

Face Cluster Radiosity

Andrew J. Willmott, Paul S. Heckbert and Michael Garland¹

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Abstract. An algorithm for simulating diffuse interreflection in complex three dimensional scenes is described. It combines techniques from hierarchical radiosity and multiresolution modelling. A new face clustering technique for automatically partitioning polygonal models is used. The face clusters produced group adjacent triangles with similar normal vectors. They are used during radiosity solution to represent the light reflected by a complex object at multiple levels of detail. Also, the radiosity method is reformulated in terms of vector irradiance and power. Together, face clustering and the vector formulation of radiosity permit large savings. Excessively fine levels of detail are not accessed by the algorithm during the bulk of the solution phase, greatly reducing its memory requirements relative to previous methods. Consequently, the costliest steps in the simulation can be made sub-linear in scene complexity. Using this algorithm, radiosity simulations on scenes of one million input polygons can be computed on a standard workstation.

1 Introduction

The hierarchical radiosity algorithm in its various forms is probably the most promising radiosity method in existence. The best hierarchical radiosity methods, using clustering, permit scenes of moderate complexity (several hundred thousand input polygons) to be simulated in a few hours. Unfortunately, current radiosity techniques, even with clustering, use excessive memory and their speeds are not competitive with other, less realistic rendering methods. We would like to be able to apply radiosity methods to the complex scenes common in special effects. Such scenes routinely use objects each employing 100,000 polygons or more. We therefore seek an enhancement to the hierarchical radiosity algorithm that will permit very complex scenes — scenes with millions of input polygons — to be economically simulated on a standard computer.

One of the greatest difficulties with existing radiosity methods is that their memory use is at least linear in the number of input polygons. This is not a problem if the scene is small, but if the input polygons cannot fit in physical memory, the algorithm will thrash and performance will degrade dramatically. To deal with very complex scenes, we need methods which in practice have memory and time cost that is sub-linear in the number of input polygons.

In this paper we describe the face cluster radiosity algorithm, a technique that achieves this goal. Its three main phases are preprocessing, solution, and postprocessing. Preprocessing converts the scene description into a multiresolution, hierarchical model. The time cost of this is super-linear in the number of input polygons, but preprocessing can be done on an off-line, object-by-object basis, so its memory costs are

1. {ajw|ph|garland}@cs.cmu.edu

modest and its time costs can be amortized over multiple solutions. Next, one or more radiosity solutions are found. This is the costliest step, in practice. The solution phase is sub-linear in cost because it accesses only the coarsest levels of detail from the hierarchy that are necessary. Consequently, often large portions of the hierarchy need never be paged in during this phase, with huge physical memory savings. After solution, post-processing evaluates the radiosity of the finest details of the scene. This requires linear time. The overall cost, being dominated by solution, is thus sub-linear in practice.

A preview of the technique is shown visually in Figure 7. If one of the best existing radiosity algorithms (hierarchical radiosity with volume clustering) is used on a detailed model, a solution takes over ten minutes (Figure 7a). If, on the other hand, the input geometry is simplified by cutting the number of triangles by a factor of 100, and the same algorithm is applied to the simplified model, a solution can be calculated much more quickly (Figure 7b, 7 seconds). This is fast, but the accuracy and visual quality are poor. Our face cluster radiosity technique allows a solution not much more expensive than this to be calculated and propagated to the fully detailed model, yielding Figure 7d. This is much faster than the full solution and almost as accurate.

2 Hierarchical Radiosity

We review previous work, and describe at a high level how our new method differs. Hierarchical radiosity [3, 9] has a cost linear in the number of elements, n . Unfortunately, because the initial light transport ‘link’ from each polygon to every other polygon must be computed, the cost is also quadratic in the number of input polygons, k . The cost is thus $O(k^2 + n)$. Classical hierarchical radiosity algorithms work well on scenes with a small number of large polygons, but they become impractical in time and memory consumption for scenes of several hundred polygons.

Hierarchical Radiosity with Volume Clustering. To combat this problem, clustering methods for hierarchical radiosity were developed [2, 7, 14, 15, 16]. 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 octree or k-d tree boxes containing a set of disconnected polygons with potentially varying normal vectors and reflectances. The use of clusters reduces the number of links needed from quadratic to linear.

Several methods for handling the light incident on a cluster have been explored. The simplest is to sum the incoming light. This approach, called beta links by Smits, turns out to be fast, $O(k + n)$, but inaccurate. A more successful alternative is to push the light down to the leaves of the tree (Smits’ alpha links) [15, 16, 17]. This raises the cost of the algorithm to $O(k \log k + n)$. A third alternative, proposed by Sillion and Christensen [2, 14], is to represent a cluster as a point that emits and reflects light according to a directional distribution. Both latter methods require light to be pushed down the tree, as with alpha links. Christensen’s algorithm appears to be asymptotically the fastest, achieving good quality results in $O(k + n)$ time. Although any of these clustering methods is significantly faster than classical hierarchical radiosity, the need to touch all of

the input polygons on each solver iteration causes their working set to be excessively large for complex scenes.

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 and 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. Figure 1b shows hierarchical radiosity with volume 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 push light down to polygons C-J.

Face Cluster Radiosity. We propose that volume clusters be replaced by multiresolution models for all groups of input polygons that represent a surface. The use of such models allows pushing of light to the leaves to be avoided, and they often provide a better fit to the original surfaces than volume clusters. The particular multiresolution representation that we use, face clustering, groups adjacent faces that have similar normals, and thus can approximate largely planar surfaces well.

With this scheme, the data structures for elements coarser than and finer than the input polygons are more similar, since both face clustering and quadtree refinement yield a contiguous piece of surface with a normal. Hierarchical radiosity is now free to subdivide below the level of the input polygons and to “unsubdivide” above this level. This reduces the hitherto inordinate role of the input polygons, permitting hierarchical radiosity to represent light transport at more natural levels of detail, and to operate more efficiently in complex scenes.

The use of multiresolution models 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 tree nodes below T and above C-J are now face clusters, not volume clusters, while the highest levels of the tree, above connected objects, are volume clusters. Note that the subtree below T can be paged out during simulation, saving time and memory. The hierarchy is seen schematically in Figure 2.

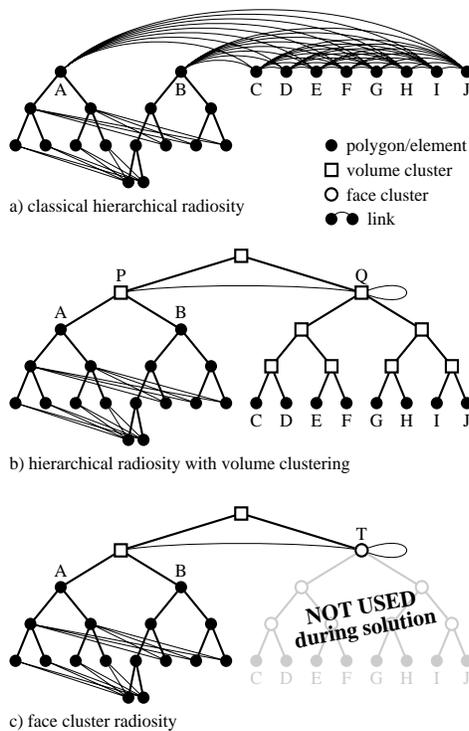


Fig. 1. Three approaches to hierarchical radiosity.

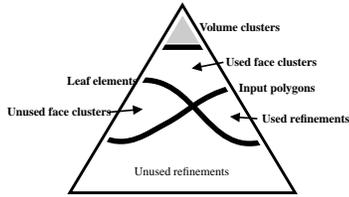


Fig. 2. Schematic of a hierarchy using face clusters.

The use of simplified models in radiosity has previously been proposed. Rushmeier et al. demonstrated the feasibility of the concept, but their method employed manually-constructed simplified models, making it impractical for complex scenes [13]. Greger et al. showed how a radiosity simulation of a simple scene could be applied to a more detailed version of that same scene by using an *irradiance volume* [8]. Both these methods re-

quire the user to judge the level of pre-simulation simplification; simplification and simulation are two separate steps, whereas in our algorithm, the level of simplification is driven by the radiosity simulation.

3 The Face Cluster Radiosity Algorithm

In this section we describe our algorithm for simulating radiosity on multiresolution models that use face clustering. We discuss our approach to building the necessary hierarchies, and then show how the standard radiosity equations can be modified to better suit the use of these hierarchies, starting with the standard formulas governing diffuse interreflection [3].

3.1 Face Cluster Hierarchies

Recent work from the area of surface simplification provided a starting point for our research into radiosity using multiresolution models. Iterative edge contraction, one of the currently popular simplification techniques, can be used to construct a hierarchy of progressively larger vertex neighbourhoods on the surface. Each edge contraction collapses the two vertices at either end of the edge to a single, new vertex. The hierarchy is created by treating the endpoints as the children of this new vertex. These vertex hierarchies have been used primarily for view-dependent refinement of models for real-time rendering [10].

In our original algorithm we used vertex hierarchies directly, but they sometimes proved problematic, because vertex nodes don't have a well defined area and normal. These properties are easier to establish for face hierarchies than for vertex hierarchies.

To address this problem, we developed an algorithm for generating *face cluster hierarchies* [4] that is a dual form of the quadric-based simplification algorithm of Garland and Heckbert [5]. Rather than iteratively merging pairs of vertices to directly simplify the mesh, we iteratively merge groups of topologically connected faces, which we refer to as face clusters, thereby partitioning the original mesh (see Figure 8). This process does not change the original geometry of the surface in any way; it merely groups surface polygons into progressively larger clusters. These merge operations can be used to create a simplification hierarchy in the same manner as edge collapses; an example is shown in Figure 3.

In [5] quadric functions of position, $\mathbf{v}^T \mathbf{A} \mathbf{v} + 2\mathbf{b}^T \mathbf{v} + c$, are used to represent the set of planes associated with a vertex node, and can be used to find the best-fit point to those planes. When we are working in the dual space, we instead use quadric functions of the face normal, $\mathbf{n}^T \mathbf{A} \mathbf{n} + 2\mathbf{b}^T \mathbf{n} + c$, to represent the set of vertices in a cluster node, and to find the best-fit plane to those points. Because of this, it can be shown that by applying the quadric error approach to the dual problem, face clusters can be made to preserve planarity where possible, in the same way that vertex simplification tries to preserve shape. We can also add an additional error term to ensure the clusters are well-shaped, in the sense of being as close to circular as possible. More detail can be found in [4].

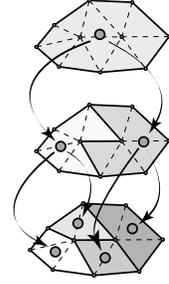


Fig. 3. A Face Cluster Hierarchy

For each node i in the face cluster hierarchy, we calculate an oriented bounding box for the faces it contains using principal component analysis [11]. We also store the sum of the area-weighted normals of those faces

$$\mathbf{S}_i = \sum_j A_j \hat{\mathbf{n}}_j \quad (1)$$

where $face\ j \in cluster\ i$. As we will show later, this is a useful approximation of the reflective qualities of the faces within the node.

3.2 Notation

Each element i has a known reflectance ρ_i , emittance e_i , area A_i , normal $\hat{\mathbf{n}}_i$, and unknown radiosity b_i , where bold symbols denote vectors. The distance between elements i and j is r_{ij} , and the unit vector from i to j is $\hat{\mathbf{r}}_{ij}$. The average visibility between two points on these elements is \bar{v}_{ij} : 1 for no occlusion, and 0 for total occlusion.

The *irradiance* E_i is the incident power per unit area. Calculating it exactly is typically intractable, so we approximate it using the point-to-point approximation of the form factor [3],

$$E_i = \sum_j \frac{(\hat{\mathbf{n}}_i^T \hat{\mathbf{r}}_{ij})(\hat{\mathbf{r}}_{ji}^T \hat{\mathbf{n}}_j)}{\pi r_{ij}^2} \bar{v}_{ij} A_j b_j. \quad (2)$$

We have written the traditional dot products in matrix notation because we will shortly be exploiting the associativity of matrix multiplication to rewrite this formula.

3.3 Vector-based Radiosity

The classical radiosity method assumes piecewise constant (Haar) basis functions and planar surfaces. In a hierarchy, the children are coplanar with the parent. For the purposes of projecting radiosities up and down the tree, radiosities are scalar quantities. If this method is applied to a multiresolution model, it causes curved or bumpy portions of the model to be shaded a flat colour, leading to a faceted appearance that hides the geometric detail (Figure 7c). This is similar to the step-function effect in constant-basis radiosity, but applying a post-process smoothing step at the leaves is no longer sufficient to cover up these discontinuities.

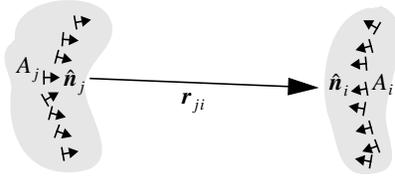


Fig. 4. Radiosity transfer between clusters of faces.

We now adapt the radiosity method to multiresolution models that use face clustering. Consider the light transfer from one cluster to another (Figure 4). We let j be an element in the source cluster and i be an element in the receiver cluster. If we assume that all (i,j) pairs are inter-

visible and that the sources are close together and far from the receiver, then $\hat{\mathbf{r}}_{ij}$, $\hat{\mathbf{r}}_{ji}$, r_{ij} , and \bar{v}_{ij} are independent of j , and we can approximate the irradiance from a single cluster as:

$$E_i \approx \hat{\mathbf{n}}_i^T \left[\frac{-\hat{\mathbf{r}}_{ij}^T}{\pi r_{ij}^2} \sum_j \hat{\mathbf{n}}_j A_j b_j \right] \bar{v}_i \quad (3)$$

This allows us to rewrite the transfer in terms of two vector quantities, so that:

$$E_i = \hat{\mathbf{n}}_i^T \mathbf{E}_i \quad (4)$$

where

$$\mathbf{E}_i = \bar{v}_i \left[\frac{-\hat{\mathbf{r}}_{ij}^T}{\pi r_{ij}^2} \right] \mathbf{P} \quad (5)$$

and

$$\mathbf{P} = \sum_j \hat{\mathbf{n}}_j A_j b_j = \sum_j \mathbf{S}_j b_j \quad (6)$$

We refer to \mathbf{E} as the irradiance vector [1], and \mathbf{P} as the power vector.

The irradiance vector \mathbf{E}_i 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 modelled. This eliminates most of the faceting effects of Figure 7c, as seen in Figure 7d.

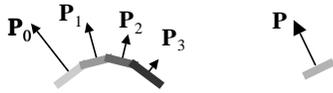


Fig. 5. The power vector. The radiosity emitted by the surfaces on the left is approximated by the elemental surface on the right, whose direction is the power-weighted sum normal.

Standard hierarchical radiosity effectively assumes that outgoing power is diffuse (isotropic) over the hemisphere above a planar surface. But the outgoing power from a cluster can be anisotropic due to occlusion. To permit nonplanar clusters to approximate their outgoing power compactly and quickly, we employ a 3-dimensional power vector. The magnitude of the vector approximates the total power leaving the cluster,

and the direction of the vector indicates the hemisphere toward which most of the energy is directed (Figure 5).

These formulas are generalizations of the standard radiosity equations; in the case of co-planar clusters, they reduce to the familiar hierarchical radiosity push-pull formulas.

We can substitute these vector quantities directly for the irradiance and radiosity in a standard hierarchical radiosity algorithm, although as \mathbf{P} is already area-weighted,

when pulling radiosity up the hierarchy we sum the power vectors of a node’s children, instead of averaging them. Instead of a single transfer coefficient, we store the *transfer vector*, $\mathbf{m} = \mathbf{r}/\sqrt{\pi r^2}$, which allows us to apply Equation 5 more simply as

$$\mathbf{E} = -\bar{v}\mathbf{m}\mathbf{m}^T\mathbf{P}. \quad (7)$$

At the leaves of the hierarchy, where we must transform the accumulated irradiance vectors into the power vector we apply the equation

$$\mathbf{P}_j = \mathbf{S}_j(\rho_j\hat{\mathbf{S}}_j^T\mathbf{E}_j + e_j). \quad (8)$$

As with standard radiosity, whenever these equations are applied, dot products must be clipped to zero to account for occlusion by the tangent plane to the surface.

The above treatment of vector-based transfer assumes a monochromatic world. It can easily be extended to the familiar RGB colour model; we simply store $\mathbf{P}_R, \mathbf{P}_G, \mathbf{P}_B, \mathbf{E}_R, \mathbf{E}_G, \mathbf{E}_B$, and operate on each pair of irradiance and power vectors independently. Note that the transport vector, \mathbf{m} , is still wavelength independent.

3.4 Algorithm Description

There are three types of nodes in the hierarchy used by our algorithm. At the top, volume clusters contain all unconnected parts of the scene. In the middle, face clusters contain connected surface meshes. At the bottom, there are polygonal elements, and refinements of those elements. In our implementation, all of these nodes use a common object-oriented interface to communicate with each other. Usually much of each face cluster hierarchy remains unused; only those face clusters at the top of the tree are paged in during the solution phase.

Radiosity using vector-based transfer proceeds in much the same manner as the irradiance/radiosity method first popularised by [6], and outlined in Figure 6. In Gershbein’s method, irradiance is gathered to each node in the hierarchy, and then pushed down to the leaves, whereupon it is converted to radiosity by the application of reflectance and emittance operators, and pulled back up the hierarchy. In our algorithm, the irradiance vector, rather than scalar, is pushed to the leaves, and the power vector, not scalar radiosity, is pulled back up the hierarchy. Below is an outline of the algorithm.

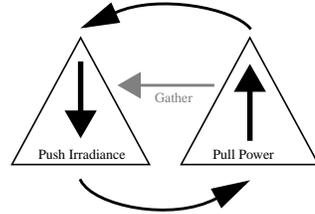


Fig. 6. Schematic of the Face Cluster Radiosity Cycle.

Preprocessing. Face cluster hierarchies are generated for the input models. These hierarchies are dependent only on the geometry of the models, and can be reused over multiple instantiations of each model. This process is done off-line, and typically only once, whenever a new model is acquired.

Initialisation. The scene description and its constituent models are read in. Hierarchical radiosity elements are then created for the small number of root face clusters in the scene, and these are volume clustered to complete the initial element hierarchy.

```

solve()
  while (not converged)
    gather(root)
    pushpull(root, 0)
    refine(root, ε )

gather(element i)
   $\Delta \mathbf{E}_i = - \sum_{\text{links } j \rightarrow i} \bar{v}_{ij} \mathbf{m}_{ij} \mathbf{m}_{ij}^T \mathbf{P}_j$ 
  for (all children c of i)
    gather(c)

pushpull(element i, vector  $\mathbf{E}$ )
  //  $\mathbf{E}$  is irradiance on i from parent
   $\mathbf{E} = \mathbf{E} + \Delta \mathbf{E}_i$ 
  if (i is a leaf)
    // convert irradiance to power
     $\mathbf{P}_i = \mathbf{S}_i (\text{Max}(\hat{\mathbf{S}}_i^T \mathbf{E}_p, 0) \rho_i + e_i)$ 
  else
     $\mathbf{P}_i = 0$ 
    for (all children c of i)
      pushpull(c,  $\mathbf{E}$ ) // push irradiance
       $\mathbf{P}_i = \mathbf{P}_i + \mathbf{P}_c$  // pull power

```

Solution. The solver proceeds according to the pseudocode at left.

The **refine** procedure follows that outlined in [3]. When a face cluster element needs to be subdivided, its two child elements are created using position and average normal information retrieved from the cluster file on disk. Storage for power vectors and other such element information is only required for those face cluster nodes actually being used by the solver.

Post Processing. After the algorithm has terminated, the radiosity solution is propagated to the leaves of the model by applying the irradiance vectors at the leaves of the transport tree to all their descendents (e.g., node T of Figure 1c). This final process is again $O(k)$, but is typically insignificant compared to the solution time. The radiosities of the vertices of the models in the scene

are then written out to disk. An alternative that we don't use is to perform a final gather on the hierarchy generated by the solution. This can generate extremely good quality results, but is view-dependent and is quite slow.

3.5 Estimating Transfer Coefficients and Error

We must handle a number of different types of transfer; radiosity exchange between face clusters, standard planar elements, and volume clusters. This can lead to problems in choosing an error metric that is consistent for all transfers. To handle these in a common framework we use sampling across the receiver to estimate the error in the transfer at the same time as we estimate the transfer itself. We use the L_1 norm to measure error, i.e., the *BFA*-weighted refinement discussed in [16]. A fixed number of sample points are generated across the cluster or face, and used both for estimating fractional visibility and determining bounds on the transfer [7, 12]. For links that are partly occluded, the refinement epsilon is reduced to encourage subdivision at shadow boundaries. (The refinement epsilon controls link subdivision; links with transport error greater than this are split.)

We estimate the transport vector as

$$\bar{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^n \mathbf{m}_i = \frac{1}{n} \sum_{i=1}^n \frac{\mathbf{r}_i}{\sqrt{\pi r_i^2}} \quad (9)$$

where \mathbf{r}_i is the vector from sample point i of n on the source s to sample point i on the

destination d . The L_1 norm can then be estimated by using the samples \mathbf{m}_i to evaluate

$$\Delta BFA = \left(\lceil \mathbf{S}_i^{sT} \mathbf{m}_i \mathbf{m}_i^T \mathbf{S}_i^d \rceil - \lfloor \mathbf{S}_i^{sT} \mathbf{m}_i \mathbf{m}_i^T \mathbf{S}_i^d \rfloor \right) \frac{\|\mathbf{P}^s\|}{\|\mathbf{S}^s\|}, \quad (10)$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote upper and lower bounds. This effectively measures the range of FA , weighted by the average emitted radiosity of the source. The problem of generating correlated area-weighted normal samples \mathbf{S}_i and transport samples \mathbf{m}_i can be addressed by using a constant number of children of each face cluster in question to generate the samples.

3.6 Implementation Details

To build volume clusters, we followed the methods described in [7, 15]. An octree that encloses the scene is created, and scene polygons are placed within that octree according to their size and position. Visibility is sampled using ray-tracing; the spatial data structure used for acceleration is a nested grid data structure.

Compared to the storage required for a face in standard hierarchical radiosity, we store 9 real numbers per hierarchical element instead of 3, and 3 reals per link instead of 1. Although the face cluster hierarchical elements are more expensive than standard Haar elements, they are in general more lightweight than volume cluster elements, which require 8 child pointers in our implementation, and much more lightweight than storing a general radiance distribution.

4 Results

We have compared face cluster radiosity to our own implementation of hierarchical radiosity with volume clustering, which follows those of Sillion and Gibson [7, 15].

4.1 The Museum Scene

We designed an indoor scene typical of those seen in the radiosity literature. This scene is lit by both sun and sky, and internally by three spotlights. Much of the light in the room is reflected from the overhead skylight by the detailed stone floor, providing a good test of complex interreflection.

The scene contains a number of high resolution scanned models, a polygonized implicit surface (the podium), and a displacement-mapped surface (the floor). These models range in complexity from 4,140 polygons to 1,000,000 polygons; it took from 1s to 600s to generate their associated face cluster hierarchies, for a total time of 1500s.

4.2 Empirical Complexity

Using the quadric-based surface simplification method [5] we applied progressive radiosity, hierarchical radiosity with volume clustering, and our face cluster radiosity algorithm to versions of the museum scene with varying polygon counts, to demonstrate the effect of using ever more detailed models on these algorithms. While previous experiments have investigated the effect of increasing the lighting or geometric complexity of a scene on radiosity algorithms [18], this experiment shows the effect of increasing model complexity in a scene with fixed geometric layout. The results were collected on

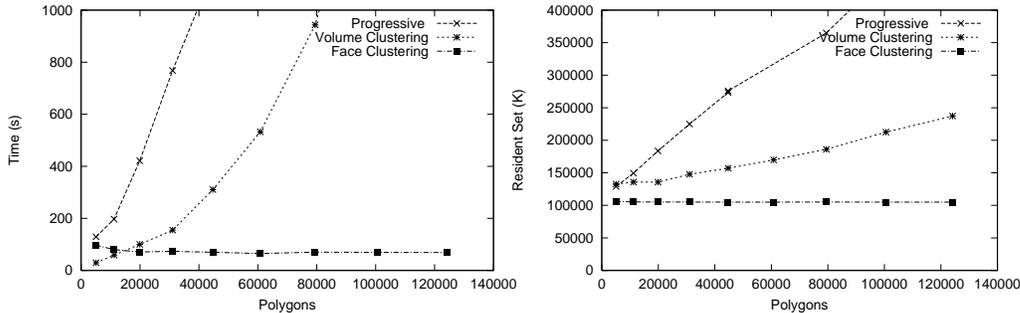


Fig. 7. Performance of hierarchical, progressive, and face cluster radiosity as model complexity increases. The graphs show (left) solution time and (right) maximum resident set size (memory usage).

an SGI Power Challenge with a 195MHz R10000 MIPS processor and 1Gb of main memory. While exact error comparisons between the solutions were not available due to the difficulty in generating a reference solution, we took care to use similar parameter settings for all three methods. When compared by eye, the solution meshes from the two hierarchical radiosity methods looked much the same. The progressive radiosity method did somewhat better on generating first order shadows, but very poorly on simulating second bounce illumination. At higher resolutions, almost no secondary illumination was detectable.

Figure 7 shows graphs of solution time and memory use for the methods tested. Notably, the time cost of the face cluster radiosity algorithm stays approximately constant, while previous methods have a cost that is super-linear in the number of input polygons. As suggested in Figure 1c, this is because fine levels of detail of the cluster hierarchy are never accessed, in spite of the increasing polygon count. In fact the solution time for face cluster radiosity is slightly greater for very low resolution versions of the test scene where face clustering provides little benefit and a significant number of leaf polygons must be refined.

The memory use of the face cluster radiosity algorithm also stayed constant at around 100MB maximum resident set size, and 120MB total memory use, significantly below that of the other methods. While the total memory use of the progressive radiosity method is typically better than that of hierarchical methods, it is apparent from this graph that its memory locality is much poorer. Also, because the highly tessellated floor accounts for much of the indirect illumination in the scene, the number of shooting steps is $\Theta(k)$, and the performance of progressive radiosity is the worst case $\Theta(k^2)$. The HRVC curve is approximately $\Theta(k \log k)$.

4.3 A Highly Complex Museum Scene

The use of face cluster radiosity enables an increase in quality and scene size over previous methods. We were able to run a radiosity simulation on the full museum scene, containing 2.7 million polygons, in under two minutes, including both solution and pre-processing time. (Neither of the two other radiosity methods could handle this complex a scene on our test machine.) The result is shown in Figure 9. Initialisation and solution

for the illumination in the scene, i.e., running the core face cluster radiosity algorithm, took 56 seconds. A further 53 seconds was devoted to postprocessing, which included both the final push-to-leaves phase, and refining the mesh locally for better visibility results when viewing the output mesh.

Preprocessing time for the models in this scene took a total of 1500s. We think that preprocessing time could be sped up significantly; our current algorithm is $O(k^2)$, which could be reduced to $O(k \log k)$ by using a convex hull algorithm to help calculate bounding boxes. As pointed out previously, this cost can be heavily amortized; the same models can be reused over multiple scenes and radiosity simulations. Indeed, this is the reason we have not yet optimized our face cluster hierarchy code.

4.4 Discussion

Our algorithm gives the greatest benefit for finely tessellated objects. For more intricate, space-filling objects such as trees, we rely on volume clustering to provide a good simulation. This is because of the limitation in our algorithm that each topologically connected surface has its own face cluster tree. For instance, a pile of pebbles might need a separate tree for each pebble, if those pebbles were modelled separately. While we currently aggregate disconnected components by using volume clustering, better methods for tightly packed components such as the pebble pile are an avenue for future research.

Many of the input polygons in Figure 9 are invisibly small. Arguably a similar picture could be generated using a much simpler scene. However, this ignores the possibility of using the output mesh for a walk-through, where we might wish to pass much closer to some of the models in the scene. Also, it is tedious to manually preprocess a scene in a view-dependent manner to optimise radiosity simulations. Ultimately, we wish the radiosity simulation to be as independent of the model resolution chosen for viewing as possible.

5 Conclusions

We have designed and implemented a new hierarchical radiosity algorithm based on face clustering and a vector formulation of radiosity. The algorithm yields sub-linear performance in the number of input polygons, an improvement over the linear performance of the previous fastest volume clustering algorithms. The principal reason for this speed-up is that the face cluster radiosity algorithm greatly reduces memory use. Unlike previous clustering methods for hierarchical radiosity, it does not require that information be pushed down to the leaves of the hierarchy during the solution phase.

This is possible because of the combination of face clustering and vector-based radiosity. Face clustering ensures that each cluster of faces has a (reasonably) small range of normal vectors, so it can more accurately be approximated by a single normal than could a volume cluster containing a set of faces with a wide range of normals. Vector-based radiosity provides an inexpensive representation for gross directional variation in irradiance and outgoing power. The vector representation is intermediate in complexity between the traditional representation of radiosity algorithms, a scalar quantity, and the more sophisticated directional techniques employed in radiance algorithms [2, 14].

Face cluster radiosity paves the way for the application of radiosity to high-com-

plexity scenes, implicit surfaces, and displacement-mapped polygons. Together with the ability to handle textures, this makes the creation of animation using radiosity solutions more practical.

Several ideas for future work present themselves: A more complete error analysis of the approximations made in this algorithm should be done. This could help reduce the cluster artifacts that sometimes appear with this method. We are currently exploring the use of conservative bounds in vector radiosity transfer, and results look promising. By generalizing the method to work with directional distributions, not just vector irradiance, global illumination in non-diffuse scenes could be simulated.

Acknowledgements

Funding was provided by NSF grants CCR-9505472, CCR-9357763, and DMI-9813259, and the Schlumberger Foundation. Thanks to Joel Welling for machine access and Francois Sillion for discussions.

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] Per H. Christensen, Dani Lischinski, Eric J. Stollnitz, and David H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1):3–33, January 1997.
- [3] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston, MA, 1993.
- [4] Michael Garland. *Quadric-Based Polygonal Surface Simplification*. PhD thesis (Technical Report CMU-CS-99-105), Carnegie Mellon University, 1999. <http://www.cs.cmu.edu/~garland/thesis>.
- [5] 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.
- [6] Reid Gershbein, Peter Schröder, and Pat Hanrahan. Textures and radiosity: Controlling emission and reflection with texture maps. In *Computer Graphics (Proceedings of SIGGRAPH '94)*, pages 51–58, July 1994. <http://www-graphics.stanford.edu/papers/texture>.
- [7] S. Gibson and R. J. Hubbard. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5):297–310, 1996.
- [8] Gene Greger, Peter Shirley, Philip M. Hubbard, and Donald P. Greenberg. The irradiance volume. *IEEE Computer Graphics & Applications*, 18(2):32–43, March–April 1998.
- [9] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (SIGGRAPH '91 Proceedings)*, July 1991.
- [10] Hugues Hoppe. View-dependent refinement of progressive meshes. In *SIGGRAPH 97 Proc.*, pages 189–198, August 1997.
- [11] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, New York, 1986.
- [12] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and Error Estimates for Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 67–74, 1994.
- [13] Holly Rushmeier, Charles Patterson, and Aravindan Veerasamy. Geometric simplification for indirect illumination calculations. In *Proceedings of Graphics Interface '93*, pages 227–236, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.
- [14] François Sillion, G. Drettakis, and Cyril Soler. A clustering algorithm for radiance calculation in general environments. In *Eurographics Rendering Workshop 1995*. Eurographics, June 1995.
- [15] François 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.
- [16] 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.
- [17] 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.
- [18] Andrew J. Willmott and Paul S. Heckbert. An empirical comparison of radiosity algorithms. Technical Report CMU-CS-97-115, School of Computer Science, Carnegie Mellon University, April 1997. <http://www.cs.cmu.edu/~radiosity/emprad-tr.html>.



(a) HRVC, 108,000 triangles, 707s



(b) HRVC, 1000 triangles, 7 s



(c) Scalar FCR, 108,000 triangles, 7s



(d) Vector FCR, 108,000 triangles, 8s

Fig. 7. Face cluster radiosity (FCR) and hierarchical radiosity (HRVC) algorithms applied to a detailed dragon model.

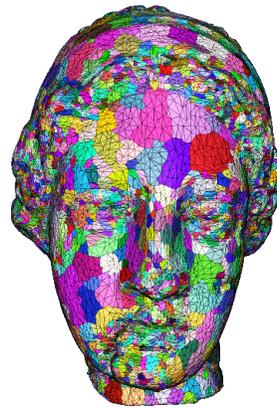


Fig. 8. Face clusters on the Venus head model. A total of 8000 randomly coloured clusters are shown.



Fig. 9. The museum scene: diffuse interreflection simulated with face cluster radiosity. The input scene contained 2,700,000 polygons. Solution plus post-processing took two minutes and 120MB of memory to generate.