Galerkin Radiosity:
A Higher Order Solution Method for Global Illumination

Harold R. Zatz[1]
Cornell Program of Computer Graphics

Abstract

This paper presents an alternative radiosity formulation using piecewise smooth radiance functions that incorporates curved surfaces directly. Using the Galerkin integral equation technique as a mathematical foundation, surface radiance functions are approximated by polynomials. This model eliminates the need for *a posteriori* rendering interpolation, and allows the direct use of non-planar parametric surfaces. Convergence problems due to singularities in the radiosity kernel are analyzed and rectified, and sources of approximation error are examined. The incorporation of a shadow masking technique vastly reduces the need for meshing and associated storage space—accurate radiosity calculations can often be made with no meshing. The technique is demonstrated on traditional radiosity scenes, as well as environments with untessellated curved surfaces.

CR Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism; I.3.3 [**Computer Graphics**]: Picture/Image Generation.

Additional Keywords and Phrases: global illumination, radiosity, integral equations, Galerkin methods, curved surfaces, progressive refinement.

1 Introduction

The behavior of light interacting with a macroscopic environment is extremely complex. Despite considerable effort spent searching for a closed-form solution to global illumination problems [10, 22], it seems unlikely that such an approach will be found. To produce computer-generated pictures in a reasonable amount of time, approximations must be used. Typical approximation techniques include the use of direct lighting only, tessellation of the simulated environment into polygonal surfaces, constant or linear shading of surfaces, and sampling the intensity distribution at a limited number of points.

Goral *et al.* [7] introduced the conventional radiosity approximations to computer graphics, assuming surfaces have purely diffuse reflectance distributions, and that finite regions on these surfaces have locally constant radiosity values. Intensity variations across a surface are accounted for by meshing it into a large number of smaller pieces.

Although these assumptions are effective, recent research has demonstrated their limitations. Conventional radiosity techniques generally require that objects be flat or polygonal [1, 7, 3], even though Wallace has demonstrated [21] that radiosity transfers can be computed between non-planar surfaces. Generating images with accurately placed shadows involves a lengthy meshing process, whether surfaces are divided along arbitrary lines [15, 2, 9] or along actual lines of shadow discontinuity [13, 12].

In finite element analysis, it is often possible to trade off a large number of lower-order elements for a smaller number of higher-order elements. Sparrow [18] and Heckbert [10, 11] have successfully applied higher-order radiosity techniques to special-case geometries. Max and Allison [14] explored some of the difficulties of using a linear elements in more general radiosity meshes. In this paper we reformulate the radiosity equations with the goal of applying higher-order *Galerkin* techniques to more general environments, paying particular attention to the difficulties caused by singularities and shadow discontinuities. Benefits of this approach include the direct incorporation of curved surfaces into the solution technique, as well as a significant memory savings due to a drastic reduction of mesh size.

The Galerkin method does have its disadvantages; dealing with shadows and extremely bright light sources can be tricky, and computationally expensive singularities can appear in many places in a complex environment. However, the use of higher-order functions to replace meshing provides a different perspective on the difficulties of the global illumination problem, avoiding some of the difficulties of conventional methods.

2 Background

The radiosity model of global illumination is based on the principle of energy conservation. All light energy emitted within an enclosure is tracked as it reflects off surfaces within that environment, until it dissipates into heat. Conventional radiosity methods [1, 2, 3, 4, 7, 9, 15, 21] generally simplify the solution procedure by using the Constant Radiosity Assumption [20]—the primary assumption that radiosity values are constant over finite regions, and subsidiary assumptions that emittance, reflectivity, and surface normals are also constant over finite regions. Unfortunately, this constant, polygonal approach to the radiosity problem limits the solution accuracy. Conventional radiosity methods attempt to compensate by increasing the mesh density, assuming that the environment can be accurately approximated if enough polygons. However, the number of polygons needed often exceeds the memory and computational resources available.

Tampieri and Lischinski [20] further explain that the Constant Radiosity Assumption leads to fundamental errors in radiosity computations. A solution computed on a tessellated surface can only be as accurate as the tessellation. The Constant Radiosity Assumption also presents inconsistency between its illumination and rendering phases. During the energy transfer phase, radiosity is assumed constant across each polygon. However, radiosity renderings are made by sampling each polygon at a few points and then interpolating brightness values between these points. Basic signal processing shows that while interpolating a solution may make an image look more accurate, all such interpolation can do is mask error by blurring the image. A consistent radiosity solution must incorporate the interpolation into the energy transfer calculations.

2.1 The Radiosity Integral Equation

In order to apply the appropriate mathematical tools to the solution of radiosity problems, it is convenient to express the radiosity equation in parametric form. Parametrically, the key radiosity variables (radiosity, emittance, reflectivity, *etc.*) are represented as functions of two variables, (s, t) or (u, v), over each surface i or j. By abstracting all the complexity of surface interaction into a single kernel function $K_{ij}(s, t, u, v)$, the radiosity equation can be written as an integral equation,

$$B_i(s, t) = E_i(s, t) + \sum_j \iint K_{ij}(s, t, u, v) B_j(u, v)\, du\, dv, \qquad (1)$$

where the kernel function $K_{ij}(s, t, u, v)$ is the product of the double-differential form factor $F_{i-j}(s, t, u, v)$, reflectivity $\rho_i(s, t)$, area $A_i(s, t)$, and visibility $\mathrm{VIS}_{ij}(s, t, u, v)$

$$K_{ij}(s, t, u, v) = \rho_i(s, t) F_{i-j}(s, t, u, v) \mathrm{VIS}_{ij}(s, t, u, v) A_j(u, v). \qquad (2)$$

[1] now at Rhythm and Hues Studios, Inc., 910 North Sycamore Ave., Hollywood, CA 90038. E-mail: hzatz@rhythm.com or hzatz@alumni.caltech.edu

The form factor and area functions can be further expanded in terms of the functions describing surface geometry $\vec{x}_i(s,t)$ and normals $\hat{n}_i(s,t)$:

$$F_{i-j}(s,t,u,v) = \frac{(\hat{n}_i(s,t) - \hat{n}_j(u,v)) \cdot (\vec{x}_j(u,v) - \vec{x}_i(s,t))}{\pi \|\vec{x}_i(s,t) - \vec{x}_j(u,v)\|^4}$$

$$A_j(u,v) = \left\| \frac{\partial \vec{x}_j(u,v)}{\partial u} \times \frac{\partial \vec{x}_j(u,v)}{\partial v} \right\| \qquad (3)$$

3 Mathematical Background

The Galerkin method provides a method for solving integral equations in terms of a basis set of non-constant functions across each surface. This section provides the mathematical background necessary to apply the Galerkin method to the radiosity equation.

3.1 Basis Set Projection

To approximate the radiosity distribution by a combination of functions, we first need formal tools to manipulate an appropriate two-dimensional basis set. We denote this basis set $\{\mathcal{T}_k(s,t) | k = 0, 1, \ldots\}$, where s and t are the parametric variables across a surface, and k specifies a particular function in the set.

Just as geometric vectors have a dot product that projects one onto the other, the inner product of two functions $f(s,t)$ and $g(s,t)$ can be defined,

$$\langle f | \, g \rangle_{\mathcal{W}} = \int_{-1}^{1} \int_{-1}^{1} f(s,t) g(s,t) \mathcal{W}(s,t) ds \, dt. \qquad (4)$$

$\mathcal{W}(s,t)$ is some weighting function that describes the importance of different positions to the inner product. To apply the Galerkin method to radiosity, we use an orthonormal set of basis functions, $\{\mathcal{T}_k(s,t)\}$—a set designed so that for a particular inner product weight function $\mathcal{W}(s,t)$,

$$\forall_{k,l} \quad \langle \mathcal{T}_k | \, \mathcal{T}_l \rangle_{\mathcal{W}} = \delta_{kl}. \qquad (5)$$

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

$$B_i^k = \langle B_i | \, \mathcal{T}_k \rangle_{\mathcal{W}} \, . \qquad (6)$$

The original function can be approximated by the weighted sum,

$$B_i(s,t) \approx \sum_k B_i^k \mathcal{T}_k(s,t). \qquad (7)$$

3.2 Legendre and Jacobi Polynomials

The Galerkin method is usually solved using an orthonormal polynomial basis set, defined on the interval $[-1, 1]$. Legendre and Jacobi polynomials are one-dimensional, orthonormal polynomials which can be combined into a two-dimensional basis set by multiplying two polynomials in different variables. We limit our analysis in the next two sections to polynomials of one variable.

When the inner product has a weight function equal to one, the polynomials formed are the Legendre polynomials. The unnormalized Legendre polynomials are generated by a recursion rule [8],

$$P_0(x) = 1 \qquad P_1(x) = x$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x). \qquad (8)$$

The normalized Legendre polynomials are

$$\bar{P}_n(x) = \sqrt{n + \frac{1}{2}} P_n(x) \qquad (9)$$

Polynomial sets can also be created with non-constant inner product weight functions $\mathcal{W}(x)$. Later in this paper (section 4.2), a set of polynomials will be needed with a weight function that has a multiple zero at its endpoints. The Jacobi polynomials $P_i^{(\alpha,\beta)}$ have such behavior, with the weight function,

$$\mathcal{W}(x) = (1-x)^{\alpha}(1+x)^{\beta}, \qquad (10)$$

where α and β are the degree of multiplicity.

The unnormalized Jacobi polynomials have a more complex recursion rule than the Legendre polynomials [5]:

$$P_0^{(\alpha,\beta)}(x) = 1 \qquad P_1^{(\alpha,\beta)}(x) = \frac{\alpha-\beta}{2} + \frac{2+\alpha+\beta}{2}x$$

$$P_{n+1}^{(\alpha,\beta)}(x) = \frac{A_n^{(\alpha,\beta)}x - B_n^{(\alpha,\beta)}}{C_n^{(\alpha,\beta)}} \qquad (11)$$

where

$$\begin{aligned} A_n^{(\alpha,\beta)} &= (2n+\alpha+\beta+1)(\alpha^2 - \beta^2 + 2n + \alpha + \beta + 2) \\ &\quad \times (2n+\alpha+\beta)P_n^{(\alpha,\beta)}(x) \\ B_n^{(\alpha,\beta)} &= 2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)P_{n-1}^{(\alpha,\beta)}(x) \\ C_n^{(\alpha,\beta)} &= 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta) \qquad (12) \end{aligned}$$

These polynomials can be normalized by the factor [8]:

$$\sqrt{\frac{\Gamma(n+1)\Gamma(\alpha+\beta+1+n)(\alpha+\beta+1+2n)}{\Gamma(\alpha+1+n)\Gamma(\beta+1+n)2^{\alpha+\beta+1}}} \qquad (13)$$

3.3 Quadrature Rules

An informative explanation of one-dimensional quadrature rules has been compiled by Delves and Mohamed [6]. A condensed version is presented here.

A quadrature rule is a method for approximating the integral of a function by a weighted sum of function samples at particular points. Quadrature rules can be used to approximate inner product integrals, like that in (4). Given a fixed function $\mathcal{W}(x)$ and another function $f(x)$, we can choose points ξ_i and weights w_i such that:

$$\int_a^b f(x)\mathcal{W}(x)dx \approx \sum_i w_i f(\xi_i) \qquad (14)$$

Quadrature rules can be designed to be exact for a certain class of functions. The Gaussian quadrature rules, by computing optimal positions for the N sample points ξ_i, are exact for polynomials up to order $2N - 1$. The Gauss quadrature rule with weight function $\mathcal{W}(x)$ is closely tied to the set of orthogonal polynomials with the same weight function.

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

$$\int_{-1}^{1} \mathcal{W}(x) f(x) \, dx \approx \sum_{i=1}^{N} w_i f(\xi_i), \qquad (15)$$

start by choosing a set of orthogonal polynomials $\mathcal{T}_i(x)$ with the same weight function $\mathcal{W}(x)$, and expressed in terms of recursion rules [17] so that:

$$\mathcal{T}_{-1}(x) \equiv 0, \qquad \mathcal{T}_0(x) \equiv 1,$$

$$\mathcal{T}_{i+1}(x) \equiv (x - \delta_{i+1})\mathcal{T}_i(x) - \gamma_{i+1}^2 \mathcal{T}_{i-1}(x). \qquad (16)$$

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

$$\begin{bmatrix} \delta_1 & \gamma_2 & & 0 \\ \gamma_2 & \delta_2 & \ddots & \\ & \ddots & \ddots & \gamma_N \\ 0 & & \gamma_N & \delta_N \end{bmatrix} \qquad (17)$$

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. The eigenvectors and eigenvalues for tridiagonal symmetric matrices can be found using QR factorization [17].

To create the Gauss-Legendre rule of order N, exact for polynomials up to degree $2N - 1$, the γ_i and δ_i coefficients are [22]:

$$\delta_{i+1} = 0, \qquad \gamma_{i+1} = \sqrt{\frac{i^2}{(2i+1)(2i-1)}}, \qquad (18)$$

and for general Jacobi polynomials $P_i^{(\alpha,\beta)}$:

$$\begin{aligned} \delta_{i+1} &= \frac{(\alpha+\beta)(\beta-\alpha)}{(2i+\alpha+\beta+2)(2i+\alpha+\beta)}, \\ \gamma_{i+1} &= \sqrt{\frac{4(i+\alpha)(i+\beta)i(\alpha+\beta+i)}{(\alpha+\beta+2i)^2(\alpha+\beta+2i+1)(\alpha+\beta+2i-1)}}. \qquad (19) \end{aligned}$$

When using these quadrature rules to project a function into a basis set using (6), it is important to use a sufficiently accurate quadrature rule. If a one-dimensional polynomial basis set includes terms up to order n, the projection integral (6) must be accurate up to order $2n$—since the function is represented as a polynomial of order n, the projection integrand will be a polynomial of order $2n$. Therefore, a one-dimensional Gaussian quadrature rule must have at least $N + 1$ sample points to integrate accurately [6].

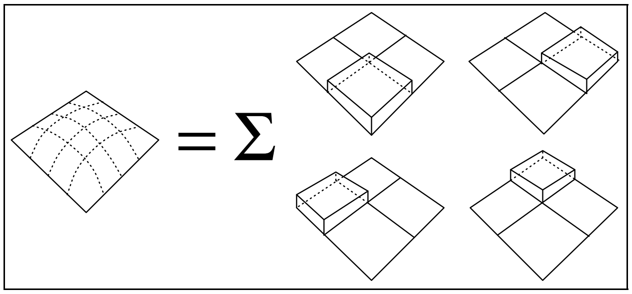

Figure 1: Conventional radiosity methods approximate a surface's radiosity by meshing it into a large number of constant intensity patches. Radiosity is represented by height above the surface.

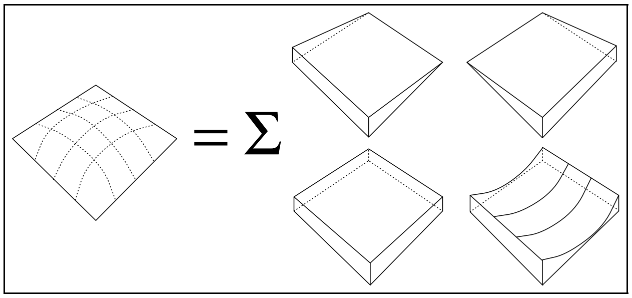

Figure 2: Higher-order radiosity approximates a surface's radiosity by dividing it into several different smooth functions. These smooth functions are scaled and combined to approximate the original radiosity distribution.

4 Non-Constant Radiosity

Consider the effect of meshing a single surface into constant radiosity patches (Figure 1). Although the radiosity is smooth on individual patches, combinations describe a discontinuous, stair-step radiosity function. To produce a smooth, consistent solution, we need to formulate radiosity in terms of smooth functions across an entire surface, instead of disjoint patches on parts of a surface.

Figure 2 shows a hypothetical decomposition of a radiosity function. Constant, linear, and higher-order functions are combined to produce a smooth approximation to the radiosity function. If the radiosity of every surface were represented by a combination of these functions, the radiosity problem would reduce to finding their relative weights.

To properly compute these proportions, we use a radiosity formulation based on a linear combination of orthonormal basis functions $\{\mathcal{T}_l(s,t)\}$. Instead of radiosity values, we use radiosity coefficients $\{B^l\}$—the relative contribution of each function $\mathcal{T}_l(s,t)$. The full radiosity distribution on a surface becomes the function

$$B_{\text{total}}(s,t) = \sum_l B^l \mathcal{T}_l(s,t). \tag{20}$$

Functions on different surfaces must interact in a manner analogous to the way conventional patches interact through form factors. Just as conventional radiosity uses form factors to describe the interaction between patches, here the kernel function $K_{ij}(s,t,u,v)$ from (1) details how energy is transferred between functions on different surfaces. When two constant functions on different surfaces interact, the kernel function interaction is equivalent to a classical form factor. Other kernel functions describe higher-order interactions.

4.1 The Galerkin Method

Given an orthonormal basis set, the Galerkin technique finds a good [6] fit to the integral equation's solution within that set. Heckbert [10, 11] suggested that the Galerkin method and meshing could be used to solve the radiosity integral equation in a plane. This and subsequent sections demonstrate how it can be applied to three-dimensional radiosity.

Starting with the parametric radiosity equation (1),

$$B_i(s,t) = E_i(s,t) + \sum_j \int\!\!\int K_{ij}(s,t,u,v) B_j(u,v)\,du\,dv, \tag{21}$$

expand the $B_j(u,v)$ term inside the integral in terms of the basis set $\{\mathcal{T}_l(u,v)\}$ using (7). 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

$$B_i(s,t) = E_i(s,t) + \sum_{j,l} B_j^l \int\!\!\int K_{ij}(s,t,u,v)\mathcal{T}_l(u,v)\,du\,dv. \tag{22}$$

Now, take the inner product of both sides with the kth basis set function $\mathcal{T}_k(s,t)$. Using bilinearity and the relation described in (6),

$$B_i^k = E_i^k + \sum_{j,l} B_j^l \left\langle \int\!\!\int K_{ij}(s,t,u,v)\mathcal{T}_l(u,v)\,du\,dv \,\middle|\, \mathcal{T}_k(s,t) \right\rangle_{\mathcal{W}}. \tag{23}$$

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 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, requiring four integrations—two explicit, and two in the inner product. However, once the kernel matrix has been computed for each value of i, j, k, and l, the radiosity equation can be written as a matrix equation,

$$B_i^k - E_i^k = \sum_{j,l} B_j^l K_{ij}^{kl}. \tag{24}$$

Just as a conventional form factor matrix relates constant radiosities on different elements, the kernel matrix relates radiosity functions across different surfaces. The K_{ij}^{kl}, B_i^k and E_i^k values are analogous 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. Note also that even though (24) is written in terms of four indices, since the surface indices i, j and function indices k, l are independent of each other, (24) is still a two-dimensional matrix equation.

This equation can be solved using any standard matrix technique, such as Gaussian elimination, or progressive refinement techniques [4]. Cohen *et al*'s progressive refinement technique requires slight modification with Galerkin radiosity, because the radiosity coefficients B_j^l may have negative values. These negative values do not indicate negative energies; they are a weight applied to the basis function. The shooting order should be based on unshot magnitude:

$$M_j^l = \|B_j^l\| \int\!\!\int |\mathcal{T}_l(u,v)|\, dA_j(u,v)\,du\,dv. \tag{25}$$

4.2 Edge Singularities

Near the common edge of two non-coplanar surfaces, the double-differential form factor approaches infinity as a pole of order two[22]. Although the function still has a finite integral, the singularity can cause serious convergence problems. If the singularity is ignored, Galerkin solution methods converge extremely slowly for a mediocre basis set, and may fail entirely for a bad basis set.

To insure reasonable convergence, the basis set must compensate for the singularity. In (23), the singularity appears inside the quadruple integral that generates K_{ij}^{kl}. This integral also includes the 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} goes to infinity, the two features can cancel and the integral will converge. Since the kernel singularity grows as a pole of order two, the weight function should have zeroes of multiplicity two at its edges. The Jacobi polynomial sets $\mathcal{P}^{(0,2)}$ and $\mathcal{P}^{(2,0)}$ (see section 3.2), have appropriate weight functions.

By using a hybrid Galerkin method, the edge singularities are cancelled. For non-singular light transfers between surfaces that do not touch, a Legendre basis set is used. 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 computing the K_{ij}^{kl} coefficients and the associated radiosity transferred in a singular shot, project this polynomial function in s and t is back into a Legendre basis set for storage. An empty box computed with this hybrid method is shown in Figure 6.

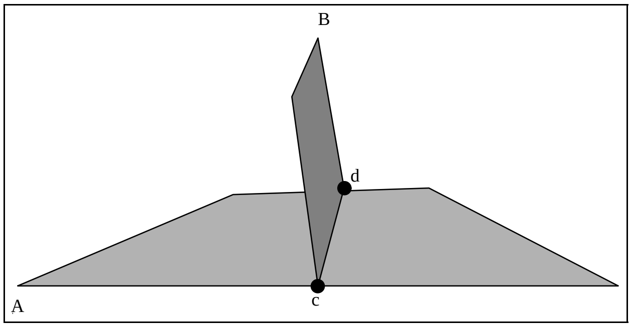

Figure 3: Surfaces A and B meet in a T-intersection; surface B divides surface A into two regions along the line $\overline{cd}$.

Because singularities can be produced at any non-parallel intersection, geometries with T-intersections (the three-dimensional analog to Heckbert's T-corners [11]) like those in Figure 3 make singularities difficult to handle. 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. More effective approaches include subdividing significant T-intersections into distinct singular intersections, or ignoring the singularity altogether when possible.

4.3 Computing the Energy Transfers

In order to generate radiosity solutions, entries in the kernel matrix (24) must be computed. Each entry is computed by applying a quadrature rule (15) to approximate the inner product of (23) for particular values of i, j, k and l. For non-singular energy transfers—those between surfaces that do not share a common edge—the inner product weight function is unity, and the quadrature rule is a Gauss-Legendre quadrature rule constructed with (17) and (18). If the Gauss-Legendre quadrature points and weights are denoted p^L_α and w^L_α respectively, then each kernel matrix element is approximated by the summation,

$$K^{kl}_{ij} \approx \sum_{\alpha,\beta,\gamma,\delta} K_{ij}(p^L_\alpha, p^L_\beta, p^L_\gamma, p^L_\delta)\mathcal{T}^L_k(p^L_\alpha, p^L_\beta)\mathcal{T}^L_l(p^L_\gamma, p^L_\delta) w^L_\alpha w^L_\beta w^L_\gamma w^L_\delta. \tag{26}$$

Since each kernel sample requires a full intersection test with the environment, caching samples $K_{ij}(p^L_\alpha, p^L_\beta, p^L_\gamma, p^L_\delta)$ or results of the associated intersection tests can save significant CPU time, at the expense of additional storage.

Singular energy transfers—those between two surfaces that meet in a singular edge—require additional processing. Kernel matrix elements for a singular transfer are computed using a Gauss-Jacobi quadrature rule matching the Jacobi basis set. Once the quadrature rule's points p^J_α and weights w^J_α have been computed using (17) and (19), the kernel matrix elements can be computed by the summation,

$$K'^{kl}_{ij} \approx \sum_{\alpha,\beta,\gamma,\delta} K_{ij}(p^J_\alpha, p^J_\beta, p^L_\gamma, p^L_\delta)\mathcal{T}^J_k(p^J_\alpha, p^J_\beta)\mathcal{T}^L_l(p^L_\gamma, p^L_\delta) w^J_\alpha w^J_\beta w^L_\gamma w^L_\delta. \tag{27}$$

When this weighted sum is evaluated, the resulting matrix entries K'^{kl}_{ij} are in terms of a 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, they can be converted from Jacobi coefficients to Legendre coefficients by expanding the Jacobi coefficients into an ordinary polynomial in s and t, and then converting that polynomial back into a sum of Legendre polynomials.

5 Shadow Discontinuities

As with any illumination algorithm, dealing with occlusions presents a special challenge. The easiest way to deal with shadows is to let the basis functions find a best fit. Unfortunately, shadows produce sharp edges which cannot be expressed in terms of a few polynomials. Attempting to model such edges with a small polynomial basis set produces a fuzzy shadow with ripples around it—the Gibbs behavior visible in Figure 7.

Shadow edges come from discontinuities in the radiosity function [10]. One way to remove these discontinuities is to mesh the environment along curves of discontinuity [13, 12], a process which eliminates the occlusion difficulties of Galerkin radiosity. Unfortunately, discontinuity meshing methods magnify the number of surfaces in the scene, vastly increasing computation time. Even though shadows are primarily an interaction between a light source and a receiving surface, subdividing the receiving surface to produce accurate shadows complicates interactions with the rest of the environment.

5.1 Shadow Masking

To smooth the shadow discontinuities out of the radiosity distribution seen by the Galerkin method, we propose using a *shadow mask* approximation. For the majority of emitter-receiver pairs, where shadows do not have a high-frequency effect on the solution, traditional visibility calculations can be used. However, for a select group of emitter-receiver pairs, we move the visibility term $\text{VIS}_{ij}(s, t, u, v)$ out of the kernel function and integral in equations (2) and (1), and replace it with a normalized shadow mask function $M_{i\leftarrow j}(s, t)$,

$$M_{i\leftarrow j}(s, t) = \frac{\iint \text{VIS}_{ij}(s, t, u, v)du\,dv}{\iint du\,dv}. \tag{28}$$

This function approximates the fraction of the light originating from emitter j that arrives at a particular location on receiving surface i. The shadow mask is one where the emitter is fully visible, zero where the emitter is fully occluded, and takes on intermediate values when the light is partially occluded. It is essentially a texture map for painting the shadow onto the receiving surface.

During the radiosity pass, if the energy transfer from emitter j to receiver i involves a shadow mask, the radiosity is accumulated without visibility calculations in the special coefficients $B^k_{i\leftarrow j}$ instead of B^k_i. When light is re-emitted from surface i's basis functions, the kernel samples are multiplied by the shadow mask across surface i, restoring some of the occlusion information. The radiosity across a surface, $B_i(s, t)$, becomes the combination of ordinary Galerkin basis functions and shadow mask-weighted basis functions. If h represents all light sources casting a shadow on surface i,

$$B_i(s, t) = \sum_k B^k_i \mathcal{T}_k(s, t) + \sum_{h,k} M_{i\leftarrow h}(s, t)B^k_{i\leftarrow h}\mathcal{T}_k(s, t). \tag{29}$$

By using coefficients $B^k_{i\leftarrow h}$, radiosity in the shadow mask is maintained separately from radiosity coming from other parts of the environment. When a receiving surface has shadow masks associated with it, every surface interacts either with a shadow mask, or with the standard surface description—not both.

In this implementation, shadow masks were computed from equation (28) using multiple point-to-point visibility samples regularly spaced in the parametric dimensions. Values of $M_{i\leftarrow j}(s, t)$ were computed by linear interpolation between these sample points. Shadow mask samples could conceivably be taken along lines of discontinuity, or in some more complicated non-regular structure to improve efficiency or accuracy.

In all environments tested, even accounting for the time spent constructing shadow masks, the time required to compute a radiosity solution using shadow masks was significantly smaller than that for a full discontinuity mesh. For the simple environment in Figure 8, the shadow mask was a regular 40 by 40 grid of sample points on the floor. Without Gibbs phenomena to transfer energy into higher order basis functions as in Figure 7, the radiosity pass actually required fewer shots and less time to converge than the non-shadow masked version.

Since a shadow mask only adds one surface to the rows (but not the columns) of the radiosity matrix for each associated emitter-receiver pair in the environment, shadow masks add relatively little to radiosity solution time compared to discontinuity meshing methods. Shadow masks can be precomputed for portions of the environment where shadow details are expected to be significant. Furthermore, since shadow masks are defined in parametric space, a single implementation can cast shadows to and from any type of surface.

Unfortunately, shadow masks also have significant disadvantages. By moving the visibility term out of the radiosity equation's integral, any correlation between the emitter's light distribution and the shape of the occluding

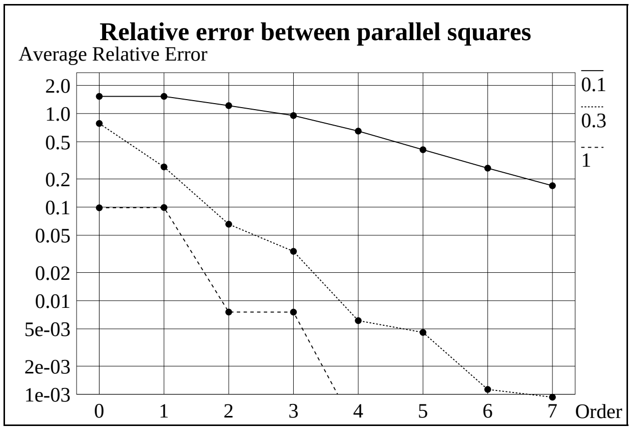

Figure 4: Average relative error for Galerkin radiosity transfers between two parallel squares of width l, distance l, $0.3l$, and $0.1l$ apart.

surface is destroyed. Because of this, a shadow mask solution will not converge to the "true" solution. Placement of the shadow masks is currently left to the user; some criteria is needed for determining whether or not to use shadow masks. Although shadow masks can be stored in a simple grid fashion, such a grid may not produce the best results when used with a particular quadrature rule. Finally, any attempt at increasing the spatial accuracy of shadow masks can duplicate many of the difficulties of storing a mesh on a surface.

However, there is a significant difference between increasing the density of a mesh and increasing the density of a shadow mask—every element of a mesh becomes another surface interacting with the environment, while even the most complex shadow mask is still only part of one surface. Shadow masks do facilitate the generation of approximate radiosity solutions with the Galerkin method, by smoothing out shadow discontinuities. Further research may suggest ways to avoid their associated disadvantages.

6 Sources of Error

The principal cause of error is not using a large enough basis set; as more basis functions are used, the Galerkin method produces a more accurate solution. Improper treatment of shadows can also cause significant inaccuracies in a Galerkin solution; if shadow discontinuities are ignored, they produce Gibbs-behavior ripples, and if shadow masks are used, they introduce approximation error. Additional errors come from inaccuracies in the quadrature rule used to evaluate kernel matrix integrals, or from approximate matrix solution techniques like progressive radiosity.

In this section, error analysis is provided at two different scales. At the level of surface-to-surface energy transfer, Galerkin radiosity results are examined for a few simple cases where comparison with an exact analytical solution is possible. At the level of picture generation, conventional and Galerkin radiosity solutions are compared for a standard radiosity test environment.

6.1 Energy Transfer Error

For the simple environment used by Sparrow's variational radiosity solution [18], a fourth-order solution produced a relative error of less than one percent. Using the method of this paper, error computations for a single energy transfer between parallel and perpendicular squares produce similar levels of accuracy. All comparisons in this section are made against an analytic solution using the formulation of Sparrow and Cess [19]. The relative error metric used is

$$E = \left\langle \frac{|B_{\mathrm{Galerkin}}(s,t) - B_{\mathrm{exact}}(s,t)|}{B_{\mathrm{exact}}(s,t)} \right\rangle_{s,t}, \tag{30}$$

where the error is evaluated on a 500 by 500 grid of sample points on the receiving surface. Transmitting and receiving squares are the same size, and are computed at the same solution order (although for numeric reasons, this often produces the worst results [22]).

The simplest case is for a radiosity transfer between parallel squares with sides of length l, as shown in Figure 4. With the distance between the squares equal to their size, the fourth-order transfer gives a relative error of only 0.04%. 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.

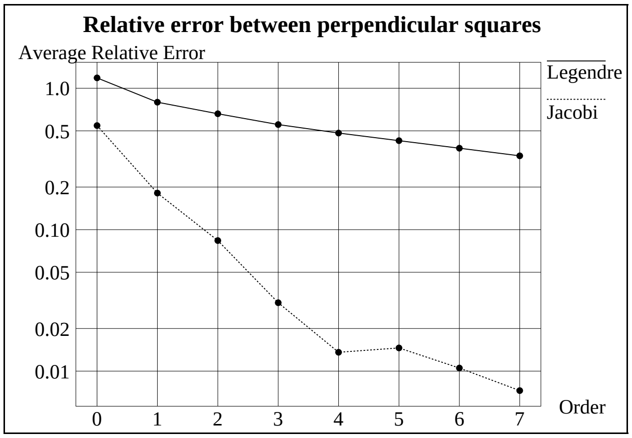

Figure 5: Average relative error for Galerkin radiosity transfers between two adjacent perpendicular squares at a corner. The Jacobi basis set computation produces significantly less error for this singular transfer than the Legendre basis set.

As the squares move closer, the relative error becomes much higher. When the distance between the squares is reduced to one-tenth their width, even a seventh order solution produces an average relative error of 16.9%. Unfortunately, when surfaces are extremely close relative to their size, to achieve reasonable accuracy, the surfaces must still be subdivided.

Experiments with perpendicular rectangles (Figure 5) illustrate the importance of proper treatment of singularities. Using a non-singular Legendre basis set to compute the energy transfer produced large error even at high order; a seventh order transfer produced a relative error of 33.2%. Using a singular Jacobi basis set, results are much more accurate. Fourth and fifth order transfers both produce about 1.4% relative error.

6.2 Comparison with Conventional Radiosity

Lischinski and Tampieri provided a reference solution to a two-box radiosity environment. This solution was computed using the discontinuity meshing techniques of [13], with adaptive integration using Wallace [21] point-to-point form factors. Individual triangles in the mesh were treated consistently as quadratic elements, limiting error in their reference solution to a few meshing artifacts, visible near the corners of the top wall. This solution is used as a comparison baseline for images generated with Galerkin radiosity.

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 depends on the sum of the highest orders used in each dimension. 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.

Figure 9 shows pictures of a simple test environment solved with different solution orders. Shadows were created using a 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. Meshing (Figure 10) was only performed to eliminate T-intersections; the three boxes and light were meshed to 26 polygons.

Figure 11 shows difference images between the different order Galerkin solutions of the test environment and the reference solution. These images were created by 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 slightly before comparison, so that outlines of the boxes and floors would be visible in the difference images. As would be expected, the difference images get progressively darker as the solution

Figure	Description	CPU time	Shots
6	Empty box	5.4s	7
7	Box with single occluder	59.7 s	33
8	Shadow masked box	42.8 s	22
12	Clay teapot	6.71 h	53

Table 1: Timings for the shadow generation and radiosity pass combined for various pictures computed with this algorithm. All timings are for an HP 9000/720 workstation.

order increases; the regions where the solution is least accurate tend to be near singular edges.

In this particular test case, the method of [13] 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 paper, 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.

7 Results

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 radiance at a given point (s, t) on surface i is recovered from these coefficients using (7). If shadow masks were used, the additional coefficients B_{ih}^k are incorporated with (29).

In this implementation, environments are rendered by a simple ray-tracing/scanline technique. When a ray intersects a surface, that intersection point is projected back into the surface's parametric space, and the result is used to compute a radiosity value for the appropriate pixel.

7.1 Curved Surfaces

Curved surfaces can be easily incorporated into Galerkin radiosity; the kernel term's form factor as expressed in (3), includes surface normals explicitly. To implement curved surfaces, replace the traditional constant surface normal value with a function, computable at any parametric location. Sample pictures are shown with bicubic patches (Figure 12) and other curved surfaces (Figure 13). The Galerkin radiosity method was applied directly to these environments; the curved surfaces were *not* tiled.

For comparison purposes, the teapot environment was also computed using a commercially-available radiosity package [16]. This package uses the point-sampling algorithm of Wallace *et al.* [21] to compute form factors, but does not perform 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. The radiosity solution took 6.2 hours, and over 54 megabytes of memory to compute; this simple forty-patch scene became a relatively complex, eight thousand polygon environment. In contrast, the Galerkin computation took 6.7 hours, but only required 3.9 megabytes of memory during the radiosity pass. Over 90% of this computation time was spent computing visibility samples.

The significant point of this comparison is that given approximately equivalent amounts of time to produce a solution, conventional and Galerkin methods produced similar results. But since Galerkin methods needn't maintain the detailed geometric structure of a mesh, they use significantly less memory.

7.2 Parallelization

Galerkin radiosity environments are not meshed into large, complicated data structures, so it is relatively easy to maintain copies of the environment in memory on multiple hosts. Since each individual light transfer between two surfaces depends only on the geometry and shadow masks, they can be computed on independent machines. 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 13 was computed in parallel on five DECstations and five HP 700's as a background process over two days.

8 Conclusions

Using the Galerkin method, this paper has presented an alternative method for producing radiosity simulations. Through special treatment of the radiosity equation's singularities and discontinuities, the Galerkin technique's dependency on smooth kernels 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 represented as a smoothly varying function. Pictures are rendered directly from the radiosity solution, without an additional blurring step.

- Adequately sampled curved surfaces can be used directly. Since curved surfaces don't need to be tessellated, they can be incorporated into a scene cheaply. Issues of approximating a surface's geometry and approximating a surface's radiosity are separated.

- Energy transfer error analysis shows that meshing is only essential 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.

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

9 Deficiencies of the Method

As with any rendering algorithm, Galerkin radiosity 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.

Shadow masking is only a rough approximation to the true occlusion behavior; it eliminates any correlation between variations in light source intensity and the intensity of the shadow, virtually returning to the Constant Radiosity Assumption for a shadow's light source. Furthermore, the distribution of the shadow mask sample points can have a significant impact on the accuracy of the shadow they generate.

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. In general, an order of 4 or 5 is sufficient, but self-intersecting or highly curved surfaces may require a higher-order solution.

The method does not mathematically guarantee radiosity continuity between adjacent coplanar surfaces. However, such surfaces appear much less frequently in a shadow masked environment than in a meshed environment. If such continuity is needed, it can be generated by using a high enough order on the adjacent surfaces that the error on each surface is reduced until their radiosity values along their common boundaries match visibly—usually 8 or 9 in our tests.

Finding all the singularities in a system can also be difficult. Environments usually have a large number of T-intersections (see Figure 3), each of which could require a separate 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, requiring recomputation.

10 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 generalization of shadow masks is needed to account for variations in light source intensity.

A means 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

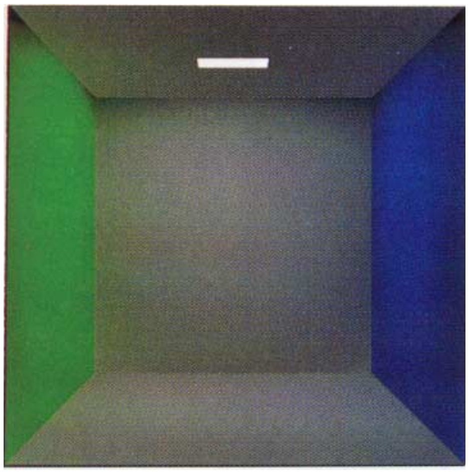

Figure 6: An empty box computed with up to fourth order polynomials, or 15 basis functions across each surface. On an HP 9000/720, the radiosity pass took 5.4 CPU seconds.

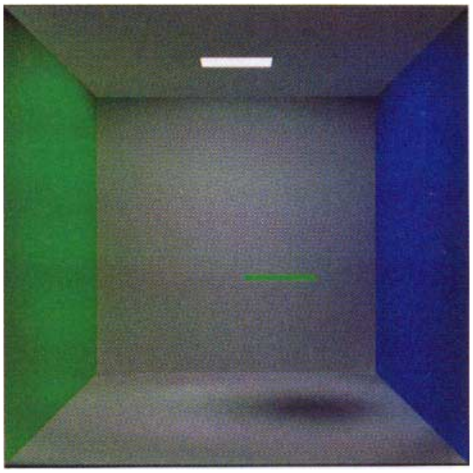

Figure 7: A box with an occluding rectangle computed with a fourth order basis on all surfaces except the floor, which has an eighth order basis. The ripples on the floor of the box appear because shadow discontinuities cannot be accurately described by a low frequency Galerkin basis set.

Hanrahan's hierarchical multigridding technique [9] to higher order functions could produce a means to do this. Some method must also be found to automatically determine an appropriate solution order for each surface, instead of the current area-based heuristic.

The method of this paper 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 paper 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.

Acknowledgements

The Program of Computer Graphics at Cornell is one of five sites of the NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (Grant # ASC-8920219). Thanks to HP/Apollo and DEC for equipment donations. This research was partially conducted under a National Science Foundation grant entitled "Interactive Input and Display Techniques." The author was funded by a National Science Foundation Graduate Student Fellowship, a Dabnicorp Computer Graphics award, and by a Cornell University Sage Fellowship. I would like to thank Don Greenberg for providing insightful comments as to what was needed to write up this research, Dani Lischinski and Filippo Tampieri for providing the reference solution of Section 6, and especially Jim Arvo for his advice in developing the research and this text. I'd also like to thank everyone at the Cornell Program of Computer Graphics, Rhythm and Hues, Inc., and the Caltech Graphics Lab for their support and encouragement.

References

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

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

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

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

[5] Philip Davis, *Interpolation and Approximation*, Blaisdell, New York, 1963.

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

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

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

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

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

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

[12] Paul Heckbert, "Discontinuity Meshing for Radiosity", *Third Eurographics Worshop on Rendering*, Bristol, UK, May 1992.

[13] Dani Lischinski, Filippo Tampieri, and Donald Greenberg, "Discontinuity Meshing for Accurate Radiosity", IEEE CG&A, 12(6), Nov. 1992.

[14] Nelson Max and Michael Allison, "Linear Radiosity Approximations using Vertex-to-Vertex Form Factors", *Graphics Gems III*, Academic Press, 1992, p. 319

[15] 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.

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

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

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

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

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

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

[22] Harold Zatz, *Galerkin Radiosity: A Higher Order Solution Method for Global Illumination*, Master's Thesis, Cornell University, Ithaca, New York, 1992.

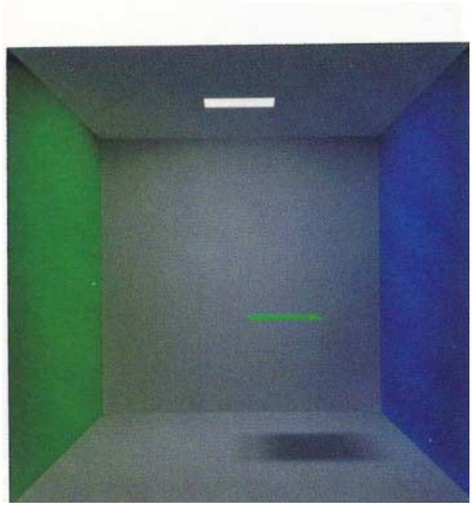

Figure 8: A box with the transfer from light source to floor shadow masked, computed to fourth order on all surfaces except the floor and light source, which are computed to eighth order.

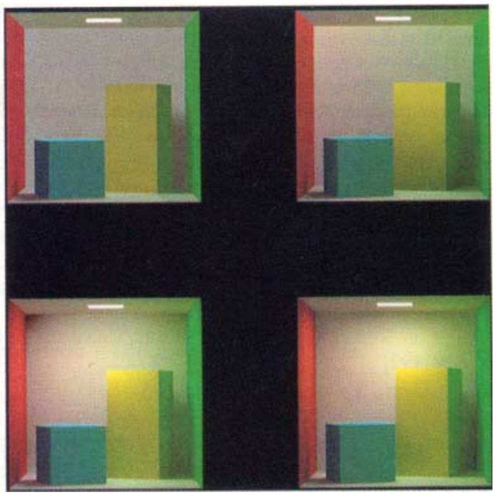

Figure 9: Solving the two box test environment, with solution orders zero, one, three, and seven.

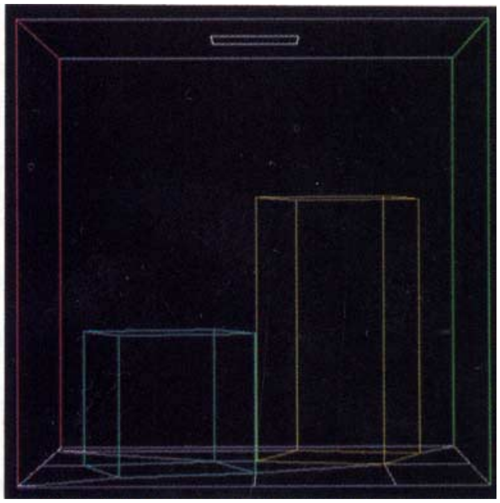

Figure 10: Mesh used for Figure 9. Only the floor has been meshed, to eliminate T-intersections. The boxes, walls, and ceilings were each solved using functions over the entire surface.

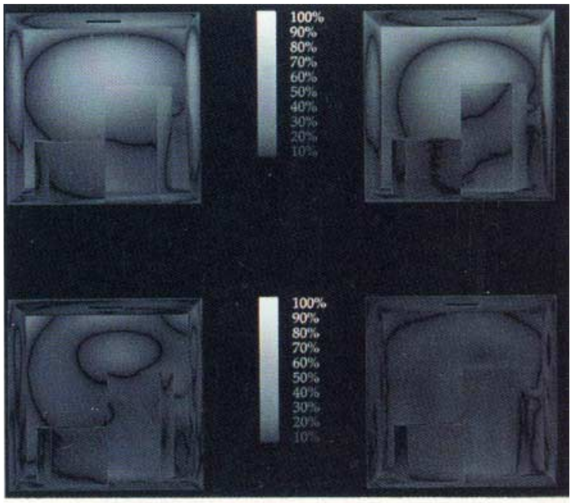

Figure 11: Difference images between the two box test environment and the reference solution, with solution orders zero, one, three, and seven.

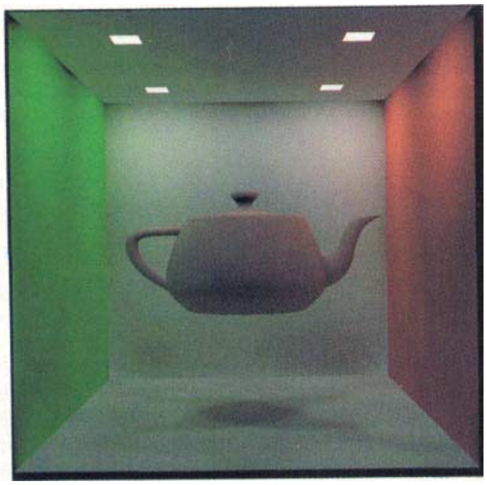

Figure 12: The radiosity function across the clay teapot was solved directly, with a sixth-order basis set for each bicubic patch. The floor, walls, and portions of the teapot received shadow masks from the four lights.

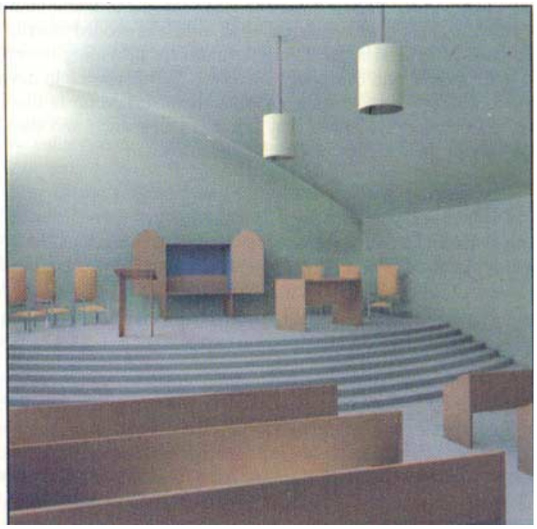

Figure 13: This picture shows the interior of a temple containing 607 parametrically defined, non-meshed surfaces, including polygons, bicubic patches, cylinders, and cubic extrusions. Most surfaces were computed with a fourth or fifth order solution, except for the walls and roof at seventh order, and the cylindrical light fixtures at thirteenth order.