

Radiosity Algorithms Using Higher Order Finite Element Methods

Roy Troutman, Nelson L. Max

Lawrence Livermore National Laboratory

Abstract

Many of the current radiosity algorithms create a piecewise constant approximation to the actual radiosity. Through interpolation and extrapolation, a continuous solution is obtained. An accurate solution is found by increasing the number of patches which describe the scene. This has the effect of increasing the computation time as well as the memory requirements. By using techniques found in the finite element method, we can incorporate an interpolation function directly into our form factor computation. We can then use less elements to achieve a more accurate solution. Two algorithms, derived from the finite element method, are described and analyzed.

CR Categories and Subject Descriptors: 1.3.3 [Computer Graphics]: Picture/Image Generation - Display Algorithms. 1.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism.

Additional Key Words and Phrases: finite elements, form-factor, global illumination, radiosity.

1 Introduction

The traditional radiosity algorithm computes the form factors at a collection of points [5]. There have been several techniques used to enhance and speed up the algorithm. Cohen described an algorithm which enabled more complex environments to be rendered by placing a half cube or “hemicube” at each evaluation point and sampling through pixels on the hemicube surface [3]. We can improve the accuracy of the solution by increasing the resolution of the hemicube or by analytically determining the form factors [1], [13]. Further improvements can be made by producing a mesh which follows the discontinuities introduced by shadow boundaries and surface intersections [9], [10].

Rather than assuming the radiosity arrives from piecewise constant patches, Max and Allison introduced an algorithm which assumed a piecewise linear approximation [11]. This algorithm works by placing an interpolation function directly into the form factor computation. Using this technique, a more accurate solution can be obtained with less patches [9]. An extension to this algorithm is to increase the order of the interpolation function to quadratic, cubic or even higher [16]. These interpolation functions are what the finite element method refers to as basis functions [2].

2 Basis Functions

The details of the basis functions, elements and nodes can be found in [2]. We will only give a brief overview to establish our terminology.

2.1 Approximation Function

The finite element method associates a basis function for each of the local nodes in a representative element. The basis function for a

global node becomes a combination of basis functions defined on the local nodes of all elements which contain the global node [2], [9], [14]. In this paper a node is “contained” or “in” an element if it is on the boundary or interior of the element. Using the basis function f_i and radiosity B_i associated with each node i , we can approximate the radiosity at a point x in our environment as a linear combination of the radiosities of each node or

$$\tilde{B}(x) = \sum_{i=1}^n B_i f_i(x) \quad (\text{EQ 1})$$

2.2 Element Construction

Due to its compatibility with triangulation and orientation independence using Gouraud shading, we have chosen the triangle as our element. We also need to concern ourselves with the connectivity of this element. Our solution will be much more accurate if we align our mesh to the D^0 and D^1 discontinuities as described by [9] and [10]. We can obtain elements with C^0 continuity by using the same nodes on the boundary of adjacent elements [2], [4]. By definition, the C^0 elements can accurately model D^1 discontinuities. A D^0 discontinuity would result from surface intersection, discrete changes in emissivity or discrete changes in reflectivity. These can be modeled by aligning our edges to the discontinuities and duplicating the nodes along the edge [9]. Higher order discontinuities could be modeled by selectively enforcing higher derivative continuity across the common edges between adjacent elements, but this is quite complex [12] so we approximate them by using smaller elements.

3 Finite Element Methods

This section will give a very brief introduction to finite element mathematics to provide us with modifications needed for the radiosity algorithm to incorporate higher order elements and the previously discussed basis functions.

3.1 Residual Error

We start by reiterating an equation from [8] which describes the radiosity for all points in the environment

$$B(x) = E(x) + \rho(x) \int_{\Omega} ds \kappa(x, s) B(s) \quad (\text{EQ 2})$$

where

$$\kappa(x, s) = V(x, s) \frac{\cos \theta_i \cos \theta_j}{\pi r^2} \quad (\text{EQ 3})$$

Exact solutions to (EQ 2) are known only in the simplest of geometries [8]. The exact solution can be approximated using the linear combination in (EQ 1). Traditional radiosity methods can be thought of as having a constant basis function of $f_i(x) = 1$ for all points x inside patch i . These *constant basis* radiosity algorithms will not be reiterated. The method introduced in [11] uses linear basis functions centered on the vertices. We will be presenting algorithms for extending *polynomial basis* radiosity to higher order polynomials.

The traditional radiosity method assigns an emissivity and reflectivity to each patch. We can enhance our radiosity algorithms by claiming that the exact emissivity and reflectivity are also defined by an approximation function similar to (EQ 1). This would allow us to describe variations in emissivity and reflectivity up to the degree of the basis function. For the sake of brevity, we will assume that the

Address: P.O. Box 808, Livermore, Ca. 94550
email: roy@nrl.gov, max2@llnl.gov

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
©1993 ACM-0-89791-601-8/93/008...\$1.50

emissivity e_k and reflectivity ρ_k are constant across each individual surface, k .

We can replace $B(s)$ in (EQ 2) by our approximation function to obtain an approximate solution for $B(x)$. If our approximation is good, then the approximate solution for $B(x)$ and the value obtained by applying (EQ 2) at point x should be close. If our approximate solution is exact, the difference between these two approximations will be zero. This gives us a measure of the accuracy of our approximation and is defined as the *residual error*. More specifically, it is expressed as

$$r(x) = E(x) + \rho(x) \int_{\Omega} ds \kappa(x, s) \tilde{B}(s) - \tilde{B}(x) \quad (\text{EQ 4})$$

3.2 Method of Weighted Residuals

A general approximation technique is the method of weighted residuals. This technique requires the residual error to be orthogonal to a set $\{w_i(x)\}$ of weighting functions over the domain Ω . It was shown in [8] that the resulting equations could be expressed in matrix form as

$$[\mathbf{M} - \mathbf{K}] \mathbf{B} = \mathbf{E} \quad (\text{EQ 5})$$

where \mathbf{B} is a column vector containing the coefficients to our approximation and

$$\begin{aligned} M_{ij} &= \int_{\Omega} dx w_i(x) f_j(x) \\ K_{ij} &= \rho_k \int_{\Omega} dx w_i(x) \int_{\Omega} ds \kappa(x, s) f_j(s) \\ E_i &= e_k \int_{\Omega} dx w_i(x) \end{aligned} \quad (\text{EQ 6})$$

where k is the index of the surface supporting weight function w_i .

4 Higher Order Algorithms

We have presented a set of interpolation functions in section 2 and combined them with our radiosity integral using the finite element method in section 3. This gave us a matrix equation where each component of the matrix contained a weighting function. By replacing the weights with different functions we obtain the point collocation and Galerkin methods [8].

4.1 Point Collocation Method

The traditional gathering algorithm as well as the linear vertex radiosity method introduced in [11] are examples of the point collocation method. This method replaces the weighting function in (EQ 6) with the dirac delta [8]. This simplifies the \mathbf{M} in (EQ 5) to be the identity matrix and \mathbf{E} to be a column vector containing the emissivities at each node. The elements of \mathbf{K} have the value

$$K_{ij} = \rho_k \int_{\Omega} ds \kappa(x_i, s) f_j(s) \quad (\text{EQ 7})$$

The contents of the integral describes a differential area to weighted area form factor where the area is defined by the domain of the basis associated with node j . We will call this a differential area to basis form factor. This integral can be solved using the approach specified in [11]. The pseudocode is as follows

```
Initialize  $\mathbf{F}$  to 0
For each pixel  $h$  in hemicube
   $k$  = index of patch at  $h$ 
   $Q$  = point on surface of  $k$ 
  For each node  $j$  in patch  $k$ 
     $F_{ij} = F_{ij} + f_j(Q) \Delta_h$ 
  EndFor
EndFor
```

4.2 Galerkin Method

The Galerkin method replaces the weighting functions with the basis functions giving us the following definitions for the matrices of (EQ 5).

$$\begin{aligned} M_{ij} &= \int_{\Omega} dx f_i(x) f_j(x) \\ K_{ij} &= \rho_k \int_{\Omega} dx f_i(x) \int_{\Omega} ds \kappa(x, s) f_j(s) \\ E_i &= e_k \int_{\Omega} dx f_i(x) \end{aligned} \quad (\text{EQ 8})$$

We'll start by looking at the equation for M_{ij} . We only need to concern ourselves with the area where f_i and f_j are both non-zero. This will only occur if an element can be found which has the nodes i and j on the boundary or interior. Clearly this occurs if i equals j . We can easily compute M_{ij} by considering only the elements which contain node i and looking through that small set of elements for the elements which also contain node j . We then integrate across these elements individually and sum the results. The formula for 2-D change of variables from the global triangle to the representative element gives us the Jacobian determinant which is the area of the global element. Therefore, the integral across an element is the same as the integral across the representative element multiplied by the area of the element. The final result is a constant multiplied by the area of the global element. The constant is dependent upon the relative positions of the local nodes corresponding to i and j . We can store these constants in a matrix \mathbf{M}_c . This matrix is symmetrical, which is what we would expect by looking at the equation. The local node numbers for i and j correspond to the row and column of a location this matrix.

Solving for E_i follows a similar path. In this case, we must integrate across the domain of the basis function. We can form a vector \mathbf{E}_c which contains the integral of all of the local nodes across the representative element. To compute E_i , we look at each element which has node i , use the local node number as an index into \mathbf{E}_c , multiply that array element by the area and then add it to the current value of E_i . After we visit each element, we multiply our result by the node emission e_i .

Computing K_{ij} is slightly more involved. We know from (EQ 7) that the inner integral is a differential area to weighted area form factor. In the Galerkin case, the weighted area still corresponds to the domain of a basis function, but the differential area corresponds to some point x in the domain of f_i . We will express this differential area to basis form factor as F_{xj} . Our equation simplifies to

$$K_{ij} = \rho_k \int_{\Omega} dx f_i(x) F_{xj} \quad (\text{EQ 9})$$

The contents of the integral describe a basis to basis form factor. This integral is in a form that is appropriate for Gaussian quadrature [2], [16]. The problem of computing K_{ij} is now reduced to computing a set of form factors, adding the results multiplied by the appropriate weight and multiplying by the reflectivity.

To compute the basis to basis form factor with gaussian quadrature we start by specifying the degree of precision [2]. This provides us with a collection of gauss points on each element. We compute an array of differential area to basis form factors FF (computed by the algorithm in section 4.1) at each of the gauss points. This hemicube will affect the basis to basis form factors associated with each node in the element. After we have completed computing the entire matrix of form factors, we multiply each row by the reflectivity to obtain \mathbf{K} . This gives us the following algorithm

```

Initialize F to 0
For each patch p
  ap = area of patch p
  For each gauss point l
    Ql = global coordinate of point l
    wl = weight assigned to point l
    FF = array of form factors computed at Ql
    For each node i in p
      For j = 1 to total number of nodes
        Fij = Fij + ap × wl × fi(Ql) × FF(j)
      Endfor
    Endfor
  Endfor
Endfor

```

The matrix **M** is very sparse. To avoid using an excessive amount of memory due to random access, the matrix is computed one row at a time. This requires us to visit a node and find the patches which share this node. The winged edge data structure allows us to easily determine adjacent elements. The algorithm for computing **M** and **E** is as follows.

```

Initialize M and E to 0
For each node i
  P = set of patches containing i
  For each patch p ∈ P
    ap = area of patch p
    J = set of nodes in p
    li = local node number of node i
    For each j ∈ J
      lj = local node number of node j
      Mij = Mij + ap × Mc(li, lj)
    Endfor
  Endfor
  Ei = Ei + ap × Ec(li)
Endfor

```

To solve for **B**, we multiply each row of the form factor matrix by the reflectivities and solve the matrix using the Gauss-Seidel iteration method.

5 Analysis

A quantitative measurement of the accuracy of our algorithms are obtained by applying an error metric. We will apply this metric to images generated by our collocation and Galerkin algorithms.

5.1 Error Metric

We determine the RMS radiosity reconstruction error by rendering each surface individually at the same distance. Our reference image was obtained by using quadratic basis elements on a discontinuity meshed version of our scene and applying a hemicube with a resolution of 314 × 314 × 157. Since 157 is a prime number, the chance of a correlation with a lower resolution hemicube is reduced. The floating point radiosity of each pixel is compared to the same pixel in each rendered surface of the reference. The error is measured using

$$\sqrt{\frac{\sum_{i=1}^n \left(\frac{r_i - e_i}{r_i} \right)^2}{n}} \quad (\text{EQ 10})$$

where r_i is the radiosity value at pixel i in the reference scene, e_i is the radiosity at pixel i in a test scene and n is the total number of pixels occupied by a rendered image of every surface. Computing the error in this manner reduces the chance of bias since it is improbable that any particular node in our scene will be on a pixel center.

5.2 Collocation Results

The collocation algorithm was implemented on a Cray YMP/C90. Timing information was obtained using the Unix *times* function. The computation time was considered to be the time spent executing the code added to the time spent completing system calls. The total time

includes the time spent generating and solving the matrix. We do not include I/O times. We could not include the time spent generating the mesh since some of the following meshes were generated by hand. Because of its flexibility, Heckbert's software z-buffer [7] was used to project the environment onto the hemicube. Although the program computed the radiosity for the red, green and blue components, only the blue component was used to determine the error.

We start by analyzing the source of error in Figure 1. Most of the error in this image is due to the shadow edges on the floor. By rendering the radiosity of the floor as a 3-D shaded surface, we enhance the radiosity discontinuities that would not be visible when the scene is rendered as an image. Figure 1 also shows the radiosity of the floor of the reference scene rendered in this manner. This gives us more information about where errors occur as well as how close our approximation is to a converged reference.

The reference image appeared to be extremely smooth. However, when we looked at the floor rendered as a shaded surface, slight discontinuities due to hemicube aliasing were detected. These artifacts are referred to as plaid patterns in [1] and [15]. When the edge of a light source is parallel or at a 45 degree angle to the edge of the hemicube, the amount of aliasing is greatly enhanced. In some scenes it may be possible to determine an ideal rotation for the hemicube in order to reduce aliasing, but when we introduce occlusion, the apparent edge of a source changes. In general, we can reduce the chance of a poor alignment by introducing a random rotation to the hemicube. This gives a slight improvement in terms of numerical error and a big improvement in terms of visually perceptible error. To improve our reference even further, we solved for the radiosity several times and averaged the results.

We applied a uniform and discontinuity mesh to Figure 1. By increasing the resolution of a mesh, the amount of computation time increases as the error decreases. We did not have access to triangulation software that would easily allow us to create a variable sized discontinuity mesh. An interactive mesh generator called Maze [6] was used to produce a set of quadrilaterals which were then split into triangles. One of the goals used in producing this mesh was to limit the number of slices or poorly formed elements. Once the mesh was created, we were easily able to further subdivide the resulting triangles to improve the accuracy of our solution.

Figure 2 shows the results of the algorithm when applied to the scene shown in Figure 1 with a 100 × 100 × 50 hemicube. A log error of less than -1.3 generated an image which was very difficult to distinguish from the reference. A log error of less than -1.6 generated an image which could not be distinguished from the reference even with high quality display devices. A log error of approximately -2 was mostly due to hemicube aliasing. Note that the linear and constant uniform elements did not obtain these error levels in the time frame shown. The effect on the form factors due to visibility changes is basically quadratic, so we did not expect or see a great deal of improvement in the cubic element over the quadratic element. Discontinuity meshing showed the most impressive results. The linear discontinuity elements produced the same error as the best uniform elements in half the time. The quadratic and cubic discontinuity elements produced an error level so small that further reduction could only be obtained by increasing the resolution of the hemicube.

The collocation algorithm was applied to other simple scenes. In some cases, even the higher order C^0 elements did not conform well to the radiosity solution along edges of high variance. These edges can be found near dimly lit corners of a closed room.

5.3 Galerkin Results

The Galerkin method was also implemented on the Cray YMP/C90. This method required only minor modifications to the existing collocation algorithm. The program was implemented so that the user could specify the number of degrees of precision. It was shown in [14] that in the case of the uniform mesh, the optimal degrees of

precision for the constant, linear, quadratic and cubic basis were one, one, four and four, respectively. At these levels the algorithm produced less error in less time. We also found that the discontinuity mesh produced these same optimal degrees of precision. The constant basis Galerkin algorithm is identical to the constant basis collocation algorithm [14], so we don't present these results.

Figure 2 shows the results of applying the Galerkin method using a uniform and discontinuity mesh for the scene in Figure 1. We used the same $100 \times 100 \times 50$ resolution hemicube for these tests. The quadratic and cubic basis achieved a lower error than the linear for both meshes. The quadratic and cubic discontinuity mesh again produced very small error levels immediately. In comparison to the collocation method, the Galerkin method as we implemented it took considerably more time to achieve the same error level.

In general, the number of patches for a scene with triangular elements is larger than the number of vertices. Since the number of hemicubes used in Gaussian quadrature depends on the number of patches, our Galerkin implementation required many more hemicubes to compute the matrices for the same scene. Figure 1 was computed using the collocation method with a quadratic basis. This required 2282 hemicubes. Our Galerkin implementation would require 3042 hemicubes if we used just 2 degrees of precision.

6 Conclusions and Future Work

We have presented two radiosity algorithms which use the finite element method and higher order basis functions to produce a more accurate solution in less time. The collocation method proved to be easier to implement and converged faster than the Galerkin method for the scenes presented. By applying discontinuity meshing, our algorithms computed an accurate solution in only a fraction of the time used by traditional methods.

Applying adaptive meshing to these algorithms could present a challenging problem. In a more traditional approach, we would subdivide our patches if the gradient became too large. The higher order

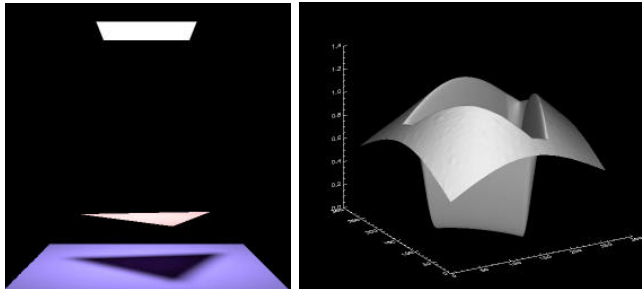


Figure 1 Reference scene used for convergence test. On the left is the rendered scene. On the right is a height field showing the radiosity of the floor.

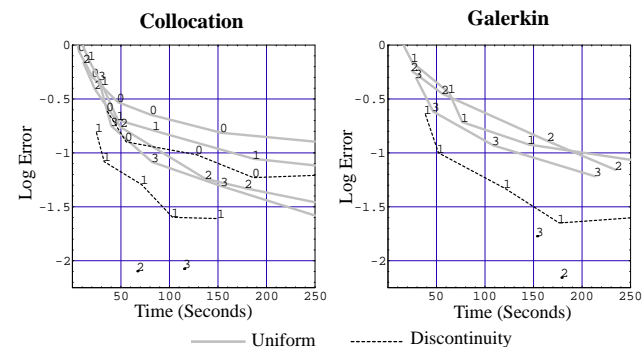


Figure 2 Error plots. The solid lines are the results from the uniform mesh. The dashed lines and single points are from the discontinuity mesh. The numbers indicate the degree of the basis functions. The error is reduced by increasing the number of elements.

elements can model these large gradients. The criteria for subdivision would require some investigation [10].

We restricted our elements to planar C^0 triangles. We could use square or even curved elements. The curved element may require ray tracing for computing the form factors, but it would allow us to use exact geometries. An additional improvement to the accuracy might be possible by using C^1 reconstruction technique [12].

Modifications to the algorithms for progressive radiosity is possible. The collocation method could be used with a ray tracing algorithm similar to [15]. In this case we would have to sample the entire basis domain rather than a single patch. Creating an efficient algorithm would be another puzzle.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

- [1] Baum, Daniel R., Holly E. Rushmeier, James M. Winget, *Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors*, Computer Graphics 23(3), July 1989
- [2] Burnett, David S., *Finite Element Analysis*, Addison Wesley Publishing Co., Reading, Massachusetts, May 1988
- [3] Cohen, Michael F., Donald P. Greenberg, *The Hemicube: A Radiosity Solution for Complex Environments*, Computer Graphics 19(3), July 1985
- [4] Farin, Gerald, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1990.
- [5] Goral, Cindy M., Kenneth E. Torrance, Donald P. Greenberg, Bennett Battaile, *Modeling the Interaction of Light Between Diffuse Surfaces*, Computer Graphics 18(3), July 1984
- [6] Hallquist, John O., *MAZE - An Input Generator for DYNA2D and NIKE2D*, LLNL Tech. Report, UCID-19029, Rev. 2.
- [7] Heckbert, Paul S., *Generic Convex Polygon Scan Conversion and Clipping*, Graphics Gems, Academic Press, 1990
- [8] Heckbert, Paul S., James M. Winget, *Finite Element Methods for Global Illumination*, U.C. Berkeley, Jan. 1991
- [9] Heckbert, Paul S., *Discontinuity Meshing for Radiosity*, Third Eurographics Workshop on Photorealism, Consolidation Express, Bristol, England, May 1992
- [10] Lischinski, Dani, Filippo Tampieri, Donald P. Greenberg, *Combining Hierarchical Radiosity and Discontinuity Meshing*, Computer Graphics, Annual Conf. Series, Aug. 1993
- [11] Max, Nelson L., Michael J. Allison, *Linear Radiosity Approximations using Vertex-to-Vertex Form Factors*, Graphics Gems III, Academic Press, 1992
- [12] Salesin, David, Dani Lischinski, Tony DeRose, *Reconstructing Illumination Functions with Selected Discontinuities*, Third Eurographics Workshop on Photorealism, Consolidation Express, Bristol, England, May 1992
- [13] Siegel, Robert, John R. Howell, *Thermal Radiation Heat Transfer*, McGraw-Hill Book Co., N.Y., 1972
- [14] Troutman, Roy, *Parallel Radiosity Algorithms using Higher Order Finite Elements*, Master's thesis, U.C. Davis, Dec. 1992
- [15] Wallace, John R., Kells A. Elmquist, Eric A. Haines, *A Ray Tracing Algorithm For Progressive Radiosity*, Computer Graphics 23(3), July 1989
- [16] Zats, Harold R., *Galerkin Radiosity: Higher Order Global Illumination*, Computer Graphics, Annual Conf. Series, Aug. 1993