A Clustering Algorithm for Radiosity in Complex Environments

Brian Smits *James Arvo* *Donald Greenberg*

Program of Computer Graphics*
Cornell University

Abstract

We present an approach for accelerating hierarchical radiosity by clustering objects. Previous approaches constructed effective hierarchies by subdividing surfaces, but could not exploit a hierarchical grouping on existing surfaces. This limitation resulted in an excessive number of initial links in complex environments. Initial linking is potentially the most expensive portion of hierarchical radiosity algorithms, and constrains the complexity of the environments that can be simulated. The clustering algorithm presented here operates by estimating energy transfers between collections of objects while maintaining reliable error bounds on each transfer. Two methods of bounding the transfers are employed with different tradeoffs between accuracy and time. In contrast with the $O(s^2)$ time and space complexity of the initial linking in previous hierarchical radiosity algorithms, the new methods have complexities of $O(s \log s)$ and $O(s)$ for both time and space. Using these methods we have obtained speedups of two orders of magnitude for environments of moderate complexity while maintaining comparable accuracy.

CR Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism.

Additional Key Words: clustering, error bounds, hierarchical radiosity, global illumination.

1 Introduction

Recent trends in realistic image synthesis have been towards a separation of the rendering process into two or more stages[10, 2, 9]. One of these stages solves for the global energy equilibrium throughout the environment. This process can be very expensive and its complexity grows rapidly with the number of objects in the environment. These computational demands generally limit the level of detail of environments that can be simulated. Furthermore, a solution to this problem must be computed before anything useful can be displayed.

Radiosity algorithms attempt to solve the global illumination problem by discretizing the environment and solving a linear system to approximate the transfer of energy between the elements [5]. For complex environments, the large number of interactions is expensive to compute, as each requires form factor and visibility calculations. The challenge is to reduce the computational complexity of this process. In one of the early approaches, Cohen et al. [3] reduced the number of interactions by imposing a two-level hierarchy of patches and elements on the environment. Although the number of interactions was reduced, the approach was still $O(p^2)$ in the number of elements in the environment.

Currently, the best radiosity algorithms are analogous to linear-time algorithms for charged particle simulation [6]. These algorithms work by clustering particles together so that the mutual effect of well-separated collections can be approximated with a single interaction. Hanrahan et al. [7] used a similar strategy to reduce the number of interactions needed for a radiosity solution. A hierarchical structure was imposed on each surface in an environment and interactions were allowed to occur between the appropriate levels on each. This approach works well when the number of initial surfaces is small, as hierarchical radiosity (HR) algorithms can "subdivide" large surfaces into smaller ones, but cannot "group" smaller elements into larger ones. The initial linking phase of HR must check all pairs of initial surfaces for potential interactions; without grouping surfaces together the algorithm is quadratic in the number of initial surfaces. Because complexity in environments is often a result of replacing large surfaces with many small surfaces, this inability to group objects together is a major obstacle in conventional HR algorithms.

Several methods have been developed to increase the efficiency of HR algorithms for complex environments. Global visibility [13] is effective for computing the initial links for environments in which only a small number of "cells" see each other. Many environments, however, also contain large collections of objects that are mutually visible. In such cases, global visibility algorithms do not suffice.

Importance [12] is another method that reduces computation in areas that have little or no noticeable effect on the surfaces of interest. This approach is effective once the initial interactions have been computed, but does nothing to reduce the time of the initial linking phase. Importance driven hierarchical radiosity stands to gain even more from clustering than standard hierarchical radiosity. Because it attempts to compute the radiance very coarsely in some regions of the environment, some means of clustering is needed if it is to do less work than that required for the initial linking.

An approach to clustering was developed by Rushmeier et al. [11] in which the effect of complex groups of surfaces is approximated by simpler representations resembling BRDF's obtained through Monte Carlo sampling. One disadvantage of this approach is that it is not automatic; appropriate clusters must be specified and approximated in advance. Also, no hierarchy is maintained, so the coarsest representation used by the algorithm is that of the selected clusters. Another clustering approach was proposed by Kok[8], which extended progressive radiosity so that patches could transfer energy to

*580 ETC Building, Cornell University, Ithaca, NY 14853
E-mail: {bes | arvo | dpg}@graphics.cornell.edu

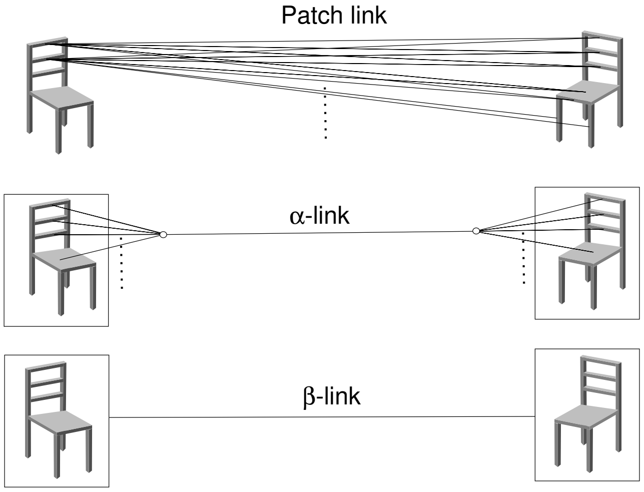

Figure 1: *Two collections of objects can interact at different levels: a) conventional HR links, b) an α-link requiring linear time and space, and c) a β-link requiring constant time and space.*

groups of surfaces at once. Neither of these approaches analyze the error of the approximations used for the transfers.

To illustrate why clustering is important, consider a simple configuration consisting of two chairs, each containing 100 surfaces. HR would check the 10,000 potential interactions between the two chairs and would create in the neighborhood of 2,500 patch links (Figure 1). However, if the chairs are well separated, then it is unlikely that the energy transfer between them will have a significant effect on the illumination of either. If we can guarantee this throughout the entire solution, we can gain efficiency by coarsely approximating the transfer of energy between them. We shall present a new strategy for linking that would handle this configuration by creating a single link with a cost comparable to either 1 or 200 conventional links, depending on the separation of the objects. By reducing the total cost of the links created between two collections of objects, we reduce the algorithmic complexity of the $O(s^2)$ initial linking step in HR.

1.1 Overview

Hierarchical algorithms for radiosity have three components: 1) a hierarchical description of the environment, 2) a criterion for determining the level in the hierarchy at which two objects can interact, and 3) a means of estimating the energy transfer between the objects.

As our criterion for interaction, we use bounds on the potential error in the transfer of energy between two objects. We first describe hierarchical radiosity using this approach, and then present two efficient techniques for bounding the transfers between clusters. The first technique is a fairly accurate approach which we call α-linking. The second is a faster but less accurate approach which we call β-linking. Corresponding to these methods for determining bounds are two ways of estimating the transfer of energy between clusters. We also describe the method used to create the hierarchy of clusters, which is the starting point for the cluster-based linking strategies. Finally we give theoretical bounds on the complexity and results of our implementation demonstrating dramatic speedups over conventional HR in complex environments.

2 Hierarchical Radiosity

The value of the radiance function L on a surface at a point x due to another surface S_i can be expressed as

$$L(x) = \int_{S_i} k(x, y)L(y)dy$$

where the kernel $k(x, y)$ expresses the radiance at x due to a differential area at the point y. The kernel can be written

$$k(x, y) \equiv \rho(x)\frac{\cos\theta_1 \cos\theta_2}{||x - y||^2}\text{vis}(x, y)$$

where θ_1 is the angle formed by the normal of the differential area at x and the direction given by $y - x$ and θ_2 is the angle formed by the normal of the differential area at y and the the direction given by $x - y$. Also ρ is the bidirectional reflectance distribution function (BRDF). In this paper we shall only address ideal diffuse scattering, although much of the analysis extends to general reflectance functions.

To compute the exact radiance function across the receiver, $L(x)$ must be evaluated at every point on the receiver. In general, some set of basis functions are used to represent the radiance functions and quadrature rules are used to compute the coefficients for the basis functions. This reduces the problem to evaluating $L(x)$ at some fixed number of points.

For an environment meshed to a fixed number of patches p, there are potentially p^2 interactions. HR maintains a hierarchical representation of each initial surface; for example, by means of a quadtree. HR then allows surfaces interact at a level in the hierarchy where all the interactions have an error less than some bound. This requires bounding the error in the transfer between two patches, which can be done by bounding the difference between the maximum and minimum transfers[12]. The maximum transfer can be computed by finding the maximum value of $k(x, y)$ over all points on the two patches and then integrating the product of this with the radiance of the source. We will denote the maximum value of a function f over some range $A \times B$ as

$$\lceil f \rceil_{A,B} \equiv \max_{\substack{x\in A\\ y\in B}} f(x, y).$$

Using this notation the bound on the energy transfer may be written

$$\max L(x) \leq \lceil k \rceil_{R_j,S_j} \int_{S_i} L(y)dy.$$

This expression allows for arbitrary distributions of radiance across the source. The minimum can be computed similarly, with the minimum set to zero if any two points are occluded. The maximum and minimum values of k are computed by taking a set of jittered samples on both the source and receiver and computing the maximum and minimum values of the kernel between all pairs of samples [12]. Occasionally, this approach greatly underestimates the maximum, such as when a small patch is very close to a large patch. Such an underestimate usually will not prevent further subdivision, however, and the problem diminishes as the two patches approach the same size. Bounding the error in the transfer of energy and only computing interactions to a given accuracy results in a linear number of interactions [7]. The transfer of energy between the two surfaces can then be determined using any of the various form-factor methods.

The main deficiency of HR is that the coarsest representation is the initial surfaces. By collecting a group of these surfaces together, we obtain an even coarser representation. To make use of this coarser representation, we require a bound on the transfer of energy between two such clusters. This bound must be computed in less

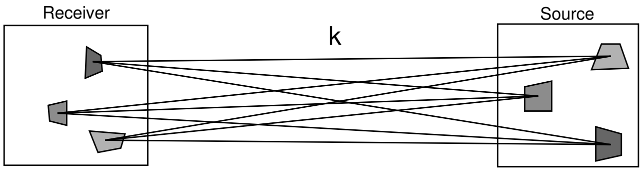

Figure 2: *Straightforward approach to bounding error. Each link represents a maximum value for k.*

than $O(s^2)$ time if we are to gain an improvement over the brute force method of checking all pairs of interactions. We now describe two ways of bounding error, each with a time complexity better than $O(s^2)$.

3 Bounding error on clusters

In this section we describe two strategies for computing bounds on the energy transfer between clusters; one requiring linear time and space, and one requiring constant time and space. The strategies are derived by systematically introducing approximations into the exact expression for energy transfer. The first level of approximation results in a type of link we call an α-link. By introducing further approximations we produce β-links. Although these bounds are coarse, they often suffice for a very large fraction of the interactions.

3.1 α-links

The transfer of energy from a cluster of patches $\mathcal{S}$ to a point can be computed by summing over the n patches in the source cluster

$$L(x) = \sum_{i=1}^{n} \int_{S_i} k(x,y)L(y)dy.$$

Since $k(x,y)$ is a function of the position and orientation of the receiver as well as the source and of the visibility between them, computing the transfer of radiance between two clusters requires evaluating $L(x)$ at least once on each of the m receivers, resulting in $O(mn)$ work.

Applying the same approach used for HR to two clusters is relatively expensive. To bound the transfer between the clusters with this approach, we can use the maximum value of $k(x,y)$ over pairs of patches to bound the transfer. The maximum radiance over all receivers due to the source cluster can be bounded by

$$\max L(x) \le \max_j \sum_{i=1}^{n} \lceil k \rceil_{R_j,S_i} \int_{S_i} L(y)dy.$$

The maximum value of the kernel function must be computed for each pair of patches, so the time complexity is $O(mn)$. (See Figure 2.)

To improve the time complexity, we split the kernel function into two simpler functions $k^r(x,y)$ and $k^s(x,y)$ given by

$$\begin{aligned} k^r(x,y) &\equiv \rho(x)\cos\theta_1 \\ k^s(x,y) &\equiv \frac{\cos\theta_2}{||x-y||^2}. \end{aligned}$$

Then

$$k(x,y) = k^r(x,y)k^s(x,y)\mathrm{vis}(x,y).$$

Each of the new functions depends on two points, but each requires information about the orientation of only one surface. k^r accounts for the projected area and reflectivity of a differential area around x and k^s corresponds to the differential solid angle subtended by a differential area around y. We now show that the above separation of the kernel can be used to conservatively bound the interaction between the source cluster and the receiver cluster. This is done by bounding the energy that reaches the receiving cluster from the source, and then determining how much of this energy potentially reaches each receiving patch. (See Figure 3.)

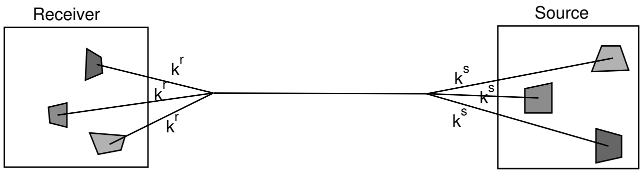

Figure 3: *More efficient α-link between two clusters.*

We now use these pieces of the kernel to bound the radiance function at a point x. First, $k^r(x,y)$ is replaced by the maximum value it attains over all y in the volume $B(\mathcal{S})$ containing all the source patches. Then, for each of the sources, we replace $k^s(x,y)$ by the maximum value it attains over all y on the source patch. We can also assume that the visibility term is always 1. These steps produce the following conservative upper bound:

$$\begin{aligned} L(x) &= \sum_{i=1}^{n} \int_{S_i} k^r(x,y)k^s(x,y)\mathrm{vis}(x,y)L(y)dy \\ &\le \lceil k^r \rceil_{x,B(\mathcal{S})} \sum_{i=1}^{n} \int_{S_i} k^s(x,y)L(y)dy \\ &\le \lceil k^r \rceil_{x,B(\mathcal{S})} \sum_{i=1}^{n} \lceil k^s \rceil_{x,S_i} \int_{S_i} L(y)dy. \end{aligned}$$

We can apply this idea to obtain upper bounds on the transfer between the two clusters. The quantity $k^s(x,y)$ can be maximized over all points in the receiving volume $B(\mathcal{R})$ making $k^s(x,y)$ independent of any particular receiver. The factor $k^r(x,y)$ can now be maximized over each receiving patch separately. Thus, the radiance function can be bounded as follows

$$\max L(x) \le \max_j \lceil k^r \rceil_{R_j,B(\mathcal{S})} \sum_{i=1}^{n} \lceil k^s \rceil_{B(\mathcal{R}),S_i} \int_{S_i} L(y)dy.$$

This expression can be separated into two pieces by defining

$$L_{S_{\max}} \equiv \sum_{i=1}^{n} \lceil k^s \rceil_{B(\mathcal{R}),S_i} \int_{S_i} L(y)dy,$$

which is a bound on the flux density incident upon the receiving bounding volume. Then

$$\max L(x) \le \max_j \lceil k^r \rceil_{R_j,B(\mathcal{S})} L_{S_{\max}}.$$

Maximizing over all of the receiving patches in $\mathcal{R}$ requires $O(m)$ work. Therefore the time complexity of computing this bound has been reduced from $O(mn)$ to $O(m+n)$.

The bounds needed for HR can be computed from upper and lower bounds on the transfer. We have given upper bounds; a trivial lower bound of zero can always be used. This lower bound is attained when the visibility between all points in the two clusters is zero. Now we have a bound on the error in the interaction between two clusters, which allows us to determine when an interaction is accurate enough. This bound on clusters determines if an α-link is an acceptable approximation for the interaction.

In addition to the error bound, we require an estimate for the energy transferred to each receiving patch across the α-link. We do this by computing an estimate as well as the bound for the $k^r(x, y)$ associated with each receiving patch and the $k^s(x, y)$ associated with each source patch. Let

$$\langle f \rangle_{A,B} \equiv \mathrm{avg}_{\substack{x \in A \\ y \in B}} f(x, y)$$

Now the radiance on receiving patch j can be estimated by

$$L_j = \langle k^r \rangle_{R_j, B(\mathcal{S})} L_{S_{\mathrm{avg}}}$$

where

$$L_{S_{\mathrm{avg}}} = \sum_{i=1}^{n} \langle k^s \rangle_{B(\mathcal{R}), S_i} \int_{S_i} L(y) dy.$$

Each average is bounded above by the maximum transfer and therefore falls within the error bounds; hence it is sufficiently accurate. In fact, it is quite likely far *more* accurate than the conservative bounds indicate.

We now show that the α-link clustering approach has time and space complexity of $O(s \log s)$. First, assume we have s initial patches stored in a hierarchy of clusters with each cluster containing two smaller clusters or, at the leaves, a single patch. This structuring results in a binary tree with a depth of $\log s$. The total number of clusters at level d is 2^d and each cluster at level d has $s/2^d$ patches. Now, following Hanrahan et al. [7] and Greengard [6], we assume each cluster is linked to a constant c_1 number of other clusters resulting in $c_1 2^d$ α-links on level d. We can also assume that each cluster is linked to other clusters at approximately the same level in the tree. An α-link between two clusters requires space and time proportional to the size of the clusters. Summing the costs for each of the $\log s$ levels of the hierarchy gives

$$\sum_{d=0}^{\log s} \mathrm{num\_links}_d \, * \, \mathrm{link\_cost}_d = \sum_{d=0}^{\log s} c_1 2^d c_2 \frac{s}{2^d} = Cs \log s.$$

Therefore, rather than the $O(s^2)$ work required for a direct approach, clustering using α-links gives a time and space complexity of $O(s \log s)$.

3.2 β-Links

For many transfers, the previous technique is still too expensive. Often large numbers of interactions are insignificant and can be treated very coarsely or ignored altogether [9]. In this section we introduce a more efficient but cruder method for bounding the interaction between two clusters. We do this using a very simple bound on the kernel $k(x, y)$ which we denote $k^d(x, y)$,

$$k(x, y) \leq k^d(x, y) \equiv \frac{1}{||x - y||^2}.$$

This bound requires no knowledge about the orientation of the surfaces. We can bound the transfer of energy between two clusters by replacing k by the maximum value of k^d over all points x in the receiving cluster and all points y in the source cluster. (See Figure 4.) Thus,

$$\max L(x) \leq \left\lceil k^d \right\rceil_{B(\mathcal{R}), B(\mathcal{S})} \sum_{i=1}^{n} \int_{S_i} L(y) dy.$$

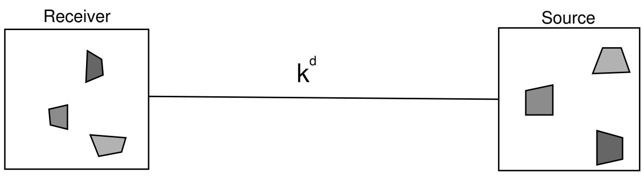

Figure 4: *β-link between clusters.*

This is a worst-case bound on the transfer between two clusters; it is only achieved when all surfaces are as close as possible to the other cluster, and each source directly faces all the receivers. The virtue of this bound is that it requires no knowledge of the surfaces. If the integral of the radiance over all the source patches has been computed in advance, during the sweep operation described in section 5.1, then the bound can be computed in constant time. As with the α-links, we can use the average value of k^d over the two clusters as an estimate of the energy transferred between them. Again, since the average value lies within the range given by the maximum transfer and a minimum transfer of zero, it meets the error tolerance.

The time and space complexity of a system with β-links is equivalent to the complexity of the standard hierarchical radiosity method. As with linking patches, the error term decreases rapidly with distance, so each cluster will be linked to a constant number of other clusters. Since the number of clusters is linear in the number of initial surfaces, and the cost of a β-link is independent of the size of the clusters, the complexity is $O(s)$ where s is the number of initial surfaces.

3.3 Strategies for linking

We have described two different approaches for linking clusters. The more accurate clustering technique tends to be too expensive for many clusters in a large environment. The faster clustering technique is too coarse to eliminate many of the negligible interactions. Our approach is to exploit the strengths of both strategies. If the coarse approach does not produce an acceptable bound then the more accurate technique is invoked. If neither are accurate enough, then the clusters are recursively subdivided. Only when the level of individual patches is reached does the algorithm resort to conventional HR.

4 Linking Criteria

Norms provide a measure of the "size" of a function. We shall use two different norms to quantify error in the energy transfer between two patches or clusters. In the previous section we computed a bound on the maximum possible difference between the approximated radiance and the actual radiance over all points on the receiving object. This corresponds to computing a bound on the ∞-norm of the radiance function on the receiver due to the source. More formally, if L is the exact radiance function over the receiver, and $\widetilde{L}$ is the computed radiance function then

$$||L - \widetilde{L}||_\infty = \max_{x \in \mathcal{R}} (L(x) - \widetilde{L}(x)) \leq M$$

where M is one of the previous bounds on the transfer.

The ∞-norm gives a bound on the variation between the computed radiance and the actual radiance. Another useful bound is the energy of the difference between the computed and actual radiance functions due to a link. This bound corresponds to the 1-norm.

More formally,

$$||L - \widetilde{L}||_1 = \pi \int_{\mathcal{R}} |L(x) - \widetilde{L}(x)|dx$$

The 1-norm can be bounded by weighting each term of the receiver by the area of the receiver and summing instead of finding the maximum value. We can write this bound for α-links as

$$||L - \widetilde{L}||_1 \le \pi \sum_{j=1}^{m} \left(A_j \lceil k^r \rceil_{R_j, B(\mathcal{S})} \sum_{i=1}^{n} \left(\lceil k^s \rceil_{B(\mathcal{R}), S_i} \int_{S_i} L(y)dy \right) \right)$$

where A_j is the area of receiver j. Coarse bounds and patch-to-patch bounds are computed similarly. In a hierarchical solution both norms are easily accommodated, but the norm used affects the subdivision strategy. After performing a local pass to display a solution, the error at each pixel is important, which implies that the ∞-norm is most appropriate. However, during the global propagation of energy throughout an environment, it may be more useful to minimize the 1-norm of the error. Intuitively, this is because large surfaces will often have a much more significant effect on the local pass than the smaller ones.

Another useful norm is obtained from the importance weighting [12]. This corresponds to a norm very similar to the 1-norm; rather than weighting each receiver by the area, the receiver is weighted by its importance.

5 Implementation

5.1 Clustering

Most ray tracing acceleration techniques collect nearby objects together into groups. Some of these, such as octrees and hierarchical bounding volumes, form a natural hierarchy. The same hierarchy can be used to identify clusters. For simplicity, each object should appear in only one cluster. Otherwise, some method of eliminating copies of objects is needed to prevent an object from contributing to a cluster twice. For these reasons we chose to use bounding volume hierarchies [4] instead of octrees to generate our hierarchy of clusters. We use axis-aligned rectangular bounding boxes for both the clusters and for ray casting.

The HR algorithm can be combined with a bounding volume hierarchy to perform clustering fairly easily. The bounds on the error given earlier fit naturally into the brightness-weighted refinement scheme of Hanrahan et al. [7]. This approach first links the surfaces together, then performs several iterations of energy propagation and refinement of the system. The error tolerance is gradually reduced until the final error tolerance is reached. This approach is a variation on the "multigridding" method for iteratively solving systems of linear equations.

Very few modifications are needed to make to the bounding volume data structures useful for clustering. For the more accurate clustering approach, we must loop over all of the patches in the bounding volume. Each link will contain four arrays that hold both the maximum and average values of k^s for all the source patches as well as the maximum and average values of k^r for all of the receiving patches. In our implementation the maximum values are approximated by taking a fixed number of jittered samples on the patch and bounding volume, then finding the maximum value of the kernel at these points. Although this method does not produce a guaranteed bound, it tends to produce a good approximation. If two bounding volumes overlap, the maximum values can be unbounded. In this case we can choose a maximum based on the maximum potential transfer of energy in the environment. We store an estimate of the inter-cluster visibility in the link as well. Visibility is estimated using ray casting. The following pseudocode shows the steps needed to compute an α-link between two clusters.

<u>Create-α-link(Src, Rec)</u>
α-link T
T.Vis = EstimateVisibility(Src,Rec)
foreach patch i in Src
    T.SrcMax[i] $\leftarrow \lceil k^s \rceil_{B(Rec),S_i}$
    T.SrcAvg[i] $\leftarrow \langle k^s \rangle_{B(Rec),S_i}$
foreach patch j in Rec
    T.RecMax[j] $\leftarrow \lceil k^r \rceil_{R_j,B(Src)}$
    T.RecAvg[j] $\leftarrow \langle k^r \rangle_{R_j,B(Src)}$
if α-Bound(T) < ϵ **then return** TRUE
else return FALSE

Procedure α-Bound(T) computes the bound on the maximum amount of energy transferred across α-link T. We have estimates of the maximum values of k^s and k^r stored in the link. The integral of the radiance function L_i over patch i for constant patches is the product of the radiance L_i and the area A_i of the patch. The procedure for computing the error incurred by an α-link between clusters is shown in the following pseudocode:

<u>α-Bound(T)</u>
SrcErr $\leftarrow$ 0
MaxErr $\leftarrow$ 0
foreach patch i in T.Src
    SrcErr $\leftarrow$ SrcErr + T.SrcMax[i] * A_i * L_i
foreach patch j in T.Rec
    MaxErr $\leftarrow$ Max(MaxErr, T.RecMax[j] * SrcErr)
return MaxErr

In addition to computing the error on the links, energy must be transferred between the two clusters. During the gather step of HR, energy is transferred across each link, including all cluster links. The gather across an α-link T is performed as follows.

<u>α-Gather(T)</u>
SrcRad $\leftarrow$ 0
foreach patch i in T.Src
    SrcRad $\leftarrow$ SrcRad + T.SrcAvg[i] * A_i * L_i
SrcRad $\leftarrow$ SrcRad * T.Vis
foreach patch j in T.Rec
    $L_j \leftarrow L_j$ + T.RecAvg[i] * SrcRad

For the coarse approach to clustering using β-links, we store the sum of the areas of all the patches in the bounding volume. We also store the radiance of the bounding volume for use as a source, and the irradiance striking the bounding volume for use as a receiver. If the radiance on each patch of the receiving cluster were updated each time a link was used in a gather, then gathering through a single link would require linear time, resulting in worse complexity bounds for coarse links. As in HR, after transferring energy across the links the radiances must be swept to each object's parents and children. For clusters, this sweep is a slightly different from the sweep described in HR since each cluster holds the irradiance I incident upon it, and the radiance leaving it. Irradiance is pushed down the bounding volume hierarchy to the patches. The incident irradiance is then weighted by the reflectance of the patch and the resulting radiance is pushed down the patches as in HR. Pulling radiance up from the children to the parents is the same as in HR. Each parent receives the area-weighted average of the radiances of its children. The following pseudocode is used when solving the linear system to redistribute the energy gathered across the links. HSweep is the conventional HR sweep procedure.

Sweep(C, I_{down})

$I_{down} \leftarrow I_{down} + I_C$
if C is a leaf **then**
  foreach patch R_j in C
    $L_j \leftarrow L_j + I_{down} * \rho_j$
    $L_{up} \leftarrow L_{up}$+ HSweep(R_j) $* A_j / A_C$
else foreach C' in C
    $L_{up} \leftarrow L_{up}$ + Sweep(C', I_{down}) $* A_{C'}/A_C$
$L_C \leftarrow L_{up}$
return L_{up}

The β-links between clusters are computed in a similar fashion to the α-links. They are easier to compute, however, as no looping over the contents of the bounding volume is required. Also, the maximum value for k^d is the inverse of the square of the minimum distance between the clusters, which can be computed with little work for axis aligned bounding volumes. Each β-link stores only the maximum value and the best computed value of k^d for the two clusters.

Create-β-link(Src,Rec)

β-link T
vis $\leftarrow$ EstimateVisibility(Src,Rec)
T.Max $\leftarrow \lceil k^d \rceil_{B(Rec),B(Src)}$
T.Avg $\leftarrow$ vis * $\langle k^d \rangle_{B(Rec),B(Src)}$
if β-Bound(T) $< \epsilon$ **then return** TRUE
else return FALSE

The error in a β-link between clusters is also straightforward, following directly from the bounds on the transfer and the previous discussion for higher accuracy links. Computing the error on a β-link T can be implemented as:

β-Error(T)

return T.Max * $L_{\text{T.Src}} * A_{\text{T.Src}}$

During the gather phase of the algorithm, each β-link transfers energy to the receiving cluster. It is stored there as irradiance until the sweep phase when it is scaled by the BRDF.

β-Gather(T)

$I_{\text{T.Rec}} = I_{\text{T.Rec}}$ + T.Avg $* L_{\text{T.Src}} * A_{\text{T.Src}}$

The above procedures allow the clusters and patches of the environment to be linked together. When two bounding volumes are to be linked it is possible to first check for complete occlusion. This can be done conservatively by checking to see if a convex object completely obstructs the shaft between the source and the receiver. If the two bounding volumes are not completely occluded, then the β-bound between the clusters is checked. When that is not sufficiently accurate the α-bound is checked. If this is still not accurate enough, then the children of the larger cluster are recursively refined against the smaller cluster. If neither of the clusters has children, then the patches in both are refined against each other using the patch refinement from HR. The clustering algorithm begins by using the top-level bounding box both as source and as receiver, which triggers the recursive refinement.

5.2 Local Pass

Once a global solution has been computed, displaying the result usually involves smoothing the values computed for each patch so that patch boundaries are not visible. This can be done by Gouraud-shading the surfaces, however this is very prone to artifacts in hierarchical systems [12]. A better approach is to use the techniques of Lischinki et al. [9] and create a separate mesh for display purposes, using the information obtained by the global solution, and recomputing some parts of the illumination on each surface. Our approach is based on this method as well as on the method proposed by Reichert [10]. Given a view, we approximate the intensity

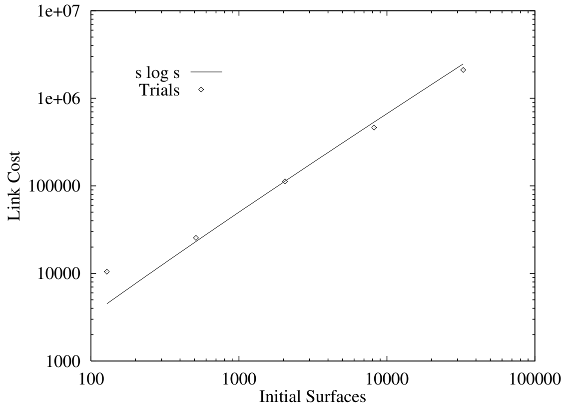

Figure 5: *Link cost for environments of different sizes compared with the function* $s\log(s)$.

by reconstructing the radiance function at each visible point in the environment. This is done by taking the collection of links at all levels of the hierarchy that directly affect this point, and for each link recomputing the form factor from the source to the point. The accuracy is determined by the magnitude of the error on the links, and can result in either recomputing just the unoccluded form factor, or both the form factor and the visibility from each source to the point. Although this method produces high quality results, it is computationally expensive. The techniques of this paper only address methods for efficiently computing the global pass. Further research is required to accelerate the local pass.

6 Results

HR has a complexity bound (ignoring the visibility term) of $O(s^2 + p)$ where s is the number of initial surfaces and p is the number of resulting patches. The $O(s^2)$ term comes from computing the initial links between all initial surfaces. Using clustering we do not need to check all pairs of surfaces even if they are mutually visible. Clustering replaces the expensive initial linking step with an algorithm that is $O(s\log s)$ in the number of initial surfaces. This results in an algorithm with an overall complexity of $O(s\log s + p)$. The sections that follow report results for several different environments and refinement strategies.

6.1 Link Complexity

We first show how the cost of the links grows with the size of the environment. As a first example, the inside of a sphere was tessellated into triangles. Each triangle was given the same emissive power and reflectance. This allowed us to vary s over a large range of values without really changing the geometry. Also, because every surface can see every other surface, it is easy to determine exactly how many initial links would be needed without clustering. It is also a challenging environment for clustering because every cluster overlaps with several other clusters. We assign a cost of 1 for patch links and β-links. The cost for α-links is $m + n$ where m is the number of patches in the receiving cluster and n is the number of patches in the source cluster. The total link cost is the sum of the link costs for each link computed for the environment. The model was refined using the ∞-norm criterion on the links. The graph in figure 5 shows the link cost for five tessellations ranging from 128 patches to 32768 initial patches, as well as the function $y = s\log s$. The costs closely match the function.

For the largest tessellation a conventional HR algorithm would need to create and store over one billion initial links before it could

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7: Solutions at different accuracies. With clustering (a-d), without clustering (e-h).

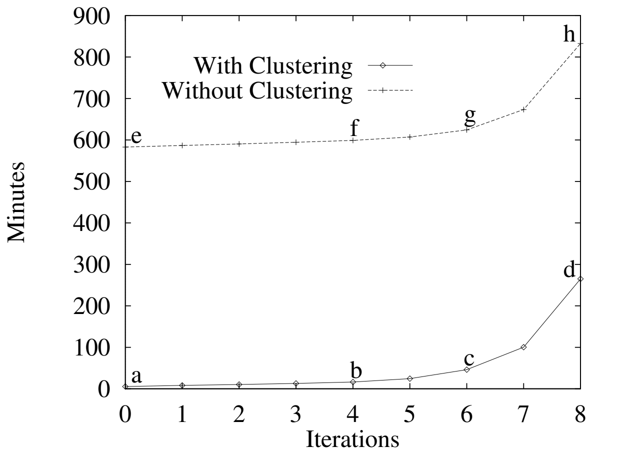

Figure 6: *Time for solutions of increasing accuracy, both with and without clustering.*

compute a solution. With clustering the links cost only about 0.2% of this, a reduction in both time and space of approximately 500.

6.2 Refinement Complexity

We now show how a solution refines over time both with and without clustering. We start with an environment of 4170 initial surfaces, illuminated by three directional lights and 8 emitting cubes, and refine the solution from the initial linking stage through eight iterations of successively smaller error tolerances. In this example we used a 1-norm bound on the error for each interaction. A graph of the times for the global solution as the accuracy increases appears in figure 6. These solutions were computed on an HP 755 with 384 megabytes of memory to prevent swapping while computing the solution without clustering. Images of the flat shaded patches are shown for several different levels of accuracy in figure 7. The long thin polygons in the model do not create ideal clusters because of the large variation in position. Neither algorithm grows linearly with the number of iterations, the algorithms are simply linear with the number of resulting patches.

With clustering, HR becomes more of a progressive algorithm since initial results can be seen relatively quickly. Without clustering it took 583 minutes for the first solution to appear. Clustering reduced that time to 5.35 minutes, a speed up of over one hundred times. Both solutions refine at about the same rate once they are linked. After eight iterations, the clustering solution resulted in 114554 patches. Without clustering there were 114410 patches. These numbers are very close because subdivision can only occur by refining a patch link, and this is done the same way in both algorithms. Bounds on the various kernels were estimated using 81 samples. Visibility was checked with 81 rays.

Clustering amortizes the cost of initial linking over many refinements and except for simple environments, the amortization continues well beyond the required levels of accuracy. This amortization has some cost associated with it, but it is compensated for by reduced link maintenance; that is, there is no need to maintain the $O(s^2)$ initial links through all the early iterations where high accuracy is unnecessary

6.3 Importance

We also tested our algorithm using importance weighted refinement. We used the same environment shown in figure 7, with a view of the group of chairs near the stairs. The upper image shown in figure 8 was created with the view-dependent local pass described earlier in the paper. The global solution used for the local pass is shown below the image. The global pass with clustering took about 3 minutes for the initial solution and 53 minutes for the entire global solution, creating 27318 patches. Without clustering, the initial linking took 728 minutes and the entire global pass took 790 minutes. (Estimates of form factor and visibility error were computed with more samples to make the local pass more effective, causing the initial linking to take longer than it did in the previous example.)

Figure 8: *Image resulting from local pass (above) and solution used for the local pass (below)*

7 Conclusions and Future Work

Clustering is an effective technique for accelerating HR. By using two different error bounds on the energy transfer between collections of surfaces, two types of links between cluster were described. Using both of these approaches in an HR algorithm reduces the asymptotic complexity from $O(s^2+p)$ to $O(s \log s + p)$ in addition to making HR a more progressive algorithm. In environments of moderate complexity we obtained speedups of two orders of magnitude. The approach presented here enables HR to work effectively even when there are many mutually visible initial surfaces.

The hierarchy of clusters presented in this paper were fairly straightforward modifications to traditional bounding volume hierarchies. Our current research shows that it is beneficial to build 5D hierarchies in the spirit of [1], to bound direction of the patch normals as well as position. This can be used to obtain tighter bounds on the transfers since patch orientations are constrained. In the implementation described here, two clusters with even one pair of patches facing each other may be linked with a relatively expensive α-link. A 5D hierarchy seems to reduce this problem.

Although a lower bound of zero on energy transfers is always valid, tighter lower bounds between two clusters would improve overall performance of the algorithm. Unlike upper bounds, this requires computing guaranteed bounds on visibility, a potentially expensive operation. Currently, our implementation uses a sampling approach to determine the upper bound on the transfer between two patches as well as the transfer between patches and clusters. Although we have not noticed any serious problems as a result of this approximation, analytic bounds on these transfers would produce guaranteed bounds. Guaranteed bounds on transfers make it possible to examine the global error in the solution. As mentioned in [12] bounds on transfers between surfaces do not immediately provide a bound on the total error in the solution.

Acknowledgments

We would like to thank Dani Lischinski for discussions on complexity and for helpful comments on the paper. Also thanks go to Greg Spencer for his software for handling high dynamic range images. This work was supported by the NSF grant "Interactive Computer Graphics Input and Display Techniques" (CCR-8617880), and by the NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219). The authors gratefully acknowledge the generous equipment grant from Hewlett-Packard Corporation.

REFERENCES

[1] ARVO, J., AND KIRK, D. Fast ray tracing by ray classification. *Computer Graphics 21*, 4 (July 1987), 55–64.

[2] CHEN, S. E., RUSHMEIER, H. E., MILLER, G., AND TURNER, D. A progressive multi-pass method for global illumination. In *Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 28–August 2, 1991)* (July 1991), vol. 25, ACM, pp. 165–174.

[3] COHEN, M. F., GREENBERG, D. P., IMMEL, D. S., AND BROCK, P. J. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications 6*, 2 (March 1986), 26–35.

[4] GOLDSMITH, J., AND SALMON, J. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications 7*, 5 (May 1987), 14–20.

[5] GORAL, C. M., TORRANCE, K. E., GREENBERG, D. P., AND BATTAILE, B. Modeling the interaction of light between diffuse surfaces. *Computer Graphics 18*, 3 (July 1984), 213–222.

[6] GREENGARD, L. *The Rapid Evaluation of Potential Fields in Particle Systems*. MIT Press, Cambridge, Massachusetts, 1988.

[7] HANRAHAN, P., SALZMAN, D., AND AUPPERLE, L. A rapid hierarchical radiosity algorithm. *Computer Graphics 25*, 4 (July 1991), 197–206.

[8] KOK, A. J. Grouping of patches in progressive radiosity. In *Proceedings of the Fourth Eurographics Workshop on Rendering (Paris, France, June 14–16, 1993)* (June 1993), pp. 221–231.

[9] LISCHINSKI, D., TAMPIERI, F., AND GREENBERG, D. P. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics Proceedings* (1993), Annual Conference Series, ACM SIGGRAPH, pp. 199–208.

[10] REICHERT, M. C. A two-pass radiosity method driven by lights and viewer position. Master's thesis, Program of Computer Graphics, Cornell University, Ithaca, New York, January 1992.

[11] RUSHMEIER, H. E., PATTERSON, C., AND VEERASAMY, A. Geometric simplification for indirect illumination calculations. *Graphics Interface '93* (May 1993), 227–236.

[12] SMITS, B., ARVO, J., AND SALESIN, D. An importance-driven radiosity algorithm. *Computer Graphics 26*, 4 (July 1992), 273–282.

[13] TELLER, S., AND HANRAHAN, P. Global visibility for illumination computations. In *Computer Graphics Proceedings* (1993), Annual Conference Series, ACM SIGGRAPH, pp. 239–246.