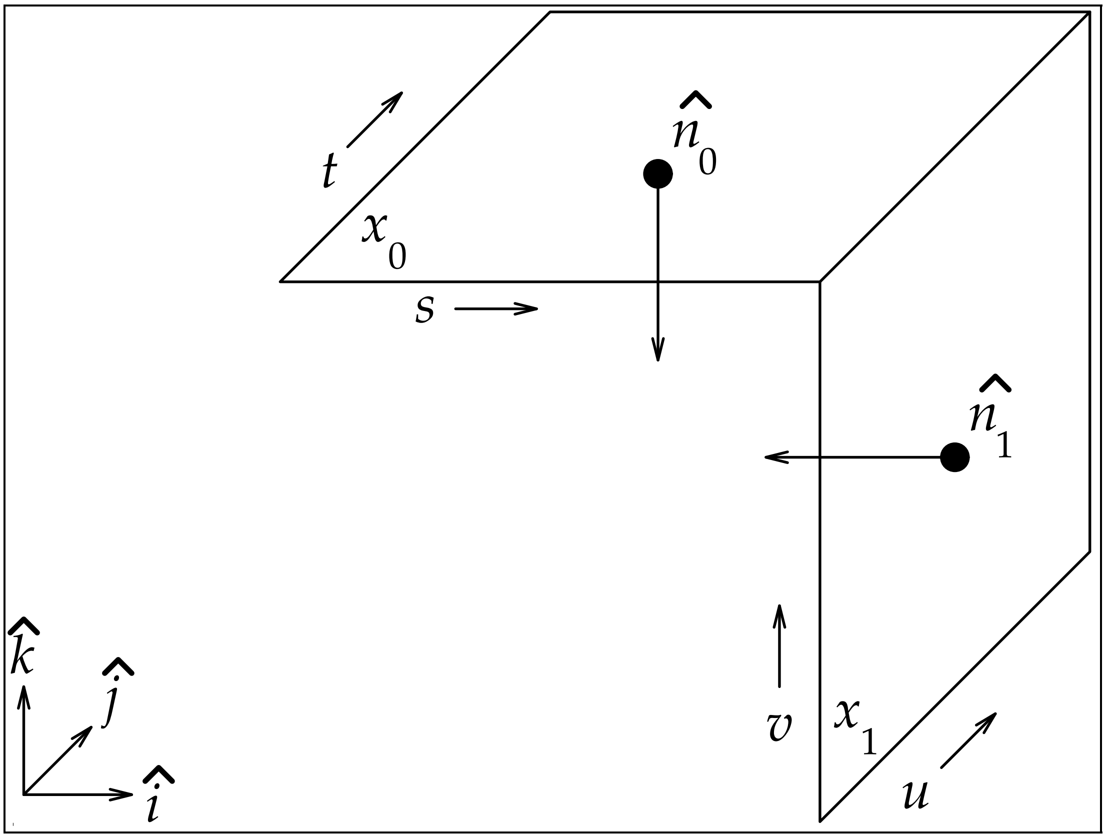


Figure B.1: There is a singularity at the corner of surfaces x_0 and x_1.

Using standard quadrature rules, v and s will approach the corner at the same rate, so take the further limit as $v = s$.

$$
\begin{aligned}
\lim_{v \to s} \lim_{u \to t} K_{01}(s,t,u,v) &= \frac{\rho(s,t)}{\pi} \frac{(s-1)\,(s-1)}{\left((s-1)^2 + (1-s)^2\right)^2} \\
&= \frac{\rho(s,t)}{\pi} \frac{1}{4(s-1)^2}
\end{aligned}
\tag{B.4}
$$

So, as we approach the corner ($s = 1$), we have a pole of order two. This singularity could be canceled by multiplying by zero twice, as the Jacobi polynomials used in this thesis do.

Bibliography

[BAUM 89] Daniel R. Baum, Holly E. Rushmeier, and James M. Winget, "Improved Radiosity Solutions Through the Use of Analytically Determined Form-Factors", *Computer Graphics*, 23(3), pp. 325-334, 1989.

[CAMP 90] A. T. Campbell, III and Donald S. Fussell, "Adaptive Mesh Generation for Global Diffuse Illumination", *Computer Graphics*, 24(4), pp. 155-164, 1990.

[COHE 85] M. Cohen and Donald P. Greenberg, "The Hemi-Cube: A Radiosity Solution For Complex Environments", *Computer Graphics*, 19(3), 1985, pp. 31-40.

[COHE 88] M. Cohen, S. Chen, John R. Wallace, Donald P. Greenberg, "A Progressive Refinement Approach to Fast Radiosity Image Generation", *Computer Graphics*, 22(4), 1988, pp. 75-84.

[COHE 90] M. Cohen and John R. Wallace, "Basic Radiosity Formulation for Diffuse Reflections", in *SIGGRAPH '90 Radiosity Course Notes*, 1990.

[CRC 59] Charles D. Hodgman, ed., *C.R.C. Standard Mathematical Tables–Twelfth Edition*, Chemical Rubber Publishing Company, USA, 1959.

[DAVI 63] Philip J. Davis, *Interpolation and Approximation*, Blaisdell, New York, 1963.

[DELV 81] L. M. Delves and T. L. Freeman, *Analysis of Global Expansion Methods: Weakly Asymptotically Diagonal Systems*, Academic Press, Inc., New York, 1981.

[DELV 85] L. M. Delves and J. L. Mohamed, *Computational Methods for Integral Equations*, Cambridge University Press, New York, 1985.

[DORS 91] Julie O'B. Dorsey, Francois X. Sillion, Donald P. Greenberg, "Design and Simulation of Opera Lighting and Projection Effects", *Computer Graphics*, 25(4), 1991, pp. 41-50.

[GORA 84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile, "Modeling the Interaction of Light Between Diffuse Surfaces", *Computer Graphics*, 18(3), July 1984, pp. 213-222.

[GORA 85] Cindy M. Goral, *A Model for the Interaction of Light Between Diffuse Surfaces*, Masters Thesis, Cornell University, Ithaca NY, January 1985.

[GLAS 89] Andrew Glassner, ed., *An Introduction to Ray Tracing*, Academic Press, Inc., New York, 1989.

[GRAD 65] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, 4th edition, Academic Press, Inc., New York, 1965.

[HANR 91] Pat Hanrahan, David Salzman, and Larry Aupperle, "A Rapid Hierarchical Radiosity Algorithm", *Computer Graphics*, 25(4), pp. 197-206, 1991.

[HECK 91a] Paul S. Heckbert, *Simulating Global Illumination Using Adaptive Meshing*, Report No. UCB/CSD 91/636, University of California, Berkeley, 1991.

[HECK 91b] Paul S. Heckbert and James M. Winget, *Finite Element Methods for Global Illumination*, Report No. UCB/CSD 91/643, University of California, Berkeley, 1991.

[HOTT 54] Hoyt C. Hottel, "Radiant-Heat Transmission", in William H. McAdams, *Heat Transmission*, McGraw-Hill Book Company, Inc., New York, 1954.

[KAJI 86] James T. Kajiya, "The Rendering Equation", *Computer Graphics*, 20(4), 1986, pp. 143-150.

[KANW 71] Ram P. Kanwal, *Linear Integral Equations; Theory and Technique*, Academic Press, Inc., New York, 1971.

[LISC 92] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg, "A Discontinuity Meshing Algorithm for Accurate Radiosity", IEEE CG&A, 1992, to appear.

[NISH 85] Tomoyuki Nishita and Eihachiro Nakamae, "Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection", *Computer Graphics*, 19(3), 1985, pp. 23-30.

[PENT 89] Alex Pentland and J. Williams, "Good Vibrations: Modal Dynamics for Graphics and Animation", *Computer Graphics*, 23(4), 1989, pp. 215-222.

[PRES 88] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, Cambridge University Press, New York, 1988.

[STAR 90] *Starbase Radiosity and Ray Tracing Programmer's Manual*, Hewlett Packard Co., USA, 1990.

[SILL 91] Francois Sillion, James Arvo, Steve Westin, and Donald P. Greenberg, "A Global Illumination Solution for General Reflectance Distributions", *Computer Graphics*, 25(4), July 1991, pp. 187-196.

[STOE 80] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

[STRO 71] A. H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

[SPAR 60] E. M. Sparrow, "Application of Variational Methods to Radiation Heat-Transfer Calculations", *Journal of Heat Transfer*, November 1960, pp. 375-380.

[SPAR 78] E. M. Sparrow and R. D. Cess, *Radiation Heat Transfer— Augmented Edition*, Hemisphere Publishing Corp., Washington, 1978.

[TAMP 91] Filippo Tampieri and Dani Lischinski, "The Constant Radiosity Assumption Syndrome", in the Proceedings of the Second Eurographics Workshop on Rendering, Barcelona, 1991.

[USIS 60] C. M. Usiskin and R. Siegel, "Thermal Radiation From a Cylindrical Enclosure With Specified Wall Heat Flux", *Journal of Heat Transfer*, pp. 369-374, November, 1960

[WALL 89] John R. Wallace, Kells A. Elmquist, Eric A. Haines, "A Ray Tracing Algorithm for Progressive Radiosity", *Computer Graphics*, 23(3), 1989, pp. 315-324.