January 1993

A Gathering and Shooting Progressive Refinement Radiosity Method

Min-Zhi Shao
University of Pennsylvania

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-03.

A Gathering and Shooting Progressive Refinement Radiosity Method

Abstract

This paper presents a gathering and shooting progressive refinement radiosity method. Our method integrates the iterative process of light energy gathering used in the standard full matrix method and the iterative process of light energy shooting used in the conventional progressive refinement method. As usual, in each iteration, the algorithm first selects the patch which holds the maximum unprocessed light energy in the environment as the shooting patch. But before the shooting process is activated, a light energy gathering process takes place. In this gathering process, the amount of the unprocessed light energy which is supposed to be shot to the current shooting patch from the rest of the environment in later iterations is pre-accumulated. In general, this extra amount of gathered light energy is far from trivial since it comes from every patch in the environment from which the current shooting patch can be seen. However, with the reciprocity relationship for form-factors, still only one hemi-cube of the form-factors is needed in each iteration step. Based on a concise record of the history of the unprocessed light energy distribution in the environment, a new progressive refinement algorithm with revised gathering and shooting procedures is then proposed. With little additional computation and memory usage compared to the conventional progressive refinement radiosity method, a solid convergence speedup is achieved. This gathering and shooting approach extends the capability of the radiosity method in accurate and efficient simulation of the global illuminations of complex environments.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-03.

A Gathering and Shooting Progressive Refinement Radiosity Method

MS-CIS-93-03
GRAPHICS LAB 52

Min-Zhi Shao
Norman I. Badler

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

January 1993

A Gathering and Shooting Progressive Refinement Radiosity Method

Min-Zhi Shao and Norman I. Badler

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

Abstract

This paper presents a gathering and shooting progressive refinement radiosity method. Our method integrates the iterative process of light energy gathering used in the standard full matrix method and the iterative process of light energy shooting used in the conventional progressive refinement method. As usual, in each iteration, the algorithm first selects the patch which holds the maximum unprocessed light energy in the environment as the shooting patch. But before the shooting process is activated, a light energy gathering process takes place. In this gathering process, the amount of the unprocessed light energy which is supposed to be shot to the current shooting patch from the rest of the environment in later iterations is pre-accumulated. In general, this extra amount of gathered light energy is far from trivial since it comes from every patch in the environment from which the current shooting patch can be seen. However, with the reciprocity relationship for form-factors, still only one hemi-cube of the form-factors is needed in each iteration step. Based on a concise record of the history of the unprocessed light energy distribution in the environment, a new progressive refinement algorithm with revised gathering and shooting procedures is then proposed. With little additional computation and memory usage compared to the conventional progressive refinement radiosity method, a solid convergence speedup is achieved. This gathering and shooting approach extends the capability of the radiosity method in accurate and efficient simulation of the global illuminations of complex environments.

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

General Terms: Algorithms

Additional Key Words and Phrases: Gathering and shooting, global illumination, progressive refinement, radiosity

1 Introduction

Visual realism is no doubt a primary goal of the rendering research in computer graphics. In approaching this goal, it has been widely realized that the accurate and efficient simulation of the light energy interreflection between objects in the scene environment is the most important and difficult part in realistic image synthesis. By accurate and efficient, we mean that the illuminating model should be able to correctly capture the detailed physical lighting behavior in the scene environment while at the same time the rendering algorithm should use as little the computing resources as possible.

The radiosity method, which was first developed for thermal radiation computations [12], has become a major global illuminating model in computer graphics during the last few years. This model is especially capable in simulating the light energy interreflection within ideal diffuse environment. Its numerical solutions developed to date, in general, can be divided into two categories: the full matrix solution and the progressive refinement solution. For an excellent introduction to the historical and theoretical backgrounds of the radiosity method, see [11].

This paper presents a gathering and shooting progressive refinement radiosity method. Our major research goal is to accelerate the convergence of the conventional progressive refinement method. Our solution is to integrate the light energy gathering iterative process used in the standard full matrix method and the light energy shooting iterative process used in the conventional progressive refinement method so as to process as much unprocessed light energy as we could in each iteration step. As usual, in each iteration, the algorithm first selects the patch which holds the maximum unprocessed light energy in the environment as our shooting patch. But before the shooting process is activated, a light energy gathering process takes place. In this gathering process, the amount of unprocessed light energy to be shot in later iterations to the current shooting patch from the rest of the environment is pre-accumulated. Intuitively, the more light energy being processed in each refinement iteration step, the more rapid the convergence should be. This extra amount of gathered light energy, in general, is far from trivial since it comes from every patch in the environment from which the current shooting patch can be seen. However, with the reciprocity relationship for form-factors, still only one hemi-cube of the form-factors is needed to calculate both the light energy distribution from the rest of the environment patches to the shooting patch in the gathering process and the light energy distribution from the shooting patch to the rest of the environment patches in the shooting process. Based on a concise record of the history of the unprocessed light energy distribution in the environment, a new progressive refinement algorithm with revised gathering and shooting procedures is then proposed. As a result, with little additional cost in computation and memory compared with the conventional progressive refinement radiosity method, a solid convergence speedup is achieved.

2 Radiosity Method

In this section, we shall briefly review the general structures of the full matrix radiosity method and the progressive refinement radiosity method.

2.1 Full Matrix Radiosity (Gathering) Method

The full matrix radiosity method was introduced to computer graphics by Goral et al. [7] in an effort to simulate the light energy interreflection within ideal diffuse environment. The idea is first to discretize the scene environment into patches, then to compute the form-factors between each pair of the patches in the environment, and finally to calculate the patch radiosities by solving a set of linear equations which is established based on the principle of light energy equilibrium.

Suppose an environment is discretized into n patches. Mathematically, this radiosity form of the light energy equilibrium can be expressed as

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij}, \qquad i = 1, 2, \ldots, n \tag{1}$$

where B_i is the radiosity, E_i is the emission, and ρ_i is the reflectivity of patch i, respectively. F_{ij} is the form-factor from patch i to patch j which represents the fraction of light energy leaving patch i reaching patch j.

In matrix format, we have

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}. \tag{2}$$

Since the above matrix equation is diagonally dominant, the Gauss-Seidel iterative method [13] is usually recommended. As pointed out in [5], here the Gauss-Seidel iterations are physically nothing more than a series of light energy gathering processes. That is, row by row, the radiosity of the current row patch is gathered and updated based on the current estimate of the radiosity of every other patch in the environment. For this reason, the full matrix radiosity method is often referred to as the gathering method.

Cohen et al. [3] [4] later published a hemi-cube algorithm for form-factor calculations of environments with occluded surfaces and a substructuring algorithm for adaptive environment subdivision. Since the light interreflections in the environment are globally computed in this full matrix radiosity method, the physical lighting phenomena such as surface-to-surface color bleeding and soft shadow are naturally simulated.

This method, however, suffers from the computation and (in our experience) most often the storage of a matrix of $O(n^2)$ form-factors. Even with virtual memory supported in many current systems, the memory swaps will eventually slow down the program dramatically, especially during the radiosity matrix equation solution. To a certain extent, this memory requirement strictly limited our capability of rendering complex environments.

Besides the excessive memory requirement, another major disadvantage to the full matrix method is its lack of user-machine interactivity. An environment cannot be rendered until the full set of $O(n^2)$ form-factors is calculated so that the matrix equation be solved. Perhaps even worse, this whole series of computations have to be redone whenever there is a geometry change in the environment. This makes the environment modeling and rendering process of the radiosity method extremely inefficient.

Recently, a new algorithm called the hierarchical radiosity method was presented by Hanrahan et al. [8] [9]. In this method, a hierarchical tree representation of the interactions between patches and elements in the environment is constructed by adaptively subdividing patches into subpatches according to a predefined form-factor error bound. With this representation, theoretically, all form-factors are calculated within the same precision and therefore the numerical errors are forced to be uniformly distributed in the environment. As a result, for a given amount of time and memory, higher quality imagery should be generated. It is inherently a full matrix method, but the interactions between patches and elements in the environment and therefore the memory storage for the form-factors is reduced to $O(n)$ in which n is the number of elements in the environment. It seems to us, however, that in practice accurate global illumination can hardly be simulated by this method unless an accurate and efficient form-factor computation scheme is developed based on the hierarchical subdivision structure.

2.2 Progressive Refinement Radiosity (Shooting) Method

With the problems mentioned in the previous subsection, Cohen et al. [5] presented a progressive refinement procedure which greatly improved the performance of the traditional full matrix radiosity method. Instead of computing all form-factors and then solving a complete set of linear radiosity equations, the progressive refinement method computes and distributes the light energy one patch per refinement iteration step. By shooting the energy of the light sources and other patches in the order of their brightness, a rough approximation of the overall global illumination, in general, can be obtained quickly after only a few iteration steps. The local illumination details, however, still need many additional iterations to converge. Perhaps even more importantly, the $O(n^2)$ form-factor memory storage requirement is reduced to $O(n)$. Consequently, the capability of rendering complex environment is squared.

More precisely, suppose patch i is the shooting patch selected. For any patch j in the environment, the contribution of the radiosity from patch i to patch j can be expressed as

$$B_j = \rho_j B_i A_i F_{ij} / A_j, \qquad j = 1, 2, \ldots, n. \tag{3}$$

Rather than the full matrix method in which only one patch is gathering light energy from all other patches in the environment per iteration step, in the progressive refinement method, only one patch is shooting light energy to all other patches in the environment per iteration step. For this reason, the progressive refinement radiosity method is often referred to as the shooting method.

The progressive refinement scheme has also been extended to include the concepts of positive and negative light energy shootings with the applications in dynamic environment simulations [2] [6]. Users therefore can interactively add, move, remove, and even change the shape or the attribute of any object in the environment. In a sense, the radiosity modeling process and rendering process are integrated, according to the inherent interactivity of the progressive refinement method.

The progressive refinement method, however, introduces a few problems which can hardly be neglected. One is aliasing artifacts. This problem is rooted in the hemi-cube algorithm for the calculation of the form-factors. Two different form-factor computation schemes, analytical [1] and ray tracing [14], were proposed and convincingly eased the problem in general cases.

Another problem with the progressive refinement method is its convergence speed. This problem has been less mentioned and studied. The authors notice that, in almost all published images generated by the conventional progressive refinement method, a very small percentage (usually less than 5%) of the patches in the environment have been ever processed. This means that the contributions of the majority of the patches in the environment are estimated merely by a constant ambient term [5]. As a result, the details of local illuminating effects, such as color bleeding, are usually lost, though a quick preview of the overall global illumination is generally available after only a few refinement iterations. Surely, we can continuously process the rest of the patches in the environment. The problem is, as always, the improvement of the local illumination details versus the costs of time and memory. Notice that the patches whose energy have been shot earlier may have to be shot again and again later when enough energy from other patches has been gathered. Since linear memory usage is the key to process even a moderately complex environment, it leaves us no choice but to recompute the form-factors associated with these patches whenever they are needed. Practically, this makes the progressive refinement radiosity method extremely slow if an accurate simulation of the global illumination is expected. In this paper, we are particularly interested in studying and accelerating the convergence in the stage of the post light source iterations.

3 Gathering and Shooting Algorithm

In this section, we shall first report the results of an experimental study of the convergence of the conventional progressive refinement radiosity method. Then, for the further acceleration of this convergence, we shall introduce a new gathering and shooting progressive refinement algorithm as well as its experimental comparisons with the conventional method.

3.1 Progressive Refinement: A Case Study

For a concrete understanding of the convergence of the conventional progressive refinement shooting procedure, we have constructed a typical test environment in radiosity research. This test environment was initially subdivided into 803 patches and among them 25 were light source patches, as shown in Figure 1. The reflectivity of the white wall was (0.7, 0.7, 0.7) while the reflectivity of the red box, the green box, and the blue box were (0.9, 0.1, 0.1), (0.1, 0.9, 0.1), and (0.1, 0.1, 0.9), respectively. Figure 2 is the final converged solution which consists of 10,298 elements.

In Figure 3, from left to right and top to bottom, the first seven images are generated by 25, 100, 200, 400, 800, 2,000, and 5,000 iterations, respectively; for comparison, the image at the right bottom corner is the converged solution. It should be noted that the ambient term is not used in our experiments in this paper, since it is not part of the iterative solution and its only function is for a

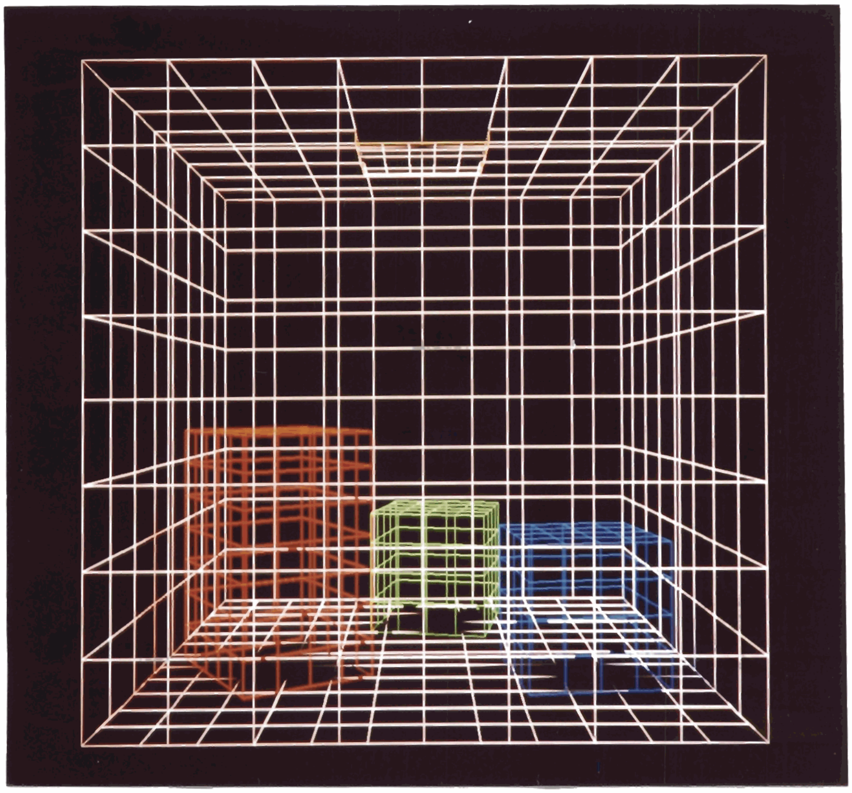


Figure 1: Test environment — initial subdivision: 803 patches; reflectivity: wall = (0.7, 0.7, 0.7), redbox = (0.9, 0.1, 0.1), grnbox = (0.1, 0.9, 0.1), blubox = (0.1, 0.1, 0.9).

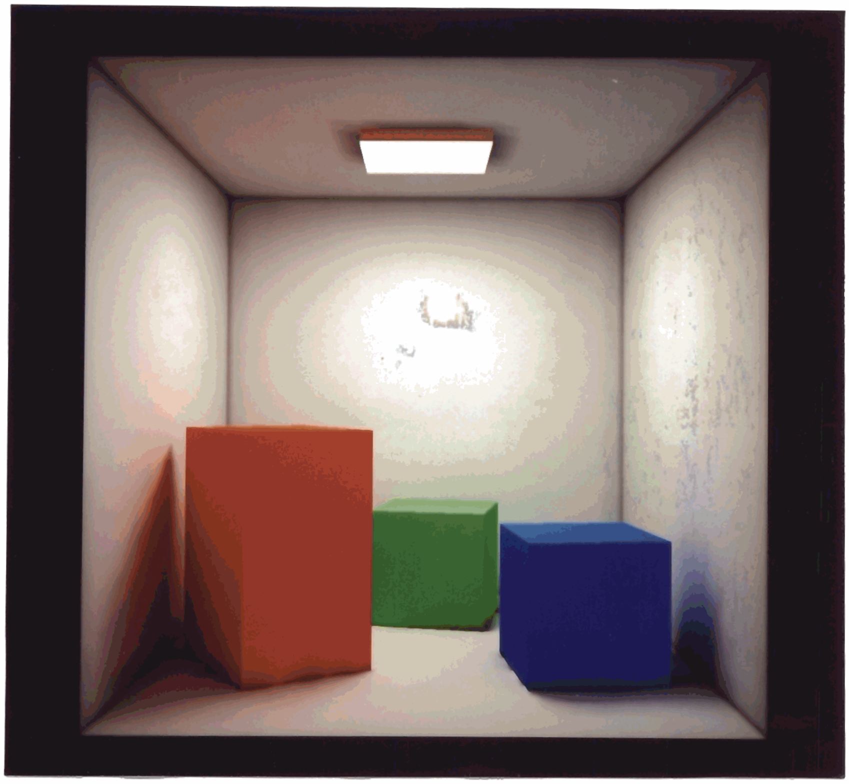


Figure 2: Test environment — converged solution: 10,298 elements.

brighter intermediate environment preview. These images have offered us a qualitative evaluation of the intermediate solutions in different stages of the progressive refinement procedure. As we can see, besides the gradually brighter overall illumination which might be partially compensated by the ambient term, differences are evident in the local illumination effects such as surface-to-surface color bleeding and shading in the shadow areas.

For a quantitative measure of the goodness of an intermediate solution, [5] uses the square root of the area weighted mean of the square of the individual element errors, or explicitly,

$$\text{RMS Error} = \sqrt{\frac{\sum_{e=1}^{m} (B_e^* - B_e)^2 A_e}{\sum_{e=1}^{m} A_e}} \tag{4}$$

where B_e^* is the converged radiosity and B_e is the intermediate radiosity of element e.

However, formula (4) only gives us an averaged overall measure. We would rather like to have an evaluation of the convergence accuracy at a relatively small scale, say the patch-element level, while at the same time keeping a global perspective. In this paper, the histogram of the patch vertex radiosity (HVR) accuracy,

$$\text{HVR Accuracy} = \text{Histogram}\{\frac{B_v}{B_v^*},\ v = 1, 2, \ldots, l\} \tag{5}$$

where B_v^* is the converged radiosity and B_v is the intermediate radiosity of patch vertex v, is used as an overview of the radiosity accuracy distribution of an intermediate solution.

The patch vertices can be viewed as a set of evenly distributed samples in the environment. Similarly, we can also use the histogram of the element vertex radiosity accuracy as an adaptive sampling. Figure 4(a) illustrates a series of histograms of the patch vertex radiosity accuracy corresponding to different iteration stages in Figure 3. As we can see, it could take a substantial number of the iterations for the progressive refinement procedure to converge to its final solution.

With these visual images and statistical graphs in mind, let us take a different view of the convergence behavior in terms of the percentages of patch shooting times in different iteration stages. For our test environment, Table 1(a) contains the percentage distributions with the series of iteration stages corresponding to Figure 3 and Figure 4(a). The first column in the table is a series of iteration stages. The first row in the table is the patch shooting times. The following rows which correspond to the different iteration stages contain the percentages of the patch shooting times. For instance, the second row tells us that, after 25 iterations, 96.9% of the patches have not been shot and 3.1% of the patches have been shot once. Let us take a closer look of these statistics. After 800 iterations, 25.3% of the patches have been shot once, 31.8% of the patches have been shot twice, and 3.6% of the patches have been shot three times. But, 39.4% of the patches have not been shot at all. After 2,000 iterations, the situation is similar: 13.4%, 7.6%, 22.2%, 14.4% of the patches have been shot two, three, four, and five times. But still 38.5% of the patches have been shot only once. The percentage distribution at the 5,000 iterations stage is even more diverse.

What do these percentages tell us? Propagation of the patch shooting times is not constant, but rather, the order and magnitude of the light energy contribution of each individual patch to the global illumination in the environment can vary significantly. Apparently, this phenomenon has something to do with the shooting patch selection criteria which says that the patch which holds the maximum unprocessed light energy

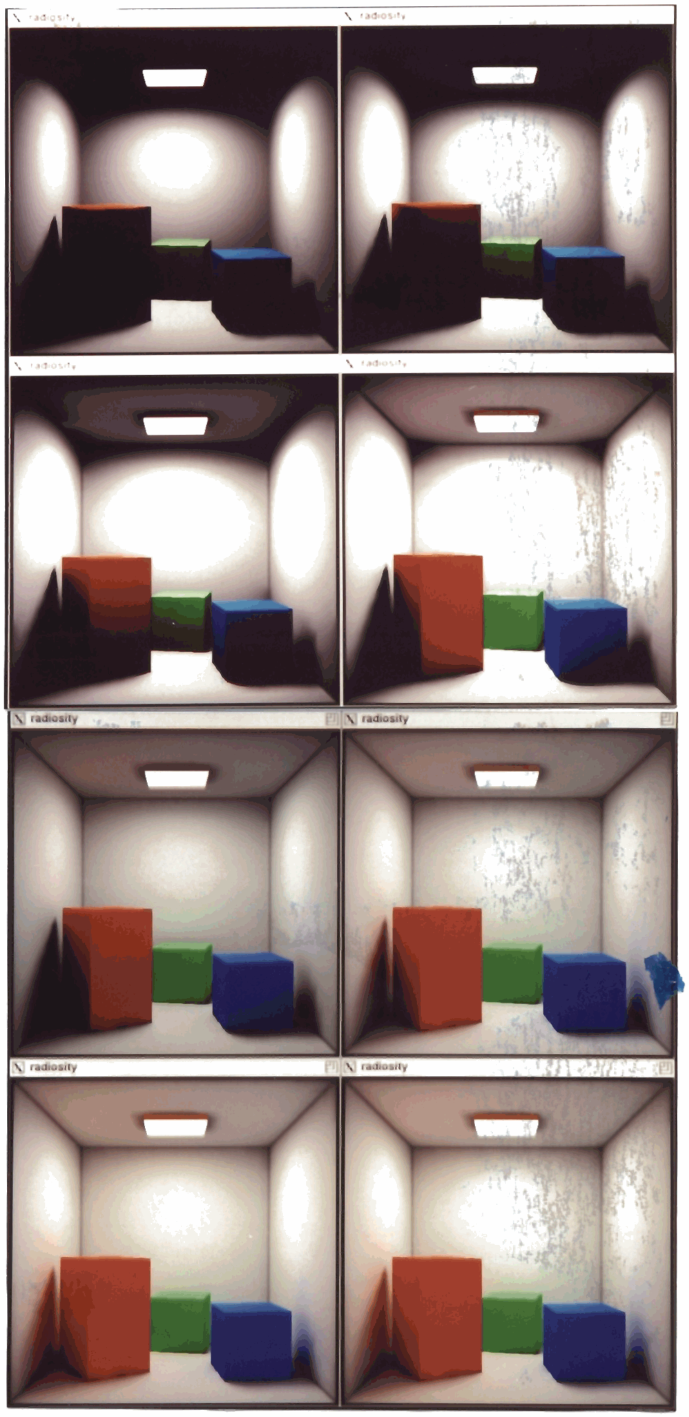
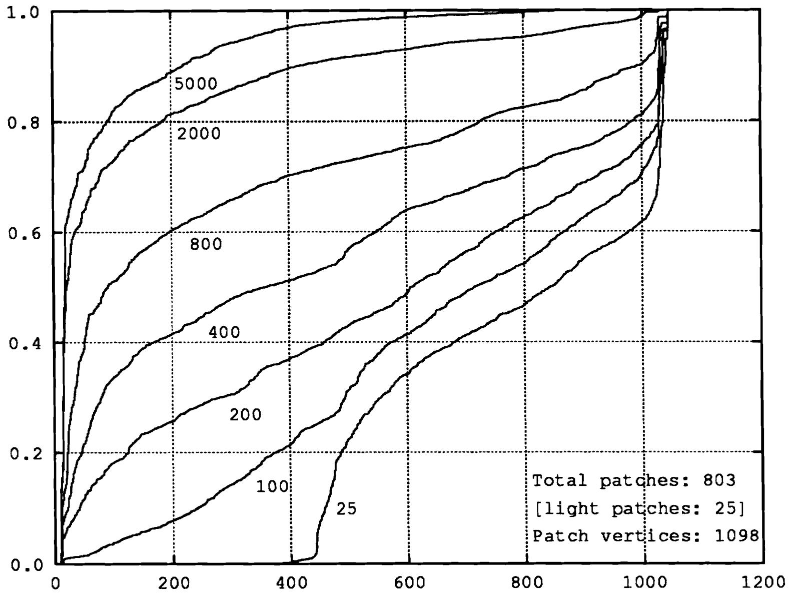
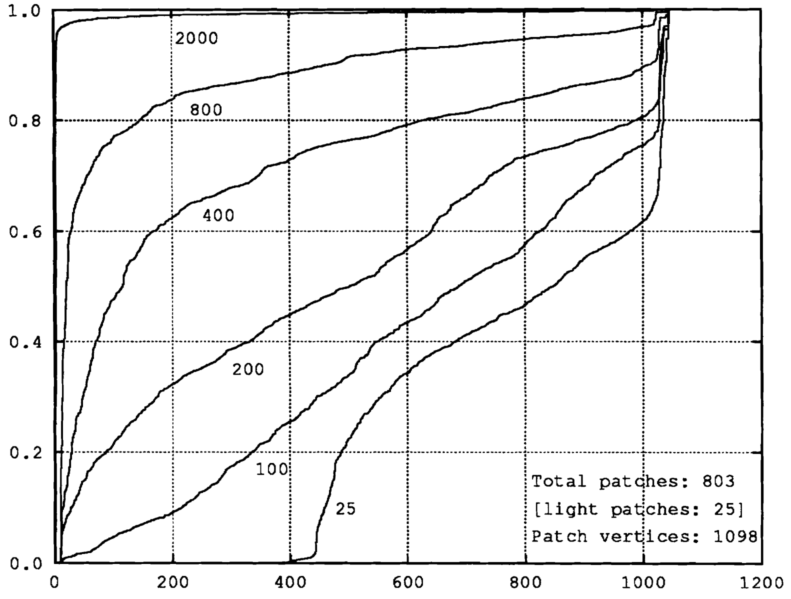


Figure 3: Test environment — conventional progressive refinement radiosity method (from left to right and top to bottom): 25 iterations, 100 iterations, 200 iterations, 400 iterations, 800 iterations, 2,000 iterations, 5,000 iterations, and converged solution.



(a) standard shooting



(b) gathering and shooting

Figure 4: Test environment — histograms of the patch vertex radiosity accuracy.

	0	1	2	3	4	5	6	7	8	9	10	11	12
25	96.9	3.1											
100	87.5	12.5											
200	75.1	24.9											
400	52.2	45.8	2.0										
800	39.4	25.3	31.8	3.6									
2000	3.9	38.5	13.4	7.6	22.2	14.4							
5000	1.3	0.9	2.0	31.6	11.1	6.1	7.7	0.0	1.6	7.2	13.1	16.3	1.0

(a) standard shooting

	0	1	2	3	4
25	96.9	3.1			
100	87.5	12.5			
200	75.1	24.9			
400	50.2	49.8			
800	32.6	35.3	32.1		
2000	1.3	19.2	36.3	15.6	27.6

(b) gathering and shooting

Table 1: Test environment — percentages of the patch shooting times in different iteration stages. The first column in the table is a series of iteration stages. The first row in the table is the patch shooting times. The following rows which correspond to the different iteration stages contain the percentages of the patch shooting times.

$$\max_i\{(\Delta B_i^r + \Delta B_i^g + \Delta B_i^b)A_i\} \tag{6}$$

should always have the highest priority [5]. It is a fair criteria for the sake of the overall convergence speedup.

Clearly enough, with (6), the priority of each patch in the waiting queue depends on the following four factors: 1) its emission; 2) its size; 3) its reflectivity; and 4) its location in the environment. Accordingly, we can roughly divide the environment patches into three categories: 1) the light patches; 2) the global patches; and 3) the local patches.

The light patches are the light sources. They are often extremely bright and will be processed in the very first stage of the iterative process. But after their first shootings, they will degenerate to global or local patches which will be described below. Light patch's function, in general, is for the direct global illumination of the environment.

The global patches are usually the enclosure of the environment, for instance, the wall patches in our test environment. They are often relatively large, with a wide spectrum distribution, and in a position which can overlook most of the environment. Therefore, usually, a global patch can not only shoot more light energy to the environment when a candidate is to be selected but also gather more light energy from the environment for the next shooting than that of non-global patch. Global patch's function, in general, is for the secondary global illumination of the environment.

The local patches comprise the rest of the environment. They are often relatively small, with a narrow spectrum distribution which makes the environment vivid, and in a position visible to a limited portion of the environment. A local patch's light energy interaction with the environment is usually relatively small, but it could have a significant influence on its neighboring patches. Its shooting chance often comes in the later iteration stages. Local patch's function, in general, is for the local illumination of the environment.

Although the patch categorization is somewhat vague and environment dependent, in real environments this kind of difference among patches (and therefore the patterns of the percentage distribution of the patch shooting times depicted in Table 1) are very common. So we have two major observations: first, the post-shooting gathering of a class of (global) patches could be substantial; second, this pattern of heavily repeated shootings of the same class of the patches could slow down the convergence dramatically since the form-factors usually have to be calculated on-the-fly in the progressive refinement method to render complex environments [5].

One might argue that we could always subdivide the so-called global patches in smaller subpatches to break up the unbalanced iterations. But doing this will usually add patches which are unnecessary for the same solution precision and thus slow down the entire iterative process. One might also suggest that we could use the maximum light energy in a single wavelength as the criteria for the shooting patch selection. By doing so, the local illumination effects might appear earlier. But, since generally less light energy is processed in each iteration step, it will also slow down the entire iterative process.

We have also noted that the number of iterations used for many images with the progressive refinement radiosity method is quite arbitrary, usually greater than the number of the light patches but far less than the number of the global patches estimated. This, in our view, makes no sense. Since there is a class of global patches and their unprocessed light energies are usually compatible, processing some but estimating the rest by the ambient term seems meaningless. We believe a better strategy should be: 1) light patches + ambient term (local illuminating model); 2) light patches + global patches + ambient term (medium illumination model); or 3) light patches + global patches + local patches (global illuminating model).

3.2 Gathering and Shooting

In some applications, for instance, architecture design, a rough overview of the environment with some kind of soft shadows might be good enough. However, accurate solutions could be critical for many other engineering applications, such as radiation computation in thermal engineering, global illuminating simulation in machine perception, and of course photorealistic image synthesis in computer graphics. Clearly, both the Gauss-Seidel gathering iteration and the progressive shooting iteration are capable of producing accurate simulations, if enough time and memory are available. The problem is, as always, the efficiency.

Suppose the whole set of $O(n^2)$ form-factors have been calculated and can be freely accessed. We know that the gathering iteration and the shooting iteration are just two different kinds of iterative processes for solving the matrix equation (2). Though it has been proved that the shooting iteration always outperforms the gathering iteration [10], we should say they are both practically effective iterative methods. As a matter of fact, the Gauss-Seidel method is one of the most popular iterative approaches in solving simultaneous linear equations.

Imagine a virtual environment with n patches. Suppose all patches were identical in this ideal environment. They have the same size, reflectivity, unprocessed light energy, and form-factors to each other. It is not difficult to prove that, after every n steps, both the gathering iteration and the shooting iteration will come up with the same solution, with regard to a patch reordering. That is, basically the two iterations have the same convergence speed. In practice, however, since environments are always unbalanced, the shooting iterative process will have a faster convergence since it processes the brightest emission first. The more unbalanced the environment is, the more favorable the shooting iteration versus the gathering iteration. This is exactly the reason that the shooting process converges much much quickly in the stage of the light source shootings than the gathering process does with the same number of iterations. However, as the iterative process continues, the imbalance of the unprocessed light energy distribution decreases, reducing the advantage of shooting versus gathering. In the gathering process, the update to one patch radiosity in each iteration is relatively significant, since light energy is gathered from every other patch in the environment. In the shooting process, every patch other than the shooting patch will be updated relatively trivially, since only one patch is shooting light energy.

To do better, we can actually combine both gathering and shooting iterations. In the progressive refinement process, suppose patch i is chosen as the current shooting patch which holds the maximum unprocessed patch light energy $\Delta B_i A_i$ in the environment. In standard shooting iteration, the formula (3) is directly applied to update the environment patch radiosities distributed by patch i [5]. We have noticed, however, with the reciprocity relationship for form-factors

$$A_i F_{ij} = A_j F_{ji} \tag{7}$$

the unprocessed light energy of every other patch in the environment $\Delta B_j A_j$ $(j = 1, 2, \ldots, n)$ which will be shot in later iterations, can be pre-shot to patch i in the current step (we do not add

$j \neq i$ since $F_{ii} = 0$). Furthermore, with the same set of form-factors, this pre-gathered light energy of patch i can also be shot back to the environment in the same iteration step as part of patch i's unprocessed light energy.

More precisely, in an iteration step of patch i,

$$\Delta B_j A_j F_{ji}, \qquad j = 1, 2, \ldots, n \tag{8}$$

or, equivalently,

$$\Delta B_j A_i F_{ij}, \qquad j = 1, 2, \ldots, n \tag{9}$$

is the energy leaving patch j ($j = 1, 2, \ldots, n$) reaching patch i; while

$$(\Delta B_i + \rho_i \sum_{j=1}^{n} \Delta B_j F_{ij}) A_i \tag{10}$$

is the total amount to be shot from patch i to the environment.

We should be careful when patch j ($j = 1, 2, \ldots, n$) later gets its turn, since a certain amount of its unshot light energy has already been pre-shot to patch i. Apparently, a record is needed. Since there are n patches thus $n \times n$ interactions, an immediate solution would be a two-dimensional array. We denote P_{ij} to be the pre-shot radiosity leaving patch i reaching patch j. At the beginning of the algorithm, we initialize P_{ij} to zero. The iteration can be described in the following pseudo-code:

```
/* initialization */
select shooting patch $i$;
calculate form-factors $F_{ij}$;
$\Delta S = \Delta B_i$;
/* gathering */
for each patch $j$:
    $\Delta S = \Delta S + \rho_i(\Delta B_j - P_{ji})F_{ij}$;
    $P_{ji} = \Delta B_j$;
/* shooting */
$B_i = B_i + \Delta S$;
$\Delta B_i = \Delta B_i + \Delta S$;
for each patch $j$:
    $tmp = \rho_j(\Delta B_i - P_{ij})F_{ij}A_i/A_j$;
    $\Delta B_j = \Delta B_j + tmp$;
    $B_j = B_j + tmp$;
    $P_{ij} = 0$;
$\Delta B_i = 0$;
```

It should be noted that in the gathering procedure $\Delta B_j - P_{ji}$ must be used instead of ΔB_j because patch j may have not been shot in the interim between two shootings of patch i.

Direct illumination from the light sources usually occupies a significant percentage of overall light energy distribution in the global illumination. After light patches been processed, in each iteration, the light energy being gathered should also be a significant percentage of the shooting patch's total unprocessed energy since it comes from every patch in the environment from which the current shooting patch is visible. In each iteration, the current patch not only updates itself (as it does in the gathering iterative process) but also shoots more light energy than it would in the conventional progressive refinement method. As a result, a solid convergence speedup should be expected.

Since the shooting patch has gathered all unprocessed light energy it can get in the current iteration stage, its light energy gathering in certain number of later interactions should be quite trivial. Therefore, the priorities of other patches are relatively increased and the phenomenon of the unbalanced percentage distribution of the patch shooting times should be eased. This is especially crucial to those local patches so that the local illumination effects should show up earlier.

Unfortunately, this is a $O(n^2)$ algorithm in terms of memory. The P_{ij} matrix records the latest history of pre-shot interactions between each pair of patches in the environment. In each iteration, one row and one column relating to the shooting patch i will be updated. However, we notice that, in a certain number of iterations, the changes in P_{ij} are rather trivial. We can thus consider a more concise record of history of the pre-shot interactions.

With a predefined integer, δ, called the gathering-shooting interval, we divide the whole series of iteration processes into a number of intervals. In each interval, there are δ iterations. In each iteration of the same interval, the pre-shot light energy of each patch is set to be a constant which only depends on the records of unshot light energies of previous intervals. In other words, the gathering process neglects the updates of unshot light energy in the current interval and leaves them for consideration in the next interval.

We denote $T_{k,i}$ to be the amount of increased radiosity of patch i in the kth interval after δ iterations, and I_i to be an integer that is the index of latest interval in which patch i has been shot. Both $T_{0,i}$ and I_i will be initialized to zero at the beginning of the algorithm.

Suppose the current iteration interval is the $k+1$. In each iteration, as usual, we first select the patch with maximum unprocessed light energy as our shooting patch, say it is patch i.

The gathering procedure in the original pseudo-code should be replaced by:

```
/* gathering */
for each patch $j$:
    $last = \max(I_i, I_j)$;
    $\Delta S = \Delta S + \rho_i(\sum_{l=last}^{k} T_{l,j})F_{ij}$;
```

In the above, we first determine the index of last interval in which patch i and patch j have a light energy interaction. We then decide how much radiosity should be gathered from patch j which is the summation of $T_{l,j}$ from the end of the interval of their last interaction to the end of last interval.

The shooting procedure in the original pseudo-code should be replaced by:

```
/* shooting */
$B_i = B_i + \Delta S$;
$\Delta B_i = \Delta B_i + \Delta S$;
```

```
for each patch $j$:
    $tmp = \rho_j(\Delta B_i - \sum_{l=I_i}^{I_j-1} T_{l,i})F_{ij}A_i/A_j$;
    $\Delta B_j = \Delta B_j + tmp$;
    $B_j = B_j + tmp$;
    $T_{k+1,j} = T_{k+1,j} + tmp$;
$\Delta B_i = 0$;
$T_{k+1,i} = 0$;
$I_i = k + 1$;
```

In the above, we first update the radiosity of the shooting patch by the radiosity just pre-gathered. We then decide how much radiosity should be shot from patch i to patch j which is its current unshot radiosity minus the summation of $T_{l,i}$ from the end of interval of patch i's last shooting to the beginning of patch j's last shooting.

The main idea is clear: first, the algorithm should gather the majority of the unprocessed light energy from the environment, that is, from previous intervals; second, the algorithm should avoid repeated light energy interactions. We have realized both of them efficiently. The total storage of $T_{k,i}$ $(i = 0, 1, 2, \ldots, n)$ depends on the value of δ and the number of total iterations. In practice, it should be almost linear to the number of patches in the environment.

Figure 4(b) and Table 1(b) illustrates the convergence behaviors of this gathering and shooting iteration in the test environment. The gathering-shooting interval, δ, is set to be 100. Obviously, the improvement is sound, especially to the final accurate solution.

4 Results

The gathering and shooting progressive refinement radiosity method described in this paper is implemented in C on a Silicon Graphics Indigo Elan workstation. The form-factors are analytically calculated [1] and vertex radiosities [14] are progressively refined in each iteration with the adaptive patch-element subdivision technique [4] [5].

The gathering and shooting method and the standard shooting method was compared with our test environments. The final converged solutions were used as references to measure the accuracy at refinement stages in both methods.

Figure 5 contains an environment which was initially discretized into 1,080 patches and further subdivided into 25,536 elements in the final converged solution. The gathering-shooting interval was set to 100 iterations. With the hemi-cube resolution of 200 by 200, the computation of a single iteration took about two seconds. The two graphs of histograms of the patch vertex radiosity accuracy in Figure 6 illustrate the convergence behaviors of the standard shooting method and our gathering and shooting method, respectively.

Figure 7 contains an environment which was initially discretized into 1,892 patches and further subdivided into 36,419 elements in the final converged solution. The gathering-shooting interval was set to 100 iterations. With the hemi-cube resolution of 400 by 400, the computation of a single iteration took about four seconds. The two graphs of histograms of the patch vertex radiosity accuracy in Figure 8 illustrate the convergence behaviors of the standard shooting method and our gathering and shooting method, respectively.

In this paper, we choose following inequality as the termination criteria for the final convergence:

$$\sum_{i=1}^{n} \Delta B_i A_i < \epsilon \tag{11}$$

where ΔB_i is the unprocessed light energy and A_i is the area of patch i. In other words, the iteration process terminates when the total unprocessed light energy in the environment is less than a predefined error bound ϵ.

Empirically, based on the test environments, this gathering and shooting method requires approximately twice as many iterations as there are patches in the environment for an accurate radiosity simulation. Surely it is environment dependent. Further experience and mathematical analysis are needed for a thorough understanding of the refinement process of this method.

Since in our gathering and shooting method, part of the radiosity of the shooting patch is pre-accumulated from the rest of the environment, in general, there will be a radiosity discontinuity among the shot patches and their neighbors in the intermediate solutions. However, this radiosity discontinuity will be gradually smoothed as the refinement process converges. When we preview an intermediate solution, those discontinuous brighter spots indicate which patches have been processed. If a continuous intermediate shading is required, for display purpose, we can always discard the pre-accumulated part of the patch radiosities.

5 Conclusions

Based on our test environments, we have conducted a qualitative and quantitative study of the convergence behavior of the conventional progressive refinement radiosity method. We have also presented a new progressive refinement approach which is an integration of the light energy gathering iterative process in the standard full matrix radiosity method and the light energy shooting iterative process in the conventional progressive refinement radiosity method.

According to our experimental results, we made the following observations of the conventional progressive refinement method:

- Though the refinement procedure, in general, converges very quickly to a certain degree in the first several iterations, further improvement can be extremely slow. The visual and statistical differences between solutions in different iteration stages and the final converged solution are evident. Particularly, those of the local illumination effects, such as surface-to-surface color bleeding and shading in the shadow areas, usually appear in later iteration stages.

- In the refinement procedure, the order and magnitude of the light energy contribution of each individual patch to the global illumination in the environment can vary significantly. Accordingly, the environment patches can be roughly divided into three categories: the light patches, the global patches, and the local patches, depending on their emissions, sizes, reflectivities, and locations. Those global patches which are often relatively large and have a wide spectrum distribution, in general, not only can shoot more light energy to the environment when a candidate is to be selected but also can gather more light energy from the environment

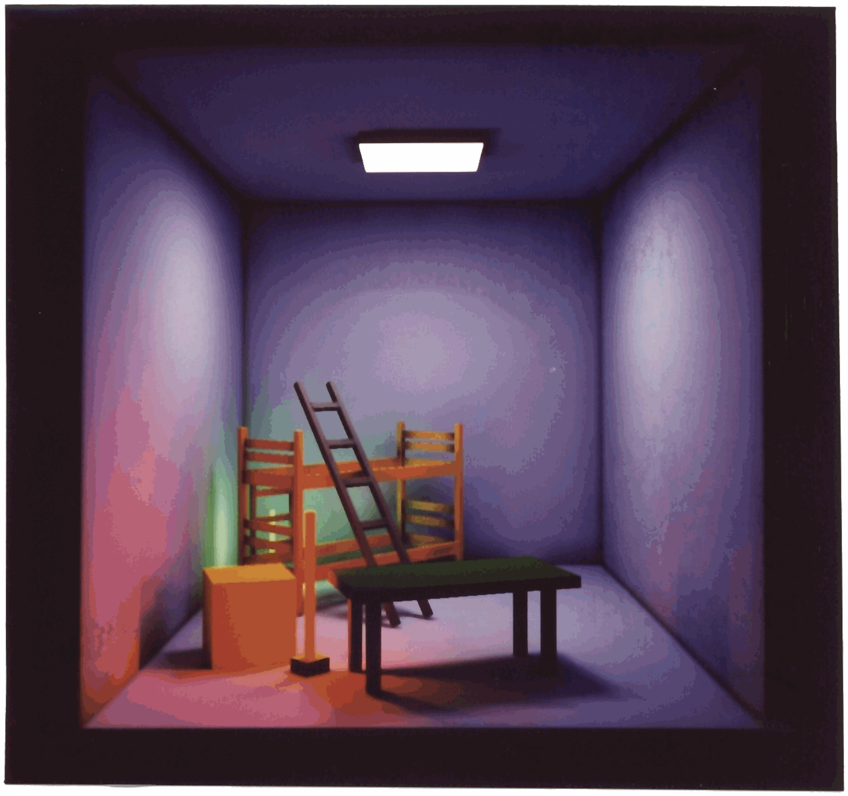
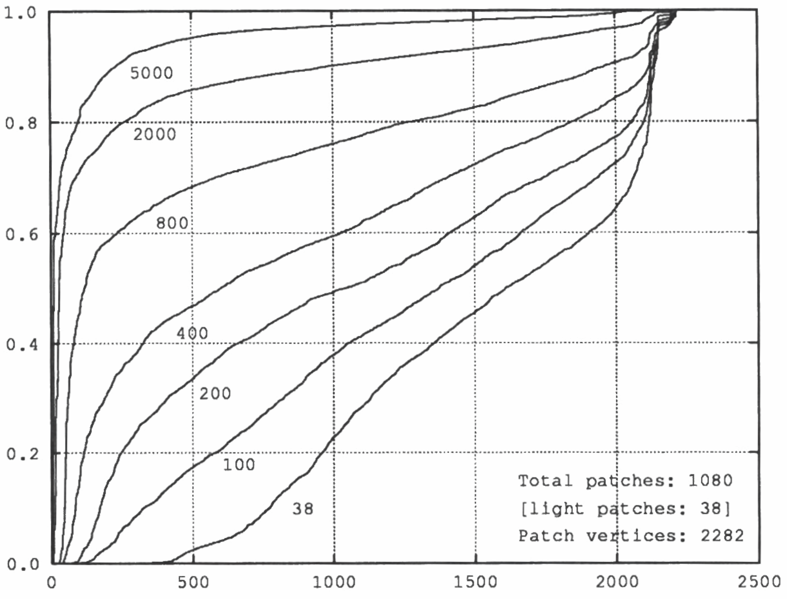
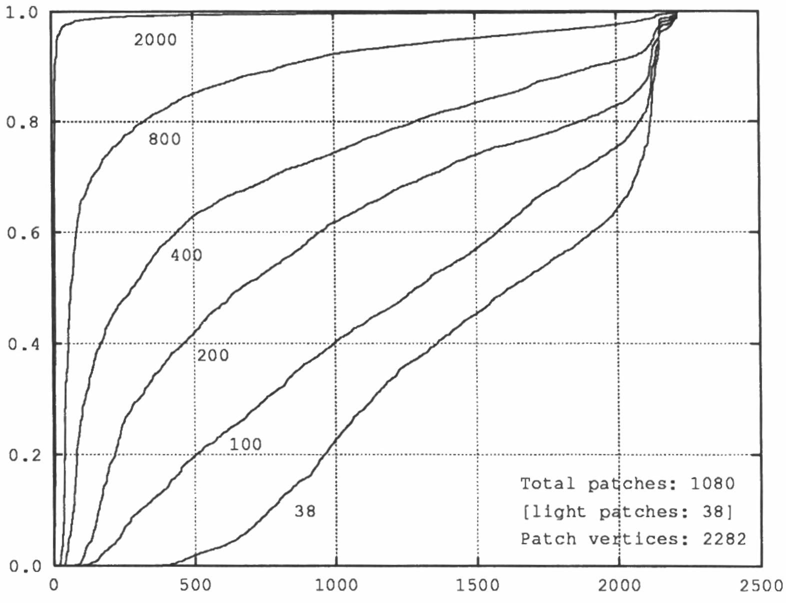


Figure 5: Kindergarten impression — initial subdivision: 1,080 patches; converged solution: 25,536 elements; hemi-cube resolution: 200×200; gathering-shooting interval: 100 iterations.



(a) standard shooting



(b) gathering and shooting

Figure 6: Kindergarten impression — histograms of the patch vertex radiosity accuracy.

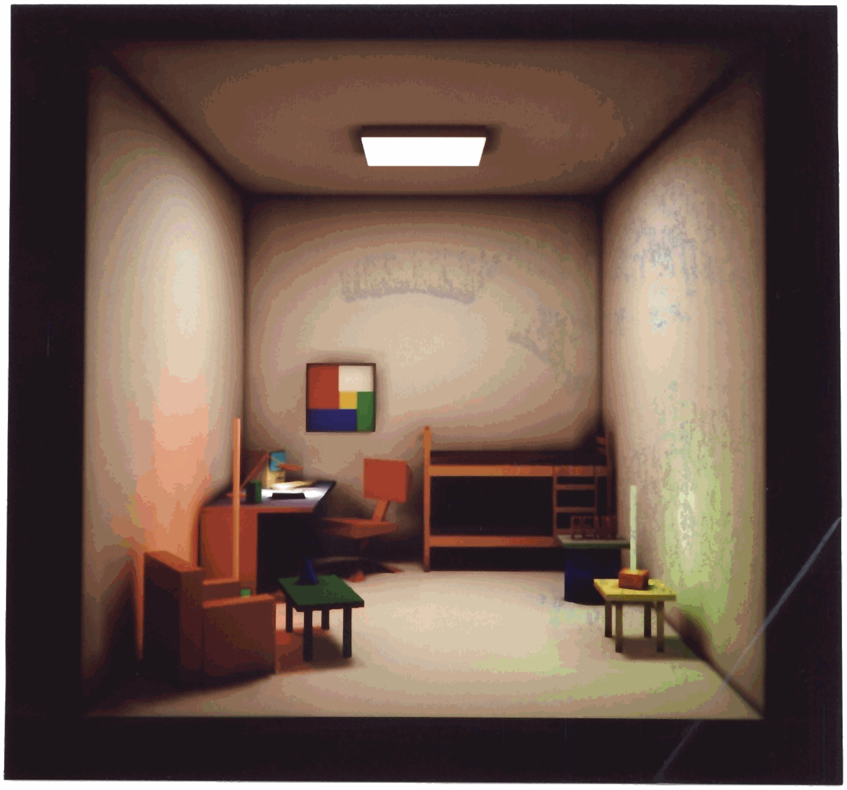
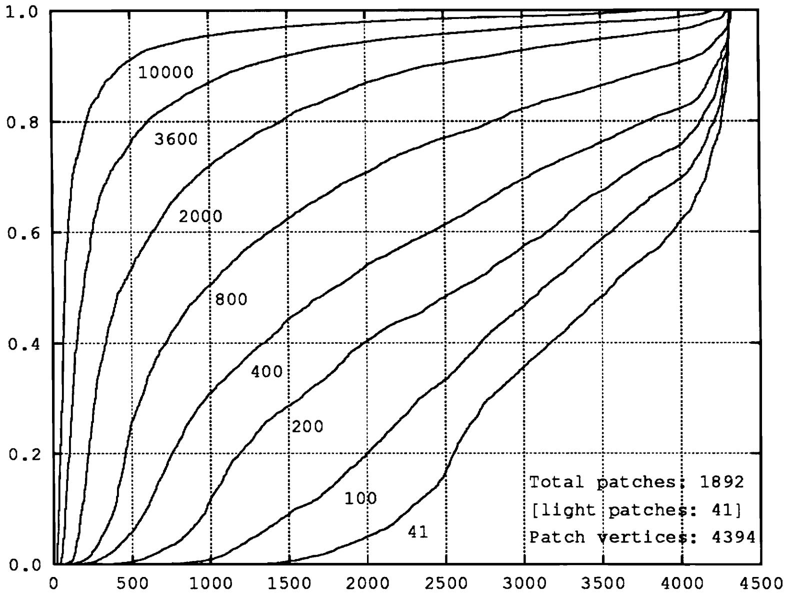
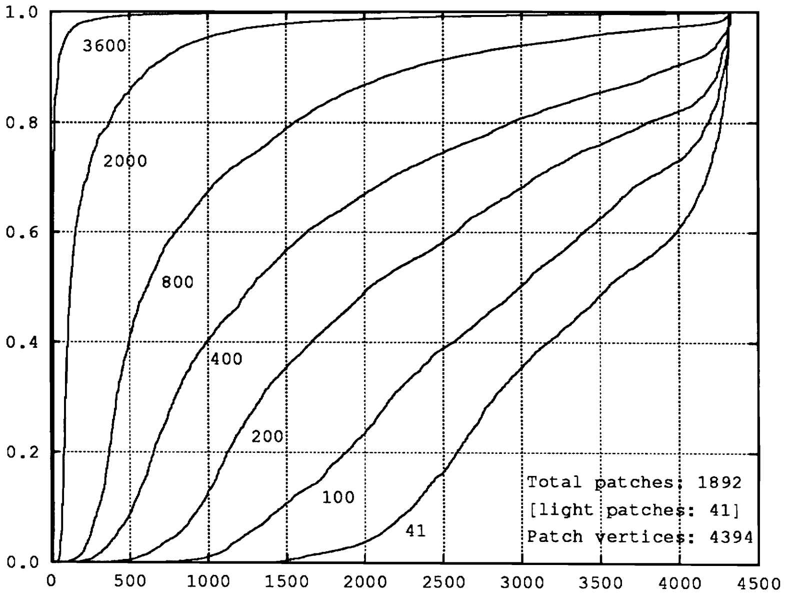


Figure 7: Dormitory room — initial subdivision: 1,892 patches; converged solution: 36,419 elements; hemi-cube resolution: 400×400; gathering-shooting interval: 100 iterations.



(a) standard shooting



(b) gathering and shooting

Figure 8: Dormitory room — histograms of the patch vertex radiosity accuracy.

for the next shootings than those of local patches which are often relatively small and have a narrow spectrum distribution. As a result, it is not unusual that some global patches are processed quite a few times while some local patches still have not yet got a chance.

Our goal is to accelerate the convergence of this iterative process. Our solution is a new iterative structure: gathering and shooting. The idea is try to process as much unprocessed light energy as we could in each iteration step so as to postpone the next shooting of the current patch as long as possible. At the same time, by doing so, we have increased the relative priorities of other patches in the waiting queue, especially those local patches. Based on a concise record of the history of the unprocessed light energy distribution in the environment, we have proposed a new progressive refinement algorithm and data structure with revised gathering and shooting procedures. As a result, though with little additional computation and memory usage comparing to the conventional progressive refinement radiosity method, a solid convergence speedup is achieved. This gathering and shooting approach extends the capability of the radiosity method in accurate and efficient simulation of the global illuminations of complex environments.

Acknowledgments

The authors wish to thank Qunsheng Peng for his critical comments during the writing of this paper and Jianmin Zhao for several helpful discussions. This research was conducted in the Computer Graphics Research Laboratory at the University of Pennsylvania and partially supported by a Dean's Fellowship from the University of Pennsylvania; ARO Grant DAAL03-89-C-0031 including participation by the U.S. Army Research Laboratory (Aberdeen), Natick Laboratory, the Army Research Institute, and NASA Ames Research Center; U.S. Air Force DEPTH contract through Hughes Missile Systems F33615-91-C-0001; DARPA through General Electric Government Electronic Systems Division; and NSF CISE Grant CDA88-22719.

References

[1] D. R. Baum, H. E. Rushmeier, and J. M. Winget, "Improving radiosity solutions through the use of analytically determined form-factors," *ACM Computer Graphics*, 23(3):325-334, July 1989.

[2] S. E. Chen, "Incremental radiosity: An extension of progressive radiosity to an interactive image synthesis system," *ACM Computer Graphics*, 24(4):135-144, August 1990.

[3] M. F. Cohen and D. P. Greenberg, "The hemi-cube: A radiosity solution for complex environments," *ACM Computer Graphics*, 19(3):31-40, August 1985.

[4] M. F. Cohen, D. P. Greenberg, D. S. Immel, and P. J. Brock, "An efficient radiosity approach for realistic image synthesis," *IEEE Computer Graphics and Applications*, 6(3):26-35, March 1986.

[5] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg, "A progressive refinement approach to fast radiosity image generation," *ACM Computer Graphics*, 22(4):75-84, August 1988.

[6] D. W. George, F. X. Sillion, and D. P. Greenberg, "Radiosity redistribution for dynamic environments," *IEEE Computer Graphics and Applications*, 10(4):26-34, July 1990.

[7] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," *ACM Computer Graphics*, 18(3):213-222, July 1984.

[8] P. Hanrahan and D. Salzman, "A rapid hierarchical radiosity algorithm for unoccluded environments," *Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, 151-171, June 1990.

[9] P. Hanrahan, D. Salzman, and L. Aupperle, "A rapid hierarchical radiosity algorithm," *ACM Computer Graphics*, 25(4):197-206, July 1991.

[10] P. Hanrahan, Private communications, 1992.

[11] P. S. Heckbert, "Simulating global illumination using adaptive meshing," Ph.D. dissertation, University of California, Berkeley, June 1991.

[12] R. Siegel and J. R. Howell, *Thermal Radiation Heat Transfer*, Washington: Hemisphere Publishing Corporation, 1981.

[13] G. Strang, *Linear Algebra and its Applications*, San Diego: Harcourt Brace Jovanovich, Publishers, 1988.

[14] J. R. Wallace, K. A. Elmquist, and E. A. Haines, "A ray tracing algorithm for progressive radiosity," *ACM Computer Graphics*, 23(3):315-324, July 1989.