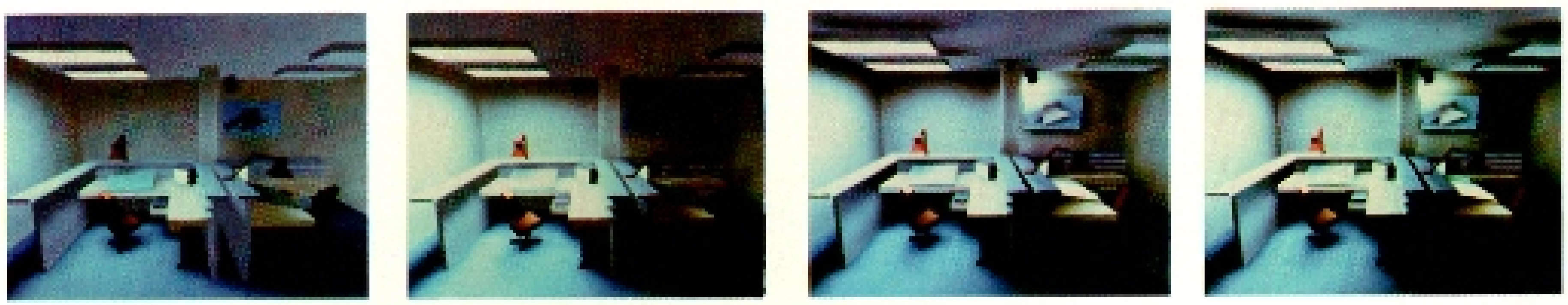
A Progressive Refinement Approach to Fast Radiosity Image Generation

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.

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:

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

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],

$$F_{ij} A_i = F_{ji} A_j \tag{2}$$

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

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

or in matrix form:

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

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:

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

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:

$$B_i \text{ due to } B_j = \rho_i B_j F_{ij} \qquad (6)$$

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

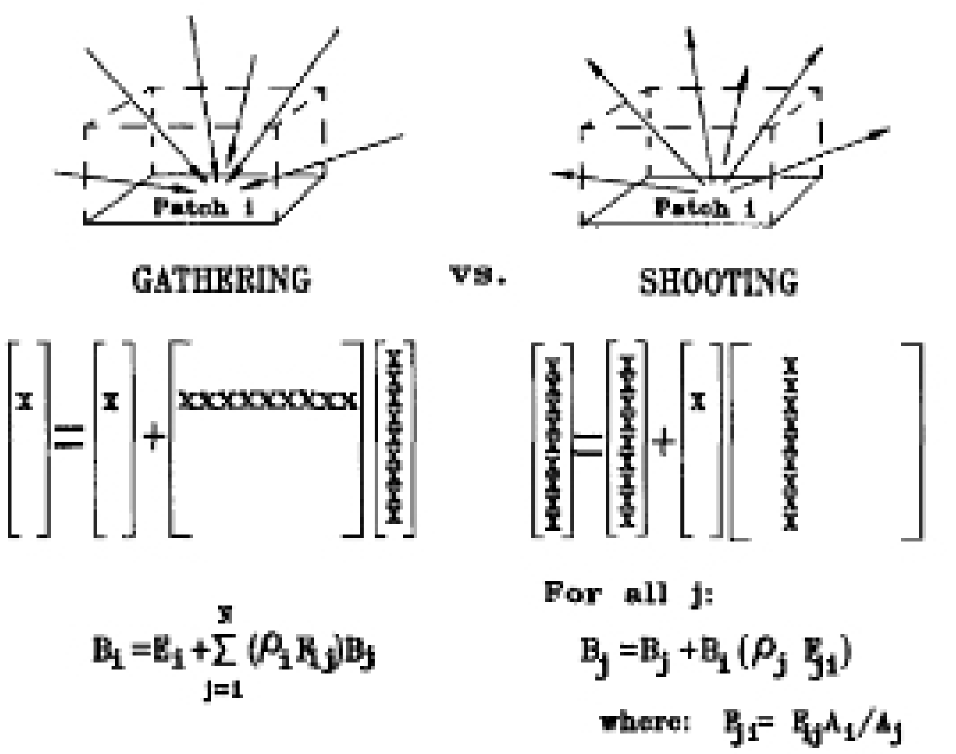


Figure 1: Gathering vs. Shooting

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.

this relationship. The contribution of the radiosity from patch i to the radiosity of patch j is:

$$B_j \text{ due to } B_i = \rho_j B_i F_{ij} A_i / A_j \qquad (7)$$

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

$$\text{For all patches } j : B_j \text{ due to } Bi = \rho_j B_i F_{ij} A_i / A_j \qquad (8)$$

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.

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