# Combining Hierarchical Radiosity and Discontinuity Meshing

*Dani Lischinski* *Filippo Tampieri* *Donald P. Greenberg*

Program of Computer Graphics
Cornell University
Ithaca, NY 14853

## ABSTRACT

We introduce a new approach for the computation of view-independent solutions to the diffuse global illumination problem in polyhedral environments. The approach combines ideas from hierarchical radiosity and discontinuity meshing to yield solutions that are accurate both numerically and visually. First, we describe a modified hierarchical radiosity algorithm that uses a discontinuity-driven subdivision strategy to achieve better numerical accuracy and faster convergence. Second, we present a new algorithm based on discontinuity meshing that uses the hierarchical solution to reconstruct an object-space approximation to the radiance function that is visually accurate. Our results show significant improvements over both hierarchical radiosity and discontinuity meshing algorithms.

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

**Additional Key Words and Phrases:** diffuse reflector, discontinuity meshing, global illumination, hierarchical radiosity, Mach bands, photorealism, quadratic interpolation, radiance function, radiosity, reconstruction, shadows, view-independence.

## 1 INTRODUCTION

Computing solutions to the global illumination problem is an essential part of photorealistic image synthesis. In this paper, we are interested in computing *view-independent* (or *object-space*) solutions for global illumination. Such solutions provide an approximation to the radiance function across each surface in the environment. Once a solution is computed, images from any viewpoint can be rendered with a relatively small additional effort. These methods are particularly attractive for applications such as architectural design, interior design, lighting design, illumination engineering, and virtual reality, in which the need for multiple views or walk-throughs of static environments arises.

So far, most view-independent methods have been derived from the radiosity method that was originally developed to solve radiative heat transfer problems [23]. Computer graphics researchers adopted this method to compute the global illumination of diffuse polyhedral environments [10, 7, 19]. Radiosity has been extended and improved dramatically since, but there is still much to be done before the method can become a useful tool for its intended users.

The goal of our research is to develop an efficient radiosity system that satisfies the following requirements:

**Objective (numerical) accuracy:** Solutions produced by the system should converge rapidly to the exact solution. This requirement may seem obvious, however, in the computer graphics community results of simulations are too often judged solely by their visual appearance.

**Subjective (visual) accuracy:** While visual appearance should not be used to judge the objective accuracy of the simulation, it is still very important, since the image is the final product. Clearly, accurate visual appearance can be achieved through numerically accurate simulation (if the underlying model is physically accurate.) Unfortunately, experience has shown that the human visual system is extremely sensitive to small perceptual errors that are difficult to quantify. The simulated environments can be very complex and, therefore, the computation of ultra-accurate solutions is generally impractical. Thus, we must have means of producing visually acceptable images even from coarse solutions.

**Ease of control:** (i) The system should be controllable by users who are not necessarily familiar with its inner workings. Therefore, the control parameters should be intuitive and small in number. (ii) In many cases (such as early design stages) the user is interested in a quick solution, even if not exceedingly accurate. At other times, one might be willing to wait overnight for a reliable solution. Therefore, the system should provide the user with the option to trade speed for accuracy.

Most radiosity systems do not satisfy any of these requirements. There are no error bounds on the solutions, because approximations are often used without justifications regarding their impact on the accuracy of the results. The resulting images typically exhibit many visual artifacts such as Mach bands, light and shadow leaks, jagged shadow boundaries, and missing shadows. Radiosity systems are seldom user-friendly and require massive user intervention: typically, a time consuming trial-and-error process is required to produce an image that looks right. Baum *et al.* [1] and Haines [12] provide good discussions of the various pitfalls of radiosity.

In this paper we present a new radiosity method, which comes closer to satisfying our goals. The new method combines two recently developed approaches: hierarchical radiosity [14] and discontinuity meshing [15, 18]. First, we present an improved hierarchical radiosity algorithm that uses a discontinuity-driven subdivision strategy to achieve better numerical accuracy and faster convergence. Second, we describe a new algorithm based on discontinuity meshing that uses the hierarchical solution to reconstruct a visually accurate approximation to the radiance function. Thus, results of high visual quality can be obtained even from coarse global illumination simulations. Previous attempts to improve the visual quality of radiosity solutions were described by Nishita and Nakamae [19], Kok and Jansen [17], Chen *et al.* [4], and Reichert [20]. In all of these cases, however, the improvement takes place in image space, after the view and the resolution have been specified. Our method, instead, operates entirely in object space, and the improved solution is view-independent.

## 2 HIERARCHICAL RADIOSITY

The traditional radiosity approach [10, 7] discretizes the environment into $n$ elements and solves a linear system of $n$ equations, where the radiosities of the elements are the unknowns. The most serious drawback of this approach is the need to compute the $O(n^2)$ coefficients of the linear system, corresponding to the interactions (transfers of light energy) between pairs of elements. In addition to the overwhelming computational complexity, most of these computations are performed to unnecessarily high accuracy, while some are not sufficiently accurate.

Hierarchical radiosity (HR) [14] overcomes these problems by decomposing the matrix of interactions into $O(n)$ blocks, for a given accuracy. These blocks correspond to interactions of roughly equal magnitude, and the same computational effort is required for computing each block. HR operates by constructing a hierarchical subdivision of each input surface. Each node in the hierarchy represents some area on the surface. Two nodes are linked together if the interaction between their corresponding areas can be computed within the required accuracy; otherwise, the algorithm attempts to link their children with each other. Each link corresponds to a block in the interaction matrix.

HR has several important advantages: it is fast, the errors in its approximations are bounded, and it is controlled by only two parameters: the error tolerance and the minimum node area. The smaller the values of these parameters, the more accurate (and expensive) the solution becomes. Thus, HR satisfies our goals of objective accuracy and ease of control.

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

The number of links created by HR is $O(n + m^2)$ where $n$ is the final number of nodes and $m$ is the number of input surfaces. As the complexity of the environment increases, the $m^2$ term eventually becomes dominant, drastically reducing the efficiency of the algorithm. As pointed out by Smits *et al.* [22], this problem could be solved by grouping the input surfaces into higher level clusters. This is an interesting research topic by itself, and it will not be pursued in this paper.

## 3 DISCONTINUITY MESHING

Radiosity methods typically attempt to approximate the radiance function with constant elements and use linear interpolation to display the result. The actual radiance function, however, is neither piecewise constant nor piecewise linear. It is usually smooth, except along certain curves across which discontinuities in value or in derivatives of various order may occur. Discontinuities in radiance functions are discussed in detail elsewhere [16, 15, 18]; what follows is a brief summary of the various types of discontinuity and their causes.

The most significant discontinuities are discontinuities in the radiance function itself (denoted $D^0$). They occur along curves of contact or intersection between surfaces. Discontinuities in the first and the second derivatives ($D^1$ and $D^2$, respectively) occur along curves of intersection between surfaces in the environment and *critical surfaces* corresponding to qualitative changes in visibility, or *visual events*. Visual events in polyhedral environments can be classified into two types [9]: EV events defined by the interaction of an edge and a vertex, where the critical surface is a planar wedge; and EEE events defined by the interaction of three edges, where the critical surface is a part of a quadric. Discontinuities of higher than second order are also possible [16].

Discontinuities are very important both numerically and visually: all the boundaries separating unoccluded, penumbra, and umbra regions correspond to various discontinuities. When a discontinuity curve crosses a mesh element, the approximation to the radiance function over that element becomes less accurate. The resulting errors usually correspond to the most visually distracting artifacts in radiosity images. The traditional radiosity approach uses adaptive subdivision [8] to reduce these errors, however there are several problems with this approach. First, the user must specify an initial mesh that is sufficiently dense, or features will be lost. Second, the shape of the mesh is determined by the geometry of the surface being meshed, and the discontinuities are not resolved exactly. As a result, many small elements are created as the method attempts to converge to shadow boundaries. Furthermore, although the resulting solution may be of adequate visual quality for some views, artifacts may become visible as the view changes (e.g., when we zoom in on a surface.)

Discontinuity meshing (DM) algorithms compute the location of certain discontinuities and represent them explicitly, as boundaries, in the mesh. This leads to solutions which are both numerically and visually more accurate. Another advantage is that higher order elements can be used much more effectively in conjunction with discontinuity meshes [16]. Several algorithms have been described that use the idea of discontinuity meshing to various extents [1, 3, 6, 15].

Recently, a progressive radiosity DM algorithm was described by the authors [18]. The meshing in this algorithm is automatic. Using analytical visibility and form factor computations followed by quadratic interpolation it has produced radiosity solutions of impressive visual accuracy. This algorithm was also shown to be numerically accurate [24].

However, this method is too expensive for computing converged solutions of complex environment and only offers limited user control in trading off speed for accuracy. The main reason for this is that all energy transfers are computed very accurately, regardless of their magnitude.

## 4 A COMBINED APPROACH

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

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

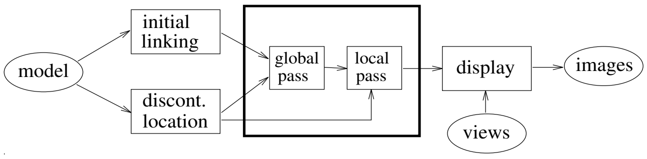


Figure 1: The structure of the new radiosity system

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

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

To test our approach we have implemented a new radiosity system whose overall structure is shown in Figure 1. The global and the local passes are discussed in detail in the next two sections. In the rest of this section we briefly describe the remaining parts.

The initial linking stage creates for each input polygon a list of links to all the polygons that are visible from it. For each link it is determined whether the two polygons are completely or partially visible to each other. This creates a starting point for the global pass, which proceeds to refine these links as needed. We test visibility between two polygons using a combination of shaft-culling [13] and the ray-tracing algorithm that Hanrahan *et al.* [14] used.

The discontinuity location stage computes the location of all the $D^0$ discontinuities, since these are typically responsible for the most severe errors (both numerically and visually.) In most environments the direct illumination by primary light sources is responsible for the most perceptible illumination details. Therefore, all of the $D^1$ and $D^2$ discontinuities caused by EV events involving the primary light sources are computed as well. The computed discontinuities are henceforth collectively referred to as *primary* discontinuities.

EEE events are more difficult to handle because their corresponding critical surfaces are curved, rather than planar. However, the resulting discontinuities always lie within penumbra regions, and never define the outer boundaries of a shadow. For these reasons, we excluded EEE events from our current implementation.

We described the discontinuity location algorithm in a previous paper [18]. Tampieri [24] provides a more detailed description of this algorithm. Heckbert [15] and Teller [25] describe alternative algorithms for locating discontinuities. Teller's algorithm is the only one capable of handling EEE events.

## 5 THE GLOBAL PASS

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

$$B_{rs}(x) \sim \hat{B}_{rs} = \rho_r B_s F_{rs} V_{rs}$$

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

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

$$E_{rs} = \sup_{x \in r} \left| B_{rs}(x) - \hat{B}_{rs} \right| \qquad (1)$$

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

$$\begin{array}{ll} B_s^{\min} = \inf_{x \in s} B_s(x) & B_s^{\max} = \sup_{x \in s} B_s(x) \\ F_{rs}^{\min} = \inf_{x \in r} F_{xs} & F_{rs}^{\max} = \sup_{x \in r} F_{xs} \\ V_{rs}^{\min} = \inf_{x \in r} V_{xs} & V_{rs}^{\max} = \sup_{x \in r} V_{xs} \end{array}$$

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

$$\left[\rho_r B_s^{\min} F_{rs}^{\min} V_{rs}^{\min},\, \rho_r B_s^{\max} F_{rs}^{\max} V_{rs}^{\max}\right]$$

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

$$E_{rs} \leq \rho_r \left( B_s^{\max} F_{rs}^{\max} V_{rs}^{\max} - B_s^{\min} F_{rs}^{\min} V_{rs}^{\min} \right) \qquad (2)$$

Three main factors affect the magnitude of the error:

1. the variation of the radiosity on the source node $s$
2. the variation of the form factor across the receiver node $r$
3. the variation in the visibility of the source from the receiver

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

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

Clearly, it would be to our advantage to use a subdivision strategy that would result in as many totally visible or totally occluded pairs, as quickly as possible. Since discontinuity lines on the receiver correspond to abrupt changes in the visibility of the sources [16, 18], subdividing the receiver along these lines should quickly resolve partial occlusion.

We have modified the HR algorithm to perform discontinuity-driven subdivision instead of regular subdivision. There are two main changes in the data structures used by the new algorithm: first, we store with each node a list of all the discontinuity segments on the corresponding polygon; second, we use a 2D binary space partitioning (BSP) tree [3] instead of a quadtree to represent the hierarchical subdivision of each initial polygon, since BSP trees allow for subdivision of polygons along arbitrarily oriented lines. Pseudocode for subdividing a node is given in Figure 2.

When a node is subdivided we choose one of its discontinuity segments and split the node using the corresponding line equation. The segment is chosen such that the split is as balanced as possible. Priority is given to $D^0$ discontinuities over higher order ones, since the former typically bound areas totally occluded from the rest of the

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

Figure 2: Pseudocode for the *Subdivide* routine

environment. The subdivision is completed by splitting the list of segments into two new lists, one for each child. If no segments are stored with the node, we split the node by connecting the midpoint of the longest edge to a vertex or another midpoint chosen so that the resulting children have roughly equal areas.

### 5.1 Results

Figure 4 demonstrates the improved hierarchical algorithm using a simple environment illuminated by two small triangular light sources. A 3D view of the environment is shown in image a1. The radiance function on the floor polygon is shown in image a2. Image a3 shows the discontinuity segments on the floor. $D^0$ discontinuities are drawn in red; $D^1$ and $D^2$ discontinuities in yellow. In rows b and c, we compare the subdivision produced by the discontinuity-driven algorithm to the one produced by regular subdivision. The level of subdivision shown increases from left to right: the leftmost pair shows the subdivision at level 2, then level 4, 6, and 8.

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

In order to compare the rates of convergence of the two strategies we computed a set of approximations to the direct illumination on the floor using a successively larger number of elements. Figure 3 shows the RMS and the maximum absolute errors versus the number of elements for the two strategies. These errors were computed with respect to an analytical solution at the vertices of a 400 by 400 grid on the floor. All the values were scaled to set the maximum brightness on the floor to 1.

Our algorithm converges faster in both error metrics. Note that the convergence of the regular subdivision is particularly poor in the maximum absolute error metric. The reason is that there are $D^0$ segments on the floor that are not aligned with the subdivision axes. Thus, there are always elements that are partially covered by the pyramid while the remaining part is brightly illuminated by the light sources. The algorithm assigns a single constant value to each

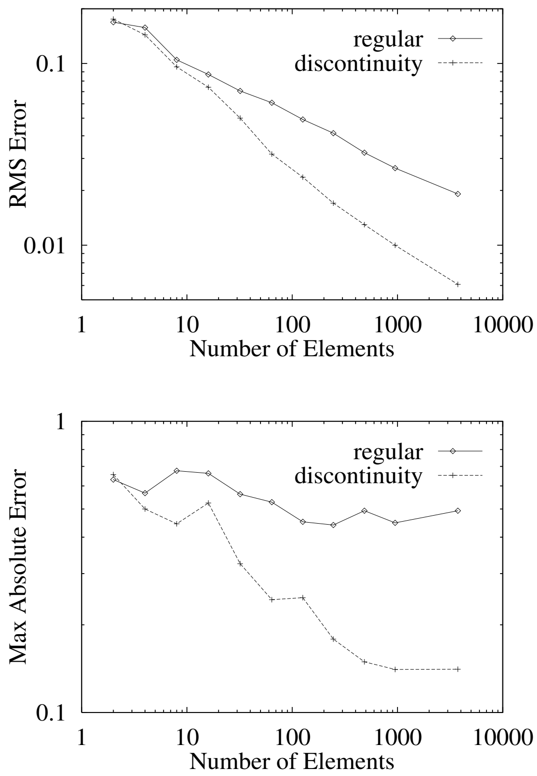


Figure 3: A comparison of errors between the two subdivision strategies using log-log plots

such element, and this results in a large error there. Our algorithm, on the other hand, resolves $D^0$ discontinuities and therefore does not suffer from this problem.

In the RMS error metric regular subdivision does converge, because the elements that contain the errors become progressively smaller, and this is accounted for by the metric; however, the convergence is slower.

## 6 THE LOCAL PASS

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

Our experience with discontinuity meshing [18] has shown that reproducing the discontinuities in the radiance function, while maintaining a smooth approximation elsewhere is key to achieving visual accuracy, especially when multiple views of the same solution are to be rendered. Therefore, we construct a discontinuity mesh containing the precomputed primary discontinuities for each polygon. Mesh nodes are assigned radiance values using the hierarchical solution. This mesh is then used for the shaded display of the environment. Thus, the local pass essentially performs an additional light gathering operation over the environment. However, instead of gathering to the nodes in the hierarchy, we gather to the elements of the discontinuity mesh.

The discontinuity mesh is constructed using constrained Delaunay triangulation (CDT) [5]. The Delaunay triangulation (DT) of a point set maximizes the minimum angle over all possible triangulations of that set and has a number of other desirable properties [2].

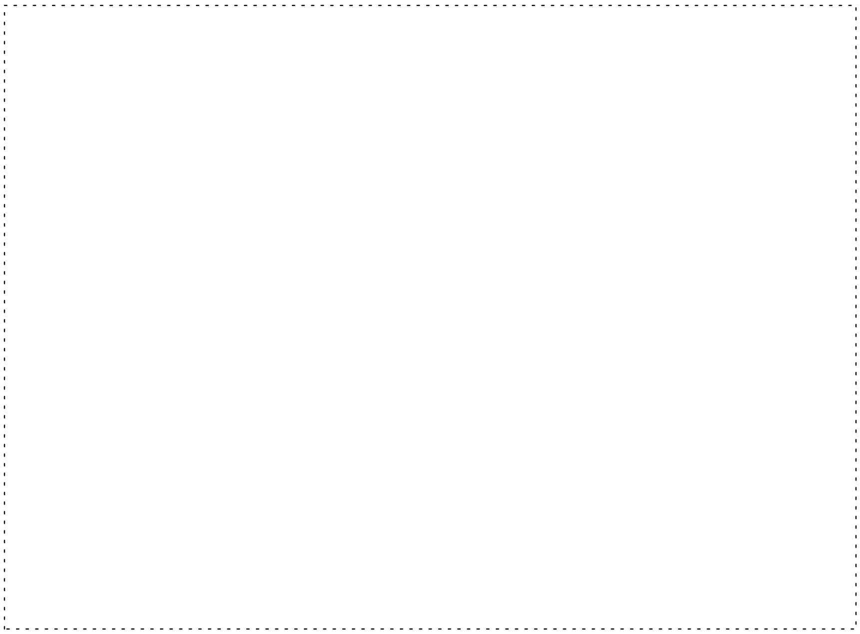


Figure 4: Discontinuity-driven vs. regular subdivision

These properties are important because they result in well-shaped elements that yield more accurate approximations and reduce visual artifacts during display [1]. CDT takes as input a point set and a set of edges connecting some of the points, and creates a triangulation of the points that is constrained to include all the input edges. CDT preserves the properties of DT over all the constrained triangulations. We have implemented an incremental CDT algorithm that is a simple extension of the incremental DT algorithm described by Guibas and Stolfi [11]. An alternative easy-to-implement algorithm is described in the excellent survey by Bern and Eppstein [2].

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

The radiance across each triangle is approximated using a standard quadratic element commonly used in finite element methods [27]. Six radiance values are computed for each element: three at the vertices, and three at the edge midpoints. Except for $D^0$ edges, these values are shared between adjacent faces (our CDT algorithm constructs a topological data structure suitable for such information sharing [11].) The six values are then interpolated by a quadratic bivariate polynomial. This scheme yields a $C^0$ piecewise quadratic interpolant to the radiance on each polygon. This interpolant was found to provide approximations that look smoother and are less prone to Mach bands than the traditional piecewise linear interpolation [18]. Salesin *et al.* [21] describe a piecewise cubic interpolant that can be used instead, if $C^1$ interpolation is desired.

To obtain a radiance value at a point $x$ we use the information available to us from the hierarchical solution. Below we describe four different methods that we have experimented with. Pseudocode for the last three methods is given in Figure 5.

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

The accuracy of the resulting value depends on the accuracy of the global pass solution. Consider the path from the root of the hierarchy to the leaf containing the point $x$. Every node along this pass has zero or more links to other nodes, representing areas on primary or secondary sources that illuminate $x$. The error at $x$ due to one such link between a containing node $r$ and an illuminating node $s$ is bounded by equation (2). The total error at $x$ is the sum of the errors over all the contributing links.

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

$$E_{rs}(x) \leq \rho_r F_{xs} \left( B_s^{\max} V_{rs}^{\max} - B_s^{\min} V_{rs}^{\min} \right) \tag{3}$$

**Method C.** The next logical step is to recompute both the form factor and the visibility of each source as seen from $x$. In order to obtain an accurate visibility value the visible parts of the source are computed analytically [18]. As a result, the error bound shrinks further:

$$E_{rs}(x) \leq \rho_r F_{xs} V_{xs} \left( B_s^{\max} - B_s^{\min} \right) \tag{4}$$

However, the computation becomes more expensive.

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

```
Spectrum Shade(node, x)
rad ← 0
foreach l ∈ node.links do
    ff ← FormFactor(x, l.source)
    v ← Visibility(x, l)
    rad ← rad + ff * v * l.source.radiosity
end for
if IsInterior(node) then
    if Contains(node.left, x) then
        rad ← rad + Shade(node.left, x)
    else if Contains(node.right, x) then
        rad ← rad + Shade(node.right, x)
    else
        rad ← rad + 0.5 * (Shade(node.left, x)
                          +Shade(node.right, x))
    end if
end if
return rad

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

Figure 5: Pseudocode for the *Shade* routine

### 6.1 Results

We compared methods A, B, C, and D using a simple model of a square exhibit room displaying a modern sculpture illuminated by two small square light sources.

Three global pass solutions of the exhibit room are shown at the top row of Figure 6, in order of increasing accuracy starting from the left. For each solution, the elements (leaf nodes) of the hierarchical subdivision are shown as flat shaded, outlined polygons. The bottom row of the same figure shows the corresponding local pass meshes. Table 1 reports statistics for both passes.

The results of the global pass were fed to the local pass four times, once for each of the methods A, B, C, and D, yielding a total of twelve radiosity solutions shown in Figure 7. Columns 1, 2, and 3 were computed respectively from the low, medium, and high accuracy global pass solutions shown in Figure 6. Each row corresponds to a different shading strategy starting with method A for the top row.

As demonstrated in the top row, method A is prone to visual artifacts: the shading on walls is flat or not sufficiently smooth; some shadows are entirely missing (image A1), while others have incorrect boundaries. These artifacts are the result of interpolating radiance values obtained by sampling the piecewise constant global pass solution.

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

| | Solution Accuracy | | |
|---|---|---|---|
| | low | medium | high |
| input polygons | 47 | 47 | 47 |
| disc. segments | 559 | 559 | 559 |
| initial links | 652 | 652 | 652 |
| total links | 720 | 1316 | 21805 |
| total nodes | 147 | 803 | 6041 |
| total leaf nodes | 97 | 425 | 3044 |
| CDT elements | 1538 | 2384 | 8177 |
| shading calls | 3799 | 5674 | 17984 |
| initial linking | 6 | 6 | 6 |
| discontinuity comp. | 1 | 1 | 1 |
| hierarchical sol. | 1 | 4 | 60 |
| triangulation | 0.53 | 0.81 | 2.78 |
| method A | 1 | 2 | 13 |
| method B | 6 | 10 | 80 |
| method C | 405 | 624 | 2633 |
| method D | 20 | 26 | 110 |

Table 1: Statistics for images in Figures 6 and 7. Timings are in seconds for execution on an HP 9000/720 workstation.

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

Method D yields results that are almost indistinguishable from those given by method C. However, as can be seen from the timings reported in Table 1, method D takes only a fraction of the time required by method C. In fact, it is not much more expensive than method B.

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

When comparing the computation times reported in Table 1, it can be seen that the local pass is in most cases costlier than the global pass. It may be argued that the time used by the local pass could be better spent in further refinement of the subdivision hierarchy in the global pass. One might expect that if the hierarchy were sufficiently refined, even a very simple shading strategy would have sufficed for visually accurate results. Figure 7, however, demonstrates that this is not the case. Image D2, computed from the medium accuracy global pass followed by method D for the local pass, is visually more accurate than images A3 and B3; yet, it took considerably less time to compute (38 versus 83 and 150 seconds, respectively.)

Another set of comparisons was made to illustrate the importance of including discontinuity segments in the mesh for the local pass. Figure 8 shows a view of the floor of the exhibit room. The top row shows the mesh in wireframe with $D^0$ discontinuities in red and $D^1$ and $D^2$ discontinuities in yellow. The bottom row shows the shaded floor as reconstructed by the local pass. All images were computed from the medium accuracy global pass solution shown in image a2 of Figure 6 and all of them used method D in the local pass. As can be seen from the top row of Figure 8, no discontinuity segments were included in the left mesh, only $D^0$ discontinuities were included in the middle mesh, and all the discontinuity segments were included in the right mesh.

When comparing the corresponding images in the bottom row, the higher quality of the right image stands out. Image b1 presents many


Figure 6: Exhibit Room. Global pass solutions (top row) and the corresponding local pass meshes (bottom row). The accuracy of the solutions increases from left to right.

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

| | Discontinuities in the Mesh | | |
|---|---|---|---|
| | none | $D^0$ | $D^0$ $D^1$ $D^2$ |
| triangulation | 0.39 | 0.39 | 0.81 |
| shading | 9 | 10 | 26 |
| disc. segments | 0 | 36 | 559 |
| CDT elements | 1170 | 1190 | 2384 |
| shading calls | 2739 | 3027 | 5674 |

Table 2: Statistics for the comparison of meshing strategies shown in Figure 8. Timings are in seconds for execution on an HP 9000/720 workstation.

As the statistics reported in Table 2 show, building a mesh that incorporates discontinuity segments takes longer than building one without discontinuities. Furthermore, including the discontinuities generally results in a larger number of elements and consequently shading the mesh takes longer. We believe, however, that the increased computation time is well justified.

## 7 A FINAL COMPARISON

In this section we demonstrate the performance of our combined approach on an environment of moderate complexity (1,688 input polygons.) Figure 9 shows a rendered view of the scene. There are two primary light sources: a small distant polygonal source outside the room simulates sunlight, and another polygonal source close to the ceiling provides the artificial illumination.

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

As can be expected in a complex environment, the initial linking stage results in a very large number of initial links, most of which represent interactions of very small magnitude. For efficiency, we use a simple culling strategy: we ignore all the initial links that do not involve a primary light source and whose form factor falls below a user specified threshold. We found that by using a small threshold it is possible to eliminate most of the initial links, without any noticeable change in the resulting images. As was mentioned in Section 2, clustering of input surfaces in the initial linking stage should provide a more comprehensive solution to this problem.

Table 3 reports various statistics for the two solutions from which the images in Figure 9 were rendered. The two solutions have roughly the same number of final triangles, yet the HDMR solution looks dramatically better than the HR solution; while the latter exhibits many of typical problems of radiosity images, HDMR produces sharp shadow boundaries and correct penumbrae, eliminates shadow and light leaks, and captures some small features that are entirely missed by HR. Furthermore, the total computation time was almost twice as long for the HR solution.

We attempted to perform a similar comparison with our progressive DM algorithm [18]. However, we were not able to obtain a converged solution for this environment: after four hours of computation the DM algorithm was still in its fourth iteration.

### 7.1 Complexity of Discontinuity Meshing

A legitimate concern regarding discontinuity meshing is that, in theory, $l$ light source edges and $m$ polygon edges can result in $O(lm)$ distinct EV visual events. In the worst case, each event intersects $O(m)$ polygons, resulting in a total number of $O(lm^2)$ discontinuity segments. In such a case each polygon has $O(lm)$ discontinuity segments, which can result in as many as $O(l^2m^2)$ elements in the discontinuity mesh for that polygon.

We have found that this worst case analysis is too pessimistic in practice. Consider, for example, the environment shown in

A

B

C

D

1 2 3

Figure 7: Exhibit Room. A comparison of shading strategies. Columns 1, 2, and 3 were computed respectively from the low, medium, and high accuracy global pass solutions shown in Figure 6. Each row corresponds to a different shading strategy; starting from the top: method A, method B, method C, and method D.

Figure 9. In this environment $l$ is 8, and $m$ is 6,744. The worst case upper bound on the number of discontinuity segments on a single polygon is 215,808. In practice, there were 18,664 discontinuity segments in the entire environment, an average of roughly 11 segments per polygon. The highest number of segments on a single polygon (the floor) is 2,175, resulting in only 7,627 triangles in the floor's discontinuity mesh.

## 8 CONCLUSIONS

By combining hierarchical radiosity with discontinuity meshing we have created a new radiosity method that is superior to both of its ancestors: it is more accurate than the HR algorithm, both numerically and visually, and it is faster and more flexible than DM algorithms. The new algorithm is capable of producing high quality images even from quick simulations.

Hierarchical radiosity has been recently extended to deal with very complex environments by introducing the notion of importance into the solution process [22]. This improvement is readily applicable to our algorithm as well: the global pass would simultaneously solve for radiosity and for importance as described by Smits *et al.* [22]; the local pass would only reconstruct the radiance on surfaces which are direct receivers (or emitters) of importance.

There are several aspects of our algorithm that can be substantially improved:

**Visibility computations.** Our implementation uses shaft culling [13] to reliably determine complete visibility between polygons, but point sampling is used to determine whether two polygons are entirely occluded from each other. Our method could be improved by using the accurate and reliable visibility algorithms described by a

b

1 2 3

Figure 8: Exhibit Room Floor. A comparison of meshing strategies. Mesh (top row) and computed radiance (bottom row) on the floor using simple CDT (left), CDT with $D^0$ discontinuity segments (middle), and CDT with $D^0$, $D^1$, and $D^2$ discontinuity segments (right).

Figure 9: A comparison of Hierarchical Discontinuity Meshing Radiosity (left) vs. Hierarchical Radiosity (right)

| | Radiosity Algorithm | |
|---|---|---|
| | HDMR | HR |
| initial linking | 2 : 16 : 27 | 2 : 16 : 27 |
| discontinuity computations | 0 : 09 : 06 | 0 : 00 : 00 |
| hierarchical solution | 0 : 16 : 42 | 3 : 58 : 01 |
| triangulation | 0 : 00 : 21 | 0 : 00 : 16 |
| shading computations | 0 : 33 : 49 | 0 : 00 : 51 |
| total time (hr:min:sec) | 3 : 16 : 56 | 6 : 15 : 35 |
| input polygons | 1,688 | 1,688 |
| discontinuity segments | 18,664 | 0 |
| initial links | 165,814 | 165,814 |
| links after culling | 27,002 | 27,002 |
| total links | 39,056 | 161,668 |
| total nodes | 5,778 | 35,454 |
| total leaf nodes | 3,733 | 18,571 |
| avg. depth of hierarchy | 1.31 | 2.04 |
| CDT elements | 41,090 | 41,284 |
| shading calls | 109,885 | 101,208 |
| recomputed form factors | 3,609,941 | 0 |
| recomputed visibility terms | 128,705 | 0 |

Table 3: Statistics for the comparison of hierarchical discontinuity meshing radiosity (HDMR) vs. hierarchical radiosity (HR) shown in Figure 9. All timings are for execution on an HP 9000/720 workstation.

Teller and Hanrahan [26].

We need to be able to compute tight bounds on the visibility between two partially occluded polygons. This would improve the efficiency of the global pass by eliminating unnecessary subdivision in penumbral areas.

**Choice of sources.** Our algorithm is particularly effective for environments with a few primary light sources that are responsible for the most noticeable shadows. In general, however, primary light sources do not dominate the illumination on all the surfaces in an environment. Our algorithm should be extended to compute a set of the most dominant sources, primary or secondary, with respect to each receiving surface. This set should be used both for computing the discontinuities on that surface and for determining when visibility should be recomputed in the local pass.

**Choice of discontinuities.** Not all the discontinuities are equally significant. In the global pass, for example, we should choose discontinuities that would resolve partial visibility most effectively, rather than ones that split the node most evenly. In the local pass we need to identify the discontinuities that are visually significant and insert only these discontinuities into the mesh.

## ACKNOWLEDGEMENTS

We would like to thank Brian Smits, Jim Arvo, and Kevin Novins for helpful discussions and for reviewing the manuscript. Ben Trumbore assembled and submitted the review draft when the authors were away on vacation. Suzanne Smits modeled the sculpture used in the examples in Section 6, and Matt Hyatt modeled the room used in the final comparison. This work was supported by the NSF grant, "Interactive Computer Graphics Input and Display Techniques" (CCR-8617880), by the NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219), and by generous donations of equipment from Hewlett-Packard.

## REFERENCES

[1] Baum, Daniel R., Stephen Mann, Kevin P. Smith, and James M. Winget. "Making Radiosity Usable: Automatic Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity Solutions," *Computer Graphics*, 25(4), July 1991, pp. 51–60.

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

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

[4] Chen, Shenchang Eric, Holly E. Rushmeier, Gavin Miller, and Douglass Turner. "A Progressive Multi-Pass Method for Global Illumination," *Computer Graphics*, 25(4), July 1991, pp. 165–174.

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

[6] Chin, Norman and Steven Feiner. "Fast Object-Precision Shadow Generation for Area Light Sources Using BSP Trees," in Proceedings of 1992 Symposium on Interactive 3D Graphics, March 1992.

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

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

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

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

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

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

[13] Haines, Eric A. and John R. Wallace. "Shaft Culling for Efficient Ray-Traced Radiosity," in Proceedings of the Second Eurographics Workshop on Rendering, May 1991.

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

[15] Heckbert, Paul S. "Discontinuity Meshing for Radiosity," in Proceedings of the Third Eurographics Workshop on Rendering, May 1992, pp. 203–216.

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

[17] Kok, Arjan J. F. and Frederik Jansen. "Source Selection for the Direct Lighting Computation in Global Illumination," in Proceedings of the Second Eurographics Workshop on Rendering, May 1991.

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

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

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

[21] Salesin, David, Dani Lischinski, and Tony DeRose. "Reconstructing Illumination Functions with Selected Discontinuities," in Proceedings of the Third Eurographics Workshop on Rendering, May 1992, pp. 99–112.

[22] Smits, Brian E., James R. Arvo, and David H. Salesin. "An Importance-Driven Radiosity Algorithm," *Computer Graphics*, 26(4), July 1992, pp. 273–282.

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

[24] Tampieri, Filippo. *Discontinuity Meshing for Radiosity Image Synthesis*, PhD dissertation, Cornell University, Ithaca, New York, May 1993.

[25] Teller, Seth J. "Computing the Antipenumbra of an Area Light Source," *Computer Graphics*, 26(4), July 1992, pp. 139–148.

[26] Teller, Seth and Pat Hanrahan. "Global Visibility Algorithms for Illumination Computations," *Computer Graphics*, 27(4), August 1993.

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