An Efficient Progressive Refinement Strategy for Hierarchical Radiosity

Nicolas Holzschuch, François Sillion, George Drettakis

iMAGIS / IMAG *

A detailed study of the performance of hierarchical radiosity is presented, which confirms that visibility computation is the most expensive operation. Based on the analysis of the algorithm's behavior, two improvements are suggested. Lazy evaluation of the top-level links suppresses most of the initial linking cost, and is consistent with a progressive refinement strategy. In addition, the reduction of the number of links for mutually visible areas is made possible by the use of an improved subdivision criterion. Results show that initial linking can be avoided and the number of links significantly reduced without noticeable image degradation, making useful images available more quickly.

1 Introduction

The radiosity method for the simulation of energy exchanges has been used to produce some of the most realistic synthetic images to date. In particular, its ability to render global illumination effects makes it the technique of choice for simulating the illumination of indoor spaces. Since it is based on the subdivision of surfaces using a mesh and on the calculation of the energy transfers between mesh elements pairs, the basic radiosity method is inherently a costly algorithm, requiring a quadratic number of form factors to be computed.

Recent research has focused on reducing the complexity of the radiosity simulation process. Progressive refinement has been proposed as a possible avenue [1], whereby form factors are only computed when needed to evaluate the energy transfers from a given surface, and surfaces are processed in order of importance with respect to the overall balance of energy. The most significant advance in recent years was probably the introduction of hierarchical algorithms, which attempt to establish energy transfers between mesh elements of varying size, thus reducing the subdivision of surfaces and the total number of form factors computed [4, 5].

Since hierarchical algorithms proceed in a top-down manner, by limiting the subdivision of input surfaces to what is necessary, they first have to establish a number of top-level links between input surfaces in an "initial linking" stage. This

* iMAGIS is a joint research project of CNRS/INRIA/UJF/INPG. Postal address: B.P. 53, F-38041 Grenoble Cedex 9, France. E-mail: `Nicolas.Holzschuch@imag.fr`.

results in a quadratic cost with respect to the number of input surfaces, which seriously impairs the ability of hierarchical radiosity systems to deal with environments of even moderate complexity. Thus a reformulation of the algorithm is necessary in order to be able to simulate meaningful architectural spaces of medium complexity (several thousands of input surfaces). To this end the questions that must be addressed are: What energy transfers are significant? When must they be computed? How can their accuracy be controlled?

The goal of the research presented here is to extend the hierarchical algorithm into a more progressive algorithm, by identifying the calculation components that can be delayed or removed altogether, and establishing improved refinement criteria to avoid unnecessary subdivision. Careful analysis of the performance of the hierarchical algorithm on a variety of scenes shows that the visibility calculations dominate the overall compute time.

Two main avenues are explored to reduce the cost of visibility calculations: First, the cost of initial linking is reduced by delaying the creation of the links between top-level surfaces until they are potentially significant. In a BF refinement scheme this means for instance that no link is established between dark surfaces. In addition, a form factor between surfaces can be so small that it is not worth performing the visibility calculation.

Second, experimental studies show that subdivision is often too high. This is a consequence of the assumption that the error on the form factor is of magnitude comparable to the form factor itself. In situations of full visibility between surfaces, relatively large form factors can be computed with good accuracy.

2 Motivation

To study the behaviour of the hierarchical algorithm, we ran the original hierarchical program [5] on a set of five different interior environments, varying from scenes with simple to moderate complexity (from 140 to 2355 input polygons). The scenes we used were built in different research efforts and have markedly different overall geometric properties. By using these different scenes, we hope to identify general properties of interior environments. We thus hope to avoid, or at least moderate, the pitfall of unjustified generalisation that oftens results from the use of a single scene or a class of scenes with similar properties to characterise algorithm behaviour. The scenes are: *"Full office"*, which is the original scene used in [5], *"Dining room"*, which is "Scene 7" of the standard set of scenes distributed for this workshop, *"East room"* and *"West room"*, which are scenes containing moderately complex desk and chair models, and finally *"Hevea"*, a model of a hevea tree in a room. Table 1 gives a short description and the number of polygons n for each scene. Please refer to colour section (Figs. 1, 3, 5 and 9-12) for a computed view of the test scenes.

2.1 Visibility

The first important observation we make from running the algorithm on these test scenes is the quantification of the cost of visibility calculations in the hier-

Table 1. Description of the five test scenes.

Name	n	Description
Full Office	170	The original office scene
Dining room	402	A table and four chairs
East room	1006	Two desks, six chairs
West room	1647	Four desks, ten chairs
Hevea	2355	An hevea tree with three light sources

archical algorithm. As postulated in previous work [9, 6], visibility computation represents a significant proportion of the overall computation time. In the graph shown in Fig. 1, the percentages of the computation time spent in each of the five main components of the hierarchical algorithm are presented. “Push-pull” signifies the time spent traversing the quadtree structure associated with each polygon, “Visibility” is the time spent performing visibility calculations, both for the initial linking step and subsequent refinement, “Form Factors” is the time spent performing the actual unoccluded form factor approximation calculation, “Refine” is the time spent updating the quadtree for refinement, and finally “Gather” shows the cost of transferring energy across the links created between quadtree nodes (interior or leaves) [5]. The graph in Fig. 1 shows that

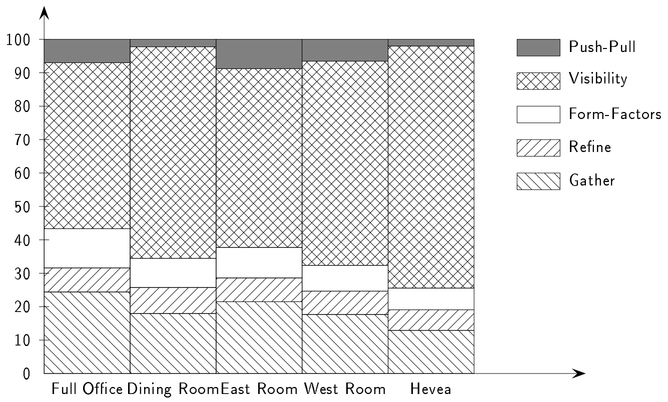


Fig. 1. Relative time spent in each procedure.

visibility calculations dominate the computation in the hierarchical algorithm[2].

[2] In its current version, the program uses a fixed number of rays to determine the mutual visibility between two polygons. The cost of visibility computation is thus roughly proportional to the number of rays used. In the statistics shown here, 16 rays were used, a relatively small number. Using more rays would increase the percentage of time devoted to visibility tests.

Of course this is relative to the algorithm used. A better approach, e.g. with a pre-processing step, as in Teller et al. [9] could probably reduce the relative importance of visibility.

2.2 Initial Linking

The second important observation concerns the actual cost of the initial linking step. As mentioned in the introduction, this cost is at least quadratic in the number of polygons, since each pair of input polygons has to be tested to determine if a link should be established. Since this step is performed before any transfer has commenced, it is a purely geometric visibility test, in this instance implemented by ray-casting. The cost of this test for each polygon pair can vary significantly, depending on the nature of the scene and the type of ray-casting acceleration structure used. In all the examples described below, a BSP tree is used to accelerate the ray-casting process.

Table 2. Total computation time and cost of initial linking (in seconds).

Name	n	Total Time	Initial Linking
Full office	170	301	5.13
Dining room	402	4824	436
East room	1006	587	194
West room	1647	1017	476
Hevea	2355	4253	1597

Table 2 presents timing results for all test scenes. The total computation time is given for ten steps of the multigridding method described by Hanrahan et al [5].[3].

These statistics show that the cost of initial linking grows significantly with the number of polygons in the scene. The dependence on scene structure is also evident, since the growth in computation time between *East room* and *West room* is actually sublinear, while on the other hand the growth of the computation time between *West room* and *Hevea* displays greater than cubic growth in the number of input polygons. For all tests of more than a thousand polygons, it is clear that the cost of initial linking becomes overwhelming. Invoking this cost at the beginning of the illumination computation is particularly undesirable, since a useful image cannot be displayed before its completion. Finally, we note that recent improvements of the hierarchical radiosity method by Smits et al. [8] and Lischinski et al. [6] have allowed significant savings in refinement time, but still

[3] The k'th step of the multigridding method is typically implemented as the k'th "bounce" of light: the first step performs all direct illumination, the second step all secondary illumination, the third all tertiary illumination etc.

rely on the original initial linking stage. Thus initial linking tends to become the most expensive step of the algorithm[4].

Another interesting observation can be made concerning the number of top-level links (links between input polygons) for which the product BF never becomes greater than the refinement threshold $\varepsilon_{\mathrm{refine}}$ over the course of the ten refinement steps[5]. Figure 2 shows the percentage of such links during the first ten iterations. A remarkably high percentage of these links never becomes a candidate for refinement: after 10 steps, between 65% and 95% of the links have not been refined. A significant number of those links probably have very little impact on the radiosity solution.

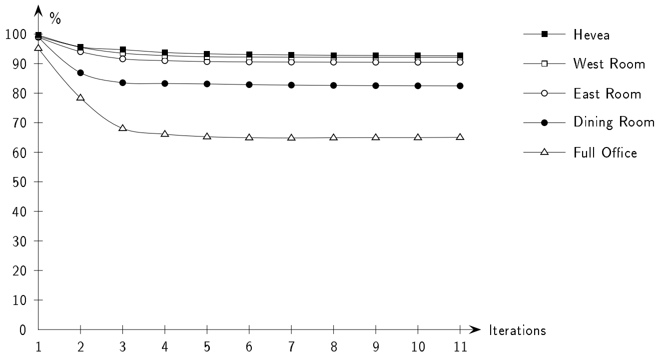


Fig. 2. Percentage of links for which BF does not exceed $\varepsilon_{\mathrm{refine}}$.

What can be concluded from the above discussion? First, if the initial linking step can be eliminated at the beginning of the computation, a useful solution becomes available much more quickly, enhancing the utility of the the hierarchical method. Second, if the top-level links are only computed when they contribute significantly to the solution, there is the potential for large computational savings from eliminating a large number of visibility tests.

2.3 Unnecessary Refinement

The third important observation made when using the hierarchical algorithm is that unnecessary subdivision is incurred, especially for areas which do not include shadow boundaries. This observation is more difficult to quantify than the previous two. To demonstrate the problem we present an image of the *Dining room* scene, and the corresponding mesh (see colour section, Fig. 1 and 2). The simulation parameters were $\varepsilon_{\mathrm{refine}} = 0.5$ and $MinArea = 0.001$.

As can be seen in Fig. 2 in the colour section, the subdivision obtained with these parameters is such that acceptable representation of the shadows

[4] For example Lischinski et al. report a refinement time of 16 minutes for an initial linking time of 2 hours and 16 minutes.

[5] This is the ε used in the original formulation.

is achieved in the penumbral areas caused by the table and chairs. However, the subdivision on the walls is far higher than necessary: the illumination over the wall varies very smoothly and could thus be represented with a much coarser mesh. In previous work it was already noted that radiance functions in regions of full illumination can be accurately represented using a simple mesh based on the structure of illumination [2].

If this unnecessary subdivision is avoided, significant gains can be achieved since the total number of links will be reduced, saving memory, and since an attendant reduction of visibility tests will result, saving computation time.

3 Lazy Evaluation of the Top-level Interactions

In this section a modification of the hierarchical algorithm is proposed, which defers the creation of links between top-level surfaces until such a link is deemed necessary. The basic idea is to avoid performing any computation that does not have a sizable impact on the final solution, in order to concentrate on the most important energy transfers. Thus it is similar to the rationale behind progressive refinement algorithms. At the same time it remains consistent with the hierarchical refinement paradigm, whereby computation is only performed to the extent required by the desired accuracy.

To accomplish this, a criterion must be defined to decide whether a pair of surfaces should be linked. In our implementation we use a specific threshold $\varepsilon_{\mathrm{link}}$on the product BF. Top-level links are then created *lazily*, only once the linking criterion is met during the course of the simulation.

3.1 Description of the Algorithm

In the original hierarchical radiosity algorithm, two polygons are either mutually invisible, and thus do not interact, or at least partially visible from each other and thus exchange energy. We introduce a second qualification, whereby a pair of polygons is either *classified* or *unclassified*. A pair will be marked *classified* when some information is available regarding its interaction. Initially, all pairs of polygons are marked as unclassified.

At each iteration, all unclassified pairs of polygon are considered: First their radiosity is compared to $\varepsilon_{\mathrm{link}}$. If they are bright enough, we check (in constant time) if they are at least partially facing each other. If not, the pair is marked as classified and no link is created. If they are facing, we compute an approximation of their form factor, without a visibility test. If the product of the form factors and the radiosity is still larger than $\varepsilon_{\mathrm{link}}$, we mark the pair of polygons as classified, and compute the visibility of the polygons. If they are visible, a link is created using the form factors and visibility already computed. Thus a pair of polygons can become classified either when a link is created, or when the two polygons are determined to be invisible. Figure 3 shows a pseudo-code listing of both the Initial Linking phase and the Main Loop in the original algorithm [5] and Fig. 4 gives the equivalent listing in our algorithm.

```
Initial Linking
for each pair of polygons (p, q)
    if p and q are facing each other
        if p and q are at least partially visible from each other
            link p and q
Main Loop
for each polygon p
    foreach link l leaving p
        if B × F > ε_refine
            refine l
    foreach link l leaving p
        gather l
```

Fig. 3. The Original Algorithm

```
Initial Linking
for each pair of polygons (p, q)
    record it as unclassified
Main Loop
    for each unclassified pair of polygons (p, q)
        if p and q are facing each other
            if B_p > ε_link or B_q > ε_link
                compute the unoccluded FF
                if B × F > ε_link
                    link p and q
                    record (p, q) as classified
        else record (p, q) as classified
    for each polygon p
        for each link l leaving p
            if B × F > ε_refine
                refine l
        for each link l leaving p
            gather l
```

Fig. 4. Pseudo-code listing for our algorithm

The threshold $\varepsilon_{\text{link}}$used to establish top-levels interactions is not the same as the threshold used for BF refinement, $\varepsilon_{\text{refine}}$. The main potential source of error in our algorithm is an incomplete balance of energy. Since energy is transfered across links, any polygon for which some top-level links have not been created is retaining some energy, which is not propagated to the environment.

When recursive refinement is terminated because the product BF becomes smaller than $\varepsilon_{\text{refine}}$, a link is always established, which carries some fraction of this energy (the form factor estimate used in the comparison against $\varepsilon_{\text{refine}}$ is an upper bound of the form factor). On the other hand, when two top-level surfaces are not linked because the product BF is smaller than $\varepsilon_{\text{link}}$, all the corresponding energy is "lost". It is thus desirable to select a threshold such that $\varepsilon_{\text{link}} < \varepsilon_{\text{refine}}$.

In the examples shown below we used $\varepsilon_{\text{link}} = \varepsilon_{\text{refine}}/5$.

The classified/unclassified status of all pairs of input surfaces requires the storage of $\frac{n(n-1)}{2}$ bits of information. We are currently working on compression techniques to further reduce this cost[6].

3.2 Energy Balance

Since radiosity is mainly used for its ability to model light interreflection, it is important to maintain energy consistency when modifying the algorithm. An issue raised by the lazy linking strategy is that "missing" links, those that have not been created because they were deemed insignificant, do not participate in energy transfers. Thus each gather step only propagates some of the energy radiated by surfaces.

If the corresponding energy is simply ignored, the main result is that the overall level of illumination is reduced. However a more disturbing effect can result for surfaces that have very few (or none) of their links actually established: these surfaces will appear very dark because they will receive energy only from the few surfaces that are linked with them.

The solution we advocate in closed scenes is the use of an *ambient term* similar to the one proposed for progressive refinement radiosity [1]. However the distribution of this ambient term to surfaces must be based on the estimated fraction of their interaction with the world that is missing from the current set of links. The sum of the form factors associated with all links leaving a surface gives an estimate of the fraction of this surface's interactions that is actually represented. Thus, in a closed scene, its complement to one represents the missing link. Using this estimate to weight the distribution of the ambient energy, the underestimation of radiosities can be partially corrected: surfaces that have no links will use the entire ambient term, whereas surfaces with many links will be only marginally affected.

However, since form factors are based on approximate formulas, the sum of all form-factors can differ from one, even for a normally linked surface. This comes from our BF refinement strategy: we accept that the form-factor on a link between two dark surfaces be over-estimated, or under-estimated. This may results in energy loss, or creation. If the error we introduced by not linking some surfaces is of the same order – or smaller – than the one due to our lack of precision on the form-factor estimation, using the ambient term will not suffice to correct the energy inbalance.

To quantify the influence of those errors on the overall balance of energy, we compute the following estimate of the incorrect energy:

$$\mathcal{E}_{E_T} = \sum_p |1 - F_p| \, B_p A_p \tag{1}$$

[6] The storage cost for the classified bit represents 62 kb for a thousand surfaces, 25 Mb for twenty thousand surfaces.

where A_p is the area of polygon p, B_p its radiosity and F_p the sum of the form factors on all links leaving p. This can be compared to the total energy present in the scene:

$$E_T = \sum_p B_p A_p \tag{2}$$

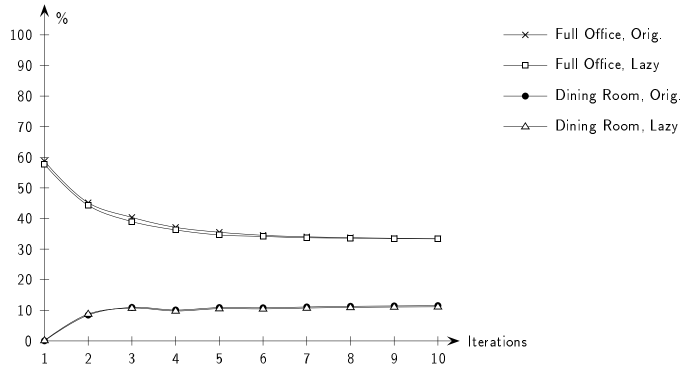


Fig. 5. Incorrect Energy $\mathcal{E}_{E_T}/E_T$

Figure 5 shows a plot of the ratio $\mathcal{E}_{E_T}/E_T$ for the *Dining Room* scene and the *Full Office*, for both the original algorithm and our algorithm. Note that the error can be significant, but is mainly due to the original algorithm.

4 Reducing the Number of Links

The refinement of a link is based on the estimation of an associated error bound. Various criteria have been used that correspond to different error metrics, including the error in the form factor [4], the error in the radiosity transfer [5], and the impact of the error in the radiosity transfer on the final image [8].

All these criteria implicitly assume that the error in the form factor estimate is equivalent to the magnitude of the form factor itself. While this is true for infinitesimal quantities, in many instances it is possible to compute a reasonable estimate of a relatively large form factor. In particular this is true in situations of full visibility between a pair of surfaces.

Consider two patches p and q. A bi-directionnal link between them carries two form factor estimates $F_{p,q}$ and $F_{q,p}$. If we refine the link by dividing p in smaller patches p_i of area A_i (e.g. in a quadtree), the definition of the form factor

$$F_{u,v} = \frac{1}{A_u} \int_{A_u} \int_{A_v} G(dA_u, dA_v)\, dA_u dA_v \tag{3}$$

where G is a geometric function, implies that the new form factors verify:

$$F_{p,q} = \frac{1}{A_p} \left(\sum_i A_i F_{p_i,q} \right) \tag{4}$$

$$F_{q,p} = \sum_i F_{q,p_i} \tag{5}$$

These relations only concern the exact values of the form factors. However they can be used to compare the new form factor estimates with the old ones, and determine *a posteriori* wether refinement was actually required. If the sum of the F_{q,p_i} is close to the old $F_{q,p}$, and they are not very different from one another, little precision was gained by refining p. Moreover, if $F_{p,q}$ is close to the average of the $F_{p_i,q}$, and the $F_{p_i,q}$ are not too different from one another, then the refinement process did not introduce any additional information. In this case we force p and q to interact at the current level, since the current estimates of form factors are accurate enough.

In our implementation we only allow reduction of links in situations of full visibility between surfaces. We compute the relative variation of the children form factors, which we test against a new threshold $\varepsilon_{\text{reduce}}$. We also check that the difference between the old form factor $F_{p,q}$ and the sum of the $F_{p_i,q}$, and the difference between $F_{q,p}$ and the average of the F_{q,p_i} are both smaller than $\varepsilon_{\text{reduce}}$.

If we note $F_{u,v}$ our current estimation of the form-factor between two patches u and v, and assuming we want to refine a patch p in p_i, we note:

$$\begin{array}{ll}
F_{p,q}^{\min} = \min_i(F_{p_i,q}) & F_{q,p}^{\min} = \min_i(F_{q,p_i}) \\
F_{p,q}^{\max} = \max_i(F_{p_i,q}) & F_{q,p}^{\max} = \max_i(F_{q,p_i}) \\
F'_{p,q} = \frac{1}{A_p}\left(\sum_i A_i F_{p_i,q}\right) & F'_{q,p} = \sum_i F_{q,p_i}
\end{array}$$

and we refine p if any of the following is true:

$$\begin{array}{ll}
\frac{F_{p,q}^{\max} - F_{p,q}^{\min}}{F_{p,q}^{\max}} > \varepsilon_{\text{reduce}} & \frac{F_{q,p}^{\max} - F_{q,p}^{\min}}{F_{q,p}^{\max}} > \varepsilon_{\text{reduce}} \\
\frac{\left|F'_{p,q} - F_{p,q}\right|}{F'_{p,q}} > \varepsilon_{\text{reduce}} & \frac{\left|F'_{q,p} - F_{q,p}\right|}{F'_{q,p}} > \varepsilon_{\text{reduce}}
\end{array}$$

The decision to cancel the subdivision of a link is based purely on geometrical properties, therefore it is permanent. The link is marked as "un-refinable" for the entire simulation.

The check whether a link is worth refining involves the computation of form factor estimates to and from all children of patch p. Thus the associated cost in time is similar to that of actually performing the subdivision. If a single level of refinement is avoided by this procedure, there will be little gain in computation time, but the reduction in the number of links will yield memory savings. But if link reduction happens "early enough", several levels of refinement can be avoided. In our test scenes, an implementation of this algorithm reduced significantly the number of quadtree nodes and links (see Fig. 6), with a slightly smaller reduction in computation time because of the cost of the extra form factor estimates (see Fig. 7).

5 Results

5.1 Lazy Linking

Figures 3 in coulour section shows the same scene as in Fig. 1, computed using the lazy linking strategy of Sect. 3. Note that it is visually indistinguishable from its original counterpart. Figure 4 plots the absolute value of the difference between these two images.

5.2 Reduction of the Number of Links

To measure the performance of the reduction criterion, we computed the ratio of the number of quadtree nodes (surface elements) obtained with this criterion, to the number of nodes obtained with the original algorithm. The graph in Fig. 6a plots this ratio against the number of iterations. Note that an overall reduction by nearly a factor of two is achieved for all scenes. Figure 6b shows a similar ratio for the number of links. This global reduction of the number of objects involved leads to a similar reduction of the memory needed by the algorithm, thus making it more practical for scenes with more polygons.

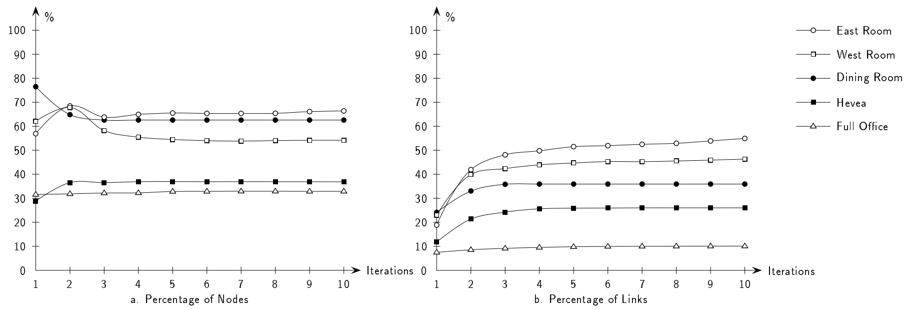


Fig. 6. Percentage of nodes and links left after reduction.

Figure 7 shows the ratio of the computation times using the improved criterion and the original algorithm. The reduction of the number of links has a dramatic impact on running times, with speedups of more than 50%.

Figure 5 and 6 in colour section shows the image obtained after link reduction. Note the variation in the mesh on the walls, and the similarity of the shaded image with the ones in Figs. 1 and 3. Figure 7 plots the absolute value of the difference between the image produced by the original algorithm and the image obtained after link reduction. Note that part of the differences are due to the lazy linking strategy of Sect. 3. So Figure 8 shows the difference between lazy linking and reduction of the number of links.

5.3 Overall Performance Gains

Timing results are presented in Table 3. As expected, a significant speedup is achieved, particularly for complex scenes. For all scenes, ten iterations with lazy

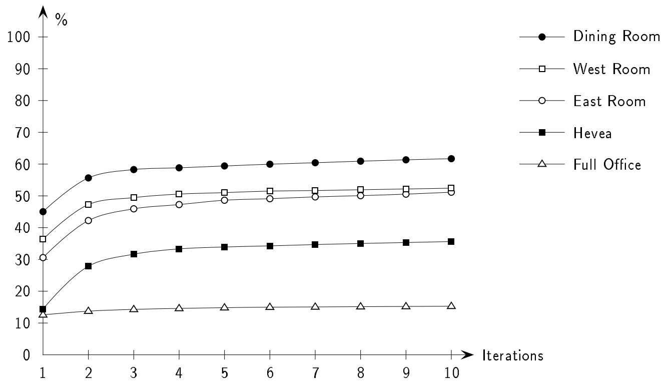


Fig. 7. Percentage of computation time using link reduction.

linking took less time to compute than the first iteration alone with the original algorithm. Finally, using lazy linking and reduction produces a useful image in a matter of minutes even for the most complex scenes in our set.

Table 3. Time needed for ten iterations (and time for producing the first image).

Name	n	Original Algorithm		with Lazy Linking...		and Reduction	
Full office	170	301 s	(242 s)	287 s	(234 s)	43 s	(30 s)
Dining room	402	4824 s	(4191 s)	4051 s	(3911 s)	657 s	(552 s)
East room	1006	587 s	(378 s)	377 s	(191 s)	193 s	(59 s)
West room	1647	1017 s	(752 s)	514 s	(277 s)	270 s	(101 s)
Hevea	2355	4253 s	(2331 s)	1526 s	(847 s)	543 s	(122 s)

6 Conclusions and Discussion

We have presented the results of an experimental study conducted on a variety of scenes, showing that visibility calculations represent the most expensive portion of the computation. Two improvements of the hierarchical algorithm were proposed. The first modification creates top-level links *lazily*, only when it is established that the proposed link will have a definite impact on the simulation. With this approach the hierarchical algorithm still remains quadratic in the number of input surfaces, but no work and very little storage is devoted to the initial linking phase. The resulting algorithm is more progressive in that it produces useful images very quickly. Note that the quadratic cost in the number of input surfaces can only be removed by clustering methods [7].

An improved subdivision criterion was introduced for situations of full visibility between surfaces, which allows a significant reduction of the number of links.

Future work will include the simplification of the hierarchical structure due to multiple sources and subsequent iterations. A surface that has been greatly refined because it receives a shadow from a given light source can be fully illuminated by a second source, and the shadow become washed in light.

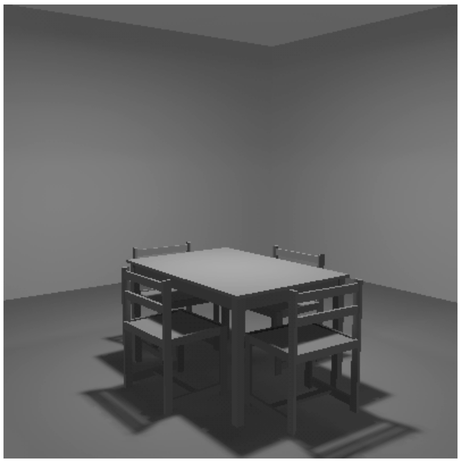
Better error bounds, both on form factor magnitude and global energy transfers, should allow even greater reduction of the number of links. Accurate visibility algorithms can be used to this end, by providing exact visibility information between pairs of surfaces.

7 Acknowledgments

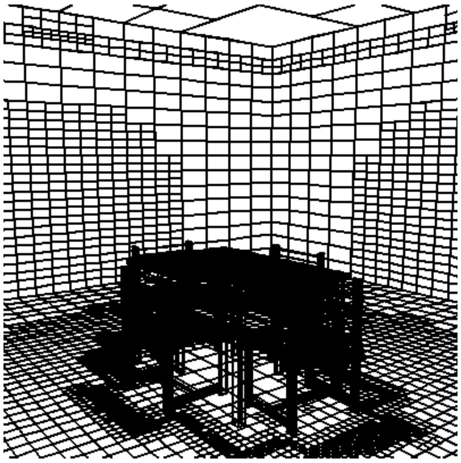
George Drettakis is a post-doc hosted by INRIA and supported by an ERCIM fellowship. The hierarchical radiosity software was built on top of the original program kindly provided by Pat Hanrahan.

References

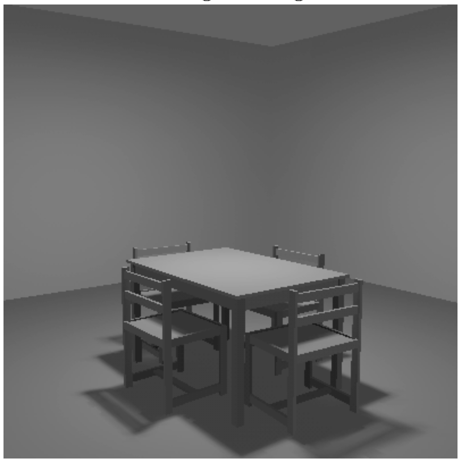
1. Cohen, M. F., Chen, S. E., Wallace, J. R., Greenberg, D. P.: A Progressive Refinement Approach to Fast Radiosity Image Generation. SIGGRAPH (1988) 75–84
2. Drettakis, G., Fiume, E.: Accurate and Consistent Reconstruction of Illumination Functions Using Structured Sampling. Computer Graphics Forum (Eurographics 1993 Conf. Issue) 273–284
3. Goral, C. M., Torrance, K. E., Greenberg, D. P., Bataille, B.: Modeling the Interaction of Light Between Diffuse Surfaces. SIGGRAPH (1984) 213–222
4. Hanrahan, P. M., Salzman, D.: A Rapid Hierarchical Radiosity Algorithm for Unoccluded Environments. *Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, June 1990.
5. Hanrahan, P. M., Salzman, D., Auperle, L.: A Rapid Hierarchical Radiosity Algorithm. SIGGRAPH (1991) 197–206
6. Lischinski, D., Tampieri, F., Greenberg, D. P.: Combining Hierarchical Radiosity and Discontinuity Meshing. SIGGRAPH (1993)
7. Sillion, F.: Clustering and Volume Scattering for Hierarchical Radiosity calculations. *Fifth Eurographics Workshop on Rendering*, Darmstadt, June 1994 (in these proceedings).
8. Smits, B. E., Arvo, J. R., Salesin, D. H.: An Importance-Driven Radiosity Algorithm. SIGGRAPH (1992) 273–282
9. Teller, S. J., Hanrahan, P. M.: Global Visibility Algorithm for Illumination Computations. SIGGRAPH (1993) 239–246



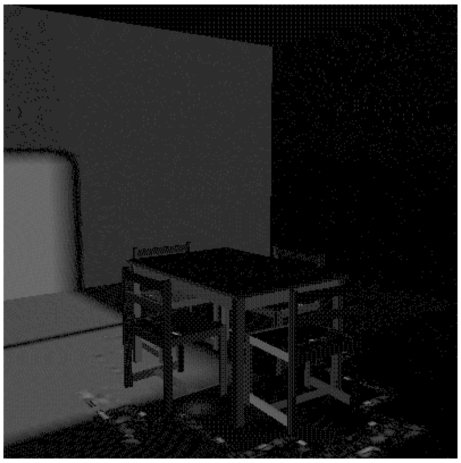
1. The Original Algorithm



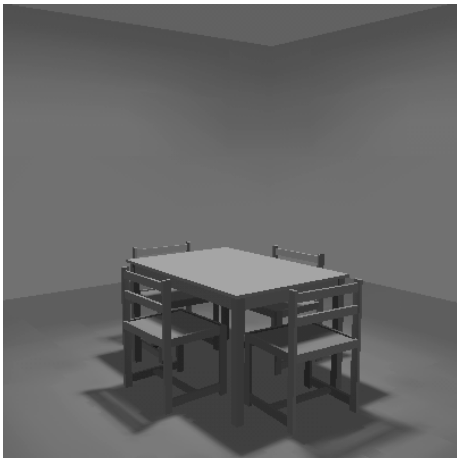
2. The Grid Produced



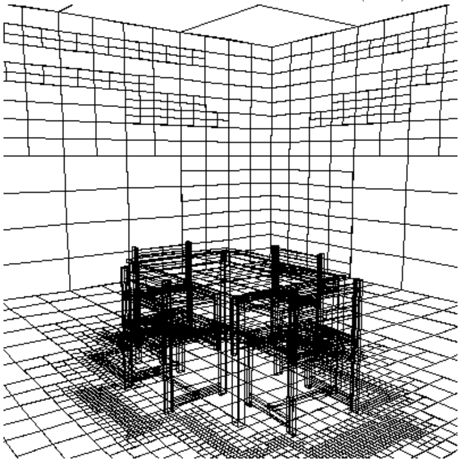
3. With Lazy Linking



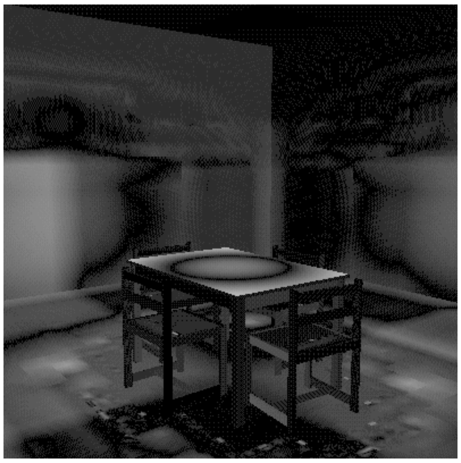
4. Diff. Between 1. and 3. ($\times 8$)

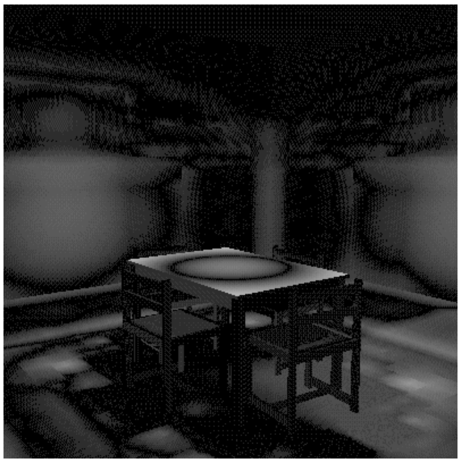


5. With Link Reduction



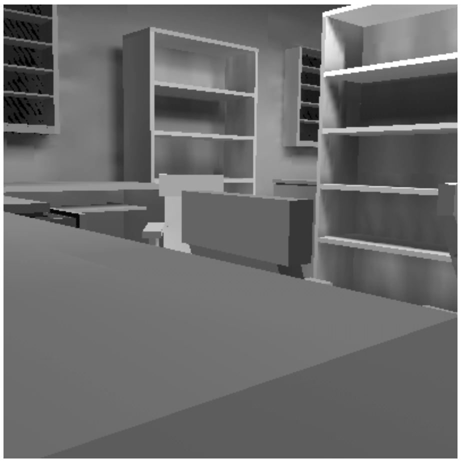
6. The Grid Produced


7. Diff. Between 1. and 5. ($\times 8$)


8. Diff. Between 3. and 5. ($\times 8$)


9. Full Office


10. East Room


11. West Room


12. Hevea

This article was processed using the LaTeX macro package with LMAMULT style