An Importance-Driven Radiosity Algorithm

Brian E. Smits
James R. Arvo
David H. Salesin

Program of Computer Graphics
Cornell University
Ithaca, NY 14853

Abstract

We present a new radiosity algorithm for efficiently computing global solutions with respect to a constrained set of views. Radiosities of directly visible surfaces are computed to high accuracy, while those of surfaces having only an indirect effect are computed to an accuracy commensurate with their contribution. The algorithm uses an adaptive subdivision scheme that is guided by the interplay between two closely related transport processes: one propagating power from the light sources, and the other propagating *importance* from the visible surfaces. By simultaneously refining approximate solutions to the dual transport equations, computation is significantly reduced in areas that contribute little to the region of interest. This approach is very effective for complex environments in which only a small fraction is visible at any time. Our statistics show dramatic speedups over the fastest previous radiosity algorithms for diffuse environments with details at a wide range of scales.

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

Additional Key Words: importance functions, adaptive meshing, hierarchical radiosity, adjoint transport equation, global illumination, view-dependence.

1 Introduction

View-independent global radiosity algorithms have two major drawbacks: they oversolve globally and undersolve locally. The algorithms oversolve globally in that they attempt to compute radiosities to a uniform precision throughout the environment—even on surfaces hidden from all useful points of view. They undersolve locally in that a single global radiosity solution is seldom adequate under close inspection—local effects such as shadowing and color bleeding among small objects are often lost when the radiosity of the entire environment has to be computed to a uniform precision. Thus, the utility of a purely view-independent solution diminishes as the environment complexity or the required accuracy increases.

To address this problem, several adaptive meshing schemes have been devised to increase the level of approximation where significant intensity gradients occur [3, 4, 6]. These algorithms can achieve good results with fewer surface elements. For complex environments, however, the cost of creating a fine mesh to capture every illumination detail, whether visible or not, may still be far too high.

In practice, radiosity implementations often rely on some form of additional intervention by the user in order to handle complex scenes. For example, in order to keep the total number of surface elements small, the user may need to supply meshing hints based on the anticipated set of views. This approach has the advantage of saving work in areas that are unimportant to the final image. However, it also tends to sacrifice global accuracy, as there is no obvious way for the user to predict the level of meshing required for distant objects to have their proper effects on the visible parts of the scene.

The converse of this problem arises during the solution process: which of the vast number of interactions in a complex environment are significant enough to evaluate? Consider, for example, computing a radiosity solution for a large building. In principle, light leaving a small surface on one floor could reach any other floor by some circuitous path of stairwells and corridors. A radiosity algorithm that accurately computed all such interactions would be highly impractical.

Hanrahan, Salzman, and Aupperle [12] have recently proposed a brightness-weighted hierarchical radiosity algorithm that goes a long way toward resolving this difficulty. The hierarchical algorithm focuses effort on the significant energy transfers, quickly approximating the insignificant interactions. However, because the algorithm is still view-independent, it does more work than necessary in complex environments when only a single view or set of views is required. Indeed, it is not difficult to construct models for which even this rapid algorithm is impractically slow.

One way to make radiosity practical for complex environments is to incorporate a notion of view-dependence. For instance, we might first use some form of visibility preprocessing to determine the surfaces that are directly visible [1, 19]. To compute the radiosities of these surfaces, all surfaces contributing energy must be taken into account. But these contributing surfaces, in turn, receive energy from surfaces that are still further away. However, in general the effect of distant interactions will be less important to the visible scene. In short, to make effective use of view-dependence it is necessary to consider *all* potential interactions among surfaces, but to compute each interaction only to an appropriate level of accuracy.

In this paper, we describe such an algorithm: an extension to hierarchical radiosity that refines the interactions contributing the most

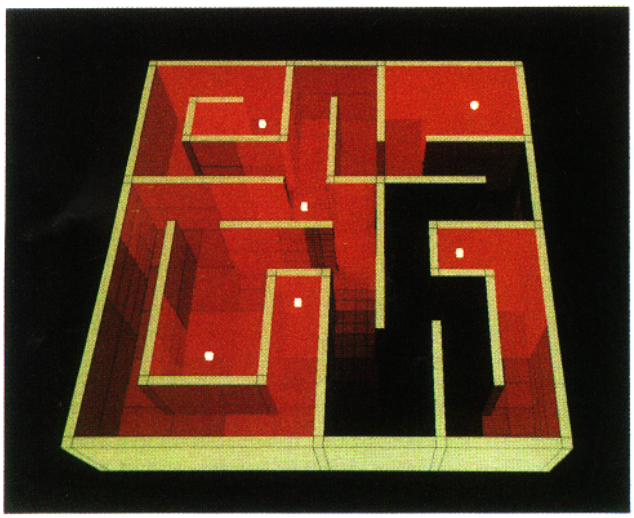

Figure 1: A radiosity solution for a labyrinth.

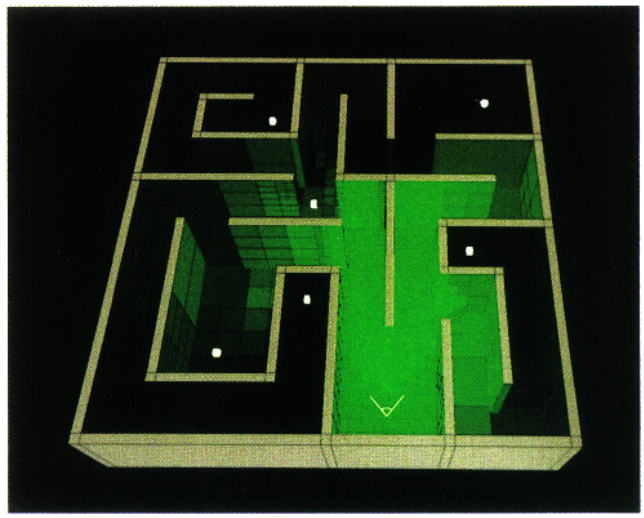

Figure 2: An importance solution for the same model.

error to the view-dependent solution. The algorithm makes use of *importance functions*, which have been studied extensively in neutron transport theory [15, 16]. In our context, importance functions describe how the radiosity originating at a given patch influences the visible surfaces. The algorithm we propose combines estimates of importance and radiosity to drive the global solution, allowing it to exploit view-dependent information as part of an adaptive refinement scheme.

2 Importance

Illumination algorithms can be divided into two categories: those that simulate the propagation of all light throughout an environment, typified by radiosity algorithms; and those that simulate only the light reaching the eye, typified by ray tracing. These two strategies are in some sense dual: the former simulate the process of photons emanating from sources of light, while the latter trace rays that emanate from the eye, but behave very much like photons in every other respect. The two strategies have advantages for modeling different modes of light scattering. Indeed, the many bidirectional ray tracing and multi-pass radiosity methods suggested in the last few years [2, 5, 13, 17, 18, 20] exploit the complementary nature of these two processes.

An analogous duality appears in neutron transport theory, where equations similar to those of radiative transfer are used to predict neutron flux [8]. If only the flux impinging on a small receiver is required, it is typically the *adjoint* of the original transport equation that is solved, in effect reversing the direction of neutron migration back toward the source. This strategy is closely related to the various ray tracing approaches for global illumination [14, 22].

The efficiency of these "backward" methods depends on the ability to quickly find paths leading back to the source. But determining these paths is equivalent to solving the transport equation in the forward direction. Thus, solving the transport equation in one direction would seem to require its prior solution in the other direction. However, through variational methods, nuclear engineers have been able to use this interdependence to advantage. These methods allow approximate solutions to the two transport equations to be combined, yielding an overall solution with higher accuracy than either component alone [8, 16].

The effectiveness of these techniques suggest that in the realm of global illumination it may also be possible to exploit the dual nature of light transport more effectively than has previously been recognized. Rather than using these dual processes in multiple passes to simulate different light scattering modes, we can instead use the two processes together—solving them simultaneously—in order to produce an accurate result more quickly.

As an illustration of our approach, consider the labyrinth model depicted in Figure 1. The model is illuminated by several light sources, shown in white. The radiosity solution due to these lights is shown in red. In an analogous fashion, the camera can also "illuminate" the scene with a new quantity, "importance." Figure 2 depicts the importance solution due to the camera in green.

Our algorithm uses radiosity and importance together to accelerate the radiosity solution for the visible scene. It does this by refining estimates of the transport equations most where the interaction of radiosity and importance is highest. Thus, in Figure 3 the algorithm uses the finest mesh for the parts of the scene that are yellow, somewhat less meshing for the parts that are orange, and even less for the parts that are red, green, or black—i.e., in regions of little importance, little brightness, or little of either.

The remainder of this section shows how the duality of radiosity and importance can be established more formally, and develops some mathematical tools for using these quantities together to drive a hierarchical algorithm.

2.1 Duality of radiosity and importance

Suppose we have a linear operator $\mathbf{L}$ governing some transport process, and a *source* term S. Then the transport equation and its adjoint can be written as follows:

$$\mathbf{L}\Phi = S \quad (1)$$

$$\mathbf{L}^*\Psi = R \quad (2)$$

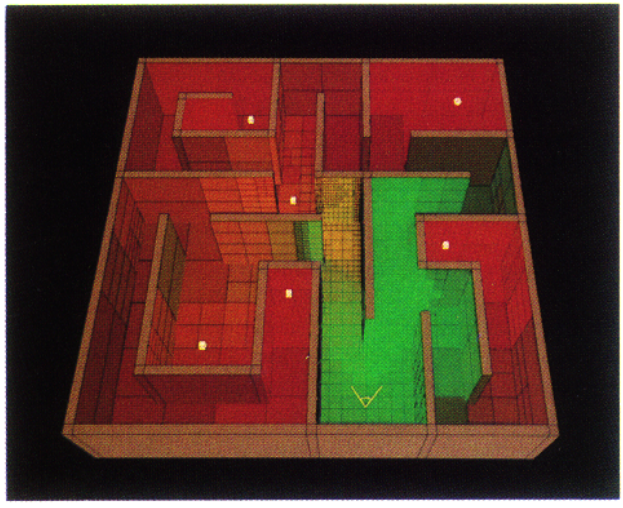

Figure 3: The radiosity and importance solutions together.

where * is the adjoint operator, and R is a *receiver*, dual to the source. In the context of radiosity, Φ is a vector of radiosities, S a vector of emittances, and $\mathbf{L}$ a discrete approximation to the continuous transport operator. Because $\mathbf{L}$ is a matrix of real numbers, $\mathbf{L}^* = \mathbf{L}^T$

In the adjoint equation (2), the vector Ψ is the *importance* vector dual to Φ. To understand the significance of R and Ψ, suppose that we wish to compute a scalar function of the radiosity solution $v(\Phi)$, rather than the entire solution Φ. For example, the function v might compute the average radiosity visible through a certain aperture, such as a pixel or the entire image plane. Since any linear function of Φ can be expressed as an inner product, we can write $v(\Phi)$ as the product $R^T\Phi$, where R_i gives the contribution of Φ_i to the scalar function v. We can then use equations (1) and (2) together to derive an alternate formulation of $v(\Phi)$ with respect to Ψ:

$$v(\Phi) = R^T\Phi = (\mathbf{L}^T\Psi)^T\Phi = \Psi^T\mathbf{L}\Phi = \Psi^T S. \qquad (3)$$

Thus, each element Ψ_j of the importance vector Ψ gives the contribution made by a unit of emittance at patch j to the scalar function $v(\Phi)$. In other words, if v is chosen for a particular view, then Ψ_j gives *the fraction of radiosity emitted at patch j that ultimately reaches the eye.*

Note that since the vector R plays a role dual to S in equations (1) and (2), we can think of R as describing the initial "emittance of importance." In this sense, we can think of every patch j as having associated with it both a steady-state radiosity Φ_j coming from all patches in the environment but originating at the lights S, and also a steady-state importance Ψ_j coming from all patches but originating at R, determined by the eye (Figure 4).

While radiosity and importance are similar in many respects, the two quantities are not exactly the same. Radiosity is a flux density, measured in watts per meter-squared, whereas importance is defined as a fraction, and is therefore a dimensionless quantity. This distinction has ramifications when we distribute the two quantities in a hierarchical system, as described in Section 2.3.

2.2 Importance-driven refinement

For most transport equations, we must use a discrete approximation to the exact transport operator. The global illumination problem is no exception; the matrix approximation of $\mathbf{L}$ generally contains imprecise form-factor estimates and other assumptions that introduce error. However, the approximate matrix can be expressed as a perturbation of the exact operator, and refined using estimates of radiosity and importance, as follows.

Let $\tilde{\mathbf{L}}$ be an approximation to $\mathbf{L}$, with $\tilde{\mathbf{L}} = \mathbf{L} + \Delta\mathbf{L}$, and let $\tilde{\Phi}$ be the solution to the corresponding transport equation, which is an approximation to the exact radiosity vector Φ. Assuming exact emittances S, the approximate radiosity transport equation can be written as

$$\tilde{\mathbf{L}}\tilde{\Phi} = S. \qquad (4)$$

which is equivalent to

$$\mathbf{L}\tilde{\Phi} = S - \Delta\mathbf{L}\tilde{\Phi}. \qquad (5)$$

Thus, we can rewrite the approximate transport equation with exact emittances S in terms of an exact transport operator with perturbed emittances $S - \Delta\mathbf{L}\tilde{\Phi}$.

Since the importance of a patch describes the fraction of emitted radiosity that reaches the eye, we can now derive an expression for how the error in the emittances affects the view-dependent function v. Writing the visible error as $v(\Phi - \tilde{\Phi})$ and using equations (3) and (5), we have:

$$\begin{aligned} v(\Phi - \tilde{\Phi}) &= R^T\Phi - R^T\tilde{\Phi} \\ &= \Psi^T S - \Psi^T\mathbf{L}\tilde{\Phi} \\ &= \Psi^T S - \Psi^T(S - \Delta\mathbf{L}\tilde{\Phi}) \\ &= \Psi^T\Delta\mathbf{L}\tilde{\Phi}. \end{aligned} \qquad (6)$$

Thus, the quantity $\Psi^T\Delta\mathbf{L}\tilde{\Phi}$ is the error that the approximations to $\mathbf{L}$ and Φ contribute to the view-dependent function.

Unfortunately, in practice it is not possible to compute the importance vector Ψ exactly, since computing the global importance solution is as difficult as computing the global radiosity solution. However, we can compute an approximation $\tilde{\Psi}$ to the exact importance vector Ψ, and refine it as we refine $\tilde{\mathbf{L}}$ and $\tilde{\Phi}$. In this case, we can use

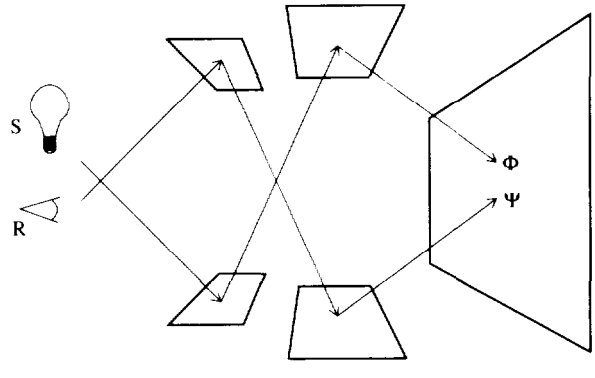

Figure 4: The duality of radiosity and importance

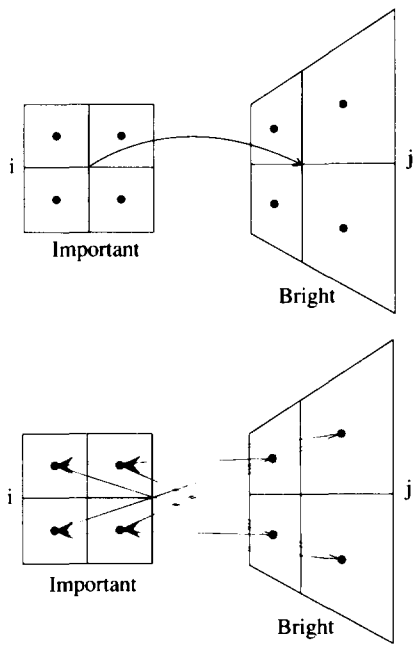

Figure 5: Transporting radiosity and importance in a hierarchy.

the quantity $\widetilde{\Psi}^{\mathrm{T}}\Delta\mathbf{L}\widetilde{\Phi}$ as a close approximation to the actual error $\Psi^{\mathrm{T}}\Delta\mathbf{L}\widetilde{\Phi}$.

To incorporate this insight into a hierarchical radiosity algorithm we begin by considering the interaction between two patches i and j. In general, computation of the energy transfer between any two patches will be approximate, due to the approximations inherent in the discrete operator $\widetilde{\mathbf{L}}$. Let δ_{ij} denote the product $|\widetilde{\Psi}_i\Delta\mathbf{L}_{ij}\widetilde{\Phi}_j|$. The quantity δ_{ij} approximates the error contributed by the interaction of patch i and j to the view-dependent function v. By reducing δ_{ij} over all pairs (i,j), we can make the magnitude of the overall error $\widetilde{\Psi}^{\mathrm{T}}\Delta\mathbf{L}\widetilde{\Phi}$ arbitrarily small.

We reduce the error in an interaction by subdividing one of the two patches involved and computing new interactions for the refined system. The net effect of this refinement is that transfers of radiosity from bright patches and transfers of importance from important patches will generally be treated with greater accuracy at a lower level in the hierarchy.

Figure 5 illustrates this idea for two patches i and j with high importance and high radiosity, respectively. The diagram on the top shows a single link from patch i to patch j. The link carries radiosity from i to j, in the direction of the arrow, and importance from j to i, against the arrow. Since the effect of these interactions on the error is not great, they take place at a high level in the hierarchy. On the other hand, the diagram on the bottom shows the transfer of radiosity from j to i and of importance from i to j. These transfers, which have greater potential effect on the error, take place at a more refined level of the hierarchy. In this way the algorithm can put the most work into refining the parts of the transport process where the impact of error is greatest.

Because of the duality of radiosity and importance, the same error criterion $|\widetilde{\Psi}_i\Delta\mathbf{L}_{ij}\widetilde{\Phi}_j|$ used to refine radiosity can be used to refine the importance solution as well. Moreover, since estimates of importance are used to drive the radiosity solution and *vice versa*, we need to refine both solutions to the same level of accuracy. Therefore, the hierarchical system used for radiosity is also appropriately refined for importance, so a single hierarchical system suffices for both.

2.3 Radiosity and importance transport

While managing radiosity in a hierarchical system is well-understood [6, 12], importance is new and subtly different. For completeness, we discuss the equations governing both importance and radiosity here.

Until now, we have described the linear operator $\mathbf{L}$ only as a general matrix. For the radiosity transport equation, the operator $\mathbf{L}$ can be expressed as follows:

$$\mathbf{L} = \begin{bmatrix} 1-\rho_1F_{11} & -\rho_1F_{12} & \cdots & -\rho_1F_{1n} \\ -\rho_2F_{21} & 1-\rho_2F_{22} & \cdots & -\rho_2F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_nF_{n1} & -\rho_nF_{n2} & \cdots & 1-\rho_nF_{nn} \end{bmatrix}, \quad (7)$$

where ρ_i is the *reflectance* of patch i, and F_{ij} is the *form factor* from patch i to patch j [10].

The form factor F_{ij} expresses the fraction of power leaving patch i that arrives at patch j. The form factor between two unoccluded differential patches di and dj is given by

$$F_{di,dj} = \frac{\cos\theta_i\cos\theta_j}{\pi r^2}dA_j, \quad (8)$$

where θ_i and θ_j relate the normal vectors of di and dj to the vector joining the two patches, r is the distance between the patches, and dA_j is the differential area of dj.

The form factor from a differential patch di to a finite-area patch j is given by

$$F_{di,j} = \int_{A_j}\left[\frac{\cos\theta_i\cos\theta_j}{\pi r^2}\right]dA_j, \quad (9)$$

where A_j is the area of patch j. The form factor between two finite-area patches i and j is the average of this quantity over all of patch i:

$$F_{ij} = \frac{1}{A_i}\int_{A_i}\int_{A_j}\left[\frac{\cos\theta_i\cos\theta_j}{\pi r^2}\right]dA_j\,dA_i. \quad (10)$$

This double integral can be approximated by the single-integral formulation (9), assuming the two patches are "well-separated" [10].

It is easy to check from equation (10) that the following reciprocity relationship holds between the form factors F_{ij} and F_{ji}:

$$A_iF_{ij} = A_jF_{ji}. \quad (11)$$

We can rewrite equations (1) and (2) in terms of the radiosity and importance arriving at a single patch i from every other patch j to give the familiar radiosity formulation, along with an analogous formulation for importance:

$$\Phi_i = S_i + \rho_i\sum_j\Phi_jF_{ij}, \quad (12)$$

$$\Psi_i = R_i + \sum_j \rho_i \Psi_j F_{ij}. \quad (13)$$

Since $\mathbf{L}$ is used for radiosity and $\mathbf{L}^T$ is used for importance, the two equations differ only in the indices of the reflectance ρ and the form factor F.

We now examine how radiosity and importance can be distributed up and down a hierarchical system. Let i be a child patch, and I its parent in the patch hierarchy. We will assume that reflectance, as well as emittance of radiosity and importance, is constant over every patch.

When the patches are well-separated, equations (9) and (11) imply the following form-factor relationships:

$$F_{ij} \approx F_{Ij}, \quad (14)$$

$$F_{ji} \approx \frac{A_i}{A_I} F_{jI}. \quad (15)$$

Substituting these equations into (12) and (13) allows us to express the radiosity and importance of a child patch in terms of those of its parent, as follows:

$$\Phi_i \approx \Phi_I. \quad (16)$$

$$\Psi_i \approx \frac{A_i}{A_I} \Psi_I. \quad (17)$$

Thus, radiosity does not change when pushed down from a parent to its children, while importance is distributed according to the proportional area of each child.

We also need to be able to solve for the radiosity and importance of a parent patch from those of its children. To derive these relationships, first note that equations (10) and (11) allow us to express the form factors for a parent patch in terms of those of its children:

$$F_{Ij} = \frac{1}{A_I} \sum_{i \in I} F_{ij} A_i, \quad (18)$$

$$F_{jI} = \sum_{i \in I} F_{ji}. \quad (19)$$

Substituting these equations into (12) and (13) gives:

$$\Phi_I = \frac{1}{A_I} \sum_{i \in I} A_i \Phi_i. \quad (20)$$

$$\Psi_I = \sum_{i \in I} \Psi_i. \quad (21)$$

Thus, radiosities must be averaged according to area when pulled up the hierarchy, whereas importances are simply summed.

3 Algorithm

The view-dependent radiosity algorithm we describe here is very similar to the brightness-weighted hierarchical algorithm proposed by Hanrahan, *et al.*, with the crucial difference that in the view-dependent algorithm, importance as well as radiosity plays a role in refining interactions.

The algorithm iteratively computes an accurate radiosity solution for visible patches by refining the interactions between any two patches i and j whose estimated error $|\tilde{\Psi}_i \Delta \mathbf{L}_{ij} \tilde{\Phi}_j|$ exceeds a given tolerance ϵ.

3.1 Overview

The algorithm takes as input the initial emittance of radiosity S and importance R for each patch, along with an initial set of interactions $\mathcal{I}$ among the patches. The set $\mathcal{I}$ contains a link between every pair of patches that are not completely occluded from one another. Each link carries radiosity in one direction and importance in the other. The set $\mathcal{I}$ can be computed once for each model and stored along with it.

The algorithm refines the initial set of interactions $\mathcal{I}$ by transforming this single-level network into a hierarchy of interactions. At each iteration of the outermost loop, the algorithm solves for both radiosity and importance using the current set $\mathcal{I}$. These solutions, in turn, are used to guide a refinement step, which improves the accuracy of the radiosities and importances with respect to a particular view, by adding more links to $\mathcal{I}$. The iteration continues until a preset error tolerance F_ϵ is met.

The basic algorithm can be described in pseudocode as follows:

```
ImportanceDrivenRadiosity(S, R, I)
(Φ̃, Ψ̃) ← (S, R)
for ε decreasing from ∞ to F_ε do
    SolveDualSystems(Φ̃, Ψ̃, S, R, I)
    for each interaction j → i of I do
        RefineInteraction(j → i, I, ε)
    end for
end for
```

The following sections describe the steps of this refinement process in more detail.

3.2 Solving for radiosity and importance

For a given hierarchical system, $\tilde{\Phi}$ and $\tilde{\Psi}$ are found by iteratively solving two systems of linear equations. Each iteration involves gathering radiosity and shooting importance between every linked pair of patches. Radiosity and importance are then distributed up and down the hierarchy so that every patch receives the appropriate contributions from its ancestors and descendants. This process is repeated until convergence—that is, until the difference between the radiosities and importances from one iteration to the next becomes smaller than some tolerance:

```
SolveDualSystems(Φ̃, Ψ̃, S, R, I)
repeat
    GatherAndShoot(Φ̃, Ψ̃, I)
    for each top-level patch p do
        SweepRadiosity(Φ̃, S, p, 0)
        SweepImportance(Ψ̃, R, p, 0)
    end for
until convergence of Φ̃ and Ψ̃
```

Gathering radiosity and shooting importance is implemented as follows:

GatherAndShoot$(\widetilde{\Phi}, \widetilde{\Psi}, \mathcal{I})$

$(\widetilde{\Phi}', \widetilde{\Psi}') \leftarrow (0,0)$
for each patch i **do**
  for each interaction $j \rightarrow i$ of $\mathcal{I}$ **do**
    $\widetilde{\Phi}_i' \leftarrow \widetilde{\Phi}_i' + \rho_i \widetilde{F}_{ij} \widetilde{\Phi}_j$
    $\widetilde{\Psi}_j' \leftarrow \widetilde{\Psi}_j' + \rho_i \widetilde{F}_{ij} \widetilde{\Psi}_i$
  end for
end for
$(\widetilde{\Phi}, \widetilde{\Psi}) \leftarrow (\widetilde{\Phi}', \widetilde{\Psi}')$

For each interaction between two patches $j \rightarrow i$, radiosity is "gathered" from j to i, and importance is "shot" from i to j. Because the importance matrix is the transpose of the radiosity matrix, the same matrix entry $\rho_i \widetilde{F}_{ij}$ appears in both these operations. Here $\widetilde{F}_{ij}$ denotes an estimate of the actual form factor F_{ij}.

Once radiosity and importance have been transferred between linked patches, the two quantities are distributed up and down the hierarchy so that every patch receives the appropriate contributions from its parents and children:

SweepRadiosity$(\widetilde{\Phi}, S, i, \phi_{\text{down}})$

if i is a leaf **then**
  $\phi_{\text{up}} \leftarrow S_i + \widetilde{\Phi}_i + \phi_{\text{down}}$
else
  $\phi_{\text{up}} \leftarrow 0$
  for each child i' of i **do**
    $x \leftarrow$ *SweepRadiosity*$(\widetilde{\Phi}, S, i', \widetilde{\Phi}_i + \phi_{\text{down}})$
    $\phi_{\text{up}} \leftarrow \phi_{\text{up}} + x * A_{i'} / A_i$
  end for
end if
$\widetilde{\Phi}_i \leftarrow \phi_{\text{up}}$
return ϕ_{up}

SweepImportance$(\widetilde{\Psi}, R, i, \psi_{\text{down}})$

if i is a leaf **then**
  $\psi_{\text{up}} \leftarrow R_i + \widetilde{\Psi}_i + \psi_{\text{down}}$
else
  $\psi_{\text{up}} \leftarrow 0$
  for each child i' of i **do**
    $x \leftarrow (\widetilde{\Psi}_i + \psi_{\text{down}}) * A_{i'} / A_i$
    $\psi_{\text{up}} \leftarrow \psi_{\text{up}} +$ *SweepImportance*$(\widetilde{\Psi}, R, i', x)$
  end for
end if
$\widetilde{\Psi}_i \leftarrow \psi_{\text{up}}$
return ψ_{up}

The two quantities are distributed in slightly different ways, as discussed in Section 2.3. The radiosity ϕ of a child patch i is the sum of its emitted radiosity S_i, the radiosity it receives directly, and the radiosity of its parent ϕ_{down}. The importance of a child patch i is the sum of its emitted importance R_i, the importance it receives directly, and an area-weighted fraction of the importance of its parent ψ_{down}.

When pulling radiosity and importance back up the hierarchy the situation is reversed. The radiosity of a parent patch is the area-weighted average of the radiosities of its children, whereas its importance is the sum of the importances of its children.

3.3 Refining the interactions

We refine any interactions whose estimated error $|\widetilde{\Psi}_i \rho_i F_{\text{err}} \widetilde{\Phi}_j|$ exceeds the tolerance ϵ. For F_{err}, we use an upper bound on the error in the form factor, computed by taking the difference between upper and lower bounds F_{ij}^+ and F_{ij}^- on the actual form factor F_{ij}, as described in the next section. If the error in the interaction exceeds the tolerance ϵ, the interaction is refined by subdividing the patch p with greater area and creating new links directly from the children of p to the other patch:

RefineInteraction$(j \rightarrow i, \mathcal{I}, \epsilon)$

$F_{\text{err}} \leftarrow F_{ij}^+ - F_{ij}^-$
if $\widetilde{\Psi}_i \rho_i F_{\text{err}} \widetilde{\Phi}_j > \epsilon$ **then**
  SubdivideAndRefine$(p, j \rightarrow i, \epsilon)$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{$new links created through subdivision$\}$
end if

The procedure *SubdivideAndRefine* is essentially the same as the refinement routine described by Hanrahan, *et al.*With importance-driven refinement, however, the minimum patch-size criterion that guarantees termination is rarely necessary, since the importance of a patch always decreases with its area.

3.4 Estimating the form factors

We estimate the form factor F_{ij} from patch i to patch j by taking a number of samples across patch i and averaging the point-to-disk form factors [21] over all samples.

Care is needed in estimating upper and lower bounds on F_{ij}, since the assumptions required by point-to-disk form factors may not always hold. We choose samples on both patch i and j and compute double-differential form factors, using equation (8). We set the upper and lower bounds F_{ij}^+ and F_{ij}^- to $A_j \max\{F_{di,dj}\}$ and $A_j \min\{F_{di,dj}\}$, respectively. These "bounds" are not necessarily strict if the samples are poorly chosen; however, they seem to work well in practice. Note that if any of the double-differential form factors encounters an occluding object, then the lower bound F_{ij}^- is 0, and the estimated form-factor error is set to the upper bound F_{ij}^+.

3.5 Assigning initial importance

The vector R determines the initial emitters of importance. Different choices of R allow us to use importance in different ways.

Typically, for a single, static view, we define an infinitesimal patch for the "camera" and make it the sole emitter of importance. In this case, we also ensure that an initial link between the camera and the rest of the environment is set up only if it falls within the camera's viewing frustum. This choice of R has the same effect as assigning an initial importance to each patch according to its visible projected area.

There is nothing, however, that limits us to a single, static view. For example, importance could be used to speed the radiosity solution for an animation or walk-through of an environment by making every patch that is visible at any time an emitter of importance. Similarly, importance could be used when the scene is visible from more than a single viewpoint, such as in stereoscopic views or in stage design applications [9].

Finally, for some applications it may not be necessary for everything visible to be important. For example, perhaps we would like to study a particular object from all angles, such as a sculpture in a museum. If we are unconcerned about the environment itself except in how it contributes to the illumination of this object, we could make the object itself the sole emitter of importance.

Note that in all cases, the overall magnitude of R is arbitrary. However, the error tolerance F_ϵ must be scaled appropriately.

4 Reconstruction and display

The algorithm described in this paper addresses only the global illumination problem—i.e., how to determine the radiosities of visible surfaces to within a certain accuracy.

However, producing an accurate global radiosity solution does not guarantee a high-quality picture. Indeed, the reconstruction and display of a radiosity solution is a tricky problem for any algorithm that assumes constant radiosities over a patch [11], and a hierarchical algorithm is no exception. Annoying artifacts become apparent if each patch is displayed with a single constant intensity, as the eye is extremely sensitive to such discontinuities.

Artifacts due to poor reconstruction are in fact even more noticeable for hierarchical algorithms than for standard adaptive subdivision schemes. Since constant radiosities are distributed over patches of different sizes at many different levels of the hierarchy, the straightforward approach of pushing constant radiosities down to the leaves and Gouraud-shading these smallest elements leaves the boundaries between higher-level patches readily apparent.

In order to avoid these artifacts, we can take advantage of view-dependence by adding extra meshing to visible surfaces as part of the reconstruction process, once the importance-weighted global solution has been computed. In our implementation, we force subdivision of all visible surfaces down to a fixed projected-area threshold, and push all interactions down to these leaves.

Displaying these smaller patches with Gouraud shading produces results comparable to a standard Gouraud-shaded adaptive subdivision solution. In order to eliminate Mach banding and z-buffer artifacts, we use ray casting with a modified form of supersampling. For each ray–surface intersection, we choose 16 Gaussian-jittered samples in the (u, v)-coordinates of the intersected surface. The 16 radiosities from the patches at the bottom of the hierarchy are then averaged together to give the intensity of the pixel.

5 Results

Figures 6 and 7 show a small maze sitting on a table. The images were computed using importance-driven refinement. Figure 6 is flat-shaded to show the meshing produced by the algorithm, while Figure 7 is displayed using the reconstruction algorithm described in Section 4. Note the color bleeding from the red wall of the maze to the wall behind it. This effect is most pronounced on the top of the wall, as the bottom half of the wall is illuminated mostly by light reflected from the maze floor.

The next pair, Figures 8 and 9, show the same computed solution, but displayed from further back. Figure 8 shows the radiosity of each patch, while Figure 9 depicts the importance of each patch, divided by its area. In the latter image, the visible parts of the scene show up very clearly as white because all wavelengths of light emitted from these surfaces are equally important to the camera. The other surfaces in the room also have importance, but to a lesser degree. The color of each surface gives the fraction of light emitted at a particular wavelength that finds its way to the camera. Due to the red and blue walls, red light is more important on the left side of the floor and back wall, while blue light is more important on the right. Note how the algorithm has done less meshing in areas with little importance.

Figures 10 and 11 show the radiosity and importance of the same solution—displayed from even further back, showing the whole environment. From this vantage point, the meshing done on the table is extremely fine. A view-independent algorithm would have great difficulty solving the entire environment to such high precision. Note that no meshing at all has taken place in the room at the lower left, which contains a complex block sculpture, since the importance of the room and everything inside it is negligible. Note also that the wall illuminated by the bright light in the center does have some importance, and is therefore refined to a certain extent. The importance of indirect interactions such as these would be very difficult for a user to anticipate in giving meshing hints to a conventional algorithm.

Some quantitative measurements of our tests are summarized in Table 1. All tests were run on an HP 720, a 55 MIPS machine with 64 megabytes of physical memory.

Figures 12 and 13 compare brightness-weighted refinement (*B-only*) to brightness-and-importance-weighted refinement (*BI*). In running the test, we refined the *B-only* solution as much as possible before running out of physical memory. We then ran a *BI* algorithm on the same environment until the solution appeared to be at a similar or slightly higher level of refinement. For this environment of moderate complexity (1002 initial polygons), the *B-only* solution was 45 times slower, requiring 16 times the total number of patches, and 22 times the total number of links. We expect that for environments of greater complexity, the speedups would be even more dramatic.

We have also tried to compare timings for the more refined solution of Figure 6 with a *B-only* solution of comparable accuracy. However, the extraordinary memory requirements for a high-accuracy global solution makes this comparison difficult. We let the *B-only* solution run for about 40 hours, at which point the virtual memory size was well over 100 megabytes. With 19 times the number of patches, 35 times the number of links, and 30 times the number of CPU-minutes as the *BI* solution, the *B-only* algorithm still had not achieved a solution close to the same level of accuracy.

As an additional test, the camera was moved inside the small maze on the table, in the same position with respect to the small maze as it was with respect to the large maze in Figure 13. The *BI* solution was more accurate than *B-only* and took only 0.5% of the time to compute.

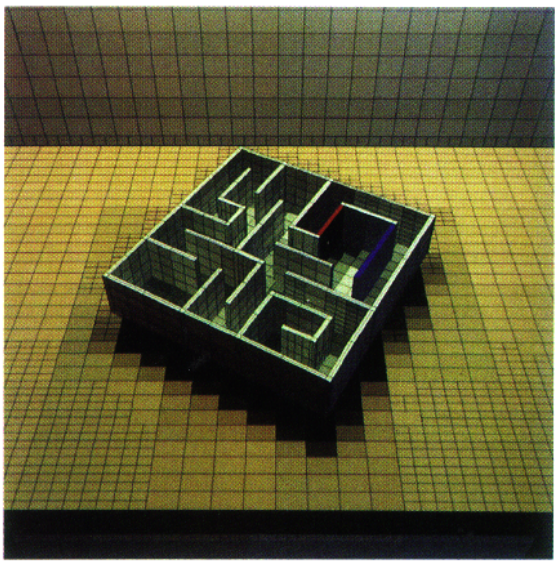

Figure 6: A view-dependent radiosity solution.

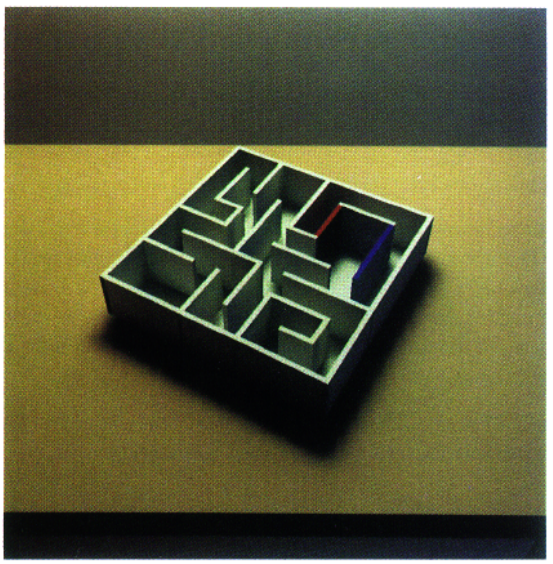

Figure 7: The same solution after reconstruction.

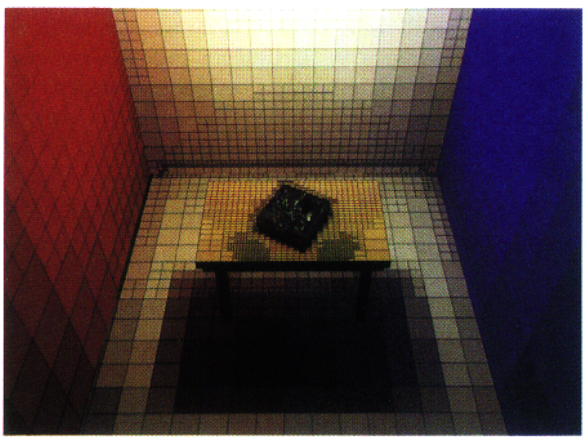

Figure 8: The radiosity solution of Figure 6, seen from further back.

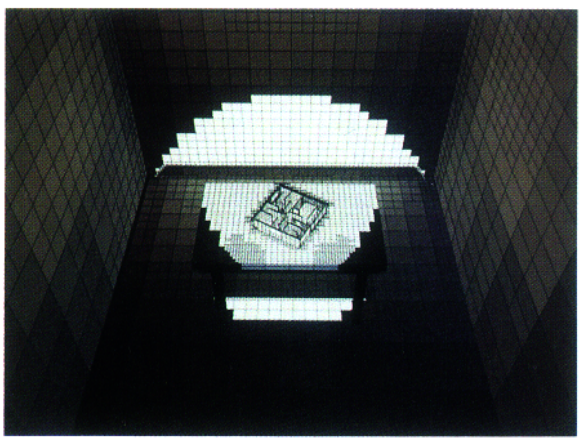

Figure 9: The importance solution for Figure 6.

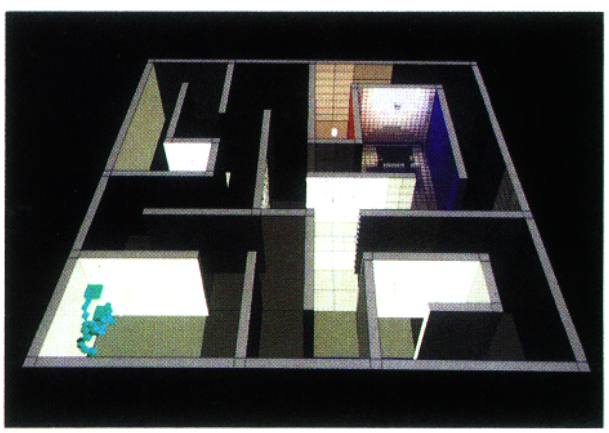

Figure 10: The radiosity solution from even further back.

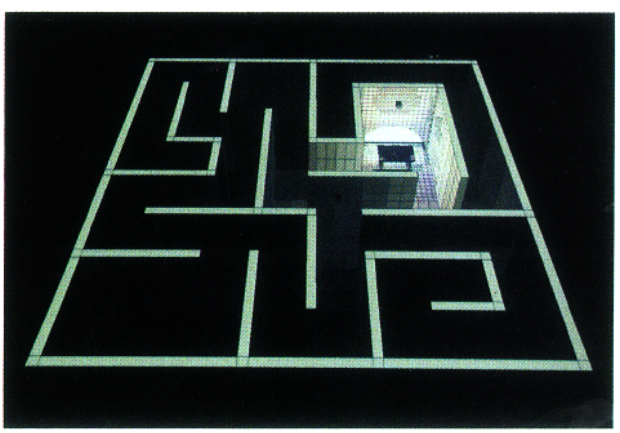

Figure 11: The importance solution from even further back.

	Maze on table Medium accuracy			Maze on table High accuracy			Inside maze Medium accuracy		
	B-only	*BI*	Gain	*B-only*	*BI*	Gain	*B-only*	*BI*	Gain
Total patches	46778	2983	16×	163053+	8779	≫ 19×	46778	1299	36×
Total links	585308	27055	22×	2803112+	79192	≫ 35×	585308	19288	30×
Time (minutes)	585	13	45×	2337+	76	≫ 30×	585	3	195×

Table 1: *B-only* versus *BI* refinement. Statistics for three test cases.

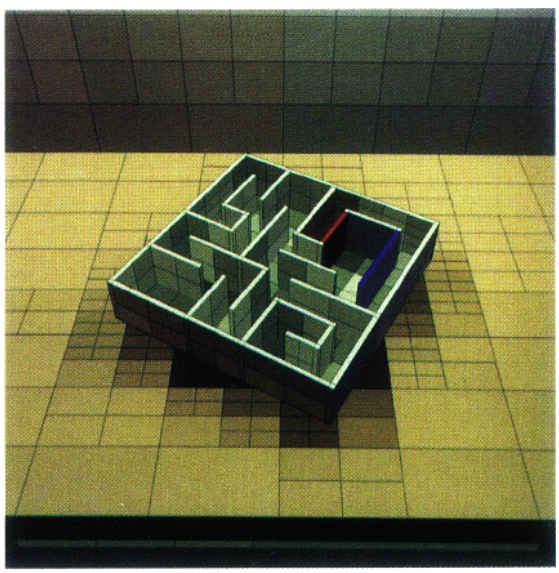

Figure 12: A *B-only* solution.

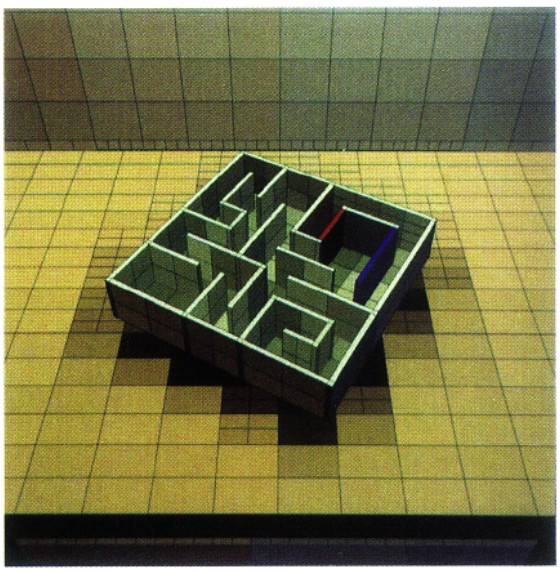

Figure 13: A comparable *BI* solution.

Finally, in order to get a sense of how well the *BI* algorithm might perform for a moving camera, we started with the solution for Figure 13 and then turned the camera 10 degrees. Solving to the same level of accuracy took an additional $3\frac{1}{2}$ minutes, about a quarter the time for the original view.

The initial-linking time, which is identical for *BI* and *B-only* refinement, is not included in the statistics in Table 1. For the 1002-polygon environment used in these tests, initial linking took 75 minutes and produced 18314 links. Note, however, that since initial linking is independent of the level of refinement, material characteristics, or view, it can performed once and stored along with the model. Our current implementation of initial linking uses a brute-force algorithm that checks every pair of patches to determine visibility; however, this step could be easily optimized using a more sophisticated visibility testing scheme.

6 Conclusions and future work

We have described a hierarchical radiosity algorithm that substantially reduces the computation required for an accurate global solution with respect to a particular view. The algorithm works by refining interactions according to the error each interaction introduces to the visible portions of a global solution. Our results show dramatic speedups, even for scenes of moderate complexity. We expect that for a truly complex environment these speedups would be even greater.

There are many aspects of the algorithm that require further research:

Walk-throughs. One way to handle an animated sequence is to make every patch that becomes visible an emitter of importance, as described in Section 3.5. Another approach would be to update importances incrementally as the animation progressed. Because there is a great deal of coherence from one frame to the next in most animations, the importance-driven solution for one frame is likely to be a very good starting point for the computation of the next. Indeed, for many walk-through applications, computing a series of view-dependent solutions may actually be faster than computing a single view-independent solution, especially if high accuracy is required. The one essential change is to add a mechanism for pruning links from the hierarchy that are no longer sufficiently important.

Error analysis. It is easy to get a trivial upper bound on the overall error in the view-dependent radiosity solution by summing the error over all the links: if the system has N links, each refined to a tolerance of ϵ, then an overall bound is given by $N\epsilon$. However, this analysis is clearly too coarse, as we have no guarantee that N grows more slowly than ϵ shrinks. In addition, a rigorous error bound requires a bound on the difference between the estimated error at each link $\widetilde{\Psi}^{\mathrm{T}}\Delta\mathbf{L}\widetilde{\Phi}$ and the actual error $\Psi^{\mathrm{T}}\Delta\mathbf{L}\widetilde{\Phi}$. Currently, we have no

guarantee that this difference will always be small.

Clustering. A major advantage of using a hierarchical approach is that it reduces the total number of interactions between patches from $O(n^2)$ to $O(n+m)$, where n is the total number of patches after subdivision, and m is the number of initial, top-level patches. Still, when m is large, which is often the case for complex environments, the number of initial interactions may be too large to be practical. In this case, we would like to group patches into higher-level clusters in order to reduce the initial interactions. This problem was mentioned by Hanrahan, *et al.* in the description of their brightness-weighted hierarchical algorithm; however, resolving this problem becomes even more crucial in an importance-driven algorithm, since the latter is better able to handle complex environments in every other respect.

Acknowledgments

We would like to thank Donald Greenberg for helpful discussions during the development of these ideas, and Eric Haines, John Wallace and Harold Zatz for reviewing the manuscript. Thanks to Suzanne Smits for creating the models, and to Susan Verba for suggesting a maze. This work was supported by the NSF grant, "Interactive Computer Graphics Input and Display Techniques" (CCR-8617880), by the NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219), and by generous donations of equipment from Hewlett-Packard. Brian Smits is supported by a National Science Foundation Graduate Fellowship.

References

[1] Airey, John M., John H. Rohlf, and Frederick P. Brooks Jr. Towards image realism with interactive update rates in complex virtual building environments. *Computer Graphics* (Special Issue on 1990 Symposium on Interactive 3D Graphics) 24(2): 41–50, 1990.

[2] Arvo, James. Backward ray tracing. In the SIGGRAPH '86 "Developments in Ray Tracing" course notes, 1986.

[3] Baum, Daniel R., Stephen Mann, Kevin P. Smith, and James M. Winget. Making radiosity usable: automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *Computer Graphics* 25(4): 51–60, 1991.

[4] Campbell, III, A. T. and Donald Fussell. Adaptive mesh generation for global diffuse illumination. *Computer Graphics* 24(4): 155–64, 1990.

[5] Chen, Shenchang Eric, Holly E. Rushmeier, Gavin Miller, and Douglass Turner. A progressive multi-pass method for global illumination. *Computer Graphics* 25(4): 165–74, 1991.

[6] Cohen, Michael F., Donald P. Greenberg, David S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications* 6(2): 26–35, 1986.

[7] Cohen, Michael F., Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics* 22(4): 75–84, 1988.

[8] Davison, B. *Neutron Transport Theory.* Oxford University Press, London, 1957.

[9] Dorsey, Julie O'B., François X. Sillion, and Donald P. Greenberg. Design and simulation of opera lighting and projection effects. *Computer Graphics* 25(4): 41–50, 1991.

[10] Goral, Cindy M., Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *Computer Graphics* 18(3): 213–22, 1984.

[11] Haines, Eric. Ronchamp: a case study for radiosity. In the SIGGRAPH '91 "Frontiers in Rendering" course notes, §3:1–28, 1991.

[12] Hanrahan, Pat, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics* 25(4): 197–206, 1991.

[13] Heckbert, Paul S. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics* 24(4): 145–54, 1990.

[14] Kajiya, J. T. The rendering equation. *Computer Graphics* 20(4): 143–50, 1986.

[15] Kalos, M. H., and Paula A. Whitlock. *Monte Carlo Methods, Volume I: Basics.* J. Wiley, New York, 1986.

[16] Lewins, Jeffery. *Importance, The Adjoint Function: The Physical Basis of Variational and Perturbation Theory in Transport and Diffusion Problems.* Pergamon Press, New York, 1965.

[17] Sillion, François and Claude Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics* 23(4): 335–44, 1989.

[18] Sillion, François X., James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics* 25(4): 187–96, 1991.

[19] Teller, Seth J. and Carlo H. Séquin. Visibility preprocessing for interactive walkthroughs. *Computer Graphics* 25(4): 61–9, 1991.

[20] Wallace, John R., Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: a synthesis of ray tracing and radiosity methods. *Computer Graphics* 21(4): 311–20, 1987.

[21] Wallace, John R., Kells A. Elmquist, Eric A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics* 23(3): 315–24, 1989.

[22] Ward, Gregory J., Francis M. Rubenstein, and Robert D. Clear. A ray-tracing solution for diffuse interreflection. *Computer Graphics* 22(4): 85–92, 1988.