Analysis of Average and Worst Case Error in Hierarchical Radiosity Matrices

M. Pellegrini
Istituto di Matematica Computazionale del CNR, Via S.Maria 46, 56126-Pisa (Italy).
pellegrini@imc.pi.cnr.it

Abstract

Hierarchical radiosity is a technique developed in the early 90's to speed up the iterative solution of systems of linear equations associated with rendering of complex scenes. The gist of the method consists in substituting a dense $k \times k$ matrix A with a second matrix A' of the same size but endowed with an additional internal block structure. Because of the block structure the product of any k-vector with A' requires only $O(k)$ operations, thus speeding up the solution of the associated linear system. In this paper we give the first analytic study of the *a priori* error introduced by the substituting the matrix A' to A. In particular, we prove upper bounds on the absolute value of any entry of the error matrix $A - A'$, both in the worst case and on average. Such bounds are an essential ingredient in any rigorous analysis of the overall performance of fast radiosity algorithms.

1 Introduction

1.1 Motivation and main result. Computing global illumination of scenes for the purpose of photorealistic rendering is logically, as well as computationally, split into several phases each of which introduces approximations and errors of different types. A very clear discussion of different types of error is in the paper of Arvo, Torrance and Smits [ATS94]. In the initial phases the radiosity integral equation[1] is transformed into a finite system of linear equations, which in matricial form we denote with $Ax = b$, where x is a vector of k real unknowns, b is a vector of k real values and A a $k \times k$ matrix (the so called *radiosity matrix*). Errors introduced in the initial phases are classified in [ATS94] either as *data perturbation* or *discretization errors*, and do not concern us in this paper. According to Arvo, Torrance and Smits errors incurred in computing the entries of A and b and in solving the linear systems are classified as *computational errors*, and are the focus of the present paper. Exact solution of the linear system, for example by Gaussian elimination, takes $O(k^3)$ operations and is considered too slow to attain the goal of real time rendering rates. Thus usually the linear system is solved iteratively, where in each iteration the most expensive task is a matrix-vector multiplication costing $O(k^2)$ operations. The gist of the hierarchical radiosity method (HR) consists in substituting the matrix A with a block structured matrix A' which has the property that the product of A' with any k-vector requires only $O(k)$ operations. In this paper we study the error matrix $E = A - A'$ and we give worst case and average error bounds on the entries of E in terms of the area of the surfaces in the scene $\mu(\mathcal{S})$ and a user controlled accuracy parameter α. The parameter α has the dimension of an area and typically $\alpha << \mu(\mathcal{S})$. Introducing the parameter β normalized on the total area, where $0 < \beta = \frac{\alpha}{\mu(\mathcal{S})} << 1$, the main result is the following:

THEOREM 1. *Given a polyhedral scene $\mathcal{S}$ with normalized accuracy parameter β, it is possible to build matrices A, A' and E as above such that:*

1. *Each entry of E has an absolute value smaller than 2β;*
2. *The average absolute value of an entry of E is $O(\beta^2 \log \beta^{-1})$;*
3. *The number k of rows and columns of A' is $O(\beta^{-1} \log \beta^{-1})$.*

[1] See sub-section 1.4 for background.

With this theorem we are able to express both the size of the matrices and the errors in terms of the ratio of the input parameters α and $\mu(\mathcal{S})$. Clearly it is possible, and sometimes useful, to reverse the relations and express the errors in terms of the area and the maximum allowable size of the radiosity matrix due to memory limitations.

1.2 State of the art in Hierarchical Radiosity. Hierarchical radiosity (HR for short) has been introduced by Aupperle, Hanrahan and Salzman [HS90, HSA91, AH93] in the early nineties to speed up the solution of the linear systems associated with a the numerical solution of the radiosity and rendering integral equations. In its simplest form the core of the HR algorithm is as follows. The input is a set of interior-disjoint surfaces (e.g. triangles) S_i for $i = 1,..,n$ representing the scene. The output is a hierarchical decomposition of the given surfaces into patches P_j, $j = 1,..,k$ for $k \geq n$. Since the patches are generated by repeated bisections, there is a natural tree structure associated with the patches (*patch-trees*). The patches produced are the geometric support of a piecewise constant expansion of the radiosity function in a Galerkin discretization scheme. The natural associated radiosity matrix is an $k \times k$ matrix A where $A[a, b]$ is the *form factor* correlating patches P_a and P_b. However the matrix A is not constructed explicitly as it would take quadratic time in k. What is constructed explicitly is a block structured matrix A' where each block requires the computation of a single form factor. Graphically this corresponds to computing the form factors among some pairs of patches associated with internal nodes (call them c and d) in the path-trees and considering these form factors (also termed *links*) as representative of all the form factors among pairs of leaves of subtrees rooted in c and d. The decision of forming a link depends on whether the value of an estimator function for the form factor is below a user-defined threshold value. Arguments in [HS90, HSA91] are given to show that the number of links computed is linear in the number of patches produced.

The basic idea is susceptible of several variations. Lischinski, Tampieri and Greenberg in [LTG93] propose the use of *discontinuity lines* to perform the bisection of the input surfaces (see also [DS96] for further developments). Lischinski, Smits and Greenberg in [LSG94] propose a new estimator function where, besides the the form factor, the importance of the patch with respect to the view point, and upper and lower bounds on the final associated radiosity value are considered. Smits, Arvo and Greenberg [SAG94] propose estimator for the energy transfer among pair of patches, rather than the form factors. A variant of such estimator is in [CS98].

While the core HR algorithm works by splitting large surfaces, when the input consist of many small surfaces, an initial clustering phase is required (see e.g. [Sil94], [SAG94] [SD95], [Sil95] [DS96], [SS96], [GH96], [CSS96]).

1.3 What is new in this paper? The main analysis in [HS90, HSA91] says that the *number of links generated by the algorithm HR* is linear in the *number of patches generated by the algorithm HR*. This is certainly an important property, however in a standard analysis of algorithm (like those in Knuth's book [Knu68]) one should express some measure of complexity of the algorithm (e.g. number of steps, memory used, number of links produced by HR) as a function of some measure of the complexity *of the input*[2]. At the best of my knowledge this paper reports the first input-output analysis of hierarchical radiosity algorithms, relating the relevant algorithm performance to the surface area of the input surfaces and the user-defined input accuracy parameter.

The basic HR algorithm in [HS90, HSA91] requires, for deciding the bisection of a patch, a user-defined threshold F_ϵ, which is meant to control the final accuracy of the method. However F_ϵ does not guarantee termination therefore a second threshold value A_ϵ is added to control the area of the smallest patch produced by the algorithm. The presence of two thresholds complicates the behaviour of the algorithm and its analysis, as some heuristic must be introduced to calibrate their relative values. This inconvenient persists in more advanced versions of HR. In our treatment a single parameter α controls both accuracy and termination thus making the analysis relatively simple. Finally, the error in-

[2] In the so called output-sensitive algorithms one can also throw in measures of complexity of the output provided these are canonical, that is, independent on the specific algorithm.

troduced by the matrix A' is studied by empirical experiments in [HS90, HSA91] but not analytically. Note that the error analysis in [LSG94] is of the *a posteriori* type.

In this paper we do not discuss in details how to build estimator functions relying on techniques in [Pel99a] to attain high precision estimates efficiently. However it is interesting to notice that the estimates using [Pel99a] are always finite, while some estimators like those used in [SAG94, CS98] may diverge.

1.4 Background: integral equations in graphics.

Rendering equation. The rendering integral equation was introduced by Kajiya [Kaj86] to model the equilibrium of radiant energy exchange in a scene under a variety of local shading models including diffuse, glossy and specular reflectance within a a non-partecipating medium. Extended description of such equations can be found in books such as [FvDFH90, CW93, SP94]. Using notation in [SP94] we define the total radiance function $L(y, \omega)$ to be the radiant energy emitted from a point y of the surfaces (we shall call $\mathcal{S}$ the collection of surfaces of the scene), in direction ω. Such function is given by the sum of several contributions. Energy might be emitted (as in light sources) by local phenomena, and this contribution is accounted by a function $L_0(y, \omega)$. Furthermore light received at y from all the other surfaces is re-emitted because of (different types of) reflection. The function $\rho(y, \omega, \eta)$ gives the fraction of the energy received at y from direction η that is re-emitted in direction ω. Calling n_y the external normal unit vector at y and $ray(y, \omega, \mathcal{S})$ the first point of $\mathcal{S}$ met by a ray shot from y in direction ω, the rendering equation can be written as follows:

$$L(y,\omega) = L_0(y,\omega) + \int_{\eta\cdot n_y>0} \rho(y,\omega,\eta) L(ray(y,\eta,\mathcal{S}),-\eta)\eta\cdot n_y d\eta \tag{1.1}$$

The surfaces $\mathcal{S}$ and functions L_0, ρ are given are part of the input. Thus the objective is to find the function L which is the solution of (1.1). Analytic closed solutions are not known even in the simplest models so the continuous equation must be transformed into a system of linear equations in a finite number of real variables by the application of finite boundary or finite elements methods. Note that the function L has a four dimensional domain.

Radiosity equation. The radiosity equation is obtained from the rendering equation with the assumption that the function $\rho(y, \omega, \eta)$, also known as *bidirectional reflectance distribution function (BRDF)* is independent of the incoming and outgoing angles. From a physical point of view this restriction corresponds to retaining in the model effects due to perfect diffusion neglecting specular and glossy effects. Next we introduce the radiosity function $B(y)$ defined as the radiant energy emitted per unit of surface at point $y \in \mathcal{S}$ (in any direction) and we obtain from (1.1) the following equation:

$$B(y) = L_0(y)+\rho(y)\int_{x\in\mathcal{S}} B(x)G(x,y)V(x,y,\mathcal{S})dx \tag{1.2}$$

with

$$G(x,y) = \frac{\cos\theta_x \cos\theta_y}{|x-y|^2}, \tag{1.3}$$

where θ_x (resp. θ_y) is the angle formed by the outer normal n_x (resp. n_y) and the line joining x and y, while the function $V(x, y, \mathcal{S})$ has value 1 if the open segment xy does not intersect $\mathcal{S}$, zero otherwise. Again functions L_0, ρ and surfaces $\mathcal{S}$ are known and the objective is to find the function B which is the solution of (1.2). Again close form solutions are not known even in the simplest models and numerical finite element and finite boundary methods are used. Note that the domain of the function B is two-dimensional.

Galerkin discretization. At an high level the Galerkin method consists of two steps. In the first step the unknown function which is the solution of the integral equation is replaced by the linear combination of known functions taken from a finite set of functions called the *basis*. The elements of the basis are known, while the real coefficients of the linear combination are the variables to be determined. We can multiply both sides of any integral equation by any function (called a *test function*) defined on the same domain of the unknown function and then integrate over such a domain to obtain a new equation. The Galerkin method consists in taking as test functions exactly the basis. In the derived equations the only variables are the real coefficients of the linear combinations, therefore we can use linearity of integral operators to take them out of the integral sign. What is left is a linear equation where the coefficients are integrals involving the integral operator and the basis functions. For more details on finite element methods applied in radiosity radiosity we refer the reader to [Heca], [Hecb] and [ATS94].

Galerkin discretization applied to equation (1.2) leads to coefficients called *form factors* (FF) when the basis is made up of piecewise constant functions, or *higher order form factors* (HOFF) when the basis is not piecewise constant. The use of piecewise constant basis in numerical solution of radiosity equations has a long history whose description is beyond the scope of this paper and we refer to [CW93][SP94]. Galerkin-type solution of radiosity equations with non piecewise constant basis have been introduced in computer graphics by Heckbert [Hec92] for two-dimensional setting, by Zatz [Zat93] and Troutman, Nelson and Max for the three dimensional setting [TM93]. The use of wavelet basis has been explored by Goertler et al. in [GSCH93].

1.5 Organization of the paper. The paper is organized as follows. Section 2 gives a new geometric interpretation of HR in terms of geometric measures of lines in space. Section 3 gives two specific measures for lines, which are related to different forms of the rendering equations Section 4 is devote to finding an explicit expressions and bounds for the entries of the error matrix Section 5 is devoted to conclusions and open problems.

2 HR: algorithmic aspects

The geometric construction of the HR algorithm can be re-phrased in the following terms. We are given a finite set $\mathcal{S}$ of compact polygonal surfaces in 3-space. Consider the set $L(\mathcal{S})$ of lines meeting those objects. Let us denote with μ a (weighted) geometric measure of the set $L(\mathcal{S})$. The task is to split $L(\mathcal{S})$ into subsets L_j of constant descriptive complexity such that

(i) Each subset has a measure $\mu(L_j)$ less than a threshold α.

(ii) We generate a number k of subsets close to the minimum necessary number of subsets (i.e. $\mu(L(\mathcal{S}))/\alpha$).

(iii) The traces (i.e. intersections of the subset of lines L_j onto the surfaces of $\mathcal{S}$) form an inclusion complex (that is, two traces are either disjoint or one is included in the other).

We give the algorithm for a generic (weighted) geometric measure of lines, then we will discuss the implication of using a specific geometric measures.

2.1 General subdivision algorithm. We are going to build a series of graphs, where patches are associated to nodes and edges are places according to the geometric measure of lines meeting the two incident patches. At the end the union of the graphs of the series has the desired properties (i), (ii) and (iii). We have a user defined threshold α which controls the generation of edges.

Given $\mathcal{S}$ we are looking for a decomposition of $\mathcal{S}$ into patches $\mathcal{P} = \{P_j | j = 1, .., k\}$ such that we have an edge (also called *link*) between patches P_x and P_y only if the measure of lines $\mu(P_x, P_y)$ is less than α. Every line meeting two surfaces is part of the set of lines being measured at some link.

To simplify the discussion we assume that there are no "small" patches. More precisely we shall assume the following that for each input surface S_i, $\mu(S_i) \geq \alpha$, that is we suppose that we do not have to start with surfaces too small with respect to α. Moreover we assume $\mu(\mathcal{S}) = \sum_i \mu(S_i) >> \alpha n$, which means that there are surfaces in the input whose associated line measure is much larger than α, otherwise the objective would be attained trivially. From a complexity point of view we consider the number of input surfaces n to be much smaller (and thus less important in complexity considerations) than the ratio $\mu(\mathcal{S})/\alpha$. To simplify the treatment let us suppose that the quantity $\mu(P_x, P_y)$ can be computed exactly. We will bound the number of patches k and the size of the graph by a function of $\mu(\mathcal{S})/\alpha$.

Let $G_0 = (V_0, E_0)$ be the graph obtained by associating each input surface S_i to a node v_i in V_0, and we put an edge (v_x, v_y) in E_0 with weight $\mu(S_x, S_y)$. Thus the initial graph G_0 has n vertices and $O(n^2)$ edges, the total weight is

$$\sum_{x,y} \mu(S_x, S_y) \leq \sum_x \mu(S_x) = \mu(\mathcal{S}).$$

Thus the number of edges whose weight is larger than α is bounded from above by $Q = \mu(\mathcal{S})/\alpha$.

We obtain the reduced graph G'_0 by deleting from G_0 all edges with weight less than α and, after this operation, all isolated nodes. Note that G'_0 has at most Q edges and therefore at most $2Q$ nodes. The first element of our sequence is $G_0 \setminus G'_0$ that is the set of edges and vertices eliminated in this reduction which are no longer available for manipulations. Next we show ho to obtain G_1 and G'_1 from G'_0.

Local refinement. Now we start a refinement procedure which takes an edge $e = (v_1, v_2) \in E'_0$ and *pushes* it down. The push operation consists in taking the surface associated with one of its two incident vertices say v_1, split this surface into two equal area pieces (we do this in a canonical way, independently of v_2) and we create two new nodes v_{11}, v_{12} and we compute the weight of the edges (v_{11}, v_2) and (v_{12}, v_2). If one link has weight larger then α repeat the split on the second surface v_2, thus from one link with two nodes we can get at most 4 new nodes and 4 new links.

Analysis. A push can generate either two or or 4 new nodes and either two, three or four new edges. So we have generated at most $4|V'_0|$ new nodes and the number of edges is at most $4Q$.

Now we eliminate the old edges which are not considered any more. The total weight of the new edges is still bounded by $\mu(\mathcal{S})$ and so the number of remaining edges of value greater than α is at most Q. We delete the edges with weight smaller than α. So we are left with at most Q edges. We eliminate isolated vertices, thus having at most $2Q$ nodes We obtaining the reduced graph G'_1.

The new graph has more nodes but the same upper bound on the number of edges. Clearly we can similarly construct the sequence $G'_0, G'_1, ...$ where the properties proved in the first step hold for any graph in the sequence. How long can this sequence be? Clearly the splitting of surface S_i will be finished at iteration a where

$$\frac{\mu(S_i)}{2^a} < \alpha$$

because this implies that all of the weights of nodes at level a from S_i have weight smaller than α!. Naturally all splitting will be over for a value b such that:

$$\frac{max_i \mu(S_i)}{2^b} < \alpha$$

and since $\mu(\mathcal{S}) \geq max_i \mu(S_i)$, we have that c also satisfy the stopping criterion:

$$\frac{\mu(\mathcal{S})}{2^c} = \alpha.$$

Therefore

$$c = \log_2 \frac{\mu(\mathcal{S})}{\alpha} = \log_2 Q$$

So to conclude the length of the sequence is a function of $\mu(\mathcal{S})$ and α, at each graph the number of nodes and the number of edges are both linear in $\mu(\mathcal{S})/\alpha$ and also linear is the number of edges and nodes deleted in the production of the next reduced graph. So in total the number of links and nodes generated is

$$O(Q \log_2 Q)$$

So the total number $k = \sum_i |V_i \setminus V'_i|$ of nodes generated and the total number $e = \sum_i |E_i \setminus E'_i|$ of edges generated are both upper bounded by $O(Q \log Q)$. This proves part 3 of Theorem 1.

3 Some specific measures for lines

3.1 Measure with occlusion. In $\mu(P_x, P_y)$ we consider only lines that meet both surfaces and no other surface between them. We have again the property:

$$\mu(\mathcal{S}) = \sum_{x,y} \mu(P_x, P_y) \leq \sum_x \pi |P_x| = \pi |\mathcal{S}|.$$

So we have k is $O(\frac{|\mathcal{S}|}{\alpha} \log_2 \frac{|\mathcal{S}|}{\alpha})$.

3.2 Measure without occlusion. In Pellegrini [Pel99b] the relevant measure on lines does not take occlusions into consideration, therefore:

$$\mu(\mathcal{S}) = \sum_{x,y} \mu(P_x, P_y) = \int_{n(L) \geq 2} \binom{n(L)}{2} dL = W$$

where the integral is extended to all lines in 3-space and the function $n(L)$ gives the number of surfaces intercepted by a given line L. In such case we express the bounds in terms of W and not of the area of the surfaces. Thus k is $O(\frac{W}{\alpha} \log_2 \frac{W}{\alpha})$.

3.3 Maximum degree. Case 1. Measure with occlusion. In this case we can bound the total weight of all edges incident to a node v_x corresponding to surface P_x with

$$\sum_y \mu(P_x, P_y) \leq \pi|P_x|.$$

Therefore the number of edges incident to v_x and of weight greater than α is at most $\pi|P_x|/\alpha$ for a graph G_i', thus the number of edges incident to each corresponding node in G_{i+1} is at most $2\pi|P_x|/\alpha$.

Case 2. Measure without occlusions.

$$\sum_y \mu(P_x, P_y) = \int_{L\cap P_x \neq \emptyset} (n(L)-1)dL = W_x$$

The above arguments can be repeated using the total weight W_x. Note that in the splitting process the sum of weights associated to the descendants of the same level is non-increasing in both cases. This is due to the fact that when we split we do not increase the total weight at a node, when we delete nodes the total weight at all other nodes can only decrease.

4 An explicit form of the error matrix

In this section first we aim at giving an explicit formula for an entry of the error matrix. Then we find upper bounds to these entries. A generic linear system $Ax = b$ of size $k \times k$ can be solved iteratively by the Jacobi iterative method:

$$x_i^z = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1, j\neq i}^{k} a_{ij} x_j^{z-1}\right] \tag{4.4}$$

where z is the iteration number, i the row index (for $i = 1, \ldots, k$), and j the column index (for $i = 1, \ldots, k$). Now we adapt formula 4.4 to the iterative solution of the radiosity equation. In this case the terms b_i are easily computed integrals related to the emissivity L_0 in equation (1.2). The terms x_i are radiosity intensities, which we denote with B_i. The term a_{ii} is the area of the patch i times π. The energy sent in all directions is $B_i|P_i|\pi$. The energy received is instead the radiosity intensity along certain lines times the measure of lines having that intensity so $a_{ij} = \mu(P_i, P_j)$. Rewriting in the problem specific notation:

$$B_i^z = \frac{1}{\pi|P_i|}\left[E_i - \sum_{j=1, j\neq i} \mu(P_i, P_j) B_j^{z-1}\right]$$

Now unless there is a link between i and j the value of $\mu(P_i, P_j)$ has not been computed by HR, however in some patch-trees there must be an ancestor of i (call it c) and one of j (call it d) such that the link has been established by HR between them, therefore $\mu(P_c, P_d)$ is available.

Summary of Notation:

1. i is a leaf of the first patch-tree,
2. j is a leaf of second patch-tree,
3. c is an ancestor of i,
4. d is an ancestor of j,
5. $A(i)$ is the set of ancestors of i, clearly $c \in A(i)$
6. $D(d)$ is the set of leaves descending from d, clearly $j \in D(d)$
7. $L(c)$ set of nodes lined to c, clearly $d \in L(c)$.
8. B_i radiosity intensity for lines leaving patch i.

If we associate an intensity of radiosity also to the intermediate nodes of the tree, we have:

$$B_i = \sum_{k\in A(i)} B_k'$$

where B_k' denotes the intensity due to energy gathered through links associated with the node, while B_i is the variable of the linear system associated with the leaf i. Now for $k = c$ we have to compute B_c'. This is given by the ratio of all the gathered energy to the area of patch c.

$$B_c' = \frac{1}{\pi|P_c|}\sum_{h\in L(c)} \mu(c, h) B_h''$$

where B_h'' is the average intensity of the leaves descendant of h. Taking $h = d$:

$$B_d'' = \frac{1}{\pi|P_d|}\sum_{l\in D(d)} \pi|P_l|B_l$$

Naturally we have that one value of the index l is j. So in terms of non-normalized system the coefficient linking i to j through the linking parents c and d is:

$$\frac{1}{\pi|P_c|}\mu(P_c, P_d)\frac{1}{\pi|P_d|}\pi|P_j|$$

Now the difference between this coefficient and the one of the Jacobi iteration matrix is:

$$\Delta_{ij} = \left| \frac{\mu(P_j, P_i)}{\pi |P_i|} - \mu(P_c, P_d) \frac{|P_j|}{\pi |P_c||P_d|} \right|.$$

Going back to the original matrix A, the corresponding error is:

$$\delta_{ij} = \pi |P_i| \Delta_{ij} = \left| \mu(P_j, P_i) - \mu(P_c, P_d) \frac{|P_i|}{|P_c|} \frac{|P_j|}{|P_d|} \right|$$

The interpretation of this formula is clear when we think of the measure of lines meeting two surfaces as a product of the area of one surface times the average solid angle. The second term representing the measure of lines meeting two larger surfaces is scaled by a factor taking into account the shrinking of the area and a second factor taking into account the shrinking of the angle.

4.1 Bounds on the entries of the error matrix

A simple error analysis the following. On each block of entries in A, corresponding to a link in A', the absolute value of the associated weight is bounded by α. Calling $\bar{f}$ the average of a function f defined over a domain D of measure $|D|$ we have that by the definition of average value:

$$\mu(S_c, S_d) = \int_{x \in D} f dx = |D| \bar{f} \leq \alpha$$

where $D = S_c \times S_d$ and f it the solid angular density G introduced in subsection 1.4 (formula (1.3)). Let $D' = S_i \times S_j$ The error is given by:

$$\begin{aligned} &\left| \mu(P_j, P_i) - \mu(P_c, P_d) \frac{|P_i|}{|P_c|} \frac{|P_j|}{|P_d|} \right| = \\ &|\int_{x \in D'} f dx - \int_{x \in D'} \bar{f} dx| \leq \\ &\int_{x \in D'} |f| dx + \int_{x \in D'} |\bar{f}| dx \leq \\ &\int_{x \in D} f dx + \int_{x \in D} \bar{f} dx \leq 2\alpha \end{aligned}$$

where we use the property that $f \geq 0$. This proves part 1 of Theorem 1

Note that the bound 2α applies also to the sum of all absolute errors for entries in a single block. The number of blocks is $O(Q \log Q)$ so the total error over all blocks is $O(\alpha Q \log Q)$. The number of elements over the full matrix A is $\Omega(Q^2)$, so the average error per element is $O(\alpha (\log Q)/Q)$. After a final normalization step by the total area, this completes the proof of parts 1 and 2 of Theorem 1.

Remark. The average sum of absolute errors on each row is also $O(\alpha)$.

4.2 Iterative solution of the linear system.

After proving bounds on the elementwise error incurred in using A' computed by the HR algorithm instead of A in the solution of the linear system $Ax = b$ the next logical step is to choose some iterative scheme. One good candidate is GMRES [GL96, page 549] since here the matrix A is used directly without modifications. Quantifying how the error $A - A'$ impacts on the iterative solution of the linear system is going to be a topic for future research.

5 Conclusion and open problems

In this paper we have analyzed the error incurred when we exchange the full radiosity matrix used in the standard Galerkin numerical solution of the rendering equation with the block structured matrix generated by the hierarchical radiosity algorithm (HR). Moreover, we give the first input-output complexity and error analysis of such exchange. We have made the analysis in the simplest geometric setting, which is view-independent and energy-independent. An obvious open problem is to extend the analysis to the more sophisticated variants of HR when the subdivision is guided by weights involving estimates of the actual energy transfer and/or of the importance for the final image on the screen. I hope this paper will be received as a valid contribution in the direction of building solid analytic foundations for computer graphics.

References

[AH93] L. Aupperle and P. Hanrahan, *A hierarchical illumination algorithm for surfaces with glossy reflection*, Computer Graphics Proceedings, Annual Conference Series, 1993, 1993, pp. 155–162.

[ATS94] James Arvo, Kenneth Torrance, and Brian Smits, *A framework for the analysis of error in global illumination algorithms*, Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–

29, 1994) (Andrew Glassner, ed.), Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, ACM Press, July 1994, ISBN 0-89791-667-0, pp. 75–84.

[CS98] Chin-Chen Chang and Zen-Chung Shih, *Tighter error bounds and weighted error metrics for hierarchical radiosity*, The Journal of Visualization and Computer Animation **9** (1998), no. 2, 53–63, ISSN 1049-9807.

[CSS96] Per H. Christensen, Eric J. Stollnitz, and David H. Salesin, *Global illumination of glossy environments using wavelets and importance*, ACM Transactions on Graphics **15** (1996), no. 1, 37–71.

[CW93] Michael F. Cohen and John R. Wallace, *Radiosity and realistic image synthesis*, Academic Press Professional, San Diego, CA, 1993.

[DS96] George Drettakis and François Sillion, *Accurate visibility and meshing calculations for hierarchical radiosity*, Eurographics Rendering Workshop 1996 (New York City, NY) (Xavier Pueyo and Peter Schröder, eds.), Eurographics, Springer Wien, June 1996, ISBN 3-211-82883-4, pp. 269–278.

[FvDFH90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes, *Computer graphics, principles and practice, second edition*, Addison-Wesley, Reading, Massachusetts, 1990, Overview of research to date.

[GH96] S. Gibson and R. J. Hubbold, *Efficient hierarchical refinement and clustering for radiosity in complex environments*, Computer Graphics Forum **15** (1996), no. 5, 297–310, ISSN 0167-7055.

[GL96] Gene H. Golub and Charles F. Van Loan, *Matrix computations*, third ed., Johns Hopkins Studies in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[GSCH93] Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan, *Wavelet radiosity*, Computer Graphics Proceedings, Annual Conference Series, 1993, 1993, pp. 221–230.

[Heca] P.S. Heckbert, *Finite element methods for radiosity*, Global illuminationcorse course notes Siggraph 93. Avalable at www.cs.cmu.edu/ ph/.

[Hecb] P.S. Heckbert, *Introduction to finite element methods*, Global illumination course notes Siggraph 93. Avalable at www.cs.cmu.edu/ ph/.

[Hec92] Paul Heckbert, *Radiosity in flatland*, Computer Graphics Forum (Eurographics '92) **11** (1992), no. 3, 181–192.

[HS90] Pat Hanrahan and David Salzman, *A rapid hierarchical radiosity algorithm for unoccluded environments*, Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics (Rennes, France), June 1990, pp. 151–71.

[HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle, *A rapid hierarchical radiosity algorithm*, Computer Graphics (SIGGRAPH '91 Proceedings) (Thomas W. Sederberg, ed.), vol. 25, July 1991, pp. 197–206.

[Kaj86] James T. Kajiya, *The rendering equation*, Computer Graphics (SIGGRAPH '86 Proceedings) (David C. Evans and Russell J. Athay, eds.), vol. 20, August 1986, pp. 143–150.

[Knu68] D. E. Knuth, *The art of computer programming I: Fundemental algorithms*, Addison–Wesley, Reading, Massachusetts, 1968.

[LSG94] Dani Lischinski, Brian Smits, and Donald P. Greenberg, *Bounds and error estimates for radiosity*, Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994) (Andrew Glassner, ed.), Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, ACM Press, July 1994, ISBN 0-89791-667-0, pp. 67–74.

[LTG93] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg, *Combining hierarchical radiosity and discontinuity meshing*, Computer Graphics Proceedings, Annual Conference Series, 1993, 1993, pp. 199–208.

[Pel99a] M. Pellegrini, *A geometric appraoch to computing higher order form factors*, Proceedings of the 15th ACM Symposium on Computational Geometry, June 1999, To appear.

[Pel99b] M. Pellegrini, *Rendering equations revisited: how to avoid explicit visibility computations*, Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms, January 1999, Available at http://www.imc.pi.cnr.it/ marcop, pp. 725–733.

[SAG94] B. Smits, J. Arvo, and D. Greenberg, *A clustering algorithm for radiosity in complex environments*, Computer Graphics (SIGGRAPH '94 Proceedings), July 1994, pp. 435–442.

[SD95] F. Sillion and G. Drettakis, *Feature-based control of visibility error: a multi-resolution clustering algorithm for global illumination*, Computer Graphics (SIGGRAPH '95 Proceedings), August 1995, pp. 145–152.

[Sil94] François Sillion, *Clustering and volume scattering for hierarchical radiosity calculations*, Fifth Eurographics Workshop on Rendering (Darmstadt, Germany), June 1994, pp. 105–117.

[Sil95] Francois X. Sillion, *A Unified Hierarchical Algorithm for Global Illumination with Scattering*

Volumes and Object Clusters, IEEE Transactions on Visualization and Computer Graphics **1** (1995), no. 3, 240–254.

[SP94] François Sillion and Claude Puech, *Radiosity and global illumination*, Morgan Kaufmann, San Francisco, 1994, excellent coverage of radiosity and global illumination algorithms.

[SS96] Cyril Soler and François Sillion, *Accurate error bounds for multi-resolution visibility*, Eurographics Rendering Workshop 1996 (New York City, NY) (Xavier Pueyo and Peter Schröder, eds.), Eurographics, Springer Wien, June 1996, ISBN 3-211-82883-4, pp. 133–142.

[TM93] Roy Troutman and Nelson L. Max, *Radiosity algorithms using higher order finite element methods*, Computer Graphics Proceedings, Annual Conference Series, 1993, 1993, pp. 209–212.

[Zat93] Harold R. Zatz, *Galerkin radiosity: A higher order solution method for global illumination*, Computer Graphics Proceedings, Annual Conference Series, 1993, 1993, pp. 213–220.