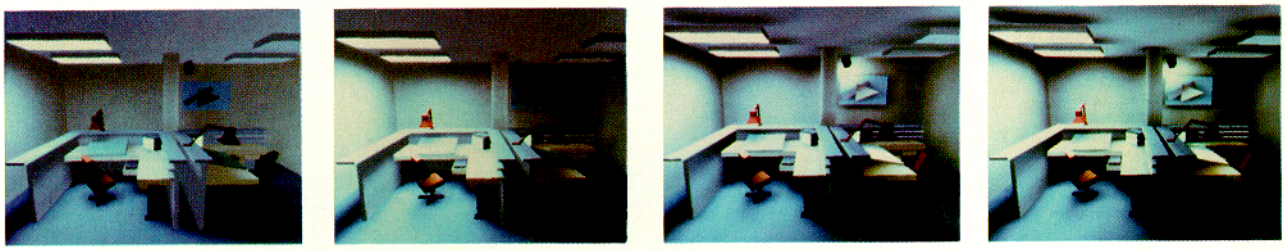
A Progressive Refinement Approach to Fast Radiosity Image Generation

Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, Donald P. Greenberg

Program of Computer Graphics, Cornell University



Abstract

A reformulated radiosity algorithm is presented that produces initial images in time linear to the number of patches. The enormous memory costs of the radiosity algorithm are also eliminated by computing form-factors on-the-fly. The technique is based on the approach of rendering by progressive refinement. The algorithm provides a useful solution almost immediately which progresses gracefully and continuously to the complete radiosity solution. In this way the competing demands of realism and interactivity are accommodated. The technique brings the use of radiosity for interactive rendering within reach and has implications for the use and development of current and future graphics workstations.

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

General Terms: Algorithms

Additional Key Words and Phrases: radiosity, progressive refinement, backward ray tracing, z-buffer, global illumination, adaptive subdivision.

 ACM-0-89791-275-6/88/008/0075 $00.75

1 Introduction

Two goals have largely shaped the field of image synthesis since its inception: visual realism and interactivity. The desire for realism has motivated the development of global illumination algorithms such as ray tracing [19], [5], [12] and radiosity [7], [13], [3], with often impressive results. However, the need for interactive manipulation of objects for geometric modeling and other computer aided design areas has generated another path of evolution. This path, dominated by speed, led from the work of early researchers [18], [8], [14] and others, to the development of current engineering workstations capable of drawing thousands of shaded polygons a second [16], [6]. In order to achieve this performance, much of what is central to the goal of realism has had to be sacrificed, including the effects of shadows and global illumination. On the other hand, algorithms like ray-tracing and radiosity are too expensive on current machines to be used as the basis of interactive rendering.

One approach to accommodating the competing demands of interactivity and image quality is offered by the method of rendering by adaptive refinement [2]. In this approach rendering begins with a simple, quickly rendered version of the image, and progresses through a sequence of increasing realism, until a change in the scene or view requires that the process start again. The aim is to provide the highest quality image possible within the time constraints imposed by the user's manipulation of the scene. It is crucial to this approach that the early images be of usable quality at interactive speeds and that the progression to greater realism be *graceful*, that is, automatic, continuous, and not distracting to the user. In the words of Bergman, what is needed is a *golden thread*, a single rendering operation that, with repeated application, will continually refine the quality of an image.

This paper presents a reformulation of the radiosity algorithm that provides such a *thread*. The radiosity approach is a particularly attractive basis for a progressive approach for two reasons. First, the process correctly simulates the global illumination of diffuse environments. Second, it provides a view-independent

solution of the diffuse component of reflection. Thus the refinement process may continue uninterrupted as the user views the scene from different directions. Unfortunately, the conventional radiosity algorithm provides no usable results until after the solution is complete, a computation of order n^2, (where n is the number of discrete surface patches). The original algorithm has the additional disadvantage of using $O(n^2)$ storage.

In the revised radiosity algorithm presented here, an initial approximation of the global diffuse illumination provides a starting point for refinement. A reorganization of the iterative solution of the radiosity equations allows the illumination of all surfaces in the environment to be updated at each step and ensures that the correct solution is approached early in the process. In addition to providing a basis for graceful image refinement, the new algorithm requires only $O(n)$ storage.

2 The Cost of Realism for the Conventional Radiosity Algorithm

The radiosity algorithm is a method for evaluating the intensity or radiosity at discrete points and surface areas in an environment. The relationship between the radiosity of a given discrete surface area, or patch, and the radiosity of all other patches in the environment is given by:

$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^{n} B_j F_{ji} A_j \qquad (1)$$

where
B_i = radiosity of patch i (energy/unit area/unit time),
E_i = emission of patch i (energy/unit area/unit time),
A_i = area of patch i, A_j = area of patch j,
F_{ji} = form-factor from j to i (fraction of energy leaving patch j which arrives at patch i),
ρ_i = reflectivity of patch i, and
n = number of discrete patches.

Using the reciprocity relationship for form-factors [15],

$$F_{ij} A_i = F_{ji} A_j \qquad (2)$$

and dividing through by Ai, the more familiar radiosity equation is obtained:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij} \qquad (3)$$

or in matrix form:

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \cdot \\ \cdot \\ E_n \end{bmatrix} \qquad (4)$$

The computation involved in the conventional hemi-cube radiosity algorithm is divided into three major sections as follows:

1. Computing the form-factors (F_{ij}). This requires determining the patches visible to each patch over the entire hemisphere of directions above the patch. For each patch, all the other patches of the environment are projected onto the five faces of a *hemi-cube* placed over the patch and a z-buffer hidden-surface operation is performed for each face [3]. Using standard scan conversion and hidden surface routines, the cost of each hemi-cube is proportional to the number of discrete patches as well as the resolution of the hemi-cube. This results in an $O(n^2)$ computation for the whole environment.
2. Solving the radiosity matrix equation (4) using the Gauss-Siedel method. Due to the strict diagonal dominance of the matrix, the solution converges in a few iterations and its cost is thus proportional to square of the number of patches [10]. The solution is performed for each color band. Since the form-factors are dependent on geometry only, this does not have a significant impact on the cost of the radiosity algorithm.
3. Displaying the results. This involves selecting viewing parameters, determining hidden surfaces, and interpolating the radiosity values. Current workstations are capable of rapidly displaying high resolution radiosity images from any vantage point through the use of Gouraud shading and z-buffer hardware.

The overwhelming cost of the radiosity method lies in the computation of the form-factors. To reduce this cost, the form-factors are calculated once and stored for repeated use during the iterative matrix solution. The total number of form-factors to be stored is potentially the number of patches squared, although the matrix of coefficients is normally quite sparse since many patches cannot *see* each other. Even so, the n by n matrix of coefficients will quickly exceed a reasonable storage size. For example, assuming a matrix that is 90 percent sparse and four bytes of memory per form-factor, an environment of 50,000 patches will require a gigabyte of storage.

For rendering by progressive refinement, an important criterion is the time required to achieve a useful as opposed to complete solution. In the conventional radiosity algorithm, all the form-factors for the entire environment are pre-calculated before the solution begins at a cost of $O(n^2)$. Furthermore, using the Gauss-Siedel solution for the system of radiosity equations, an estimate of the radiosity of all patches is not available until after the first complete iteration cycle. This clearly cannot be implemented at interactive speeds and is not the graceful first step required for progressive refinement.

3 Progressive Refinement Methods for the Radiosity Algorithm

The radiosity algorithm can be restructured to achieve the goals of progressive refinement. In the restructured algorithm, form-factors are calculated on-the-fly to eliminate the $O(n^2)$ storage and startup costs. Although the basic Gauss-Siedel approach still remains, the order of operations of the iteration cycle has been modified so that a good approximation of the final results can be displayed early in the solution process.

The restructured algorithm differs from the previous ones primarily in two aspects. First, the radiosity of all patches is updated simultaneously. Second, patches are processed in sorted order according to their energy contribution to the environment.

To further improve the quality of the images generated during the earliest stages of the algorithm, an estimate of global illumination is determined directly from the known geometric and reflective characteristics of the environment. This estimate is gradually replaced by more exact information as the solution progresses, providing a graceful and continuous convergence to a realistic image.

3.1 Simultaneous Update of Patch Radiosities: Shooting vs. Gathering Light

In the conventional radiosity algorithm, the Gauss-Siedel method is used to obtain the solution to the simultaneous equations(4). This iterative approach converges to the solution by solving the system of equations one row at a time. The evaluation of the $i'th$ row of the equations provides an estimate of the radiosity of patch i based on the current estimates of the radiosities of all other patch radiosities:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij} \tag{5}$$

In a sense, the light leaving patch i is determined by *gathering* in the light from the rest of the environment (figure 1).

A single term from the summation in (5) determines the contribution to the radiosity of patch i from patch j:

$$B_i \; due \; to \; B_j = \rho_i B_j F_{ij} \tag{6}$$

It is possible to reverse this process by determining the contribution made by patch i to the radiosity of all other patches. The reciprocity relationship (2) provides the basis for reversing this relationship. The contribution of the radiosity from patch i to the radiosity of patch j is:

$$B_j \; due \; to \; B_i = \rho_j B_i F_{ij} A_i / A_j \tag{7}$$

This is true for all patches j. Thus the total contribution to the environment from the radiosity of patch i is given by:

$$For \; all \; patches \; j : B_j \; due \; to \; Bi = \rho_j B_i F_{ij} A_i / A_j \tag{8}$$

It should be noted that while this equation adds radiosity to patches j, the form-factors used, F_{ij}, are still the form-factors calculated using the hemi-cube placed at patch i. Thus, each step of the solution now consists of performing a single hemi-cube over a patch and adding the contribution from the radiosity of that patch to the radiosities of all other patches, in effect, *shooting* light out from that patch into the environment.

During the course of the iterative solution this step may be repeated for patch i several times as the solution converges. Each time the estimate of the radiosity of patch i will be more accurate. However, the environment will already include the contribution of the previous estimate of B_i. Thus, only the difference, ΔB_i, between the previous and current estimates of B_i needs to be considered. ΔB_i represents the *unshot radiosity*.

The solution step may be restated as follows:

```
for each iteration, for each patch i:
    calculate the form-factors F_ij using a hemi-cube at
      patch i;
    for each patch j:
        ΔRad = ρ_j ΔB_i F_ij A_i / A_j;
        ΔB_j = ΔB_j + ΔRad; /* update change since last
          time patch j shot light */
        B_j = B_j + ΔRad; /* update total radiosity of
          patch j */
    ΔB_i = 0; /*reset unshot radiosity for patch i to zero*/
```

All radiosities, Bi and ΔB_i, are initialized to zero for all non-light sources and are set to the emission values for emitting patches.

The above step continues until the solution converges to within the desired tolerance. Each intermediate step simultaneously improves the solution for many patches, providing intermediate results which can be displayed as the algorithm proceeds.

This approach bears some relationship to backward ray-tracing solutions [1] which shot light out from light sources onto diffuse surfaces, but did not propagate the reflected light any further into the environment. A recursive extension of the Atherton-Weiler shadow algorithm was proposed and briefly described by Heckbert and Hanrahan [9] as a way of propagating light from light sources through the environment, but light reflected from diffuse surfaces was likewise not propagated further.

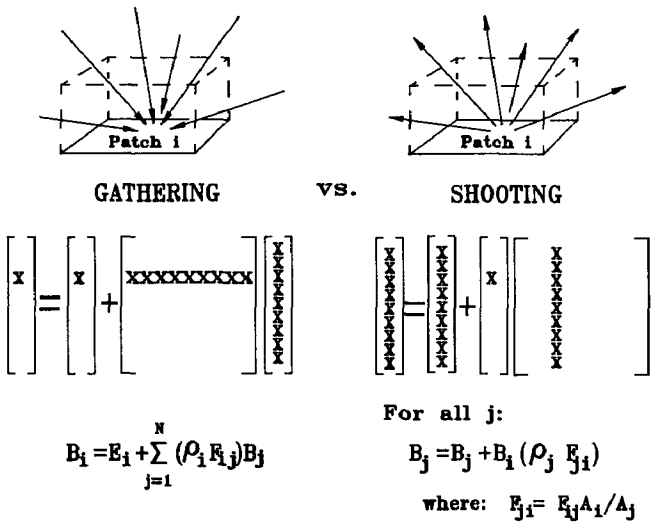


Figure 1: Gathering vs. Shooting

Gathering light through a hemi-cube allows one patch radiosity to be updated. In contrast, shooting light through a single hemi-cube allows the whole environment's radiosity values to be updated simultaneously.

3.2 Solving in Sorted Order

In addition to converging gracefully, it is desirable for the solution to improve in accuracy as quickly as possible.

The final radiosity B_j of a given patch j consists of the sum of the contributions from all other patches. The final value of this sum will be approached earliest in the process if the

largest contributions are added first. These will tend to come from those patches which radiate the most energy, i.e. have the largest product B_iA_i. Stated intuitively, those patches radiating the most light energy typically have the greatest effect on the illumination of the environment and should be treated first.

The algorithm is implemented by always shooting from the patch for which the difference, ΔB_iA_i, between the previous and the current estimates of unshot radiant energy is greatest. Most light sources are automatically processed first by this rule, since initially all other patches will have a radiosity of zero. Since lights are typically the most significant source of illumination for many patches, following the initial processing of light sources much of the environment will already be well illuminated. The next set of patches processed according to this rule will be those patches that received the most light from the light sources, and so on.

When solving in sorted order, the solution tends to proceed in approximately the same order as light would propagate through the environment. A similar approach was taken by Immel [11] in order to increase the efficiency of the view-independent specular radiosity algorithm. The reordering of the patches generally provides an accurate solution in less than a single iteration, substantially reducing computation costs.

3.3 The Ambient Term

Using the procedures described above, intermediate images will progress from a dark environment, continuously brightening to a fully illuminated scene including all diffuse interreflection. The illumination of the scene during early stages of the solution process will be inadequate, particularly for regions which do not receive direct illumination, since global illumination is not yet accurately represented. In earlier lighting models, the effect of global illumination was approximated by adding an arbitrary ambient term. Similar use is made of an *ambient* term here, but its value at any given point during the solution is based on the current estimate of the radiosities of all patches and the reflectivity of the environment. The ambient term is added for display purposes only and is not taken into account by the solution itself. The contribution of the ambient term gracefully decreases as the solution continues, providing a useful image almost immediately which unobtrusively progresses to an accurate rendering.

3.3.1 Computation of the Ambient Term

A reasonable first approximation to the form-factors can be made without any knowledge of the visibility or the geometric relationships between patches. The form-factor from any patch i to patch j can be approximated as the fraction of the total area of the environment taken up by the area of patch j. As with the correct form-factors the total will sum to unity. Thus,

$$F_{*j} \approx \frac{A_j}{\sum_{j=1}^{n} A_j} \tag{9}$$

An average reflectivity for the environment can be computed as an area weighted average of the patch reflectivities:

$$\rho_{ave} = \frac{\sum_{i=1}^{n} \rho_i A_i}{\sum_{i=1}^{n} A_i} \tag{10}$$

For any unit energy sent into the environment, ρ_{ave} will on average be reflected, and some of that will be reflected, etc. Thus, an overall interreflection factor R is simply the geometric sum:

$$R = 1 + \rho_{ave} + \rho_{ave}^2 + \rho_{ave}^3 + ... = \frac{1}{1 - \rho_{ave}} \tag{11}$$

From these assumptions an *Ambient* radiosity term is derived. It is simply the area average of the radiosity which has not yet been *shot* via form-factor computation times the reflection factor R.

$$Ambient = R \sum_{j=1}^{n} (\Delta B_j F_{*j}) \tag{12}$$

Thus at any point in the computation, the estimate of the radiosity of each patch can be improved by adding the contribution of the ambient radiosity. If B_i is the radiosity of patch i due to the radiosity received via shooting from other patches, an improved estimate is given by:

$$B_i' = B_i + \rho_i Ambient \tag{13}$$

This estimate of B_i' is used for display purposes only since the ambient contribution is not added to ΔB_i and thus is not shot during the solution. As the solution progresses the average unshot energy decreases and thus the ambient term decreases along with it. The values of B_i and B_i' converge and the initial ambient image yields gracefully to the more accurate estimate of global illumination provided by the radiosity equations.

3.4 Adaptive Subdivision: Achieving an Appropriate Surface Discretization

There are competing influences on how fine the subdivision of the surfaces of the environment should be. A finer subdivision means more computation but results in a more accurate representation of the sharp radiosity gradients that can occur at shadow boundaries. The original hemi-cube algorithm solved this problem by using a two level subdivision in which patches are further subdivided into elements [4].

In the revised algorithm as in the original algorithm, patch subdivision is kept coarse since the specific distribution of radiosity is less important for the patches, which act as the illuminators of the environment. The patches are subdivided into smaller elements. It is the elements which act as the receivers of light from the patches. The elements are projected onto a single hemi-cube for each patch to determine patch-to-element form-factors, F_{ie}. The light is thus shot from the patch to all elements. The radiosity of a patch is determined as the area weighted average of its element radiosities.

The number of patches, and thus the number of hemi-cubes, generally will grow very little during the radiosity analysis. Large patches need to be subdivided only if the radiosity varies greatly across the surface causing illumination inaccuracies or if the ratio of the areas in equation (7) causes the form-factor term $(F_{ij}A_i/A_j)$ to grow larger than unity.

The elements are free to be adaptively subdivided based on radiosity gradients without changing the patch geometry and thus no additional hemi-cube computation is required. The number of elements projected onto the hemi-cubes will grow as high gradients such as shadow boundaries are discovered. Images are generated by rendering the elements themselves as Gouraud shaded polygons with the radiosity at the vertices interpolated from adjacent elements.

4 Implementation

The complete algorithm is summarized in the following pseudo-code description:

```
/* initialization */
determine reflection factor, R;
/* determine initial ambient from given emission */
Ambient = R Σ_{i=1}^{n}(E_i A_i)/ Σ_{i=1}^{n} A_i;
/* initialize unshot radiosity to given emission */
for each patch: ΔB_i = E_i;
/* element e is a sub-unit of patch i */
for each element: B_e = E_i + ρ_i Ambient;
/* initialize change in ambient radiosity */
ΔAmbient = 0;

/* radiosity solution */
Until convergence {
  select patch i with greatest unshot energy, ΔB_i A_i;
† project elements onto hemi-cube located at patch i
    to compute patch i to element form-factors, F_ie;
  for each element e {
    /* determine increase in radiosity of element e due to
       ΔB_i */
    ΔRad = ρ_e ΔB_i F_ie A_i / A_e;
    /* add area weighted portion of increased radiosity of
       element e to radiosity of the patch j which contains
       element e */
    B_e = B_e + ΔRad + ρ_e ΔAmbient;
    ΔB_j = ΔB_j + ΔRad A_e / A_j;
  }
  interpolate vertex radiosities from neighboring elements;
  if( gradient from neighboring vertices is too high )
    subdivide elements and reshoot patch i;
  ΔB_i = 0;
  determine ΔAmbient from new unshot radiosities, ΔB_j;
† display environment as Gouraud shaded elements;
}
```

†Processes which can take advantage of current graphics hardware for scan conversion and hidden surface calculation.

The algorithms described above were implemented initially on a VAX 8700 and then on an HP 825 with an SRX graphics accelerator. The hemi-cube algorithm was performed in software and alternatively with the use of graphics hardware for the hidden surface determination and scan conversion portions of the form-factor routines. The ability to perform transformations, clipping and scan conversion on the HP workstation can potentially accelerate the hemi-cube computation and allows the intermediate results to be interactively displayed as a fully rendered image.

5 Results

The methods described above were compared experimentally in several combinations to determine the effect on the solution process. Tests included comparing the use of Gathering vs. Shooting, Sorted vs. Unsorted Patches, and With and Without Ambient effects. All the methods converged to the same final radiosity results in different amounts of time and with different intermediate results. The final converged results were used as a control with which to measure the error at stages in the image refinement. Individual errors were determined as the absolute differences between the converged and estimated radiosities of each element. (The average radiosity values of the color bands was used for the purposes of error measurement.) The square root of the area weighted mean of the square of individual errors (RMS) is used as a quantitative measure of overall radiosity inaccuracy.

$$RMS\ Error = \sqrt{\frac{\sum_{e=1}^{m}((B_e^* - B_e)^2 A_e)}{\sum_{e=1}^{m} A_e}} \qquad (14)$$

where B_e^* is the converged radiosity and B_e is the intermediate radiosity of element e. m is the total number of elements.

The images themselves offer a qualitative basis for comparison.

5.1 A Test Environment

Test were performed on a model of two office cubicles subdivided into 500 patches and 7000 elements. Four iterative approaches to solving the radiosity equations were run. After each hemi-cube, images using the current radiosity estimates were displayed as hardware Gouraud shaded polygons on a Hewlett Packard 825SRX workstation.

The four approaches were:

1. Gathering Only: This is the *traditional* radiosity method using a Gauss-Siedel solution. One hemi-cube is placed at each element.
2. Shooting Only: This method consists of reversing the process by shooting light to each element through a hemi-cube placed at each patch.
3. Shooting with Sorting: The same as the second approach, but with the patch with the largest *unshot* energy being used at each step.
4. Shooting with Sorting and Ambient: This time the radiosity due to an estimated ambient term is included for display.

Figures 2 through 5 each contain eight images from methods 1 through 4 respectively. From top to bottom they show the results after 1, 2, 24, and 100 hemi-cubes. The right hand image is a pseudo-color version. Gray indicates an accurate solution when comparing each of these images to the converged result in figure 6. The blue intensity indicates under-estimated radiosity values and red indicates an over-estimate. The inclusion of the ambient term provides an immediately useful image as illustrated in figure 5, (repeated on the cover). Note that as the algorithm progresses, the over-estimates in the shadowed regions due to the ambient term are continuously redistributed

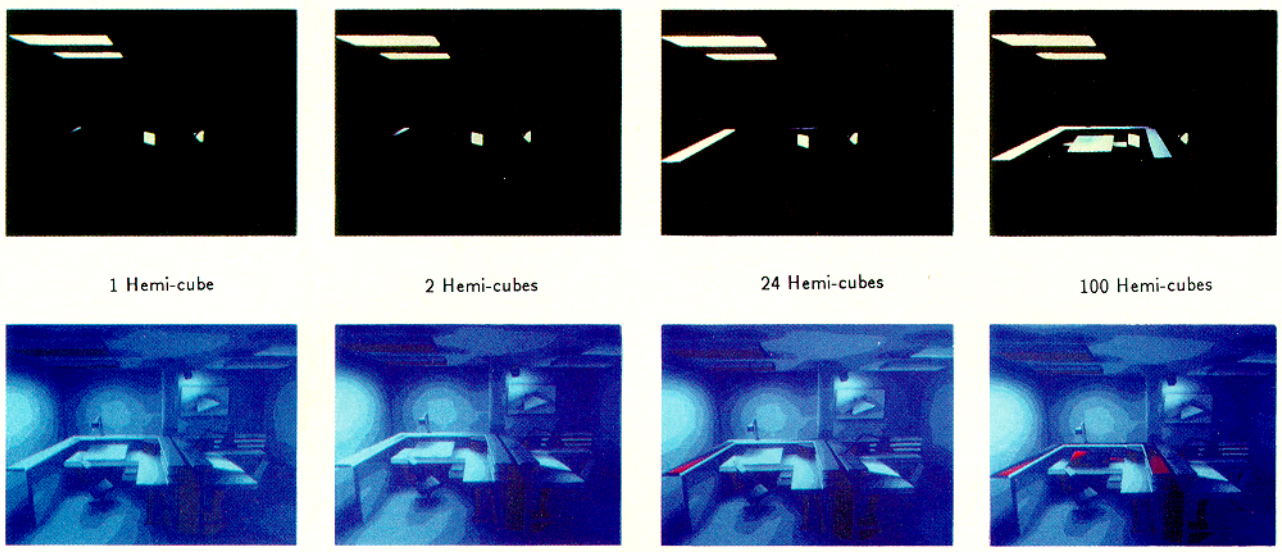


Figure 2: Gathering Only

Since the radiosity of only one patch is estimated for each hemi-cube performed, the gathering approach converges very slowly. Thus even after 100 hemi-cubes, the radiosity of very few surfaces in the environment have been estimated. The pseudo-color images on the right indicate underestimates of radiosity in blue and overestimates in red. (The small amounts of red are due to numerical differences between the hemi-cubes used for gathering and shooting.)

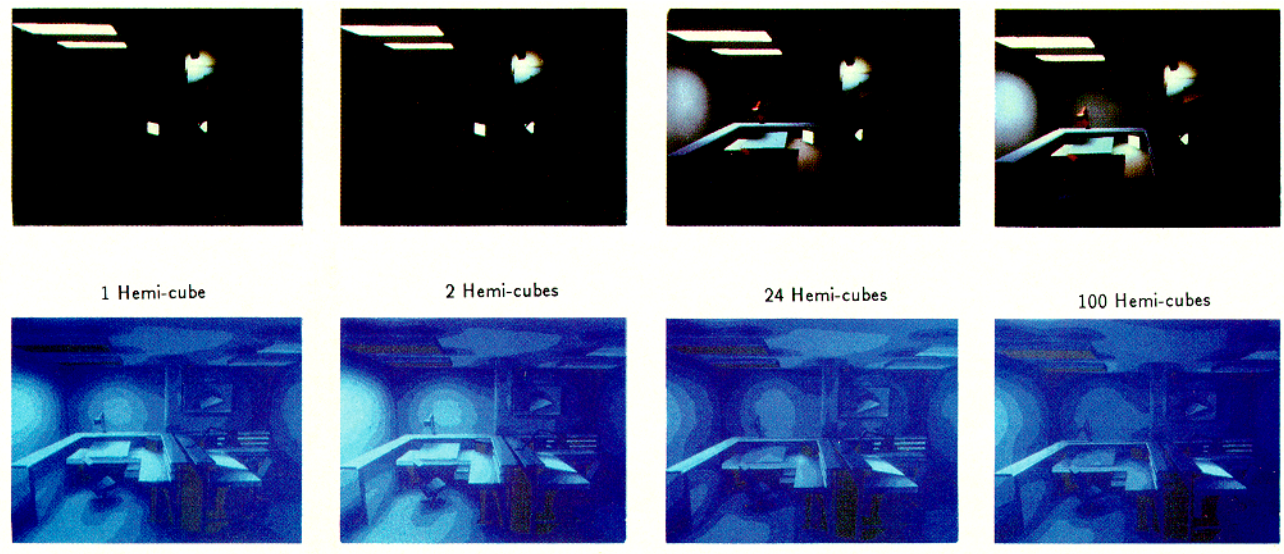


Figure 3: Shooting Only

By shooting light, more of the environment is illuminated for each hemi-cube. However, the order in which the patches shoot light is arbitrary, thus loosing potential efficiency. Note in the graph of figure 3, the jumps which occur when original light sources are processed.

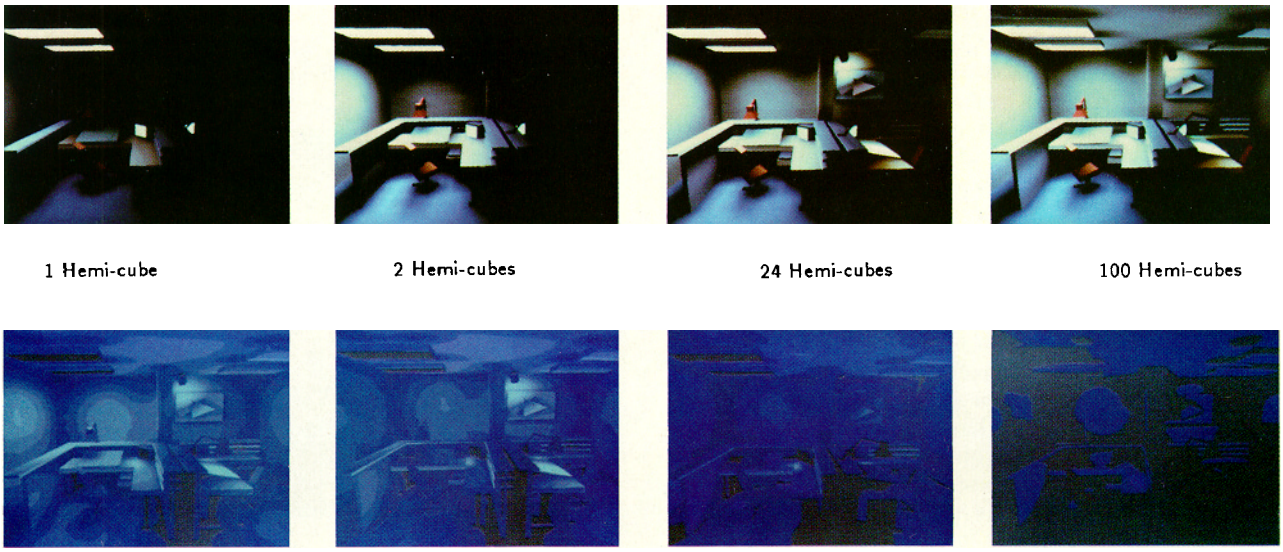


Figure 4: Shooting and Sorting

By sorting the patches according to "unshot" energy, a continuity and efficiency are achieved. Note the continual brightening of the environment as the interreflection between surfaces is accounted for. After only 100 hemi-cubes, a near complete radiosity solution has been found. Note that the under-illumination, indicated by the blueness of the pseudo-color images diminishes gradually after each step in the solution.

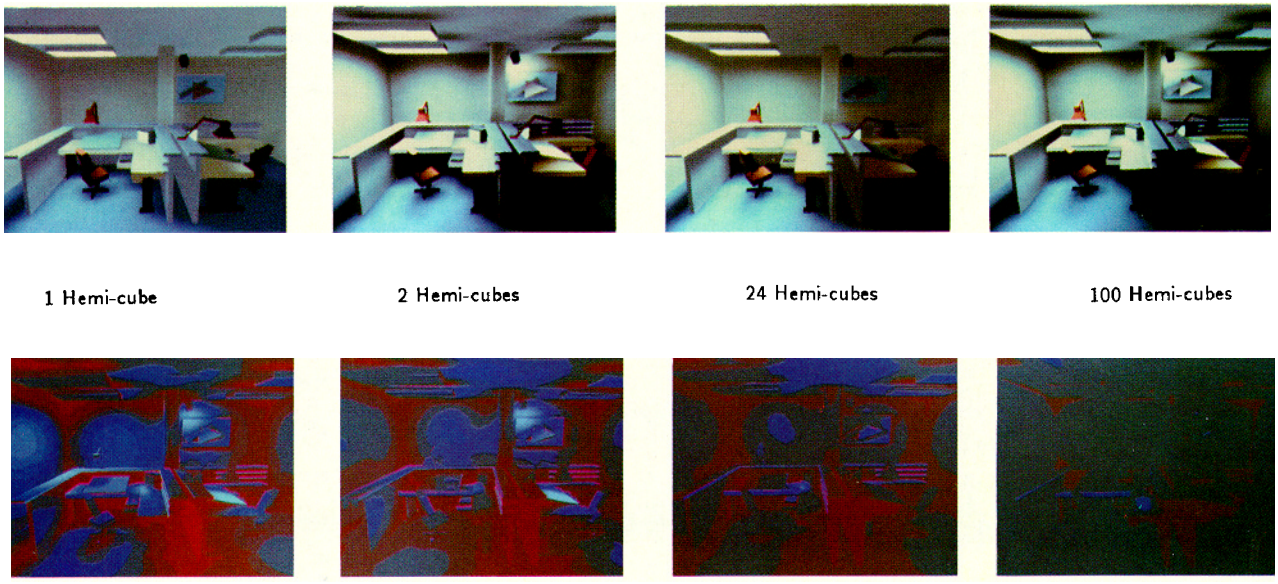


Figure 5: Shooting, Sorting and Ambient

The inclusion of the ambient radiosity provides an immediately useful image after only a single hemi-cube. Note that as the solution continues, the contrast is enhanced as the over-illumination in shadowed areas (indicated by the red pseudo-color) is transferred to the under-illuminated (blue) regions. The ambient term maintains a consistent overall illumination level allowing a more graceful transition to a final image.

to the brighter areas of the environment which were initially under-estimated.

Figure 6 contains an image produced after allowing the methods to run until convergence. The graph below, figure 7, follows the first 100 hemi-cubes and shows the RMS error of the radiosities of the elements. The graph clearly illustrates the improvements generated by the reformulation of the radiosity algorithm. In figure 8, all four methods are compared at the same point early in the solution process. At a cost of only two hemi-cubes, a radiosity image sufficient for many applications is rendered by the fourth method.

The computation of a single hemi-cube with resolution 150 by 150 for the test environment takes approximately ten seconds in the software implementation on the Hewlett Packard 825SRX workstation. The Hewlett Packard workstation was able to display each intermediate stage of the test environment in one to two seconds. Although these clearly cannot be termed interactive speeds at present, the next generation of workstation hardware should acheive near interactive speeds for an environment like the one shown. In addition, the ability to rotate or move through the environment does not depend on hemi-cube computation time. If the display of the environment and the hemi-cube calculations are performed in parallel on separate processors, walkthroughs can be performed during the iterative cycle without disturbing the radiosity computation.

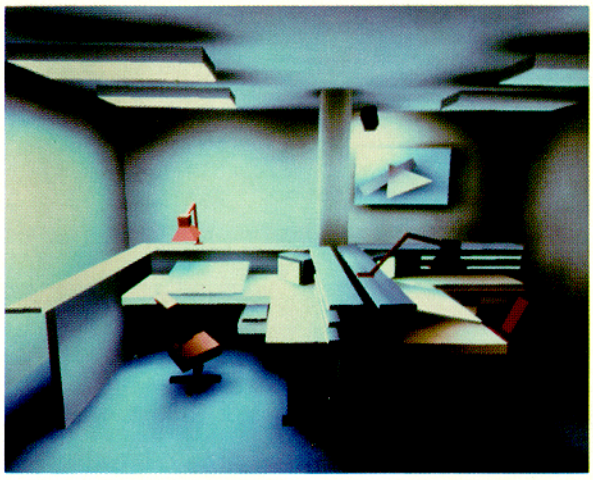


Figure 6: Two Office Cubicles: The Converged Results

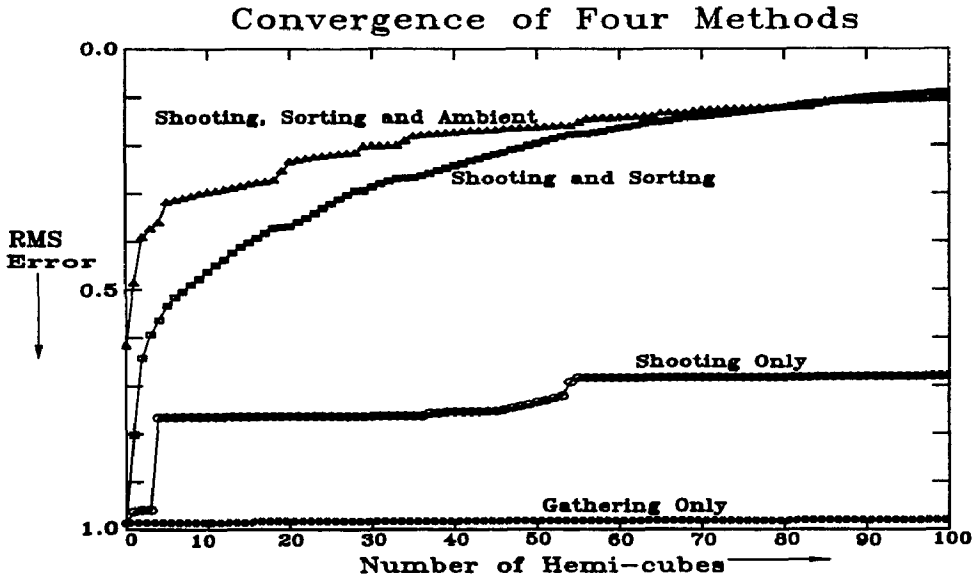
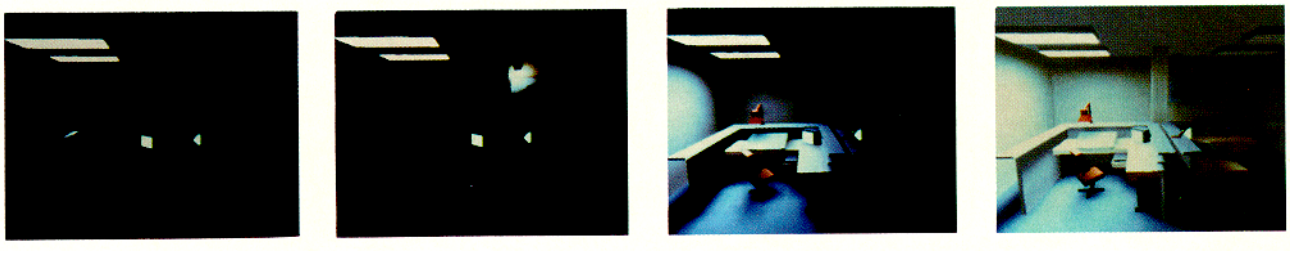


Figure 7: Plot of Normalized RMS Errors for the First 100 Hemi-Cubes

Note the initial improvement in accuracy of the fourth method due to the inclusion of the ambient term.



Gathering Only | Shooting Only | Shooting and Sorting | Shooting, Sorting, and Ambient

Figure 8: The Four Methods Compared After Two Hemi-cubes

These four images extracted from the same point in the previous four sequences illustrate the great advantage provided by the fourth method for displaying immediate results.

5.2 A Steel Mill

An early software version of the shooting and sorting algorithm described above was implemented on a VAX8700 and run on a highly complex scene to test its performance. A model of a steel mill was constructed containing 30,000 patches which were subdivided into 50,000 elements. The patch solution was run for only 2,000 of the patches in 5 hours providing a close approximation of the global diffuse illumination. This was followed by a view dependent post-process taking 190 hours in which the radiosity at the vertices of visible elements was computed by gathering light through a hemi-cube at each vertex. The results were then displayed by interpolating radiosity values across the elements. Figure 9 is the result of this process.

A traditional radiosity approach would have required the computation of 1.5×10^9 form-factors or 6 gigabytes worth of storage (sparcity would probably have reduced this by an order of magnitude). The iterative approach required the storage of only one row of form-factors or 0.12 Mbytes. In addition, the preprocess solution required only 2,000 hemi-cubes, or less than 5 percent of the 50,000 required for earlier implementations.

Figure 9: The Steel Mill

A radiosity solution for this complex environment containing 50000 elements would have been virtually impossible due to storage and computational requirements without the use of the reformulated radiosity approach described in this paper.

6 Conclusion and Future Directions

A reformulated version of the radiosity algorithm for image synthesis has been presented. Two major advantages over the traditional radiosity algorithm are evident: a useful image (although not the final image) is produced in time linear to the number of patches, and the $O(n^2)$ storage requirements for the form-factors have been eliminated. The reformulation allows the rapid generation of approximate solutions which gracefully, progressively refine themselves to accurate representation of global illumination in diffuse environments. This allows the method to be used in applications requiring interaction. It also provides a means to examine the progress of image development early in the rendering process thus providing a valuable previewing capability.

The results of the radiosity analysis make possible the display of high quality diffuse realistic images from any view point. This view independent solution provides a starting point for further adaptive refinement to add view dependent effects such as highlights and specular reflection. Such refinement might include pixel by pixel post processes as as the modified ray tracing algorithm as described in [17], or Monte Carlo methods which can take advantage of global illumination information for importance sampling.

A variety of issues arise when implementing the methods described above. How much and when should the patches and elements be subdivided? How high a hemi-cube resolution is necessary to eliminate form-factor aliasing? What is the interrelationship between patch size, element size, hemi-cube resolution, radiosity gradients, and image resolution. The answers are environment dependent and also clearly depend on the uses to which the images will be applied. Further research should be directed towards providing a body of heuristics tuned to environments, computational resources, and user needs.

Taking advantage of all information about environmental illumination at each stage in the solution process is a concept central to the ideas described in this paper. Future research should be able to apply similar ideas to the problem of rendering dynamic environments needed for geometric modeling and other applications.

Future research should also examine the possible impact of this approach on the design of graphics workstations. Hardware design can provide specialized frame buffers dedicated to hemi-cube computation or for complex reflectance computation. The goal is clear; to provide the best image possible in interactive times and to provide a continuity to a realistic image synthesis.

7 Acknowledgements

The research in this paper was carried out under a grant from the National Science Foundation #DCR8203979 with equipment generously donated by Digital Equipment Corporation and Hewlett Packard. The office model was originally created by Keith Howie and modified by Shenchang Eric Chen. The Steel Mill was modeled through a great effort by Stuart I. Feldman. The photography was done by Emil Ghinger. Special thanks to Holly Rushmeier for technical discussions and to Julie O'Brien and Helen Tahn for helping assemble the paper.

References

[1] Arvo, James, "Backward Ray Tracing," *Developments in Ray Tracing(SIGGRAPH '86 Course Notes)*, Vol.12, August 1986.

[2] Bergman, Larry, Henry Fuchs, Eric Grant, Susan Spach, "Image Rendering by Adaptive Refinement," *Computer Graphics(SIGGRAPH '86 Proceedings)*, Vol.20, No.4, August 1986, pp.29-38.

[3] Cohen, Michael F., Donald P. Greenberg, "A Radiosity Solution for Complex Environment," *Computer Graphics(SIGGRAPH '85 Proceedings)*, Vol.19, No.3, July 1985, pp.31-40.

[4] Cohen, Michael F., Donald P. Greenberg, David S. Immel, Philip J. Brock, "An Efficient Radiosity Approach for Realistic Image Synthesis," *IEEE Computer Graphics and Applications*, Vol.6, No.2, March 1986, pp.26-35.

[5] Cook, Robert L., Thomas Porter, Loren Carpenter, "Distributed Ray Tracing," *Computer Graphics(SIGGRAPH '84 Proceedings)*, Vol.18, No.3, July 1984, pp.137-145.

[6] Fuchs, Henry, et. al., "Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-Planes," *Computer Graphics(SIGGRAPH '85 Proceedings)*, Vol.19, No.3, July 1985, pp.111-120.

[7] Goral, Cindy M., Kenneth E. Torrance, Donald P. Greenberg, "Modeling the Interaction of Light Between Diffuse Surfaces," *Computer Graphics(SIGGRAPH '84 Proceedings)*, Vol.18, No.3, July 1984, pp.213-222.

[8] Gouraud, H., "Continuous Shading of Curved Surfaces," *IEEE Transactions on Computers*, Vol.20, No.6, June 1971, pp.623-628.

[9] Heckbert, Paul S. and Pat Hanrahan, "Beam Tracing Polygonal Objects," *Computer Graphics(SIGGRAPH '84 Proceedings)*, Vol.18, No.3, July 1984, pp.119-128.

[10] Hornbeck, Robert W., *Numerical Methods*, Quantum Publishers, New York, NY, 1974, pp.101-106.

[11] Immel, David S., Michael F. Cohen, Donald P. Greenberg, "A Radiosity Method for Non-Diffuse Environments," *Computer Graphics(SIGGRAPH '86 Proceedings)*, Vol. 20, No.4, August 1986, pp.133-142.

[12] Kajiya, James T., "The Rendering Equation," *Computer Graphics(SIGGRAPH '86 Proceedings)*, Vol.20, No.4, August 1986, pp.143-150.

[13] Nishita, Tomoyuki, Eihachiro Nakamae, "Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection," *Computer Graphics(SIGGRAPH '85 Proceedings)*, Vol. 19, No.3, July 1985, pp.22-30.

[14] Phong, Bui Tuong, "Illumination for Computer Generated Pictures," *Communications of the ACM*, Vol.18, No.6, June 1975, pp.311-317.

[15] Siegel, Robert, John R. Howell, *Thermal Radiation Heat Transfer*, Hemisphere Publishing Corp., Washington DC., 1981.

[16] Swanson, Roger W. and Larry J. Thayer, "A Fast Shaded-Polygon Render," *Computer Graphics(SIGGRAPH '86 Proceedings)*, Vol.20, No.4, August 1986, pp.95-102.

[17] Wallace, John R., Michael F. Cohen, Donald P. Greenberg, "A Two-pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods," *Computer Graphics(SIGGRAPH '87 Proceedings)*, Vol. 21, No.4, July 1986, pp.311-320.

[18] Watkins, G. S., "A Real-Time Visible Surface Algorithm," *University of Utah, UTECH-CSC-70-101*, 1970.

[19] Whitted, Turner, "An Improved Illumination Model for Shaded Display," *Communication of the ACM*, Vol.23, No.6, June 1980, pp.343-349.