Blockwise Refinement –
A New Method for Solving the Radiosity Problem

Günther Greiner, Wolfgang Heidrich and Philipp Slusallek
IMMD IX – Graphische Datenverarbeitung
Universität Erlangen

Abstract

Since the introduction of radiosity to computer graphics, many improvements have been made to lower the time it takes to compute a visually good approximation to the exact solution of the radiosity equation. Progressive Refinement and Overshooting are just two of the possibilities.

In this paper we present a new technique to solve the radiosity equation which provides a fast approximation to the exact solution. This technique solves the Radiosity Equation by computing an approximate solution for an initial subset of patches, considered important for the radiosity environment.

This partial result is then refined towards the final solution by choosing other sets of patches which may have a significant contribution. Thus the new technique reduces the solution of a large matrix equation to a number of solutions involving smaller matrices.

Finally we analyze and compare the different algorithms for their convergence and computational complexity. This provides some interesting new results.

1 Introduction

Since the introduction of radiosity to computer graphics by [GTGB84] many new methods to calculate form factors and solve the resulting matrix equation have been introduced [CG85] [ICG86] [WCP87] [SPL88] [WEH89] [SP89] [BEM89] [CRMT91] [HSA91].

A major advance in the solution technique for the radiosity matrix equation was the introduction of Progressive Refinement in 1988 [CCW88], which was further improved by Feda [FP92] and [GCS] using over relaxation techniques.

At first we shortly review the available solution techniques for radiosity. This includes Gauß-Seidel iteration, Progressive Refinement, successive over-relaxation (SOR), over-shooting and a new "Super-Shoot-Gather" method.

The main part of this paper will present a new family of algorithms to solve the radiosity problem. This approach solves the large set of radiosity equations by selecting in turn a small set of patches that are assumed to have a large impact on the final solution. An approximate solution is then found for this small set of patches.

This approximate solution takes into account the exact interaction between the selected patches and all other patches, but uses an approximation for the interaction between the other patches.

The approximate solution is refined by selecting other sets of patches, based on their estimated influence on the solution. Solving this small matrix system for the new set of patches will result in a correction toward the exact solution of the radiosity problem.

The last section will compare the convergence and the computational complexity of the different solution techniques for the radiosity equation. It gives some interesting insights the the convergence behavior of the different algorithms for the short- and long term convergence.

1.1 The Radiosity Equation

The radiosity method solves a restricted version of the rendering equation [Kaj86] by approximating the surfaces in the environment by planar patches and assuming a homogeneous radiosity at each patch. Additionally the surfaces are assumed to be perfectly diffuse reflectors, so that the Lambertian Law for diffuse reflection is valid.

Using the radiosity B_i the Radiosity Equation for a single patch i is given by

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

where N is the number of patches in the environment, A_i is the surface area of patch i, E_i is the radiosity emitted by the surface itself, ρ_i is the factor of incident energy that is reflected by the surface, and $F_{ji} B_j A_j$ is the part of the energy $B_j A_j$ leaving patch j and reaching patch i. F_{ji} is a purely geometric factor that obeys the reciprocity relation $F_{ij} A_i = F_{ji} A_j$. Application of this relation simplifies equation (1), now written in matrix form, to

$$B = E + \boldsymbol{\rho} \boldsymbol{F} B \ , \tag{2}$$

where $\boldsymbol{\rho}$ is a diagonal matrix given by the factors ρ_i, and $\mathbf{F}$ is the matrix of form factors. B and E are vectors given by the radiosities B_i and the emissions E_i, respectively. We write the matrix equation (2) in standard notation as

$$(\mathbf{1} - \boldsymbol{\rho}\mathbf{F}) B = E. \tag{3}$$

First, let us recall a few properties of the used variables. The reflection factor ρ_i is limited by the physical process of reflection to the range of

$$0 < \rho_i < 1. \tag{4}$$

This means that no surface is perfectly black, absorbing all light energy, nor is it a perfect mirror that reflects all incoming energy. The form factors are in the range $0 \leq F_{ij} < 1$. F_{ij} is zero if the patches cannot directly exchange light energy, or if $i = j$ because a planar surface cannot illuminate itself. The sum of all form factors $\sum_{j=1}^{N} F_{ij}$ is less or equal to one, where equality holds in a closed environment, while in a open environment energy can be lost.

2 Solution Techniques

The early algorithms for the radiosity problem used direct solution techniques to solve for the radiosities B in (3), but iterative methods have proven themselves more efficient.

There are two restrictions that prohibit the use of many of the standard solution techniques for linear systems. First of all, we cannot compute or store the complete matrix, since a typical scene contains in the order of thousands of patches. Instead the solutions should only need to look at a few entries of the matrix at a time. Using standard algorithms for form factor calculation and the reciprocity relation a row and a column of the matrix are easily available.

Secondly, our primary concern in general is not that we seek an exact solution to the radiosity problem (say in the order of 4-6 fractional digits, as in standard numerical applications) per se, rather we need a fast approximation that is visually equivalent to the correct solution.

2.1 Gauß-Seidel Iteration

One of the standard techniques used in early radiosity applications is Gauß-Seidel iteration, also called "gathering", since it is equivalent to computing the new radiosity of a patch by gathering the contributions from all other patches. The outline of the algorithm is given as:

1 for all i
2 $\quad B_i = E_i$
3 while not converged
4 $\quad$ for each i in turn
5 $\quad\quad B_i = E_i + \rho_i \sum_j B_j F_{ij}$
6 display the image using B_i as the intensity of patch i.

2.2 Progressive Refinement

The Progressive Refinement technique was especially developed for the radiosity problem. Only recently it has been shown that this technique is equivalent to a numerical method called Southwell iteration [GCS]. This kind of algorithm distributes radiosity from one patch to the environment. Because it updates all patches in the environment in each step its "visual" convergence is much faster in the beginning, making it more appropriate to the radiosity problem.

In the Progressive Refinement algorithm the patch with the largest unshot energy $\Delta B_i A_i$ is selected. Its radiosity is then "shot" to all other patches to increase their radiosity and unshot radiosity. After shooting, the unshot radiosity of the selected patch is reset to zero. The algorithm is given below:

1 for all i
2 $\quad B_i = E_i$
3 $\quad \Delta B_i = E_i$
4 while not converged
5 $\quad$ pick i, such that $\Delta B_i A_i$ is largest
6 $\quad$ for each patch j
7 $\quad\quad \Delta rad = \Delta B_i \rho_j F_{ji}$
8 $\quad\quad \Delta B_j = \Delta B_j + \Delta rad$
9 $\quad\quad B_j = B_j + \Delta rad$
10 $\quad \Delta B_i = 0$
11 $\quad$ display the image using B_i as the intensity of patch i.

2.3 Overshooting Algorithms

The first two algorithms are conservative in the sense that they only take into account the current radiosity of a patch. It is easy to see that the algorithms can be improved by considering what will happen in later steps of the iteration.

Whenever radiosity is shot from a patch, this will increase the unshot radiosity of other patches. When they are later selected for shooting, a part of this radiosity is sent back to the original patch. Of course this radiosity has to be shot again, and so forth. A similar argument holds for the gathering algorithm.

The idea of an overshooting algorithm is to take these later steps of the iteration into account and to shoot more radiosity than is actually available on the selected patch.

2.4 Successive Overrelaxation (SOR)

A variant of Gauß-Seidel iteration is successive over-relaxation (SOR), which generally gives a better convergence than the standard Gauß-Seidel algorithm. Instead of gathering the correct amount of radiosity, a factor α (in the order of 1.2 - 1.5) is used to take into account steps that would occur later in the iteration. Of course this is a very crude estimate and α has to be tuned to the problem.

The basic algorithm is identical to Gauß-Seidel except that line 5 is modified to be

$$B_i = (1 - \alpha)B_i + \alpha(E_i + \rho_i \sum_j B_j F_{ij}).$$

2.5 Ambient Overshooting

The Overshooting algorithm presented by Feda et. al. [FP92], uses the ambient term introduced by Cohen [CCW88] as an estimate for the additional amount of overshooting. From each selected patch the radiosity $\Delta B_i + \rho_i \; ambient$ is shot.

```
1  for all i
2     B_i = E_i
3     ΔB_i = E_i
4  while not converged
5     pick i, such that (ΔB_i + ρ_i ambient)A_i is largest
6     for each patch j
7        Δrad = (ΔB_i + ρ_i ambient)ρ_j F_ji
8        ΔB_j = ΔB_j + Δrad
9        B_j = B_j + Δrad
10    ΔB_i = −ρ_i ambient
11    display the image using B_i as the intensity of patch i.
```

2.6 Super-Shoot-Gather

In a paper submitted for publication, Gortler et. al. [GCS] present an algorithm that uses the form factor information that is already known to compute the amount of overshooting. Additionally, the amount of overshooting is computed separately for each other patch in the scene.

The amount is computed by setting up a simple restricted version of the original radiosity problem, which can be solved analytically. It takes into account the direct interaction between the shooting and all the other patches.

In the restricted system all form factors are set to zero except those in row and column i, where i the selected patch. The emission vector is replaced with the vector of unshot radiosities. The solution (SG) of this system gives the additional radiosity that is directly exchanged between the selected and all other patches in the scene. Please note that this does not take into account the interaction between the other patches. The algorithm is outlined below. An additional variable ∇B_{ij} is introduced, which is the shot radiosity from patch i to patch j. Since it is only nonzero for patches which have already shot radiosity, it does not require quadratic storage and need not be initialized, if only a constant number of iterations is performed. The unshot radiosity ΔB_{ij} is then derived as $\Delta B_{ij} = B_i - \nabla B_{ij}$. Please see [GCS] for details of the algorithm.

1 for all i
2 $B_i = E_i$
3 while not converged
4 pick a patch i, such that $\sum_j \Delta B_{ij}$ is largest
5 for every other patch j
6 $\Delta rad = \Delta B_{ij} \rho_j F_{ji}$
7 $B_j = B_j + \Delta rad$
8 $\nabla B_{ij} = B_i$
9 $(SG)_i = \frac{\sum_{j \neq i} \rho_i F_{ij} \Delta B_{ji}}{1 - \sum_{j \neq i} \rho_i F_{ij} \rho_j F_{ji}}$
10 $B_i = B_i + (SG)_i$
11 for every other patch j
12 $\Delta rad_j = \rho_j F_{ji} (SG)_i$
13 $B_j = B_j + \Delta rad_j$
14 $\nabla B_{ji} = B_j$
15 $\nabla B_{ij} = B_i$
16 display the image using B_i as the intensity of patch i.

The algorithm presented in the next section extends the idea of the Super-Shoot-Gather algorithm.

3 Blockwise Refinement

We extend the ideas of Gortler et. al. [GCS] in two ways:

- instead of considering a single patch, we select a collection of patches at each step.
- in order to have a reasonable estimate of the solution at the initial step, we allow approximate interaction between the remaining patches.

3.1 The Idea of the Method

We consider a scene with N patches and denote the form factors, the reflection coefficients, the emissions and the radiosities by $(F_{ij})_{1 \leq i,j \leq N}$, ϱ_i, E_i and B_i respectively. Since N is very large, a direct solution to (3) will be impossible. We select out of the N patches a considerably smaller set of n patches, called the "important patches". Typically n ranges between 5 and 50. The set of important patches will be denoted by I. The radiosity equation (3) then can be written in block form

$$\left(\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\varrho}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varrho}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix} \right) \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \end{bmatrix} \tag{5}$$

where $\mathbf{F}_{11} = (F_{ij})_{\substack{i \in I \\ j \in I}}$ contains the form factors for interaction among the n important patches. $\mathbf{F}_{11}$ is an $n \times n$ matrix. Similarly, $\mathbf{F}_{12} = (F_{ij})_{\substack{i \in I \\ j \notin I}}$, $\mathbf{F}_{21} = (F_{ij})_{\substack{i \notin I \\ j \in I}}$ are $n \times N$ and $N \times n$ matrices, respectively, containing form factors for the interaction between an important and a less important patch. Furthermore, $\mathbf{F}_{22} = (F_{ij})_{\substack{i \notin I \\ j \notin I}}$ is the $(N-n) \times (N-n)$ matrix containing the interaction among the less important patches. The n–vector $\mathbf{B}_1 = (B_i)_{i \in I}$ contains the radiosities of the important patches, while the $(N-n)$–vector $\mathbf{B}_2 = (B_i)_{i \notin I}$ contains the radiosities of the less important ones. The same is true of $\mathbf{E}_i$ and $\boldsymbol{\varrho}_i$ respectively.

The basic idea is, to replace in (5) $\mathbf{F}_{22}$ by a simpler matrix which allows to reduce the $N \times N$ system to a system of order $n \times n$. This reduced system will be solved exactly. For

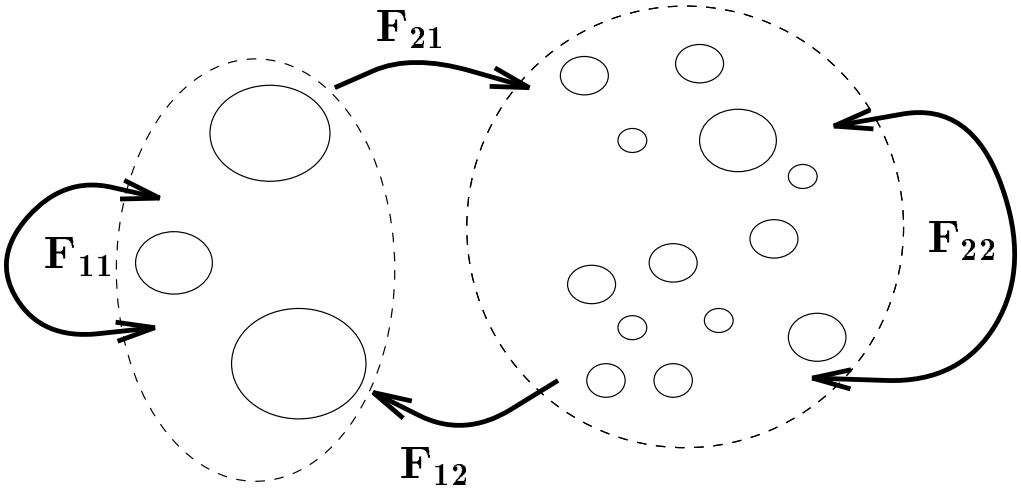

Figure 1: Scheme for the Interaction between the set of important patches (left) and the less important patches

reasonably large n this already yields a fairly good approximation to the exact solution. It can be improved by iteration. In the following sections we will present several possibilities of how $\mathbf{F}_{22}$ can be chosen, and describe the reduction, and the iteration of the algorithm.

3.2 No Interaction Between "Less Important Patches"

The simplest way to change $\mathbf{F}_{22}$ is to set it zero. That is, we approximate the original problem by a problem which has the same emissions and reflection coefficients. Also, the form factors describing the interaction of an "important patch" and any other patch are the same. However, there is absolutely no interaction among "less important patches". Therefore, instead of solving (5) we have to solve the following system:

$$\left(\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\varrho}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varrho}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{0} \end{bmatrix} \right) \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \end{bmatrix} \tag{6}$$

This $N \times N$ system can be reduced to an $n \times n$ system (see below) and can then be solved exactly.

3.2.1 Iteration

In order to improve this approximate solution, one has several possibilities. Of course, Gauß-Seidel or SOR can be applied. Also, Progressive Refinement or Super-Shoot-Gather is possible. In this case, one has to initialize the unshot radiosities ΔB_i and ΔB_{ij} as follows:

$$\Delta B_i = \begin{cases} 0 & \text{for} \quad i \in I \\ B_i & \text{for} \quad i \notin I \end{cases} \qquad \Delta B_{ij} = \begin{cases} 0 & \text{for} \quad i \in I \text{ or } j \in I \\ B_i & \text{for} \quad i \notin I \text{ and } \quad j \notin I \end{cases} \tag{7}$$

Another possibility is to perform the iteration in blockwise fashion. This can be done in the following way: As in Super-Shoot-Gathering, we keep track of unshot radiosities. ΔB_{ij} denotes the radiosity which has to be shot from patch i to patch j. It is initialized as in (7). In the k^{th} step we select another set of n patches, say $i \in I_k$, consisting of those with the highest unshot radiosity $\sum_j \Delta B_{ij}$. Then we decompose the matrices and vectors $\mathbf{F}$, $\boldsymbol{\varrho}$, etc. according to the selected set of patches. Again we denote the corresponding submatrices by $\mathbf{F}_{ij}$ and so on. Then we solve

$$\left(\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\varrho}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varrho}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{0} \end{bmatrix} \right) \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}\mathbf{B}_1 \\ \boldsymbol{\Delta}\mathbf{B}_2 \end{bmatrix} \tag{8}$$

where $\Delta B_i := \frac{1}{n}\sum_{j\in I_k}\Delta B_{ji}$. Then B_i and ΔB_{ij} are updated as follows:

$$B_i = B_i + D_i \qquad \Delta B_{ij} = \begin{cases} 0 & \text{for} \quad i \in I \text{ or } j \in I \\ \Delta B_{ij} + D_i & \text{for} \quad i \notin I \text{ and } \quad j \notin I \end{cases} \tag{9}$$

The algorithm is outlined below.

```
1  for all i
2     ΔB_i = E_i
3     for all j
4        ΔB_ij = E_i
5  while not converged
6     Select n patches i ∈ I with largest Σ_j ΔB_ij
7     for all i
8        ΔB_i = (1/n) Σ_{j∈I} ΔB_ji
9     Solve equation (8)
10    for all i
11       B_i = B_i + D_i
12       for all j ∈ I
13          ΔB_ij = ΔB_ij − ΔB_i
14          ΔB_ji = ΔB_ji − ΔB_j
15    for all i ∉ I
16       for all j ∉ I
17          ΔB_ij = ΔB_ij + D_i
18 display the image using B_i as the intensity of patch i.
```

3.2.2 Reduction to a small system

Now we are going to describe how the $N\times N$-system (6) can be reduced to an $n\times n$ system. Of course, the same applies to (8).

Equation (6) can be written as a system of two equations

$$\begin{array}{lcllcll} \mathbf{B}_1 & - & \varrho_1\mathbf{F}_{11}\mathbf{B}_1 & - & \varrho_1\mathbf{F}_{12}\mathbf{B}_2 & = & \mathbf{E}_1 \\ \mathbf{B}_2 & - & \varrho_2\mathbf{F}_{21}\mathbf{B}_1 & & & = & \mathbf{E}_2 \end{array}$$

From the second equation we get $\mathbf{B}_2 = \varrho_2\mathbf{F}_{21}\mathbf{B}_1 + \mathbf{E}_2$ and replacing this in the first one we obtain the following equation for $\mathbf{B}_1$:

$$\mathbf{B}_1 - (\varrho_1\mathbf{F}_{11} + \varrho_1\mathbf{F}_{12}\varrho_2\mathbf{F}_{21})\mathbf{B}_1 = \mathbf{E}_1 + \varrho_1\mathbf{F}_{12}\mathbf{E}_2 \tag{10}$$

This is an $n\times n$-system. Once it is solved, $\mathbf{B}_2$ is obtained by

$$\mathbf{B}_2 = \varrho_2\mathbf{F}_{21}\mathbf{B}_1 + \mathbf{E}_2$$

3.3 Averaging the Interaction Between "Less Important Patches"

Instead of setting all form factors describing the interaction between two less important patches to zero, one may try to estimate them somehow. In the simplest case one can give them all a constant value. A more sophisticated estimate for the form factors $F_{ij},\ i,j \notin I$ giving rise to a better approximate solution can be obtained as follows:
First determine the ratios $(1 - \sum_{i\in I, j\in I} F_{ij})^{-1}\sum_{i\in I} F_{ik}\ k \notin I$. For a less important patch $i \notin I$, we distribute $1 - \sum_{j\in I} F_{ij}$ to the form factors $F_{ik},\ k \notin I$ according to these ratios. Thus instead of dealing with the exact form factors $F_{ik},\ i,k \notin I$ we consider

$$\tilde{F}_{ik} := (1 - \sum_{j\in I} F_{ij})\frac{\sum_{i\in I} F_{ik}}{1 - \sum_{i\in I, j\in I} F_{ij}}\ . \tag{11}$$

Then we have to solve the following system:

$$\left(\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\varrho}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varrho}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \tilde{\mathbf{F}}_{22} \end{bmatrix} \right) \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \end{bmatrix} \tag{12}$$

with $\tilde{\mathbf{F}}_{22} = \mathbf{a} \otimes \mathbf{b} = (a_i b_j)_{i,j \notin I}$ where $a_i := 1 - \sum_{k \in I} F_{ik}$, and $b_j := (1 - \sum_{\substack{k \in I \\ l \in I}} F_{lk})^{-1} (\sum_{l \in I} F_{lj})$.

In this case the $N \times N$ system can be reduced to an $(n+1) \times (n+1)$ system (see below) and will be solved exactly. The solution is an even better approximation to the exact solution of equation (2) than in the previous case. An iteration to improve it is described in the following section.

3.3.1 Iteration

Out of the remaining patches in turn pick those having largest $\varrho_i A_i$, thus obtaining a set of n patches, say I_k. Perform in turn a gathering step with each of these patches, which yields an estimate of the unshot radiosity ΔB_i for each of the patches $i \in I_k$. Decompose the matrices and vectors $\mathbf{F}$, $\boldsymbol{\varrho}$, etc. according to the selected set I_k. Replace $\mathbf{F}_{22}$ by an approximation $\tilde{\mathbf{F}}_{22}$ as described in (11). Then solve the system

$$\left(\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\varrho}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varrho}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \tilde{\mathbf{F}}_{22} \end{bmatrix} \right) \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}\mathbf{B}_1 \\ \boldsymbol{\Delta}\mathbf{B}_2 \end{bmatrix} \tag{13}$$

Finally update $\mathbf{B}$ by $B_i = B_i + D_i$.

Generally light sources in radiosity scene are considered to have no reflection. In this case blockwise refinement reduces to normal shooting steps. To avoid the large overhead, we use initial shooting steps to distribute the radiosity such light sources.

The algorithm is briefly outlined below:

```
// initialization
1  for all i
2     B_i = 0
// shooting from all "pure" light sources
3  for all patches i with E_i ≠ 0 and ϱ_i = 0
4     for all patches j
5        B_j = B_j + ϱ_j F_ji E_i
6        ΔB_j = ΔB_j + ϱ_j F_ji E_i
7     ΔB_i = 0
8     B_i = E_i
// iteration
9  while not converged
10    Select out of the remaining patches a set I of n patches
                 with largest ϱ_i A_i
11    if not the first iteration step
      // gather towards the selected patches
12       for all i ∈ I
13          ΔB_i = E_i + ϱ_i Σ_j F_ij B_j − B_i
14    solve equation (13)
15    for all i
16       B_i = B_i + D_i
17 display the image using B_i as the intensity of patch i.
```

3.3.2 Reduction to a small system

We describe how (12) and (13) respectively can be reduced to a $(n+1) \times (n+1)$-system. We assume that $\mathbf{F}_{22}$ is replaced by $\mathbf{a} \otimes \mathbf{b} = (a_i b_j)_{\substack{i \notin I \\ j \notin I}}$. Writing (12) as system of two equations

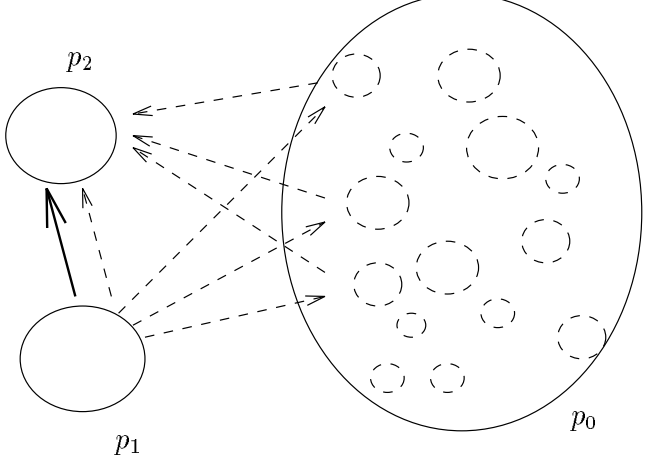

Figure 2: Interpretation of the reduced system. The interaction between p_0 and p_1 is composed of their direct interaction plus the interaction via the virtual patch p_0

yields

$$\begin{array}{lcllcll} \mathbf{B}_1 & - & \boldsymbol{\varrho}_1\mathbf{F}_{11}\mathbf{B}_1 & - & \boldsymbol{\varrho}_1\mathbf{F}_{12}\mathbf{B}_2 & = & \mathbf{E}_1 \\ \mathbf{B}_2 & - & \boldsymbol{\varrho}_2\mathbf{F}_{21}\mathbf{B}_1 & - & \langle\mathbf{B}_2|\mathbf{b}\rangle\boldsymbol{\varrho}_2\mathbf{a} & = & \mathbf{E}_2 \end{array}$$

where $\langle\cdot|\cdot\rangle$ denotes the usual inner product, that is $\langle\mathbf{B}_2|\mathbf{b}\rangle = \sum_{i\notin I} B_i a_i$. In the first equation we substitute $\mathbf{B}_2$ by $\boldsymbol{\varrho}_2\mathbf{F}_{21}\mathbf{B}_1 + \langle\mathbf{B}_2|\mathbf{b}\rangle\boldsymbol{\varrho}_2\mathbf{a} + \mathbf{E}_2$. Moreover, we take the inner product of the second equation with $\mathbf{b}$. $B_0 := \langle\mathbf{B}_2|\mathbf{b}\rangle$ will be considered as an additional unknown quantity. This results in the following system

$$\begin{array}{lcllcll} \mathbf{B}_1 & - & (\boldsymbol{\varrho}_1\mathbf{F}_{11}+\boldsymbol{\varrho}_1\mathbf{F}_{12}\boldsymbol{\varrho}_2\mathbf{F}_{21})\mathbf{B}_1 & - & B_0\boldsymbol{\varrho}_1\mathbf{F}_{12}\boldsymbol{\varrho}_2\mathbf{a} & = & \mathbf{E}_1 + \boldsymbol{\varrho}_1\mathbf{F}_{12}\mathbf{E}_2 \\ B_0 & - & \langle\boldsymbol{\varrho}_2\mathbf{F}_{21}\mathbf{B}_1|\mathbf{b}\rangle & - & \langle\boldsymbol{\varrho}_2\mathbf{a}|\mathbf{b}\rangle B_0 & = & \langle\mathbf{E}_2|\mathbf{b}\rangle \end{array} \tag{14}$$

This is a linear system in the $n+1$ unknowns $B_i,\ i \in I \cup \{0\}$. Once it is solved, one obtains $\mathbf{B}_2$ by

$$\mathbf{B}_2 = \boldsymbol{\varrho}_2\mathbf{F}_{21}\mathbf{B}_1 + B_0\mathbf{a} + \mathbf{E}_2 \tag{15}$$

3.4 More Averaging

The main computational effort (of course depending on the size of n) in the methods described in the previous sections consists in determining the product of the matrices $\mathbf{F}_{12}$ and $\mathbf{F}_{21}$. This needs roughly $2n^2N$ floating point operations. To overcome this problem one may try not only to replace $\mathbf{F}_{22}$ by a simpler matrix, but $\mathbf{F}_{21}$ as well. In case $\mathbf{F}_{21}$ has constant columns, the matrix multiplication $\mathbf{F}_{21}\mathbf{F}_{12}$ needs only $O(nN)$ floating point operations. Appropriate constant values are the averages of each column. Thus we consider

$$\left(\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\varrho}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varrho}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \tilde{\mathbf{F}}_{21} & \tilde{\mathbf{F}}_{22} \end{bmatrix}\right) \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}\mathbf{B}_1 \\ \boldsymbol{\Delta}\mathbf{B}_2 \end{bmatrix} \tag{16}$$

where $\tilde{\mathbf{F}}_{22}$ is as in Section 3.3 and $\tilde{F}_{ij} = \frac{1}{N-n}\sum_{k\notin I} F_{kj}$ for $i \in I$ and $j \notin i$. Iteration and reduction are done in the same way as described in the previous section.

Instead of $\mathbf{F}_{21}$ one may as well approximate $\mathbf{F}_{12}$ by a matrix having constant rows.

3.5 Physical Interpretation

We point out that the reduced systems (10) (14) and the one corresponding to (16) both can be considered as radiosity equation for a scene consisting of n and $n+1$ patches respectively. We describe the situation more detailed for (14). Physically one may think of

Algorithm	Computational Complexity ($\times$ N FLOPs)
Gauß-Seidel (gathering)	2
Successive Overrelaxation (SOR)	2
Progressive Refinement (shooting)	5
Ambient Overshooting	9
Super-Shoot-Gather	13
Blockwise Refinement (Sec. 3.3)	$2n^2 + 9n + 5$
Blockwise Refinement (Sec. 3.4)	$13n + 6$

Table 1: Computational Complexity of Radiosity Algorithms

this as combining all less important patches to form a single patch with index 0. The corresponding emission $E_0 := \langle \mathbf{E}_2 | \mathbf{b} \rangle$ is a weighted average of the radiosities of the less important patches. As reflection coefficient we choose $\varrho_0 = \max_{i \notin I} \varrho_i$. The form factor describing the interaction between two patches $i, j \in I$ now is $F_{ij} + \sum_{k \notin I} F_{ik} \varrho_k F_{kj}$. The first term is the direct interaction, while the second stems from "indirect" interaction through one of the less important patches (Fig. 2). Moreover, $F_{i0} = \sum_{k \notin I} F_{ik} \varrho_k a_k$ and $F_{0i} = \sum_{k \notin I} F_{ki} \frac{\varrho_k}{\varrho_0} b_k$ are the form factors describing interaction between patch i and the additional patch. It is not difficult to verify that the standard hypotheses for radiosity equations posed in Section 2 are satisfied, except for $F_{ii} \neq 0$.

This observation opens the possibility for different procedures to solve (14). Instead of a direct solution via gaussian elimination one may use Progressive Refinement, or one of the shooting methods described in Section 2.

4 Comparisons of Convergence

In order to compare the different algorithms we had to define a suitable error metric and a common performance metric. To compare the algorithms we have chose the normalized total error. This is not the most suitable metric however, since we are more interested in the visual equivalence of the resulting pictures.

The Blockwise Refinement algorithm has a completely different structure than the other algorithms, so that the definition of one 'step' in the iteration is not suitable for a comparison. Also the computational cost of each step in the other algorithms varies significantly as we will show. The use of the elapsed time for comparing the algorithms would include differences due to non optimal implementation of an algorithm and was therefore considered unsuitable.

Finally we choose to use the number of floating point operations (FLOP) as a common and exact metric for the computational complexity of the algorithms. These numbers were computed directly from the presented code segments and we counted each addition, multiplication and division as one operation. We did not account for assignments, logical operations etc.

The computational complexity of one step in each of the algorithms is given in Table 1. The complexity for the initial setup is not given, as it is neglectable and similar for all algorithms.

4.1 Example Scenes

We have selected two different scenes for comparing the performance of the algorithms. In the figures, we plot the normalized total error over the number of FLOPs on the abscissa normalized to units of Progressive Refinement steps to provide an intuitive base for comparison.

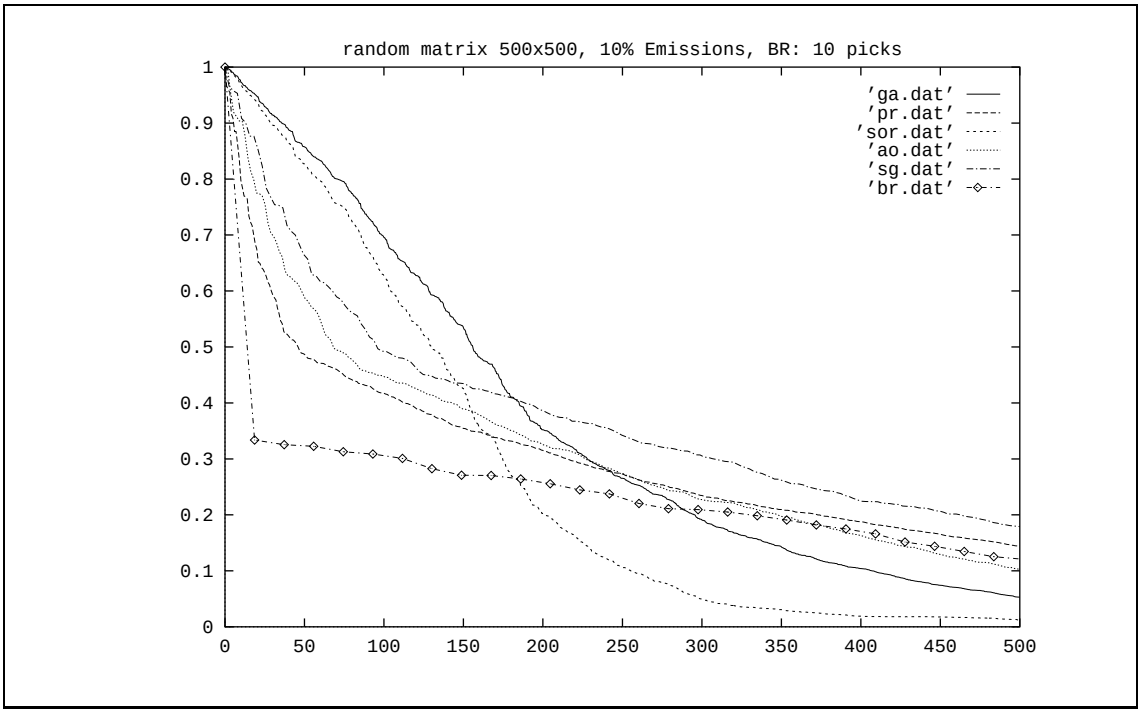

Figure 3: Convergence for a random scene. ga: Gauß-Seidel, pr: Progressive Refinement, sor: Successive Overrelaxation ($\alpha = 1.2$), ao: Ambient Overshooting, sg: Super-Shoot-Gather, br: Blockwise Refinement (Sec. 3.4, $\diamond$ denoting one step)

The first scene (Fig. 3) is an artificial scene computed by randomly setting the entries of the form-factor matrix, the reflectivity and the emissions. The algorithm used takes care to keep the row sums within the correct range (equal to 1.0 in this case) and to obey the reciprocity relation. The number of primary light sources, the mean reflectivity and emission can be adjusted.

In the scene presented the mean reflectivity is 0.5, the mean form factor is 1/500 and 10% of the patches have unit emission. Additionally the area of the largest patch is 10 times that of the smallest and the scene is closed ($\sum_{j=0}^{N} F_{ij} = 1$). Unlike in real radiosity scenes, light sources can (and generally do) have nonzero reflectivity. Therefore no initial shooting steps are required and we can immediately start with blockwise refinement.

The second scene (Fig. 4) is a simple room environment with 874 patches, 6 of which act as light sources with no reflectivity. In this example we used 6 important patches in each step of blockwise refinement.

4.2 Comparison of the Algorithms

It is interesting to note that in both cases the Gauß-Seidel and SOR iteration perform remarkably well in the long term. However, they provide very bad initial results, which makes them unsuitable for interactive radiosity calculations.

It is also remarkable that in respect to their computational complexity the convergence of the shooting algorithms (Progressive Refinement, Ambient Overshooting and Super-Shoot-Gather) are similar. They all perform very well in the first few steps. The classical Progressive Refinement technique even performs better in the initial stage of the iteration, while the others gain ground later on. This is due to the fact that each of their steps is twice and three times as costly, although they give better results in each step.

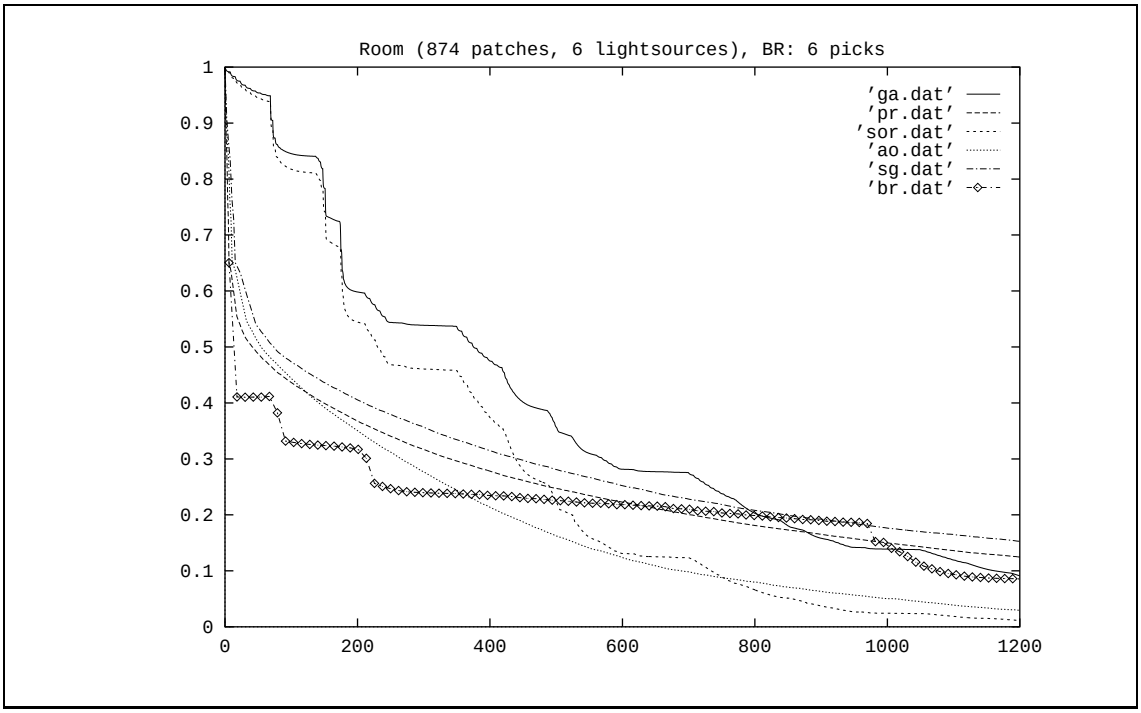

Figure 4: Convergence for a sample scene. ga: Gauß-Seidel, pr: Progressive Refinement, sor: Successive Overrelaxation ($\alpha = 1.2$), ao: Ambient Overshooting, sg: Super-Shoot-Gather, br: Blockwise Refinement (Sec. 3.4, $\diamond$ denoting one step)

For blockwise refinement we used $n = 10$ for the random and $n = 6$ for the room scene. In the room scene we used six initial shooting steps to distribute the radiosity from the primary light sources. This is necessary, because their reflectivity is zero and thus blockwise refinement has no effect and would be too costly.

Blockwise Refinement performs very well during the initial stage of the iteration, while the performance is greatly reduced later on. This is due to the large error in the approximated form factor matrix, which prohibits a fast convergence, as it reintroduces new errors in each step. However, during the first few iterations the error is small in comparison to the large residuum in the radiosity values. This large residuum is greatly reduced in just a few block iterations.

5 Conclusion

In this paper we presented a new approach for solving the Radiosity Equation by approximating and reducing the large $N \times N$ form factor matrix to a system of size $n \times n$, where n is in the order of 5-50. Although the approximation in the form factors introduces new errors and does not converge well in the long run, it gives very good results for the first few steps. A possible strategy would be to perform a few blockwise iterations, (maybe even just one with moderately high n) and then switch to one of the simple iteration procedures.

Comparison of the algorithms based on their computational complexity shows that the shooting algorithms perform well during the initial stage of the iteration, while the gathering based methods are better in the long run.

6 Acknowledgement

We would like to thank Ralf Sonntag from the University of Tübingen, for his support and for providing the room test scene. We would also like to thank the members of the Computer Graphics Lab at Erlangen University for many stimulating discussions.

References

[BEM89] Daniel R. Baum, Rushmeier Holly E., and Winget James M. Improving radiosity solutions through the use of analytically determined form-factors. *ACM Computer Graphics*, 23(3):pp. 325–334, July 1989.

[CCW88] Michael F. Cohen, Shenchang Eric Chen, and John R. Wallace. A progressive refinement approach to fast radiosity image generation. *ACM Computer Graphics*, 22(4):pp. 75–84, August 1988.

[CG85] Michael F. Cohen and Donald P. Greenberg. The hemi-cube: A radiosity solution for complex enviroments. *ACM Computer Graphics*, 19(3):pp. 31–40, July 1985.

[CRMT91] Shenchang Eric Chen, Holly E. Rushmeier, Gavin Miller, and Douglass Turner. A progressive multi-pass for global illumination. *ACM Computer Graphics*, 25(4):165–174, July 1991.

[FP92] Martin Feda and Werner Purgathofer. Accelerating radiosity by overshooting. In Alan Chalmers, editor, *Proceedings of the Third Eurographics Rendering Workshop*. Eurographics, 1992.

[GCS] Steven Gortler, Michael F. Cohen, and Philipp Slusallek. Radiosity and relaxation methods: Progressive refinement is southwell relaxation. submitted for publication.

[GSCH93] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. *ACM Computer Graphics*, 1993.

[GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. *ACM Computer Graphics*, 18(3):pp. 213–222, July 1984.

[HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *ACM Computer Graphics*, 25(4):197–206, July 1991.

[ICG86] Davis S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse enviroments. *ACM Computer Graphics*, 20(4):pp. 133–142, August 1986.

[Kaj86] James T. Kajiya. The rendering equation. *ACM Computer Graphics*, 20(4):pp. 143–149, August 1986.

[SP89] François Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. *ACM Computer Graphics*, 23(3):pp. 335–344, July 1989.

[SPL88] Min-Zhi Shao, Qun-Sheng Peng, and You-Dong Liang. A new approach by procedural refinement for realistic image synthesis. *ACM Computer Graphics*, 22(4):pp. 93–101, August 1988.

[WCP87] John R. Wallace, Michael F. Cohen, and Greenberg Donald P. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *ACM Computer Graphics*, 21(4):pp. 311–320, July 1987.

[WEH89] John R. Wallace, Kells A. Elmquist, and Eric A. Haines. A ray tracing algorithm for progressive radiosity. *ACM Computer Graphics*, 23(3):pp. 315–324, July 1989.