Radiosity in Flatland*

Paul S. Heckbert

Department of Technical Mathematics & Informatics, Delft University of Technology,
Julianalaan 132, 2628 BL Delft, Netherlands

Abstract

The radiosity method for the simulation of interreflection of light between diffuse surfaces is such a common image synthesis technique that its derivation is worthy of study. We here examine the radiosity method in a two dimensional, *flatland* world. It is shown that the radiosity method is a simple finite element method for the solution of the integral equation governing global illumination. These two-dimensional studies help explain the radiosity method in general and suggest a number of improvements to existing algorithms. In particular, radiosity solutions can be improved using a priori *discontinuity meshing*, placing mesh boundaries on discontinuities such as shadow edges. When discontinuity meshing is used along with piecewise-linear approximations instead of the current piecewise-constant approximations, the accuracy of radiosity simulations can be greatly increased.

Keywords: integral equation, adaptive mesh, finite element method, discontinuity, shadow, global illumination, diffuse interreflection, thermal radiation.

1 Introduction

One of the most challenging tasks of image synthesis in computer graphics is the accurate and efficient simulation of *global illumination* effects: the illumination of surfaces in a scene by other surfaces. Early rendering programs used a *local illumination* model, assuming that surfaces could be shaded independently, typically using a finite set of point light sources. Global illumination models, on the other hand, recognize the true interdependency of the shading problem: the radiance of a surface point is determined by the radiance of all the surfaces visible from that point.

Global illumination simulation is relevant to a number of scientific and engineering fields: lighting design in architecture, shape-from-shading in computer vision, neutron transport in physics, and thermal radiation in mechanical engineering.

The visual artifacts of global illumination are familiar in the everyday world, even if they are sometimes subtle: we see *penumbras* or soft shadows from area light sources, color bleeding between surfaces, and indirect lighting.

We define several classes of materials: *diffuse* materials have a matte surface that appears equally bright from all directions, *specular* materials have a glossy surface, showing strong highlights, and *participating media* are foggy or translucent materials that emit, absorb, or scatter light volumetrically.

In full generality, radiance is a function of three-dimensional position, two-dimensional direction, wavelength, time, phase, and polarization. As is conventional in computer graphics and thermal radiation, we drop most of these dependencies, assuming that media are non-participating, surfaces are opaque and diffuse, light is incoherent (having all phases), and the geometry and material properties

* *Computer Graphics Forum* (Proc. Eurographics '92), 11(3), Sept. 1992, pp. 181–192, 464.

are static. Furthermore, we ignore polarization and assume that surfaces are *gray*, with wavelength-independent properties within each of the wavelength bands of interest. We will typically divide the visible spectrum into the three bands: red, green, and blue. Given these assumptions, radiance is a function of the two surface parameters only.

Previous Work

Existing techniques for simulating global illumination generally fall into two classes: ray tracing methods and radiosity methods. In general, ray tracing is best at simulating specular surfaces and radiosity is best at simulating diffuse surfaces.

Ray tracing simulates light transport by tracing the paths of photons through the scene in one of two directions: either from the lights into the scene or from the eye into the scene. With the addition of Monte Carlo techniques, it is possible to simulate diffuse interreflection [Kajiya86].

Radiosity methods were first developed for the simulation of thermal radiation in mechanical engineering [Sparrow63], and were later extended and optimized for the simulation of complex scenes in computer graphics [Goral et al. 84,Nishita-Nakamae85,Cohen-Greenberg85,Cohen et al. 88]. The radiosity method subdivides the scene into *elements* (subsurfaces) and creates a system of equations whose solution is the *radiosity* (the sum of emitted and reflected radiance, integrated over a hemisphere) of each element.

The simplest radiosity methods perform *meshing*, the subdivision of surfaces into elements, in a simple, uniform manner, but this results in jaggy shadow boundaries and other artifacts. Better meshes can be created by user intervention, by *a posteriori* methods that refine the mesh after solution [Cohen et al. 86], or by *a priori* techniques that create the mesh before solution [Campbell-Fussell90,Baum et al. 91,Heckbert-Winget91,Heckbert91,Lischinski et al. 91,Campbell91]. Another improvement is to vary the mesh resolution according to the range of the interaction: nearby elements need a fine mesh but a coarse mesh suffices for distant elements [Hanrahan et al. 91].

Before discussing algorithms, we will examine the phenomena and mathematical description of radiosity in some depth.

2 Understanding Radiosity in Flatland

To simplify the study of radiosity algorithms, we restrict our attention to *radiosity in flatland*: a two-dimensional world consisting of opaque objects with diffuse emittance and reflectance. A flatland world is equivalent to a three-dimensional world where all objects have infinite extent along one direction. We will further restrict ourselves to a scene with line segments and closed polygonal shapes and diffuse light sources.

In this flatland world the global illumination problem reduces to the determination of the radiosity at each point on the edges in the scene. A flatland scene is shown in figure 1. Instead of shading two-dimensional surfaces and computing two-dimensional integrals, as we do in 3-D graphics, in flatland graphics we shade one-dimensional edges and compute one-dimensional integrals. Relative to three-dimensional worlds, in flatland one finds that analytic results are easier to come by, algorithms are easier to debug, brute force techniques such as Monte Carlo integration converge faster, and it is possible to compute approximate solutions so accurate that they can be regarded as exact. This facilitates the use of quantitative error metrics for the objective comparison of algorithms.

Integral Equation for Radiosity in Flatland, Part 1

Suppose the scene consists of m edges with total length L. Line segments can be regarded as 2-sided polygons, shaded on both sides. Each edge is parameterized by arc length, and we concatenate the domains of these functions in arbitrary order to create a single function $b(s)$ parameterized by s, running from 0 to L. This concatenation introduces explicit discontinuities at edge endpoints. In the

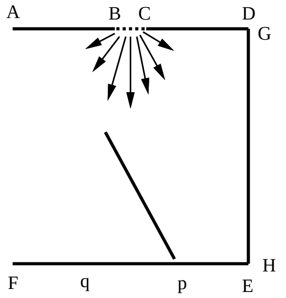

Figure 1: *Flatland test scene. All edges are reflective except light source BC, and angled edge, which is black. The angled obstacle causes a sharp penumbra at p and a gradual one at q.*

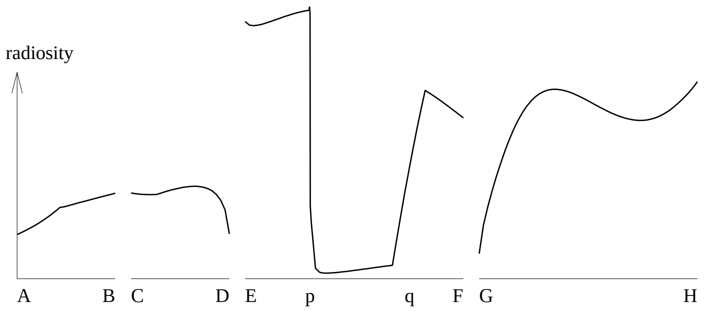

Figure 2: *Radiosity as a function of arc length along the non-black edges of test scene. Note the sharp shadow edge at p and the gradual one at q.*

following, when we refer to "the point s", what we mean is the point with parameter value s. The radiosity function can now be plotted as a piecewise-continuous function, as shown in figure 2.

The optical properties in the scene are described by several functions. The (hemi- or semi-circular) *reflectance* ρ_h is the fraction of incident radiance reflected in any direction. Reflectance is a unitless quantity between 0 (black) and 1 (perfect white). *Radiant emitted flux density* e is the power per unit distance emitted by an edge (called "emissive power" in the thermal radiation literature). Edges are reflectors if $\rho_h > 0$, and light sources if $e > 0$, and occasionally both. *Radiance* is the power per unit distance per unit angle propagating in a given direction (often called "intensity"), and *radiosity* b is the emitted or reflected power per unit distance emanating in any direction (radiance integrated over a semicircle). Each of these variables is potentially a function of arc length, but for simplicity, we assume that each edge has constant reflectance and emitted flux density. Note that the units of the above quantities differ from the standard definitions because we are operating in a 2-D world.

We use the following geometric variables (figure 3): position is parameterized by arc length variables s and s', the outgoing angle at s' is θ'_o, the incoming angle at s is θ_i, the distance between points s and s' is r. The visibility function between points s and s' is v: it equals 1 when s and s' are inter-visible and 0 when they are occluded. In 3-D, the solid angle subtended by an area at distance r is proportional to $1/r^2$, but in flatland, the angle subtended by a differential element of length ds' is proportional to $1/r$.

Radiosity consists of emitted plus reflected light, and the latter could come from any of the other edges in the scene, so the interreflection in diffuse, opaque, flatland scenes is described by the following *integral equation* [Özisik73,Heckbert91]:

$$b(s) = e(s) + \rho_h(s) \int_0^L ds' \frac{\cos\theta_i \cos\theta'_o}{2r} \, v \, b(s')$$

Integral equations similar to this have appeared in the thermal radiation literature [Özisik73], in computer vision [Koenderink-van Doorn83], where it has been called the *mutual illumination equation*, and in computer graphics, where it has been called the *rendering equation* [Kajiya86].

We digress briefly to discuss integral equations in general.

Integral Equations

Integral equations are analogous to differential equations except that they involve integrals of functions instead of derivatives. The class of integral equation of interest in this work is the *Fredholm integral equation of the second kind* [Jerri85,Delves-Mohamed85], with general form

$$b(s) = e(s) + \int_{\alpha}^{\beta} dt\, \kappa(s,t) b(t)$$

where e and the *kernel* κ are given and b is to be determined. The above equation is abbreviated as $b = e + \mathcal{K}b$, where $\mathcal{K}$ denotes the integral operator which when applied to a function b, yields a function

$$(\mathcal{K}b)(s) = \int_{\alpha}^{\beta} dt\, \kappa(s,t) b(t)$$

Formally, our integral equation $b = e + \mathcal{K}b$ can be rewritten $(\mathcal{I} - \mathcal{K})b = e$, where $\mathcal{I}$ is the identity operator, and the solution can be obtained by inverting the operator $\mathcal{I} - \mathcal{K}$:

$$b = (\mathcal{I} - \mathcal{K})^{-1} e = e + \mathcal{K}e + \mathcal{K}^2 e + \mathcal{K}^3 e + \cdots$$

where $\mathcal{K}^i$ denotes i successive applications of the integral operator $\mathcal{K}$. This *Neumann series* converges if the norm of the integral operator is less than 1 [Delves-Mohamed85].

Integral Equation for Radiosity in Flatland, Part 2

We can now apply integral equation techniques to the understanding of radiosity.

For flatland radiosity, the integral equation $b = e + \mathcal{K}b$ has integral operator $\mathcal{K}$ with kernel

$$\kappa(s, s') = \rho_h(s) \frac{\cos\theta_i \cos\theta'_o}{2r} v$$

The variables θ_i, θ'_o, v, and r are functions of s and s'.

For global illumination, physical constraints limit reflectance to be less than 1, implying that $\mathcal{K}$ will have a norm of less than 1, so the Neumann series is guaranteed to converge. The Neumann series here has a simple physical interpretation: the ith term $(\mathcal{K}^i e)(s)$ is the light that reaches the point s after exactly i 'hops' [Kajiya86]. The ith partial sum of the Neumann series $b^{(i)} = e + \mathcal{K}e + \mathcal{K}^2 e + \cdots + \mathcal{K}^i e$ is the light that reaches point s in i hops or fewer. Early illumination models (what Kajiya called the 'Utah approximation') simulated only direct illumination $b^{(1)} = e + \mathcal{K}e$; global illumination attempts to compute $b^{(\infty)}$ (though not necessarily by summing the series).

In flatland, the kernel is a bivariate function. Since we have abutted the domains of the edges in the scene, the kernel's domain consists of rectangular blocks corresponding to pairs of edges. The kernel is smooth almost everywhere, with discontinuities at the boundaries of these blocks, and along occlusion curves that trace out hyperbolas in ss' space (see figure 6). Note also that the kernel is singular at reflex corners in the scene (where touching surfaces face each other), because $\kappa \to \infty$ as $r \to 0$.

Discontinuities

We can derive many of the qualitative properties of the exact solution function $b(s)$ from the properties of the kernel and the geometry of the scene, even without solution algorithms.

We call a discontinuity in the kth derivative of a function a D^k discontinuity. A function thus has a D^k discontinuity at a point if it is C^{k-1} there but not C^k. Discontinuities in the radiosity can result from discontinuities in emitted flux density, reflectance, normal vector, or visibility.

In figure 4, we see that linear light sources cast hard shadows (causing D^0 discontinuities) when objects touch, and soft shadows (causing D^1 discontinuities) when objects do not touch. Point light

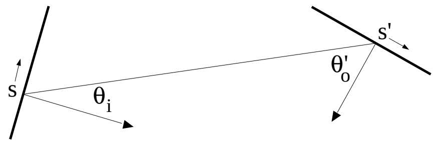

Figure 3: *Visibility geometry for edge points with parameter values s and s'.*

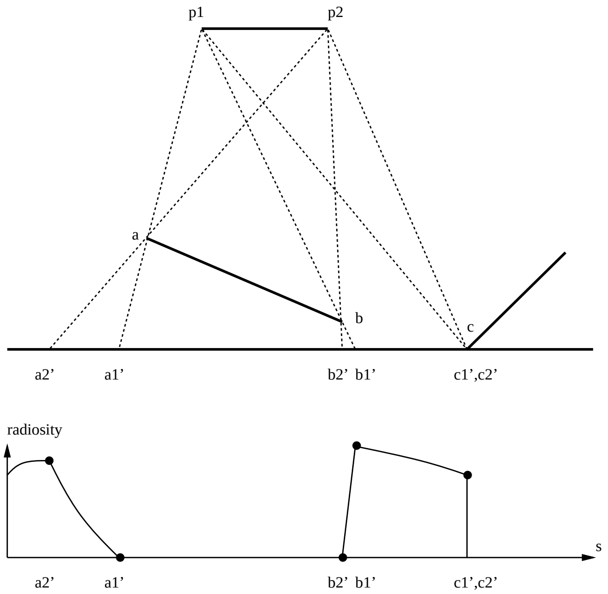

Figure 4: *D^1 and D^0 discontinuities caused by linear light source at top illuminating an occluded edge at bottom. D^1 discontinuities delimit the penumbras at a2', a1', b2', and b1'. D^0 discontinuity at the hard shadow edge c1'.*

sources, by contrast, always cause hard (D^0) shadows. The pattern generalizes to higher degree discontinuities according to the following law:

Discontinuity Propagation Law: **If there is a D^k discontinuity at point s' on one edge, then it will cause D^k discontinuities at all touching points visible to it, and D^{k+1} discontinuities at the projection of all of the silhouette points from its point of view.** A touching point is a point at which two edges touch or intersect non-tangentially. If there are no occlusions in a scene then there are no discontinuities due to changes in visibility. It can easily be proven that an upper bound on the number of discontinuities of degree i in a scene of m edges is $(2m)^{i+1}$ [Heckbert91]. In practice, the number is usually far smaller than this upper bound because of parallel edges and occlusion. One consequence of the above is that the total number of discontinuities of various degrees in the radiosity function can be infinite.

Now that we have examined some of the properties of radiosity functions, we turn to approximation algorithms.

3 Algorithms for Radiosity in Flatland

There are a variety of finite element methods for solving the integral equation governing radiosity [Heckbert-Winget91]. Figure 5 shows some of the available options: one much choose a mesh and basis functions, constraint and integration methods to reduce the problem to a linear system of equations, a system solution method, and reconstruction technique.

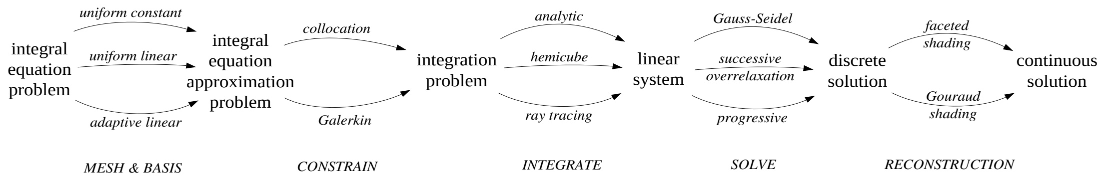

Figure 5: *Steps of finite element simulation of radiosity.*

Numerical Solution of Integral Equations

The *collocation method* is one of the simplest numerical techniques for solving integral equations [Delves-Mohamed85]. First, the exact solution $b(s)$ is approximated by a linear combination of basis functions:

$$\hat{b}(s) = \sum_{i=1}^{n} b_i W_i(s)$$

where b_i are the unknown coefficients and W_i are the chosen basis functions. The simplest basis functions are the box and hat (triangle) functions, which give rise to piecewise constant and piecewise linear approximations, respectively. The pieces of the approximation are respectively called *constant elements* and *linear elements*. Such approximations are the foundation of the finite element method [Becker et al. 81].

Of course, it is an approximation to assume that a function in this function space will solve the integral equation. To minimize the error $\hat{b} - b$, the collocation method constrains the *residual* $r = \hat{b} - \mathcal{K}\hat{b} - e$ to be zero at selected points s'_i, typically the midpoints of the intervals. A system of linear equations results, $(\mathbf{M} - \mathbf{K})\mathbf{b} = \mathbf{e}$, where $\mathbf{M}$ and $\mathbf{K}$ are $n \times n$ matrices and $\mathbf{b}$ and $\mathbf{e}$ are n-element column vectors:

$$M_{ij} = W_j(s'_i)$$

$$K_{ij} = (\mathcal{K}W_j)(s'_i) = \int_{\alpha}^{\beta} dt\, \kappa(s'_i, t) W_j(t)$$

$$e_i = e(s'_i)$$

For constant and linear elements, typically $\mathbf{M} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. More accurate approximations are possible with the more expensive Galerkin approximation method [Delves-Mohamed85,Heckbert-Winget91].

Radiosity as an Approximation Method

The radiosity algorithm, as it is used in the computer graphics field, can be derived and explained quite clearly in the context of these numerical methods. The radiosity method typically assumes, for the interreflection simulation, that each polygonal element has a constant radiosity [Tampieri-Lischinski91]. This corresponds to a mesh with constant elements.

Next, the radiosity method typically computes point-to-area *form factors* from the centers of the elements. (Variants of this method are possible, which compute form factors at polygon vertices instead of the centers, or which compute area-to-area form factors, but point-to-area form factors are the most commonly used.) The point-to-interval form factor in flatland between elements i and j is

$$F_{ij} = \int_{elem\ j} ds'\, \rho_h(s') \frac{\cos\theta_i \cos\theta'_o}{2r} v \tag{1}$$

where s' is the midpoint of element i. One then constructs an $n \times n$ matrix $A_{ij} = I_{ij} - \rho_i F_{ij}$ and solves the system of equations $\mathbf{Ab} = \mathbf{e}$ for the radiosity vector $\mathbf{b}$. But notice that this is precisely the collocation method! In the collocation method we construct the matrix $\mathbf{K}$ which is related to the form

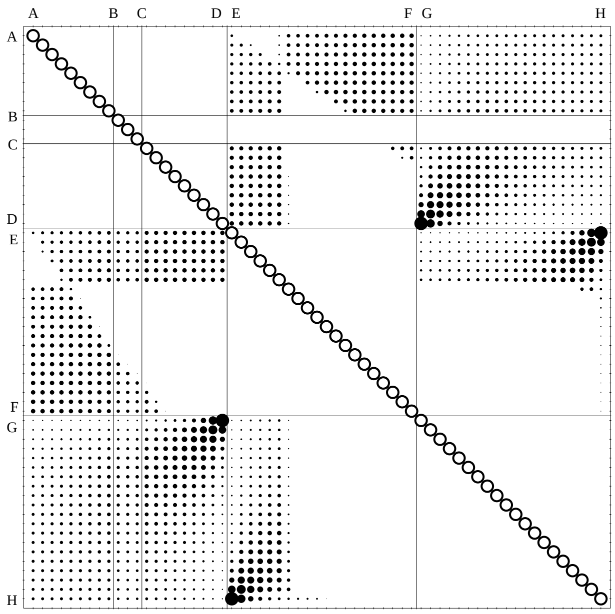

Figure 6: *Nonzero elements of the matrix* $\mathbf{I} - \mathbf{K}$ *for our test scene, with* $n = 61$ *equations. Diagonal elements (open circles) are all 1; off-diagonal elements are shown with area proportional to magnitude. White areas are zeros. Large dots indicate large form factors at reflex corners.*

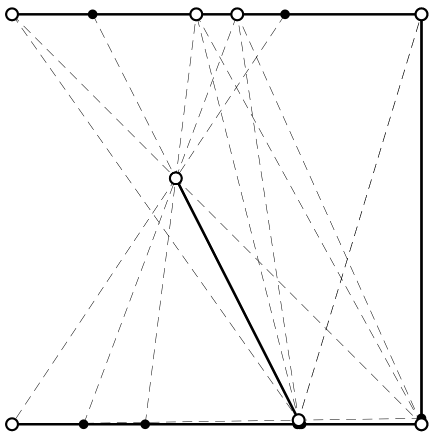

Figure 7: *Critical lines (dashed) and critical points (filled circles) for test scene in flatland. Open circles mark edge endpoints.*

factors by $K_{ij} = \rho_i F_{ij}$, and we solve the system of equations $(\mathbf{I} - \mathbf{K})\mathbf{b} = \mathbf{e}$. This is the same system of equations as is used in the radiosity method, where $\mathbf{A} = \mathbf{I} - \mathbf{K}$.

In radiosity programs, form factors are typically computed using a hemicube [Cohen-Greenberg85] or ray tracing [Wallace et al. 89]. These can be viewed as numerical methods for approximating the integrals in (1), using visible surface algorithms from computer graphics to point-sample the visibility function v. In flatland, it is possible to compute form factors analytically [Heckbert91]. This does not seem to be possible in 3-D.

In the radiosity method, the system of equations is usually solved either by constructing the matrices explicitly and solving with Gauss-Seidel iteration [Golub-Van Loan89], or with the *progressive radiosity algorithm* that interleaves form factor computation and system solving steps [Cohen et al. 88]. The latter method can be viewed as an iterative method for solving linear systems that computes one column of the matrix at a time.

Figure 6 shows the diagonally dominant, non-symmetric, moderately sparse character of radiosity matrices. They are moderately sparse (having many zeros) for most scenes of interest since each surface can "see" only a small fraction of the other surfaces.

Meshing

Early research in radiosity focused on the computation of form factors and efficient solution of the system of equations, but the issue of meshing or discretization of surfaces was little discussed; until recently it has remained a black art and a manual process for the most part [Baum et al. 91].

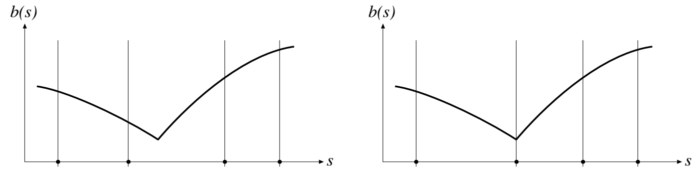

Figure 8: *Left: Mesh does not resolve the D^1 discontinuity (dots are element nodes). Right: Mesh resolves discontinuity.*

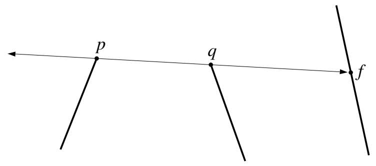

Figure 9: *f is a critical point caused by the critical line through endpoints p and q.*

Discontinuity meshing is an approach to meshing that attempts to accurately resolve the most significant discontinuities in the solution by optimal positioning of element boundaries. For the 1-D elements we use in flatland, the boundaries are the nodes.

When discontinuities in the true solution fall on element boundaries, the mesh is said to *resolve* the discontinuity (see figure 8). For the best approximation, all discontinuities with degree at or below the degree of the elements should be resolved. Thus, when using constant elements, all D^0 discontinuities should be resolved, and when using linear elements, all D^0 and D^1 discontinuities should be resolved. It is not fruitful to resolve discontinuities of degree greater than the element degree because the errors so caused are swamped by the discontinuities introduced at element boundaries.

Discontinuity Meshing Algorithm

In flatland, discontinuity meshing can be done quite easily. Possible sites of discontinuities can be predicted *a priori* (before the radiosity solution). We call an edge point which is collinear with two edge endpoints visible from it a D^1 *critical point* and the line along which they lie is a *critical line* (figure 9).

As before, assume a scene consisting of m opaque line segments. Since the lowest degree discontinuities are generally the most significant, we first resolve D^0 discontinuities, then D^1 discontinuities, and so on up to the degree of the elements being used.

The most important discontinuities to resolve are D^0. D^0 discontinuities come from intersecting edges or from the projection of silhouette points from point light sources. Intersecting edges can be found in $O(m \log m)$ time, [Preparata-Shamos85]. These intersection points are marked as D^0 critical points (points at which a D^0 discontinuity could occur). If constant elements are being used, then only D^0 discontinuities need be found, and we can stop here.

If linear or higher degree elements are being used, then D^1 discontinuities are found. D^1 discontinuities occur at critical points where there is a remote change in visibility, i.e. where an edge endpoint becomes occluded. There are $O(m^2)$ D^1 critical points in a scene.

All D^1 critical points can be found as follows:

```
for each node p
    for each node q
        if no edge intersects line segment pq then {
            e = trace_ray(p, p-q)
            f = trace_ray(q, q-p)
            if e<>NULL then critical_point(e)
            if f<>NULL then critical_point(f)
        }
```

The routine `trace_ray(p, d)` traces a ray from point `p` in direction `d` and returns the first edge point hit, if any. The above algorithm, implemented straightforwardly, has $O(m^3)$ time cost. Alternatively, visibility can be determined by applying an $O(m \log m)$ radial sweepline perspective visibility algorithm [Edelsbrunner et al. 83] at both endpoints of each edge, yielding an $O(m^2 \log m)$ algorithm. Figure 7 shows the critical points of our test scene.

Implementation

A program has been implemented to simulate radiosity in flatland which allows a choice of element degree and meshing technique. The program simulates diffuse interreflection among non-intersecting, simple polygons, with colored, diffuse reflecting and/or emitting edges. To simulate colors, samples at red, green, and blue wavelengths are used. The systems of linear equations are solved independently for each band. The program solves for radiosities using a collocation formulation with analytic form factors (no numerical integration) and either constant or linear elements [Heckbert91]. Either uniform (equispaced) or discontinuity meshing can be used. The latter places element nodes at all D^1 critical points and then creates a uniform mesh inbetween. Critical points on the m edges are found with the $O(m^3)$ ray tracing algorithm described above. Form factors are calculated with an object space visible edge algorithm using an $O(m^2)$ radial sweepline technique. For a scene discretized into n elements, the total cost of computing the matrices is $O(nm^2 + \alpha n^2)$, where α is the fraction of nonzero elements in the sparse matrix. The systems of equations are solved with successive overrelaxation (a faster variant of Gauss-Seidel) using a default overrelaxation parameter of $\omega = 1.4$. The sparse matrix data structure uses $6\alpha n^2$ bytes for each of the three (R, G, B) components. For the scenes tested, matrix density α typically ranged between 10% and 40%.

Three display views are supported, an interactive flatland view, which is a top view of the scene, a graph of the red, green, and blue solution curves as a function of arc length, and a schematic of the radiosity matrix. The screen during interaction is shown in color figure 13. The program runs on a Silicon Graphics or Hewlett Packard workstation, and is able to re-solve and redisplay a scene consisting of 100 elements in about one second, while scenes containing 1000 elements require several minutes. An intuitive understanding of the discontinuities in the radiosity function and of the errors of poor meshes is easily built up by interactively moving, creating, and deleting objects in the flatland view.

Results

Test runs have been performed to compare the speed and accuracy of six different solution methods resulting from a combination of one of the two meshing techniques, uniform meshing or discontinuity meshing, and one of three element types.

The three element types used are constant elements (box basis), linear elements (hat basis), and what we call *Gouraud elements*. Gouraud elements are constant elements for formulation and solution of the problem, followed by linear interpolation between collocation points for the 'display' step. In other words, the Gouraud approximation is computed as a post-process to the piecewise-constant approximation, by interpolation and extrapolation between neighboring elements. We call them Gouraud elements by analogy with the computer graphics shading technique [Foley et al. 90]. Gouraud elements are the most commonly used approximation technique in existing radiosity algorithms.

To measure error objectively, we use the following error norm. If the exact radiosity function is b and the approximation is $\hat{b}$, then the error is defined to be $||\hat{b} - b||_2/||b||_2$, where the L_2 norm of a function over an interval is defined as:

$$||f||_2 = \left(\int_\alpha^\beta dx\, |f(x)|^2 \right)^{1/2}$$

Since analytic solutions are not known for radiosity problems involving interreflection, the "exact" solution b is taken to be the approximate solution resulting from an extremely fine mesh. Both uniform and discontinuity meshing converge very close to this solution, verifying that it is a valid reference.

Figure 11 shows the convergence rate of the six methods for various mesh resolutions on the test scene of figure 1. As we might expect, constant and Gouraud elements show nearly the same (low)

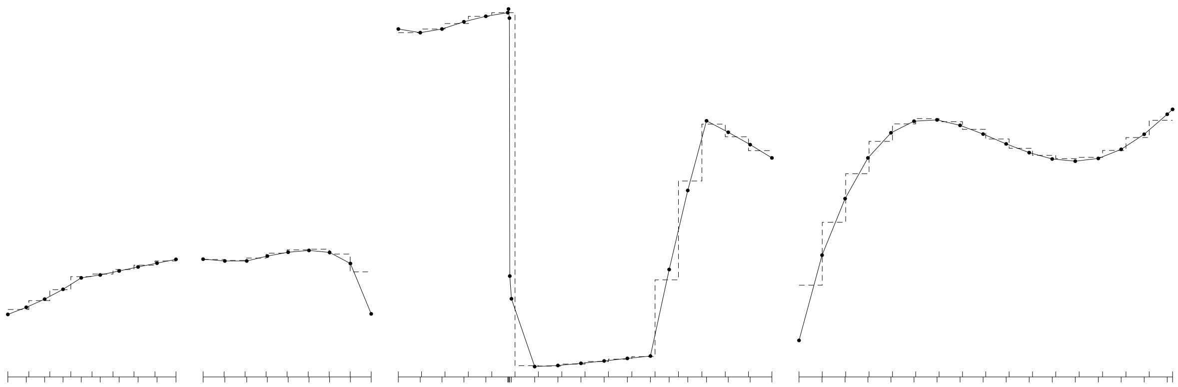

Figure 10: *Radiosity as a function of arc length. Dashed: approximation using uniform mesh and constant elements. Solid: approximation using discontinuity mesh and linear elements. Note that the latter resolves discontinuities, while the former does not. Tick marks above and below line at bottom show uniform and discontinuity meshes, respectively.*

accuracy, convergence with decreasing element size is faster with linear elements than with constant or Gouraud elements, and convergence rate is highly dependent on discontinuities. Without discontinuity meshing, solution with any of the three element types converges slowly, but with discontinuity meshing, convergence can be relatively fast. Discontinuity meshing with linear elements gives results over 10 times more accurate on fine meshes, in these experiments.

Figure 12 shows that the use of discontinuity meshing and linear elements are cost-effective; for this experiment they give the best accuracy for a given amount of total CPU time, and the fastest results at a given accuracy. Qualitatively similar plots result for other scenes tested.

Consider a specific example. When the test scene of figure 1 was simulated with discontinuity meshing with $n = 91$ equations, it required 73K bytes of memory and 1.3 seconds of total CPU time. To achieve the same accuracy with uniform meshing required $n = 775$ equations, 4.5M bytes of memory, and 74 seconds total. Discontinuity meshing, for this test case, gave results of the same quality as uniform meshing using about 1/60th the time and 1/60th the memory.

For a simple scene such as this, about 80% of the CPU time is spent computing the radiosity matrix, 5% on discontinuity meshing, and 15% on system solving. This balance could change depending on the relative number of discontinuities and elements, and system sparseness.

Conclusions

Integral equations provide a concise statement of the physics of global illumination. Global illumination can be studied without reference to integral equations, but such an approach is awkward: studying global illumination without integral equations is like studying classical dynamics without differential equations.

A fairly thorough study of radiosity in a diffuse, two-dimensional "flatland" world was made in order to better understand these integral equations and their solution functions. The solution functions and the kernel are easy to visualize in flatland. The flatland radiosity program implemented here could be generalized to simulate specular scenes, yielding a helpful global illumination visualization and teaching tool.

Radiosity solution functions have discontinuities caused by touching surfaces, intersections, creases, or occlusion. The most significant of these discontinuities are perceived as shadows.

It was demonstrated that the classic radiosity method constitutes one of the simplest finite element methods for approximate solution of the integral equation for global illumination in a diffuse scene.

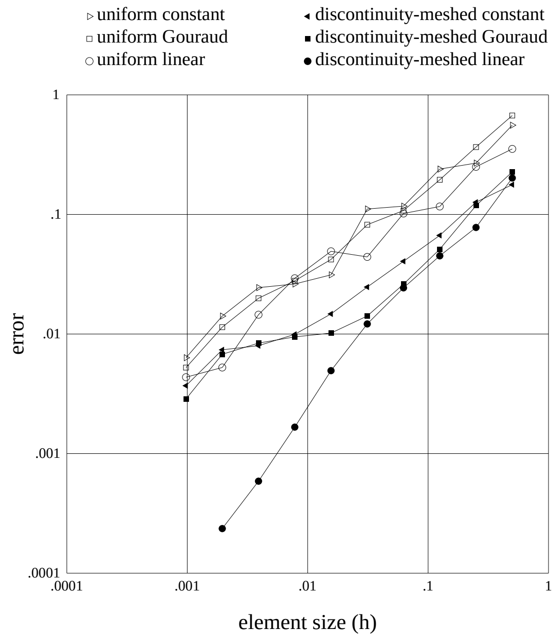

Figure 11: *Log-log plot of element size vs. error for six different solution methods. Note that discontinuity meshing with linear elements is far more accurate than the other methods, and that Gouraud elements are little better than constant elements.*

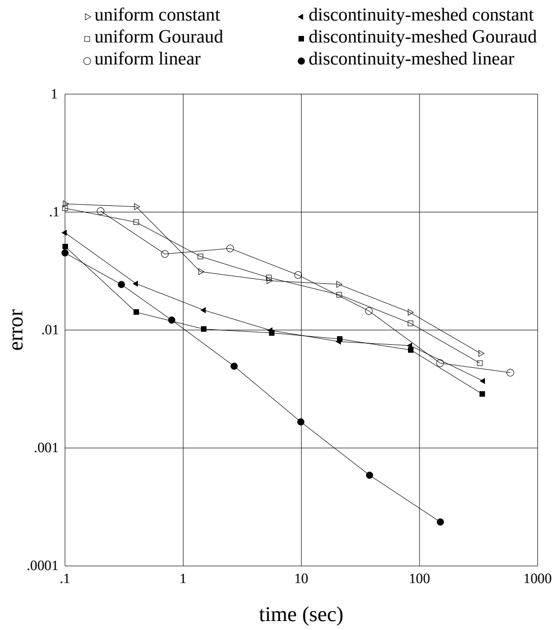

Figure 12: *Log-log plot of total CPU time (meshing, matrix computation, solution, and I/O) vs. error for six different solution methods. In this experiment, discontinuity meshing with linear elements is the most cost effective of the techniques tested except for very fast, crude simulations.*

Classic radiosity typically makes use of constant elements, a uniform mesh, collocation methods for discretization, a hemicube for numerical integration, and Gauss-Seidel or progressive iterative methods for solving linear systems. It was shown that the use of constant elements during problem formulation and solution followed by Gouraud interpolation for display gives results that are objectively no more accurate than the use of constant elements throughout (though they may look better).

We have explored several improvements on these approaches, demonstrating that both higher degree elements and a priori adaptive meshes can improve the accuracy of 2-D radiosity algorithms considerably. The most prominent discontinuities in the solution can be predicted using visibility methods from computational geometry. With this discontinuity meshing, mesh boundaries are placed on significant discontinuities. Using linear elements and discontinuity meshing, one experiment in flatland achieved a 60-fold reduction in CPU time and memory use relative to standard methods. The discontinuity meshing ideas developed here have recently been generalized to 3-D scenes [Heckbert92].

This research has suggested a number of ideas for future research: one would like to find the combination of approximations and algorithms that gives the most accurate final result in the least time. What combination of element degree (constant, linear, ...), mesh (a priori adaptive or a posteriori adaptive), discretization method (collocation or Galerkin), integration method (analytic, hemicube, Gaussian, etc.), and system solution method (successive overrelaxation, multigrid, progressive radiosity, hierarchical radiosity, etc.) are best?

4 Acknowledgements

Jim Winget and Keith Miller contributed many ideas to my global illumination work at UC Berkeley, where most of the work described here was done. In Delft, Erik Jansen has hosted my postdoctoral fellowship, and Aadjan van der Helm has given help with X windows and Latex.

References

[Baum et al. 91] Daniel R. Baum, Stephen Mann, Kevin P. Smith, and James M. Winget. Making radiosity usable: Automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):51–60, July 1991.

[Becker et al. 81] Eric B. Becker, Graham F. Cary, and J. Tinsley Oden. *Finite Elements: An Introduction*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[Campbell91] A. T. Campbell, III. *Modeling Global Diffuse Illumination for Image Synthesis*. PhD thesis, CS Dept, University of Texas at Austin, Dec. 1991. Tech. Report TR-91-39.

[Campbell-Fussell90] A. T. Campbell, III and Donald S. Fussell. Adaptive mesh generation for global diffuse illumination. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4):155–164, Aug. 1990.

[Cohen et al. 86] Michael F. Cohen, Donald P. Greenberg, David S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, pages 26–35, Mar. 1986.

[Cohen et al. 88] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22(4):75–84, Aug. 1988.

[Cohen-Greenberg85] Michael F. Cohen and Donald P. Greenberg. The hemi-cube: A radiosity solution for complex environments. *Computer Graphics (SIGGRAPH '85 Proceedings)*, 19(3):31–40, July 1985.

[Delves-Mohamed85] L. M. Delves and J. L. Mohamed. *Computational methods for integral equations*. Cambridge University Press, Cambridge, U.K., 1985.

[Edelsbrunner et al. 83] Herbert Edelsbrunner, Mark H. Overmars, and Derick Wood. Graphics in flatland: A case study. *Advances in Computing Research*, 1:35–59, 1983.

[Foley et al. 90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice, 2nd ed.* Addison-Wesley, Reading MA, 1990.

[Golub-Van Loan89] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 1989.

[Goral et al. 84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):213–222, July 1984.

[Hanrahan et al. 91] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):197–206, July 1991.

[Heckbert91] Paul S. Heckbert. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, CS Division, UC Berkeley, June 1991. Tech. Report UCB/CSD 91/636.

[Heckbert92] Paul S. Heckbert. Discontinuity meshing for radiosity. In Alan Chalmers and Derek Paddon, editors, *Third Eurographics Workshop on Rendering*, pages 203–216, Bristol, UK, May 1992.

[Heckbert-Winget91] Paul S. Heckbert and James M. Winget. Finite element methods for global illumination. Technical report, CS Division, UC Berkeley, July 1991. UCB/CSD 91/643.

[Jerri85] Abdul J. Jerri. *Introduction to Integral Equations with Applications*. Dekker, New York, 1985.

[Kajiya86] James T. Kajiya. The rendering equation. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):143–150, Aug. 1986.

[Koenderink-van Doorn83] J. J. Koenderink and A. J. van Doorn. Geometrical modes as a general method to treat diffuse interreflections in radiometry. *J. Opt. Soc. Am.*, 73(6):843–850, June 1983.

[Lischinski et al. 91] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Improving sampling and reconstruction techniques for radiosity. Technical report, CS Dept., Cornell U., Aug. 1991. TR 91-1202.

[Nishita-Nakamae85] Tomoyuki Nishita and Eihachiro Nakamae. Continuous tone representation of 3-D objects taking account of shadows and interreflection. *Computer Graphics (SIGGRAPH '85 Proceedings)*, 19(3):23–30, July 1985.

[Özisik73] Necati M. Özisik. *Radiative Transfer and Interactions with Conduction and Convection.* Wiley, New York, 1973.

[Preparata-Shamos85] Franco Preparata and Michael I. Shamos. *Computational Geometry.* Springer Verlag, 1985.

[Sparrow63] Ephraim M. Sparrow. On the calculation of radiant interchange between surfaces. In Warren Ibele, editor, *Modern Developments in Heat Transfer*, New York, 1963. Academic Press.

[Tampieri-Lischinski91] Filippo Tampieri and Dani Lischinski. The constant radiosity assumption syndrome. In *Second Eurographics Workshop on Rendering*, Barcelona, Spain, May 1991.

[Wallace et al. 89] John R. Wallace, Kells A. Elmquist, and Eric A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics (SIGGRAPH '89 Proceedings)*, 23(3):315–324, July 1989.

Figure 13: *Screen of flatland radiosity program, with flatland view at top and radiosity function graph at bottom. Scene consists of a white emitter (at left), a brown reflector (at bottom) and a polygonal reflector with one red edge, one green edge, and all other edges white. Note color bleeding along the edge between the red and green neighbors. Graphs show red, green, and blue radiosity as a function of arc length along counterclockwise traversal of polygons.*