Radiosity with Multiresolution Meshes

Thesis Proposal

Andrew J. Willmott
Computer Science Department,
Carnegie Mellon University
ajw+@cs.cmu.edu

April 14, 1998[1]

Thesis Committee:
Paul Heckbert
David O'Hallaron
Francois Sillion
Andy Witkin

The hierarchical radiosity algorithm solves for the global transfer of diffuse illumination in a scene. While its potential algorithmic complexity is superior to both previous radiosity methods and distributed ray tracing, for scenes containing detailed polygonal models, or highly tessellated curved surfaces, its time performance and memory consumption are less than ideal. Also, the density and orientation of the polygons in the input scene unduly affect the output of the method. The aim of this thesis will be to show that by using flexible surface hierarchies similar to those in the surface simplification literature, the use of regular refinement and to a large extent isotropic volume clusters can be avoided, increasing both the speed and the quality of the basic algorithm.

I will develop a radiosity system incorporating these ideas, and show that its performance is superior to existing hierarchical radiosity algorithms, in the domain of scenes containing complex models. The underlying goal of my thesis work is to make high-quality radiosity possible with such scenes.

1 Introduction

One of the greatest challenges of realistic image synthesis is the simulation of global illumination, which requires modelling the transfer of light through the environment being rendered, including the effects of interreflection between objects. Currently there are two classes of methods for calculating global illumination; Monte Carlo-based and Finite Element-based. The former is usually referred to as *ray tracing* and the latter as *radiosity*. Broadly speaking, Monte-Carlo based methods are better suited to simulating the interreflection between highly specular (mirror-like) surfaces, and finite element methods to diffuse (matte-like) surfaces. This proposal is concerned with improvements to the radiosity method.

---

[1] Also available at http://www.cs.cmu.edu/~ajw/proposal.html

Although the basic radiosity algorithm is limited to simulating scenes with diffuse surfaces, it is still worth pursuing for a number of reasons. The interreflection of diffuse light is an important phenomenon in realistic image synthesis, particularly for indoor scenes, where a significant fraction of the light illuminating a surface comes not directly from a light source, but indirectly, by bouncing off one or more other surfaces. Also, the algorithm can be extended to handle specular surfaces, and although some ray-tracing schemes that cache diffuse samples [12] are currently competitive with those radiosity methods used in industry, the hierarchical nature of state-of-the-art radiosity systems promises better performance. Finally, a global solution for the light flow in a scene can be useful, both as a guide to a ray-tracing postprocess, and for reuse in similar scenes, such as successive frames in an animation.

The *hierarchical radiosity* algorithm solves for the global transfer of diffuse illumination in a scene. While its potential algorithmic complexity is superior to both previous radiosity methods and ray tracing, it is more inflexible than pure ray-tracing algorithms, and the density and orientation of the polygons in the input scene unduly affect the output of the method. Figures 2 and 3 illustrate these particular problems. I aim to show that by using flexible surface hierarchies similar to certain model simplification hierarchies [4], the use of regular refinement and to a large extent isotropic volume clusters can be avoided, increasing both the speed and the quality of the basic algorithm.

Because these techniques allow the surface hierarchy to be extended well above those polygons in the original scene (the *input polygons*), we can eliminate the dependence of the algorithm's complexity on the number of these input polygons. The ability to refine in arbitrary orientations below the input polygons gives greater flexibility to adapt the mesh to the current radiosity solution.

I propose to base my thesis on the following modifications to the hierarchical radiosity algorithm:

1. The use of binary tree-based simplification operations, both above and below the input polygons.
2. The recasting of the radiosity equation in terms of vector irradiance [1] rather than scalar irradiance. In effect, this decouples the transfer of radiosity in the algorithm from the local surface normals of clusters of input faces. This is necessary to gain acceptable quality results when using the hierarchies generated by (1).
3. The use of directional information from the vector-based transfer links predicated by (2) to guide the orientation and position of refinement past the level of input polygons. This will allow edges in the refinement mesh to follow contours in the radiosity function as well as shadow boundaries, leading to higher quality results. (Traditional regular refinement does not adapt the shape of subfaces to fit the current solution, only their resolution.)

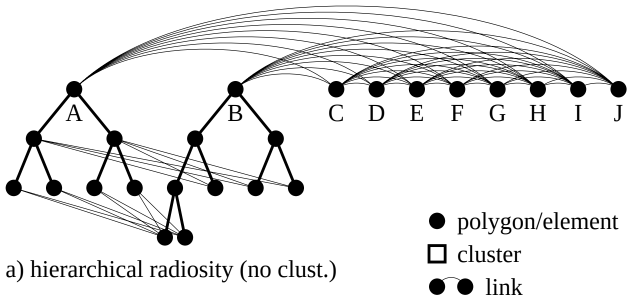
The underlying goal of my dissertation work is to help make high-quality radiosity usable for scenes containing complex models.

2 Radiosity Methods

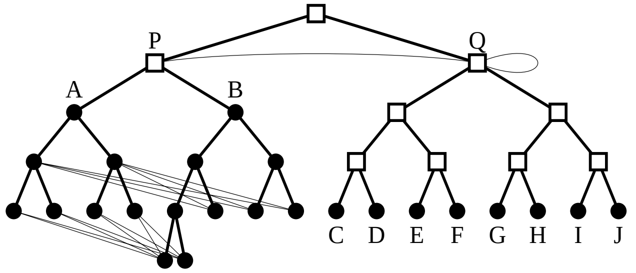
The radiosity algorithm is the most commonly used algorithm for simulating diffuse interreflection [3]. It subdivides surfaces in the environment into a mesh of *elements*, and then sets up and solves a large system of linear equations in order to compute the *radiosity* (the amount of emitted or reflected light) at each surface point. Since each element can potentially illuminate every other element, the number of element-to-element interactions is quadratic. Early radiosity algorithms used a fixed mesh of elements, and had costs that were quadratic in the number of elements in the scene. These algorithms did a poor job of exploiting the sparseness of the element-to-element interaction kernel; most objects are visible to only a small subset of the other objects in the scene, and the transfer of radiosity also obeys an inverse square law, so much of this kernel is either zero, or of small magnitude.

The most significant subsequent modification of the radiosity algorithm was the hierarchical algorithm, which takes advantage of this sparseness by employing techniques similar to the *n-body algorithm* used in gravitational simulations. By treating interactions between distant objects at a coarser level than those between nearby objects, the hierarchical radiosity algorithm reduces the cost from quadratic to linear in the number of elements used.

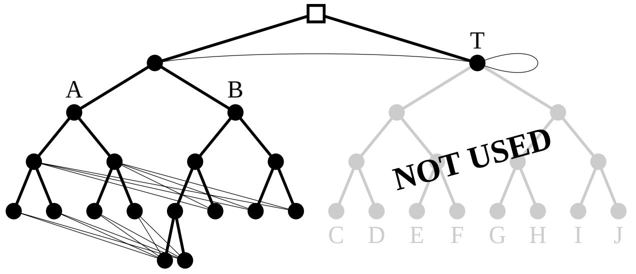
2.1 Existing Hierarchical Radiosity Methods



a) hierarchical radiosity (no clust.)



b) hierarchical radiosity with clustering



c) hierarchical radiosity with multiresolution models

Figure 1: Three approaches to hierarchical radiosity.

Classic Hierarchical Radiosity

The *hierarchical radiosity* algorithm [7] and its generalization, the *wavelet radiosity* algorithm [6], represent the radiosity on each surface with an adaptive, hierarchical mesh. The meshes consist of a quadtree or similar data structure rooted at each of the input polygons.

Illumination from geometrically distant surfaces employs a coarse level in the hierarchy, while illumination from nearby surfaces employs a finer level of subdivision. Illumination is represented by links between elements. This adaptivity makes the cost of the algorithm linear in the number of elements, n. Unfortunately, because the light transport from each polygon to every other polygon must be computed, the cost is quadratic in the number of input polygons, k. The cost is thus $O(k^2 + n)$.

Figure 1 is a schematic comparison of variants of hierarchical radiosity on a scene with two large polygons A & B in close proximity, and eight small polygons C-J more distant. Simple hierarchical radiosity (a) yields a forest of quadtrees. Polygons A & B are subdivided and some of their children are linked. The large number of links between the small input polygons C-J makes the algorithm inefficient.

Hierarchical radiosity algorithms without clustering work well on scenes with a small number of large polygons, but become impractical in time and memory consumption for scenes of several hundred polygons.

Hierarchical Radiosity with Clustering

To combat this problem, clustering methods for hierarchical radiosity were developed [9, 10, 5]. These methods group the input polygons into *volume clusters*, building a hierarchy above the input polygons that culminates in a root cluster for the entire scene. The lower nodes in this hierarchy are elements in quadtrees (small surface patches, with normal and reflectance), as before, but the upper nodes are different. They are boxes containing a set of possibly disconnected polygons with potentially varying normal vectors and reflectances. Clusters are assumed to radiate isotropically. The use of clusters reduces the number of links needed from quadratic to linear.

Unfortunately, the isotropic assumption is a poor one. Consider a large wall or a panel-type light source set in the ceiling; both radiate in one direction only. To get acceptable results it is necessary to correct for this assumption on each transfer of radiosity, both by incorporating the projected area of the child polygons of a source volume cluster in the direction of transfer, and similarly finding the projected area of the polygons in the receiving cluster [10, 9, 11]. This modification raises the cost of the algorithm from $O(k+n)$ to $O(k \log k + n)$. It also requires accessing each child polygon in the scene at least once while calculating the radiosity transfer. Although clustering is significantly faster than standard hierarchical radiosity, the need to touch all of the input polygons on each solver iteration causes its working set to be excessively large for complex scenes.

Figure 1b shows hierarchical radiosity with clustering. If cluster Q is sufficiently small and distant from cluster P then a single link between them suffices. Since a cluster can illuminate itself, a self link on Q is necessary as well. The algorithm is still slow because it is necessary to find the projected area of polygons C-J.

2.2 Meshing for Radiosity

Generally the input scene to a radiosity method is polygonized, and the polygons are then split into a mesh of triangular or quadrilateral elements. There are two methods commonly used to adapt this mesh to a radiosity solution as it progresses.

Regular Refinement

The standard mesh-refinement technique employed in hierarchical radiosity algorithms is the regular refinement of elements. For instance, a triangle can be refined into four elements by splitting each edge at its midpoint, and retesselating these points into four subelements. Such refinement operations on an original mesh element form a quadtree. Regular refinement has the advantage of simplicity, and independence—any particular element can be refined independently of another. It also makes refinement decisions straightforward; if the total estimated error across the element is above a certain threshold, we refine it. It has the disadvantage that, if an element is poorly shaped, or not well aligned to the current radiosity solution, the same will hold for its subelements. Also, quadtree meshes can feature T-vertices, which lead to interpolation problems when rendering. This can be corrected by balancing and anchoring the mesh [2].

Discontinuity Meshing

Discontinuity meshing takes adaptive methods using regular refinement a step further by computing where shadow edges or other visibility discontinuities will occur, and pre-splitting the mesh along such features. This can produce impressive results, especially when the shadows in question are relatively sharp. Its primary drawbacks are that it can be difficult to implement robustly, and it is an object space method, dependent on the edges in the scene. Most previous implementations of discontinuity meshing modify the mesh as a preprocess, and thus attempt to calculate all possible discontinuities. This makes them poorly suited to curved objects and objects with highly detailed polygonal silhouettes.

2.3 Problems with Existing Methods

The hierarchical radiosity with clustering algorithm described above, while capable of impressive performances with scenes of medium complexity, has three flaws which I wish to address with my thesis work.

- **Speed.** Large, highly-tessellated objects can cause problems when using hierarchical radiosity with volume clustering because the algorithm is least $O(k \log k)$, where k is the number of input polygons. An example of such an object can be seen in Figure 2. This is even though the general n-body finite element method has the potential to be $O(n)$, where n is the number of elements used in the simulation, and $n \ll k$ in many situations.
- **Memory.** There are similar problems with memory behavior. Because the isotropic-assumption correction forces all input polygons to be touched at least once on each solver iteration, the algorithm exhibits poor memory locality when $n \ll k$. Moreover, a radiosity sample must be stored for all polygons, rather than only for the n elements being used in the simulation, as we might wish.
- **Quality.** Shadows look poor with most existing methods. The regular refinement employed in refining input polygons means that the orientation and shape of these polygons can have a detrimental effect on interpolation results, both in areas of high radiosity gradient, and in the presence of shadows. Figure 3 demonstrates these problems.

In conjunction, these problems have largely limited hierarchical radiosity methods to the research community. There are a number of commercial rendering systems employing radiosity, including those produced by Lightscape and Lightworks, and recent 3D games have started to incorporate the algorithm for precalculation of shadows and lighting. However, these systems all use the progressive radiosity algorithm, which has the advantage that it is simple to implement, and for scenes on the order of ten thousand polygons can produce a good quality result in an acceptable time.

In this thesis I will be addressing these problems by the replacement of the standard isotropic volume cluster/regular surface hierarchy with multiresolution meshes, as described in the next section.

3 Radiosity with Multiresolution Meshes

In this section we present our new hierarchical radiosity algorithm which uses multiresolution meshes. Figure 4 shows the fundamental premise behind the algorithm; the replacement of most isotropic volume clusters with simplified models of the input polygons. With this scheme, the data structures for elements coarser than and finer than the input polygons are more similar. Hierarchical radiosity is now free to subdivide below the level of the input polygons and to cluster surfaces together above this level. This largely removes the dependence of the hierarchical radiosity algorithm's time and space complexity on the input polygons, permitting it to represent light transport at more natural levels of detail, and to operate more efficiently in complex scenes.

Specifically, by incorporating such multiresolution meshes into our radiosity simulation, we can reduce the

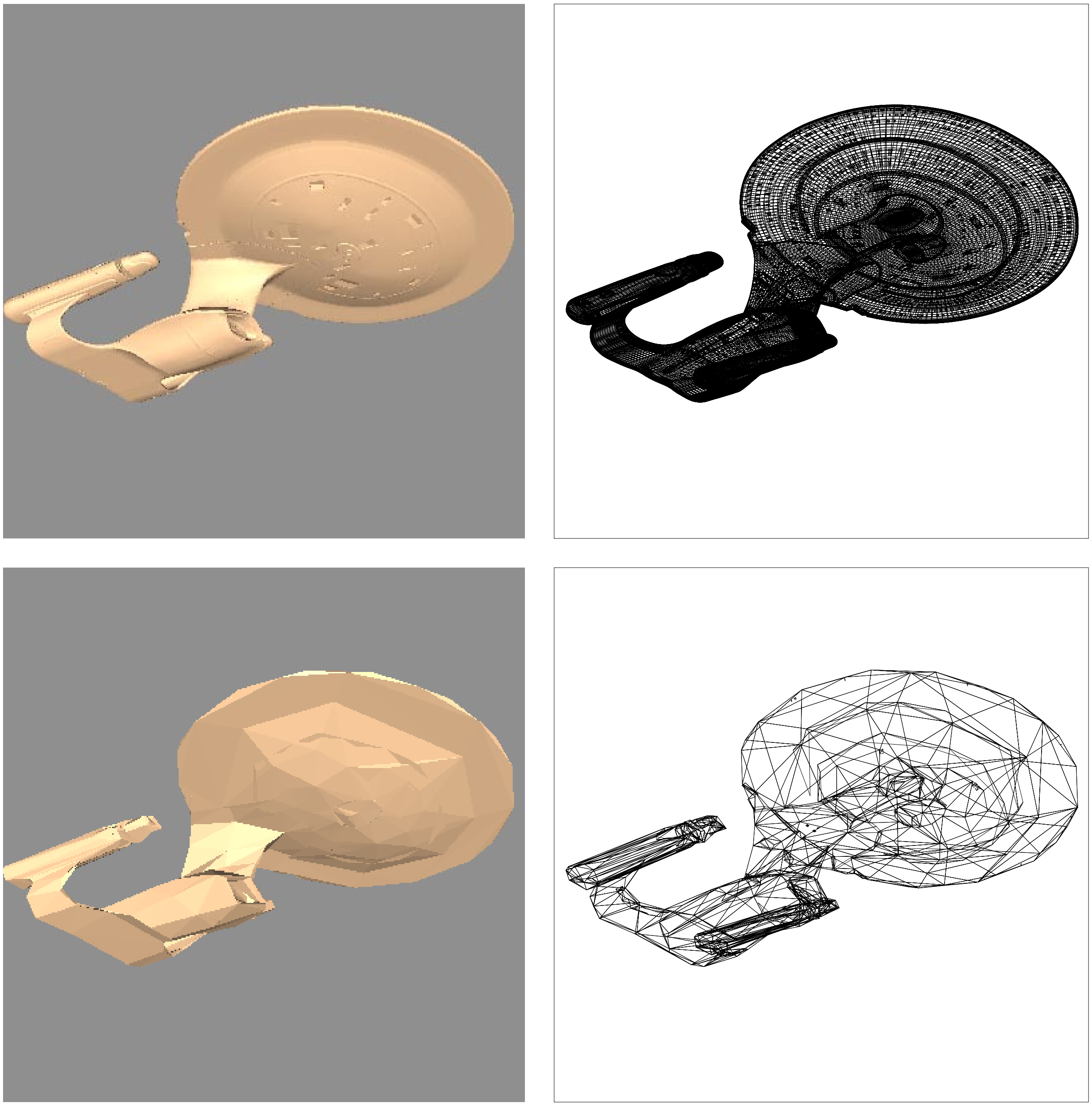


Figure 2: Large, highly-tesselated objects cause problems for existing hierarchical radiosity algorithms. **Top:** The model of the ‘starship enterprise’ pictured is composed of 203878 triangles. The model shown is a medium resolution version of the original model. (Models captured by cyberware scanner or equivalent device can sometimes reach complexity levels of the order of a million triangles.) **Bottom:** Such models can be simplified by a technique that creates a multiresolution hierarchy of the model. The simplified model shown here has 1000 triangles.

complexity of the algorithm to $O(s \log s + n)$, where s is the number of faces in the most simplified version of our scene. In complex scenes, $s < n$.

The use of simplified models also improves the accuracy of our representations and permits our algorithm to avoid touching the lowest portions of the hierarchy during iterations. In Figure 1c we see how a tree of simplified models is built above the input polygons. The subtree below T can be paged out, saving time and

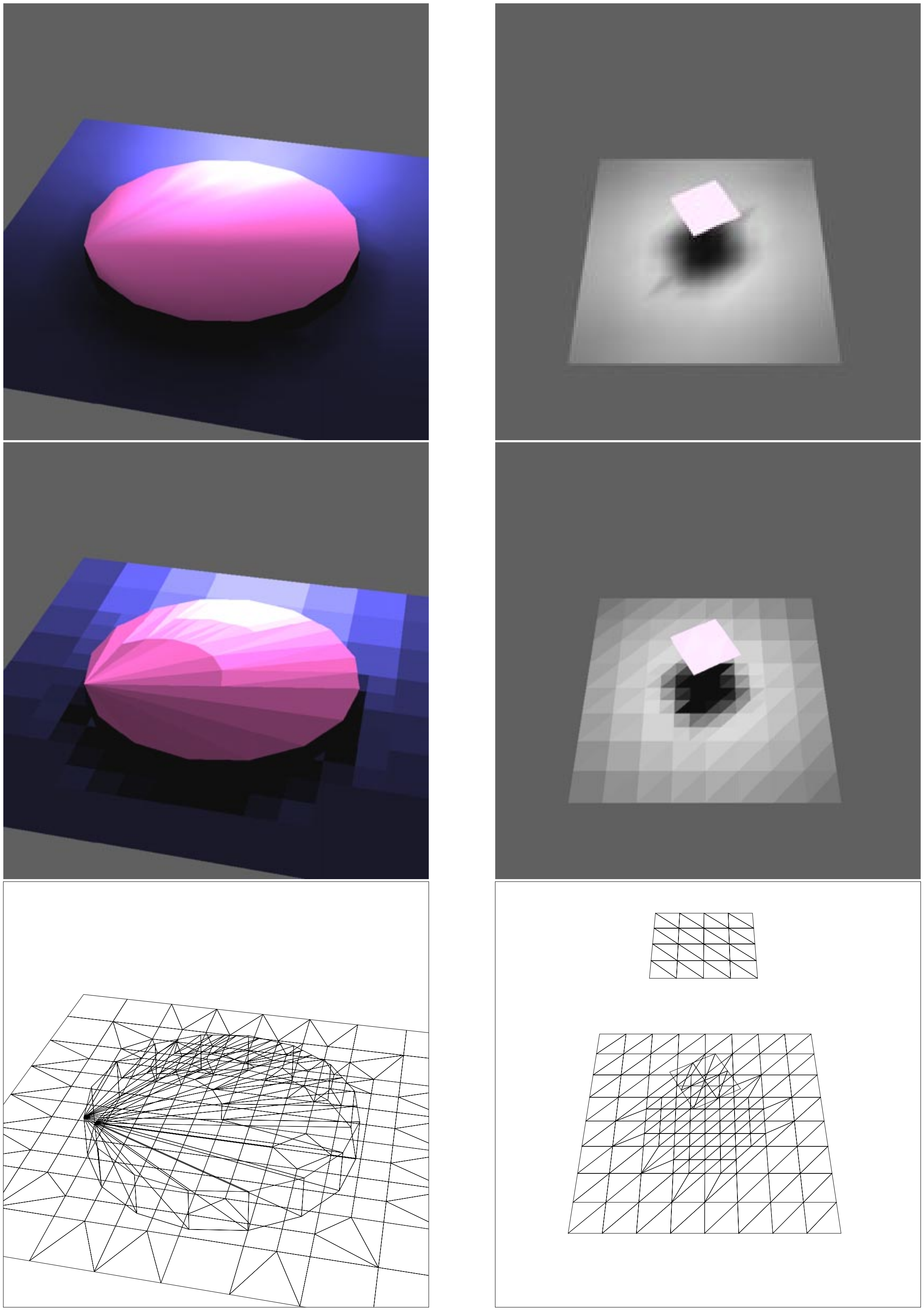


Figure 3: The underlying base mesh can have a large effect on the result of the radiosity algorithm. **Left:** a poorly-tesselated disc leads to streaking in the well-lit area. **Right:** the orientation of the initial triangulation of the ground square affects the shadow cast on it. Shown from top to bottom are the post-processed solutions, the original solutions, and the solution meshes. In post-processing, these meshes have been balanced and anchored before shading interpolation, but this still leaves us with noticeable shading artifacts.

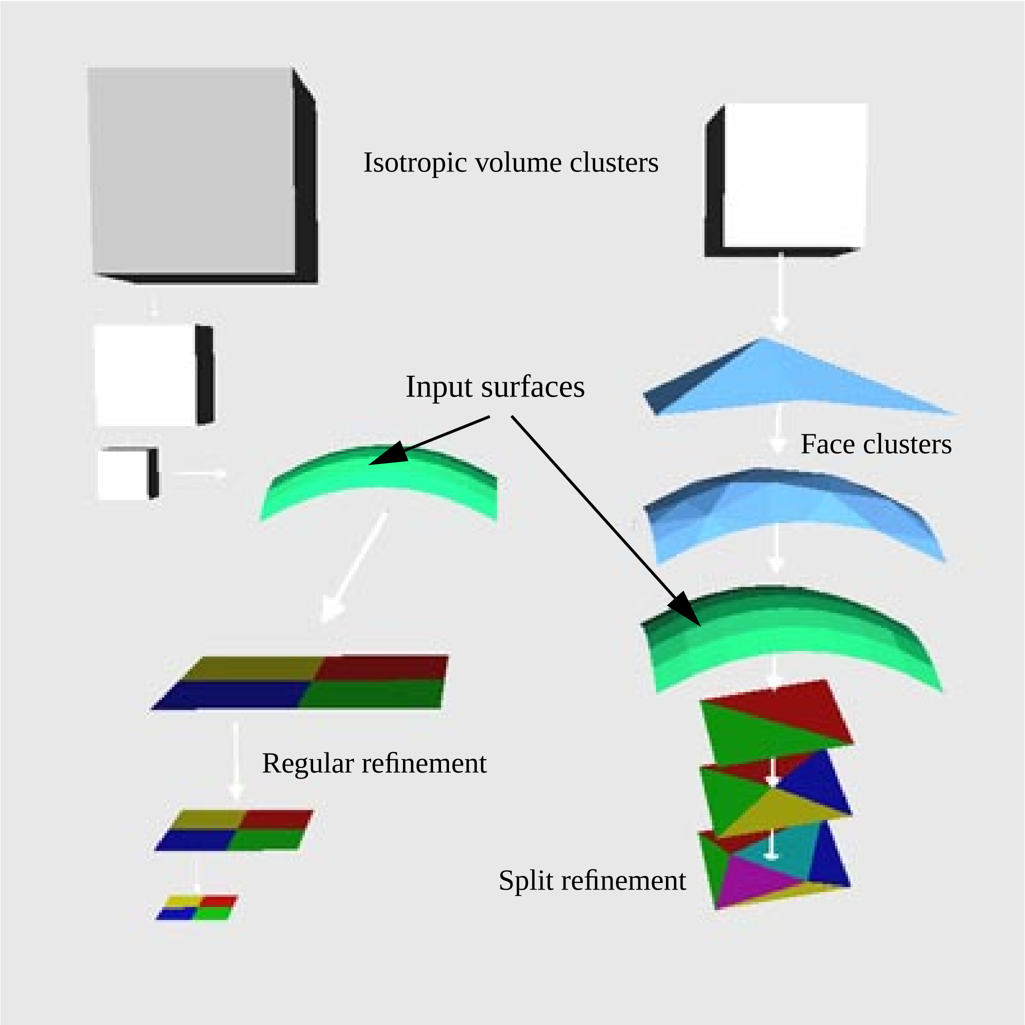


Figure 4: **Left:** traditional hierarchical radiosity with clustering builds a volume hierarchy above the input surface (light green), and then employs regular refinement below that surface to form the complete hierarchy.

Right: I propose to replace this with a multiresolution hierarchy, in which a much shallower volume hierarchy encloses a simplified form of the input surface, which is then refined into the original input surface. This gives us a better directional approximation to the surface than an isotropic volume hierarchy, and the complexity of the radiosity simulation depends on the number of simplified faces in the scene, rather than the number of faces in the original input models., Below the input surface, refinement based on the same edge-contraction operations used in the multiresolution model gives greater flexibility in adapting to local intensity gradients.

memory.

Using a multiresolution mesh in a radiosity algorithm requires two things; the construction of a suitable simplification hierarchy of *face clusters* from models in the original scene, and the adaption of the traditional radiosity equations to allow the calculation of light transfer between such face clusters. The next two sections look at how these are achieved.

3.1 Multiresolution Meshes

To approximate the light transport from a complex object, we will need shape approximations that are both geometrically accurate and compact (few polygons). There is a variety of simplification algorithms to choose from. The most promising for our purposes seems to be Garland's surface simplification technique [4], which is fast and produces reasonable quality results. His edge collapse algorithm is capable of simplifying 3-D triangulated manifolds with boundary (models each of whose edges have two or fewer adjacent triangles). The algorithm iteratively simplifies the model. At each iteration it collapses the edge (Figure 5) whose removal introduces the least error. This tends to eliminate small polygons in nearly planar regions of a surface.

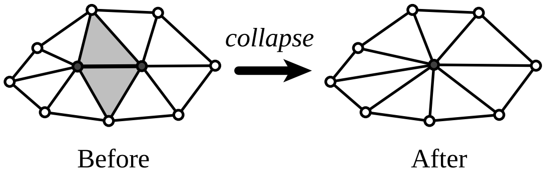
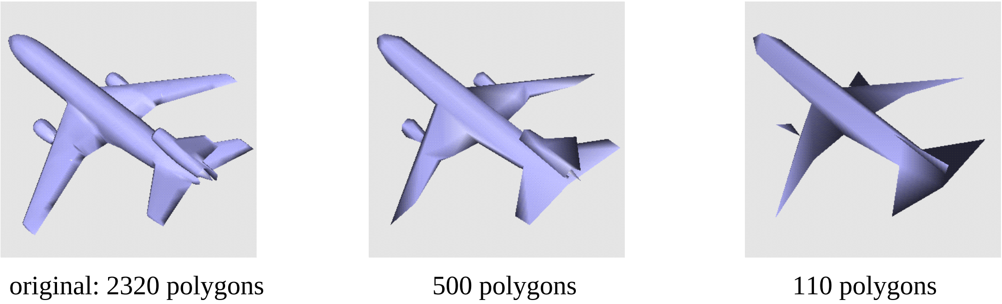


Figure 5: An edge-collapse operation.

The sequence of edge collapses builds a tree of vertices from the bottom up [8]. Cuts across this tree constitute approximations of the geometry at various levels of detail (Figure 6).


original: 2320 polygons

500 polygons

110 polygons

Figure 6: Progressively simplified models of a DC10 aircraft.

A detail is that we need a tree of face clusters, rather than the *vertex hierarchy* the above method produces. This can be rectified by treating the nodes in the vertex hierarchy as representing the equivalent faces in the dual of the polyhedral model. A promising alternative is to use similar quadric simplification methods to directly build a *face hierarchy*, consisting of a tree of faces, by iteratively clustering pairs of faces.

3.2 Vector-based Radiosity

To properly account for the transfer of light between face clusters, which may contain a number of surface faces of different orientation, it is necessary to adjust the usual formulation of the radiosity equation. Using the standard scalar radiosity formulation leads to the faceting effect shown in Figure 10c, which is obviously unacceptable.

Consider the light transfer from one face cluster to another in Figure 7 below.

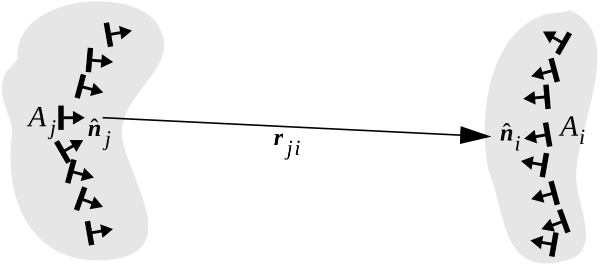


Figure 7: Radiosity transfer between two clusters of faces.

We let *j* denote an element in the source cluster on the left, and *i* an element in the receiver cluster on the right. Elements on the left have radiosities B_i and elements on the right have resulting irradiances, E_i. If we assume that all (*i*,*j*) pairs are inter-visible and that the sources are close together and far from the receiver, then we can approximate the irradiance from a single cluster as:

$$\mathbf{E} = -\mathbf{m}\mathbf{m}^T\mathbf{P}, \tag{1}$$

We refer to **P** as the power vector, **E** as the irradiance vector, and **m** as the transport vector; their relationship is shown in Figure 8.

The irradiance vector **E** is a 3-vector whose components are the irradiances on planes normal to the x, y, and z axes, respectively, positioned at the receiver. Recording this information, rather than a scalar irradiance to the average plane of the receiver, allows coarse variations in the irradiance as a function of orientation to be modeled. The irradiance of individual faces can then be recovered as:

$$E_i = \hat{\mathbf{n}}_i^T\mathbf{E}. \tag{2}$$

The power vector similarly measures the direction radiosity of the source cluster, and can be calculated as:

$$\mathbf{P} = \sum_j \hat{\mathbf{n}}_j A_j B_j, \tag{4}$$

The transfer vector **m** is simply

$$\mathbf{m} = \frac{\hat{\mathbf{r}}}{\sqrt{\pi}r} \tag{3}$$

The use of this technique eliminates most of the faceting effects of Figure 10c, as seen in Figure 10d.

We can substitute these vector quantities directly for the irradiance and radiosity in a standard hierarchical radiosity algorithm, although as **P** is area based, when pulling radiosity up the hierarchy we sum the power vectors of a node’s children, instead of averaging them.

3.3 Refinement of Input Polygons

The construction of a multiresolution mesh for a particular model leaves us with a face cluster hierarchy

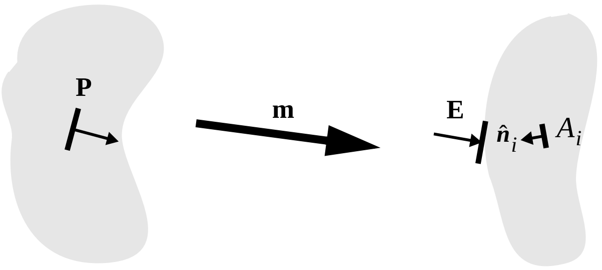
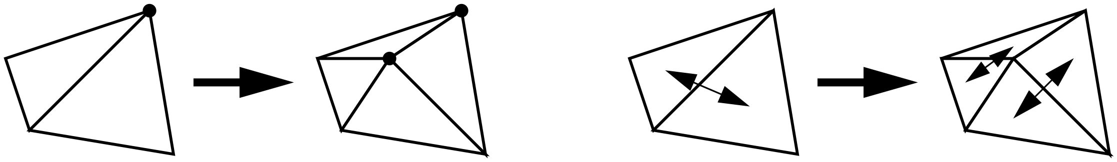


Figure 8: Approximating the radiosity transfer between clusters of faces with vector-based quantities.

above the input polygons. However, the radiosity algorithm may need to access the hierarchy at a finer resolution than this, which will require refining the input polygons to finer levels of detail. One possibility is to switch over to regular, quad-tree subdivision at this point, but a more natural alternative is to employ the same edge-split or face-split operations that make up the multiresolution mesh below the input polygons as well as above. In the case of a vertex-cluster hierarchy, the standard edge-split operation can be used, as in Figure 9a. For a face-cluster hierarchy, a cluster of two faces can be split into two such clusters, as in Figure 9b.

There are three advantages to this approach. First, there is no switch between representations, and the degree of the refinement remains the same; a binary tree, as opposed to a quadtree. Secondly, we have more freedom to adapt the refinement according to local variations in irradiance. In particular, with the directional irradiance available to us, possibly augmented with other stored information, it seems likely that we can use this information to pick an optimal direction in which to split a face. Finally, we can use information about the topology of the mesh to refine the surface in a smoother manner. If two abutting faces are tessellations of a smooth surface, the refined faces can follow that surface, rather than lying exactly in the plane of their parents.

At best, this approach may allows us to approximate the effects of a priori discontinuity meshing, without the complexity of its implementation or running time. At worst, the ability to refine the *shape* and *orientation* of the mesh in addition to its *resolution* should allow fewer elements (and thus links) to be used for a given solution accuracy. The hope is that such an approach will provide a suitable compromise between the simplicity but inflexibility of regular refinement, and the sometimes excessive refinement introduced by discontinuity meshing.



(a)

(b)

Figure 9: Multiresolution refinement operations. **(a)**, an edge split. **(b),** a face-cluster split.

3.4 Initial Results

I currently have a working prototype of the multiresolution radiosity algorithm, and initial results look very promising. Example results can be seen in Figure 10, which shows a simple scene containing a detailed dragon model. As can be seen, the use of the multiresolution algorithm reduces the time taken to that traditional hierarchical radiosity would take on a simplified model, with comparable quality. Also, Figure 10c shows why the vector transfer form of the radiosity equation is needed to model transfer of radiosity between face clusters well.

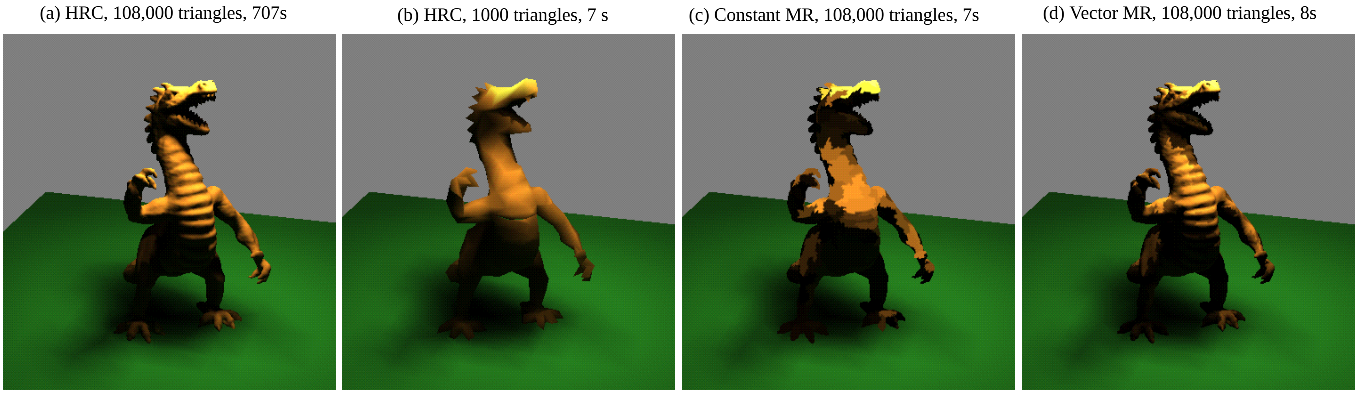


Figure 10: Hierarchical radiosity with clustering (HRC) and multiresolution radiosity (MR) applied to a detailed dragon model.

A more complex test scene, shown in Figure 11, contains a number of highly detailed models, with a total count of around 200,000 triangles. The table to the right of the scene shows the results for hierarchical radiosity with clustering and multiresolution radiosity when run on the same scene with the same refinement parameters. As can be seen from this table, the execution time speedup was significant, approaching a factor of 9. The memory efficiency was also greater than that of volume clustering, by a factor of 3.

3.5 Developing a Robust Multiresolution Radiosity System

To achieve the goal of a robust radiosity system using the methods I have outlined, and to demonstrate that it works as expected, the following tasks must be accomplished:

1. **Establish the best kind of face-cluster hierarchy to use**
   Is a vertex hierarchy or face hierarchy better? (See Section 3.1.) Michael Garland has developed new methods for creating clusters of original faces of the model, rather than vertices. This scheme has the advantage that it can produce clusters with smooth borders, and give certain guarantees about the variation in surface normal over the patch. There is a trade-off here between (assumed) better quality from direct face-clustering, and the advantage of having direct vertex-to-vertex face-cluster compatibility with simplified models used for final-stage rendering. The suitability of either method to polygon refinement will also help determine which is the better technique to use.

2. **Eliminate inter-element shading discontinuities**
   The current scheme does not attempt to fix possible discontinuities in irradiance between adjacent leaf elements that belong to different face clusters. There are no longer value discontinuities in the scalar irradiance between the borders of adjacent face clusters, but there are discontinuities in the vector irra-

Parameter	HRC	MR
Simulation Memory	31Mb	10Mb
Rays cast	2.1 million	1.7 million
Execution time	80 minutes	9 minutes
Links used	134860	117491
Volume clusters	22774	331
Face clusters/faces	199124	8800

Figure 11: A medium-complexity scene, with results for the standard hierarchical radiosity with clustering algorithm (HRC), and our new multiresolution radiosity algorithm (MR).

diance. There are two approaches to this problem. One is to use information about neighboring face clusters to interpolate the directional irradiance. The other is to use a higher-order representation of directional irradiance, analogous to the use of higher-order basis functions in hierarchical radiosity.

3. **Find a suitable representation for link visibility**
   In the initial implementation of radiosity using multiresolution meshes, the fractional visibility between two face clusters is used to scale the entire transfer. This leads to poor shadow resolution, as can be seen by comparing the shadowed area under the dragon's neck in Figure 10a to Figure 10d. One possibility is to incorporate visibility sampling into the calculation of the transfer vector **m**. Another is to store higher-order visibility information about visibility between the two clusters; linear, or even a small shadow map.

4. **Extend use of the multiresolution mesh past the input surfaces**
   The prototype implementation only uses the multiresolution mesh above the input surfaces. The mesh hierarchy needs to be extended beyond those surfaces with as outlined above.

5. **Establish the system's performance empirically**
   We expect sublinear performance from the algorithm in the number of input polygons, in the domain where the number of faces in the simplest models is smaller than the number of elements required for the radiosity solution. To establish whether this occurs, and to gain some insight into the interaction of memory, computation time, and solution accuracy, we need to run a number of experiments on the system's performance. Similarly, we would like to establish how the system performs relative to hierarchical radiosity with volume clustering, for scenes of varying complexity. I also hope to show that using radiosity for scenes larger than one million polygons is possible on a mid-sized workstation.

   To carry out these empirical studies, I will use the same systematic approach as in my earlier comparative study [13].

4 Contributions and Timetable

In this thesis I propose to deal with some of the problems with current hierarchical radiosity techniques by using a surface hierarchy based on multiresolution meshes. The expected contributions of this work would be:

- The extension of the radiosity algorithm to work on detailed, industrial-strength scenes, using the resources of a standard workstation.
- Artifact-free resolution of shadows.
- The production of a reasonably robust radiosity system, to be released to other researchers and interested parties over the internet.

Table 1 contains an approximate timetable for the tasks necessary to complete this work, as outlined in Section 3.5.

Activity	**Months**	**Start Date**
Find Best Clustering Method	1	April ‘98
Seam Elimination	2	
Visibility Scheme	2	August ‘98
Input-polygon Refinement	3	
Slack time/addition of features to system	2	January ‘99
Experiments and testing	3	
Writing Dissertation	4	June ‘99
Total/Finish	18	October ‘99

Table 1. Rough Timeline for Thesis

References

[1] James Arvo. The irradiance Jacobian for partially occluded polyhedral sources. In *Proceedings of SIGGRAPH ’94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 343–350. ACM SIGGRAPH, ACM Press, July 1994.

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

[3] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, Boston, 1993.

[4] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*

97 Conference Proceedings, Annual Conference Series, pages 209–216. ACM SIGGRAPH, Addison Wesley, August 1997.

[5] S. Gibson and R. J. Hubbold. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5):297–310, 1996.

[6] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. *Computer Graphics (SIGGRAPH '93 Proceedings)*, August 1993.

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

[8] Hugues Hoppe. View-dependent refinement of progressive meshes. In *SIGGRAPH 97 Proc.*, pages 189–198, August 1997.

[9] F. X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, September 1995.

[10] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 435–442. ACM SIGGRAPH, ACM Press, July 1994.

[11] M. Stamminger, W. Nitsch, and Ph. Slusallek. Isotropic clustering for hierarchical radiosity – implementation and experiences. In *Proc. Fifth International Conference in Central Europe on Computer Graphics and Visualization '97*, 1997.

[12] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22(4):85–92, Aug. 1988.

[13] Andrew Willmott and Paul Heckbert. An empirical comparison of progressive and wavelet radiosity. In *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 175–186, New York, NY, 1997. Springer Wien.