Error Control for Radiosity

Philippe Bekaert and Yves D. Willems

Paper presented at the 7 th Eurographics Workshop on Rendering
Porto, June 1996

Please send any correspondence to:
Philippe Bekaert
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
B-3001 Heverlee
BELGIUM
Telephone: ++32 - 16 - 32 70 94
Fax: ++32 - 16 - 32 79 96
E-mail: Philippe.Bekaert@cs.kuleuven.ac.be

Error Control for Radiosity

Philippe Bekaert[1] and Yves D. Willems

Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200 A, B-3001 Leuven, Belgium
e-mail: Philippe.Bekaert@cs.kuleuven.ac.be

Abstract: In this paper, we address the problem of computing the radiance in a diffuse environment up to a beforehand specified accuracy. We consider this problem in the context of wavelet radiosity, a hierarchical radiosity method with higher order radiance approximations. We first present an analysis of the discretisation error, which is the error introduced by projecting the problem onto a finite set of basis functions. This analysis leads to an algorithm for a-posteriori error estimation and a criterion and strategy for hierarchical refinement, generalising previous work for constant and bilinear approximations. We propose a new hierarchical radiosity algorithm in which a user controls the accuracy of the solution by specifying directly the maximum allowable absolute radiance error rather than an interaction error threshold.

1 Introduction

The numerical simulation of the propagation of light in an environment is a great tool for photo-realistic image synthesis. However, if such simulations are to be used for quantitative predictions of illumination, e.g. in illumination engineering applications, a realistic estimate of the accuracy of the computed results is needed. Moreover, it is also desirable to be able to compute the illumination in a environment up to a given precision, specified in advance, with minimal work. In this paper we address error estimation and control for illumination computations in environments exhibiting only diffuse reflection and emission of light.

We will mainly deal with discretisation error [1]. In particular, the influence of the set of basis functions will be studied. This set of basis functions can be manipulated, e.g. by subdividing surface elements, in order to obtain a higher accuracy. The automated choice of the smallest basis, allowing to approximate the true radiance distribution in a scene up to given precision, is the key to error control.

We will initially assume that computational errors do not pose major problems, e.g. by using sufficiently precise numerical integration rules and visibility estimation techniques for computing the needed coupling coefficients, and by allowing a sufficiently large number of Jacobi or Gauß-Seidel iterations to solve the system of linear equations. At the end of the paper, we will show how computational errors can be taken into account. We will also ignore errors due to perturbed boundary data. The error due to imprecise data, such as emissivities or reflectivities, or due to approximating the surfaces in a scene by triangular and/or convex quadrilateral patches, can not be influenced during the radiance computations.

In the recent past, error estimation algorithms and good refinement criteria have been developed for hierarchical radiosity methods with piecewise constant [12] or bilinear [13] approximation of radiance over the surface elements. In this paper we will

[1] Research performed with financial support by a grant from the Flemish 'Institute for the Promotion of Scientific-Technological Research in Industry' (IWT#941120).

generalise this work to a broad class of basis functions. The only restrictions placed on the basis functions in this paper is that they should be non-overlapping and be suited for hierarchical representation of radiance over surface patches. Generalisation to other bases is useful because these may lead to a more efficient computation of the radiance in a given environment. Error control is obtained by enforcing hierarchical refinement only in those places where the estimated error is larger than an a-priori specified error threshold.

The organisation of the paper is as follows: in §2 we present the equations to be solved, introducing terminology and notation. In §3 we present an analysis of the error introduced by the choice of a set of basis functions. We present an algorithm for a-posteriori error estimation which is based on the theoretical results of that section in §4. Next we present a refinement indicator and subdivision strategy in §5. In §6 we present a new algorithm for error control, obtained by combining the results of §4 and §5. In §7 we show how other sources of error can be dealt with. Finally some conclusions are drawn and directions for future research given in §8.

2 Computation of the illumination in diffuse environments

In order to compute the radiance $L(x)$ at a point x in a diffuse environment, an integral equation (see e.g. [4])

$$L(x) = L_e(x) + \int_A K(x,y)L(y)dA_y \tag{1}$$

with kernel

$$K(x,y) = \rho(x)\frac{\cos\theta_x \cos\theta_y}{\pi\|x-y\|^2}\text{vis}(x,y)$$

needs to be solved. $L_e(x)$ and $\rho(x)$ in this equation are the self-emitted radiance and the reflectivity at x. The integral is over the surfaces A of all objects in the scene. $\cos\theta_x$ and $\cos\theta_y$ are the cosine of the angle between the line connecting the two points x and y and the normal at the surface in point x and y respectively. $\text{vis}(x,y)$ is 1 if both points are mutually visible and 0 otherwise.

We assume that the scene has been discretised into patches i. In order to approximately solve (1), the self-emitted and total radiance $L_e(x)$ and $L(x)$ are approximated by a linear combination $\tilde{L}_e(x)$ resp. $\tilde{L}(x)$ of some set of basis functions $\phi_{i,\alpha}(x)$ defined on each patch i:

$$L_e(x) \approx \tilde{L}_e(x) = \sum_{i,\alpha} E_{i,\alpha}\phi_{i,\alpha}(x)$$

$$L(x) \approx \tilde{L}(x) = \sum_{i,\alpha} L_{i,\alpha}\phi_{i,\alpha}(x).$$

The error introduced by approximating the radiance function $L(x)$ by such a linear combination $\tilde{L}(x)$ will be analysed in §3.

If the basis functions $\phi_{i,\alpha}$ are non-overlapping ($\phi_{i,\alpha}(x) = 0$ if $x \notin$ patch i), projection yields a set of linear equations with the coefficients $L_{i,\alpha}$ as unknowns:

$$\sum_{\alpha}(L_{i,\alpha} - E_{i,\alpha})O_{i,\alpha,\alpha'} = \sum_{j,\beta} L_{j,\beta}K_{i,\alpha';j,\beta} \tag{2}$$

with coupling coefficients

$$K_{i,\alpha;j,\beta} = \int_{A_i} \phi_{i,\alpha}(x) \int_{A_j} K(x,y)\phi_{j,\beta}(y) dA_y \, dA_x \tag{3}$$

and basis overlap coefficients

$$O_{i,\alpha,\alpha'} = \int_{A_i} \phi_{i,\alpha}(x)\phi_{i,\alpha'}(x) dA_x . \tag{4}$$

The basis functions in our implementation were obtained by orthonormalisation of the functions $\{1, u, v, uv, u^2, v^2, u^3, u^2v, uv^2, v^3\}$ on the unit square and the standard triangle $(0,0), (1,0), (0,1)$. The bases on these master elements were then mapped to an arbitrary convex quadrilateral or triangle element using a bilinear or barycentric mapping respectively. More details are given in [3], where also the algorithm to solve (2) and to compute the coupling and overlap coefficients $K_{i,\alpha;j,\beta}$ and $O_{i,\alpha,\alpha'}$ in our implementation are described.

3 An analysis of the discretisation error

In absence of other sources of error, the accuracy of the computed radiance solution $\tilde{L}(x)$ is only determined by how well it is possible to approximate the true radiance distribution $L(x)$ by a linear combination of the basis functions $\phi_{i,\alpha}(x)$. We first present an expression for the difference $\varepsilon(x)$ between the true and the computed radiance at a given *point* x on a surface in the scene. Using the notion of a function norm, a definition of the discretisation error on a *patch* is given next. The theory in this section provides the basis for the algorithms in the next sections. Later in §7, we will discuss shortly how the analysis in this section can be extended in order to deal with other sources of error than discretisation error as well.

The discrepancy $\varepsilon(x) = \tilde{L}(x) - L(x)$ between the computed and the true radiance at a point x consists of three contributions:

- the error $\delta_e(x) = \tilde{L}_e(x) - L_e(x)$ introduced by approximating the given self-emitted radiance $L_e(x)$ by a linear combination $\tilde{L}_e(x)$ of basis functions $\phi_{i,\alpha}(x)$ at x. This term can be non-zero only for points x on light sources;
- the error introduced by approximating the radiance received at x by a linear combination of the basis functions $\phi_{i,\alpha}(x)$ defined on the patch i containing x. This term will also be nonzero when the radiance $L(y)$ on all other patches j visible from x can be written exactly as a linear combination $\tilde{L}(y) = \sum_\beta L_{j,\beta}\phi_{j,\beta}(y)$ of the basis functions $\phi_{j,\beta}(y)$ defined on those patches j;
- the error at x caused by propagation of the error $\varepsilon(y) = \tilde{L}(y) - L(y)$ from all points y visible at x.

It can be proven that $\varepsilon(x)$ is given by

$$\varepsilon(x) = \delta_e(x) + \sum_j \delta_j(x) + \sum_j \int_{A_j} K(x,y)\varepsilon(y) dA_y \, . \tag{5}$$

The three terms in this expression correspond to the three contributions mentioned above. The functions

$$\delta_j(x) = \sum_\beta L_{j,\beta} \left(\sum_\alpha R_{\alpha,\beta} \phi_{i,\alpha}(x) - \int_{A_j} K(x,y) \phi_{j,\beta}(y) dA_y \right) \tag{6}$$

are the difference between the approximated radiance $\tilde{L}_j(x)$ and the true radiance $L_j(x)$ received at x when the radiance emitted by j would be represented exactly by $\tilde{L}(y) = \sum_\beta L_{j,\beta} \phi_{j,\beta}(y)$. The coefficients $R_{\alpha,\beta}$ in (6) are obtained by solving a small set of linear equations per basis function $\phi_{j,\beta}$ defined on patch j:

$$\sum_\alpha R_{\alpha,\beta} O_{i;\alpha,\alpha'} = K_{i,\alpha';j,\beta}. \tag{7}$$

In the case of constant radiance approximation, (6) reduces to

$$\delta_j(x) = \rho_i L_j (F_{ij} - F_{dA_x,j}).$$

The function $\delta_j(x)$ then is the difference between the actual point-x-to-patch-j form factor $F_{dA_x,j}$ and the patch-i-to-patch-j form factor F_{ij}, which is the average point-to-patch form factor for points $x \in A_i$. This difference is weighted by the reflectivity ρ_i of the receiving patch and the radiance L_j of the source patch. We will show in §4.2 how $\delta_j(x)$ can be evaluated at a very low extra cost at the points x_k used to compute integrals on patch i.

We define the total discretisation error in p-norm $\varepsilon_i^{(p)}$ on a patch i as

$$\varepsilon_i^{(p)} = \left(\frac{1}{A_i} \int_{A_i} |\varepsilon(x)|^p dA_x \right)^{\frac{1}{p}}. \tag{8}$$

With $p = 1$, this expression yields the average discretisation error on the patch, with $p = 2$ the RMS error. In the limit for $p \to \infty$, the maximum

$$\varepsilon_i^{(\infty)} = \max_{x \in A_i} |\varepsilon(x)| \tag{9}$$

is taken as error on the patch. With these definitions, the dimension of the error $[W/m^2 sr]$ is always the same as for radiance. We will use error in ∞-norm in the rest of this paper and drop the superscript $^{(\infty)}$. Filling in (5) in (9) then yields

$$\varepsilon_i = \max_{x \in A_i} \left| \delta_e(x) + \sum_j \left(\delta_j(x) + \int_{A_j} K(x,y) \varepsilon(y) dA_y \right) \right|. \tag{10}$$

Deriving algorithms and results for other norms is straightforward.

4 An algorithm for a-posteriori error estimation

Expression (10) naturally leads to an algorithm for a-posteriori estimation of the error on a computed radiance solution. In §4.1 we will first give an outline of the algorithm. §4.2 is a technical section describing a simple and efficient way how to compute the needed coefficients. The algorithm is discussed in §4.3.

4.1 Outline of the algorithm

An upper bound $\overline{\varepsilon}_i$ for the discretisation error on each patch i can be obtained by solving the set of equations

$$\overline{\varepsilon}_i = \max_{x \in A_i} \left| \left(\delta_e(x) + \sum_j \delta_j(x) \right) + \sum_j \overline{\varepsilon}_j \int_{A_j} K(x,y) dA_y \right|. \tag{11}$$

The number of equations is equal to the number of patches in the scene. The terms $\delta_e(x) + \sum_j \delta_j(x)$ in these equations can be considered source terms for error, comparable with the terms describing self-emitted radiance in the set of equations to be solved in order to compute radiance. The integrals $\int_{A_j} K(x,y) dA_y$ are ρ_i times the classical point-x-to-patch-j form factors $F_{dA_x,j}$ and result from the fact that the actual discretisation error $\varepsilon(y)$ on the patches j is replaced by a constant $\overline{\varepsilon}_j \geq \max_{y \in A_j} |\varepsilon(y)|$.

The set of equations can be solved iteratively with Jacobi or Gauß-Seidel iterations. Such an iterative method will converge since, disregarding computational errors, for each point x

$$\sum_j \int_{A_j} K(x,y) dA_y = \rho(x) < 1.$$

The main difference with previous algorithms for error estimation [12, 13] is in the way the error contributions from each interacting patch j are added. In [12, 13], the absolute value of the sum in (10) is replaced by the sum of the absolute values of each term. This is the main reason why there is no need for sorting and dropping contributions in our algorithm. Because of this, equation (11) may also lead to tighter error estimates.

In order to compute the maximum $\max_{x \in A_i}$ in (11), a numerical optimisation algorithm or an estimation method such as described in [10] should be used. We have however approximated $\max_{x \in A_i}$ by taking the maximum at only a small set of points x_k on each patch i. Similar to [7], the points we used are the nodes of the numerical integration rule used to compute integrals on patch i. We show in the next section §4.2 how $\delta_j(x_k)$ and $\int_{A_j} K(x_k, y) dA_y$ can be evaluated efficiently at these points x_k while computing the coupling coefficients $K_{i,\alpha;j,\beta}$.

In the context of a hierarchical radiosity method, (11) needs to be computed for all leaf elements i in the patch hierarchies. The patches j to consider are then all patches interacting with the leaf element i and its parent elements. In our implementation, we have estimated the error at each level separately using (11). Next, the contributions from higher levels were added to the error estimate at lower levels. Finally, in order to obtain a consistent representation of the total error on each level, we have set the error at a parent level to the maximum error at the lower levels. The convergence of this simplified algorithm cannot be guaranteed in all cases. However, we believe that a more advanced and correct strategy will only be needed for environments with unrealistic high reflectivities. Moreover, as will be explained in §4.3, a much more realistic error estimate is obtained by ignoring error propagation as in [12]. In this case, the question of convergence doesn't pose itself anymore.

4.2 Computation of the coupling- and link error coefficients

Algorithm 4.1 shows how the functions $\delta_j(x)$ and the point-to-patch factors in (11) can be evaluated at the points x_k used for numerical integration over the receiving patch i at a very low extra cost during the computation of the coupling coefficients $K_{i,\alpha;j,\beta}$. The

integrals are computed by transforming the patches to their standard domain, which is the unit square for quadrilateral patches and standard triangle for triangular patches. The integrals on the the standard domain are then approximately computed using a suitable numerical cubature rule we found in [5]. More details on how the integrals were computed are given in [3]. We allowed different cubature rules to be used on i and j:

$$\int_{A_i} f(x)dA_x \approx \sum_{k=1}^{n} w_k f(u_k, v_k) J_i(u_k, v_k) \tag{12}$$

with n nodes (u_k, v_k) and weights w_k on patch i and a cubature rule with m nodes (u'_l, v'_l) and weights w'_l on j. $J_i(u_k, v_k)$ is the jacobian of the bilinear or barycentric mapping at a point (u_k, v_k) in the standard domain. Also different bases $\phi_{i,\alpha}$, with a basis functions, and $\phi_{j,\beta}$, with b basis functions, can be defined on the patches i and j respectively.

Algorithm 4.1 *Computation of the coupling coefficients $K_{i,\alpha;j,\beta}$ and link error estimation coefficients $\delta_{j,\beta}(x_k)$ between two patches i and j:*

1. For all $k = 1 \ldots n$, $l = 1 \ldots m$, compute
$$K(x_k, y_l) = \rho(x_k) \frac{\cos\theta_{x_k} \cos\theta_{y_l}}{\pi ||x_k - y_l||^2} \text{vis}(x_k, y_l).$$
2. For all $k = 1 \ldots n$, $\beta = 1 \ldots b$, compute
$$K_{j,\beta}(x_k) = \int_{A_j} K(x_k, y)\phi_{j,\beta}(y)dA_y \approx \sum_{l=1}^{m} w'_l K(x_k, y_l)\phi_\beta(u'_l, v'_l) J_j(u'_l, v'_l)$$
3. For all $\alpha = 1 \ldots a$, for all $\beta = 1 \ldots b$, compute
$$K_{i,\alpha;j,\beta} = \int_{A_i} \phi_{i,\alpha}(x) K_{j,\beta}(x) dA_x \approx \sum_{k=1}^{n} w_k \phi_\alpha(u_k, v_k) K_{j,\beta}(x_k) J_i(u_k, v_k)$$
4. For all $\beta = 1 \ldots b$,
   (a) Compute $R_{\alpha,\beta}$ by solving
   $$\sum_{\alpha} R_{\alpha,\beta} O_{i;\alpha,\alpha'} = K_{i,\alpha';j,\beta}$$
   (b) For all $k = 1 \ldots n$, compute
   $$\delta_{j,\beta}(x_k) = \sum_{\alpha} R_{\alpha,\beta} \phi_{i,\alpha}(x_k) - K_{j,\beta}(x_k)$$
5. Return the $a \times b$ coupling coefficients $K_{i,\alpha;j,\beta}$, the $n \times b$ coefficients $\delta_{j,\beta}(x_k)$ and the n coefficients $K_{j,1}(x_k)$.

The visibility factors $vis(x_k, y_l)$ in step 1 between cubature nodes x_k and y_l on patch i and patch j respectively are determined using ray casting, accelerated by using shadow caching and shaft culling [8]. In this simplified algorithm, no special care is taken of

possible partial occlusion of the patches i and j. In such situations, the functions to be integrated may not be smooth and may cause unacceptable errors in the numerical integrations. This will affect both the quality of the computed coupling coefficients and the error estimates. It is explained in §7 how partial occlusion can be handled correctly.

The coefficients $\delta_{j,\beta}(x_k)$ computed in step 4b of this algorithm are used to compute the link approximation error estimates $\delta_j(x_k)$ at the nodes of the cubature rule used on patch i:

$$\delta_j(x_k) = \sum_\beta L_{j,\beta}\delta_{j,\beta}(x_k).$$

Moreover, the bases we have used in our implementation always contained the constant function as the first function so that $K_{j,1}(x_k) = \int_{A_j} K(x_k, y)dA_y$ is computed in step 2. Algorithm 4.1 thus computes all needed ingredients in order to evaluate (11) at the cubature nodes x_k on patch i.

4.3 Discussion

Two important simplifications were made in our algorithm to estimate the error on a computed radiance solution:

- we have estimated $\max_{x\in A_i}$ in (11) by taking the maximum at only a small set of cubature points x_k;
- we have replaced the integral $\int_{A_j} K(x,y)\varepsilon(y)dA_y$ in (10) by $\overline{\varepsilon}_j \int_{A_j} K(x,y)dA_y$ with $\overline{\varepsilon}_j \geq \max_{y\in A_j} |\varepsilon(y)|$.

The first approximation was made in order to efficiently estimate the maximum over a patch. We have found this approximation to be acceptable in most of the cases if an integration rule is used with slightly higher than the minimal required precision. The minimal precision is two times the order of the approximation on the patch plus one [7]. This is illustrated in figure 1. In this figure, we have compared $|\sum_j \delta_j(x)|$ with the computed error estimate $\max_{x_k} |\sum_j \delta_j(x_k)|$ at each pixel of the image shown in (a) for various sets of basis functions. When using slightly more precise integration rules, the quality of both the computed coupling coefficients and the error estimates are visibly increased. The development of efficient techniques for better estimating the maximum with non-constant approximations is an area for future research.

The second approximation however leads to a serious overestimation of the error. We observed two reasons for this fact: First, the absolute value of the error $|\varepsilon(y)|$ is much smaller than $\varepsilon_j = \max_{y\in A_j} |\varepsilon(y)| \leq \overline{\varepsilon}_j$ on large parts of the source patches j. This can already be seen indirectly in figure 1. Second, $\varepsilon(y)$ will in general be positive on some parts of the source patch and negative on others. For error propagation (see (5)) the positive and negative parts tend to cancel each other. For these reasons, we have found it more acceptable to ignore error propagation for the estimation of the error as in [12]: $\tilde{\varepsilon}_i = \max_{x_k\in A_i} |\delta_e(x_k) + \sum_j \delta_j(x_k)|$ yields a more realistic, although non-conservative, error estimate. A realistic error estimate is important in the context of the error control algorithm presented in §6.

The main problem with our algorithm is the amount of storage required for the $n \times b$ coefficients $\delta_{j,\beta}(x_k)$ and the n coefficients $K_{j,1}(x_k)$ in addition to the $a \times b$ coupling coefficients $K_{i,\alpha;j,\beta}$. The storage overhead for our algorithm can be significantly reduced by storing only one coefficient

$$\frac{\sum_\beta L_{j,\beta}\delta_{j,\beta}(x_k)}{L_{j,1}}$$

per cubature node x_k on the receiving patch. When this coefficient is multiplied with $L_{j,1}$ the exact $\delta_j(x_k)$ is recovered. When ignoring error propagation, another n coefficients per pair of patches can be saved. With these simplifications, the storage overhead can be reduced to about 30% for a quadratic approximation and 15% for a cubic one, while the resulting error estimates in general still are realistic.

The quality of the error estimates was evaluated by comparing with the difference between a computed radiance solution and a reference radiance solution. In absence of interreflections and occlusion, the true radiance in a scene can be computed analytically and be used as reference solution. In general however, the true radiance cannot be computed exactly and a solution of higher accuracy must be used as reference solution. We found our error estimates to be in general quite trustworthy. This result will be illustrated indirectly when discussing the algorithm for error control in §6.2. Our estimates were sometimes not realistic in situations with partial occlusion. We will explain in §7 how partial occlusion can be handled correctly.

5 An indicator and strategy for hierarchical refinement

In previous hierarchical radiosity algorithms [7, 9], the accuracy of a solution is controlled with a global link error threshold Δ. Before computing light transport from a source patch j to a receiving patch i, first the error Δ_{ij} on the transport is estimated. If this estimate Δ_{ij} for the error, that would be made by computing light transport from j to i, is larger then Δ, one of the patches i or j is subdivided. The procedure is then repeated for the new candidate interactions that result.

An expression for the error Δ_{ij} over a link $i \leftarrow j$ can directly be derived from (10):

$$\Delta_{ij} = \max_{x \in A_i} \left| \delta_j(x) + \int_{A_j} K(x,y)\varepsilon(y) dA_y \right|. \tag{13}$$

The candidate interaction $i \leftarrow j$ will need to be refined if $\Delta_{ij} > \Delta$ and will be accepted for computing light transport otherwise.

If the interaction needs to be refined, a choice has to be made which patch to subdivide. By subdividing the receiving patch i, a better approximation of received radiance as a linear combination of the basis functions defined on the sub-patches can be obtained. On the other side, if the source patch j has been subdivided before, a more accurate representation of radiance will be available on its sub-patches. By subdividing the source patch, propagated error can be reduced. In order to decide which of two interacting patches i or j should be subdivided, $\max_{x \in A_i} |\delta_j(x)|$ can be compared with $\max_{x \in A_i} |\int_{A_j} K(x,y)\varepsilon(y) dA_y|$. If the former is larger, patch i should be subdivided. If the latter is larger, the error propagated from j dominates and j should be subdivided.

For constant basis functions, (13) reduces to

$$\begin{aligned} \Delta_{ij} &\leq \max_{x \in A_i} |\rho_i L_j (F_{ij} - F_{dA_x,j}) + \rho_i \varepsilon_j F_{dA_x,j}| \\ &\leq \rho_i L_j \left(\max_{x \in A_i} F_{dA_x,j} - \min_{x \in A_i} F_{dA_x,j} \right) + \rho_i \varepsilon_j \max_{x \in A_i} F_{dA_x,j} \end{aligned}$$

which is very similar to the refinement indicator proposed in [12]. The indicator (13) is also very similar to the one for bilinear approximations presented in [13].

The same techniques as in §4 can be used for approximately computing expression (13). In particular, $\max_{x \in A_i}$ is approximated by the maximum at the cubature nodes x_k on i and $\varepsilon(y)$ is replaced by an approximation $\tilde{\varepsilon}_j$ for $\max_{y \in A_j} |\delta_e(y) + \sum_s \delta_s(y)|$, where the sum is over all patches s contributing light to j.

6 An algorithm for error control

An error estimate as computed with the algorithm proposed in §4 can be used to judge the acceptability of a given radiance solution. With previous hierarchical radiosity methods, a more accurate solution is obtained by decreasing the global link error threshold Δ. In this section we present a method in which a user directly specifies the maximum allowed absolute radiance error ε instead. The outline of the method is given in §6.1. The algorithm is discussed in §6.2.

6.1 Outline of the algorithm

The basic idea is to *compute* a link error threshold Δ_i for each not further subdivided element i. This link error threshold Δ_i is such that if $\Delta_{ij} < \Delta_i$ for all interactions $i \leftarrow j$ of i and it's parent elements l, the total error $|\varepsilon(x)|$ will be smaller than the a-priori specified error threshold ε for all points $x \in A_i$. Since the real total error $|\varepsilon(x)|$ is not known, a realistic estimate $\tilde{\varepsilon}_i$, computed with the techniques of §4, is used instead.

In the algorithm we propose here, a link error threshold Δ_l is kept with *every* element l, not only the leaf elements: the link error threshold Δ_l of a parent element will be the minimum link error threshold of its sub-elements. At the start of the radiance computations, the link error thresholds Δ_l are initialised to a reasonable value. During each iteration of the radiance computations, refinement is carried out so that the link error Δ_{l,j_l} for every interaction $l \leftarrow j_l$ of each element l will be smaller than the link error threshold Δ_l for the element. After gathering radiance over each interaction, the total error $\varepsilon(x)$ is estimated on the leaf elements i of the patch hierarchies. If the error estimate exceeds the maximum allowable error ε, the link error thresholds Δ_i at the leaf element i and Δ_l at its parent elements l are decreased by a certain factor. Accumulating the error estimates at each level and possibly decreasing the link error thresholds is done with an extended push-pull operation [9, 7, 3] as shown in algorithm 6.1.

As a starting value, we have used

$$\Delta_i^{initial} = (1 - \rho_i)\varepsilon - \max_{x \in A_i} |\delta_e(x)|.$$

The starting value stems from the observation that

$$\varepsilon \leq \frac{|\delta_e(x)| + \left|\sum_j \delta_j(x)\right|}{1 - \rho(x)}.$$

This expression can be proven easily from (5) when considering $\varepsilon = \sup_x |\varepsilon(x)|$ for all points x in the scene. The starting value was chosen large enough to avoid too much refinement during the first iterations of the radiance computation. The radiance solution and error estimates become trustworthy only after a few iterations.

Decreasing the link error thresholds only by a small factor during each iteration will result in a larger number of iterations before reaching the requested accuracy ε. A large factor will eventually lead to over-refinement. In our implementation, we have used a factor 1.4. This factor should probably depend on the order of radiance approximation used. Although we obtained satisfying results with this factor 1.4, a careful analysis in order to determine the optimal factor still needs to be done.

Algorithm 6.1 *Radiosity with error control*

radiosity-with-error-control(PATCHES p, ACCURACY ε)

1. for each patch p,
   (a) initialise radiance to self-emitted radiance;
   (b) initialise the link error threshold $\Delta_p \leftarrow (1-\rho_p)\varepsilon - \max_{x \in A_p} |\delta_e(x)|$.
2. until "convergence", do for each (top-level) patch p,
   (a) refine all interactions $l \leftarrow j_l$ of p and its sub-elements l as described in §5 until all interaction errors Δ_{l,j_l} are smaller than the error threshold Δ_l;
   (b) for all elements l in the patch hierarchy,
      i. gather radiance over the interactions $l \leftarrow j_l$ of l;
      ii. compute a realistic estimate $\tilde{\varepsilon}_l$ for the error on the received radiance with the techniques of §4.
   (c) extended-push-pull(p).

extended-push-pull(ELEMENT l)

1. if l is a leaf element,
   (a) set the total radiance on $l \leftarrow$ self-emitted radiance + received radiance;
   (b) add $\max_{x \in A_l} |\delta_e(x)|$ to $\tilde{\varepsilon}_l$;
   (c) if the estimated error $\tilde{\varepsilon}_l$ on l is larger then the a-priori given error threshold ϵ, decrease the link error threshold Δ_l on l.
2. otherwise, for each child element i of l,
   (a) push down the received radiance on l and add to the received radiance of the child element i;
   (b) add $\tilde{\varepsilon}_l$ to the error estimate $\tilde{\varepsilon}_i$ for the child element;
   (c) invoke recursively extended-push-pull(i) for the child element;
   (d) pull up the total radiance on the child element i to l;
   (e) set $\tilde{\varepsilon}_l \leftarrow \max(\tilde{\varepsilon}_l, \tilde{\varepsilon}_i)$;
   (f) set $\Delta_l \leftarrow \min(\Delta_l, \Delta_i)$.

6.2 Discussion

The algorithm was verified by computing the illumination in a number of scenes at various accuracies and comparing with an unbiased reference solution. We found that the measured discrepancy was always almost everywhere smaller than or about equal to the difference in requested accuracy. This also illustrates the reliability of the error estimates of §4. A typical result is shown in figure 2.

There are however clearly some areas where the measured error is larger than estimated, leading to a less accurate solution. Sometimes, this has to be attributed to underestimation of the discretisation error caused by the point-sampling in the simple method we proposed in §4.2. In other areas, such as under the char in the corner, this is clearly due to incorrect treatment of partial occlusion. Our assumption that other sources of error than discretisation error can be ignored does not hold in these areas. In the next section we present a possible solution.

7 Other sources of error

So far in our discussion, we have assumed that other sources of error than discretisation error could be ignored. In practice however, especially computational errors on the coupling coefficients $K_{i,\alpha;j,\beta}$ may sometimes cause the computed radiance solution and

the error estimate to be unreliable if no special care is taken. The main sources of error in the computation of the coupling and error estimation coefficients, as explained in §4.2, are due to misestimation of visibly and imprecise numerical integration.

In order to deal correctly with visibility, the inner integrals in expression (3), as computed in step 2 of algorithm 4.1, should be restricted to the visible part of the source patch j as seen from cubature node x_k on the receiving patch i. As the visible part of the source patch might be concave, the integral may possibly have to be split in several parts. In the presence of occluders, the functions $K_{j,\beta}(x)$ in algorithm 4.1 will also exhibit first-order discontinuities at the shadow boundaries. In order to correctly deal with these discontinuities, the receiving patch i should be split along the shadow boundaries as described in discontinuity meshing literature. Full discontinuity meshing may however be prohibitively expensive and is not necessary in all cases. If a realistic estimate of visibility error is available, a more practical approach might be to add a term describing visibility error to expression (5) and all other expressions derived from it. When an interaction needs to be refined, the action to be taken in order to reduce the error depends on which of the terms is largest. If the visibility error appaers to be larger than both the approximation error on the receiving patch and the propagated error, partial discontinuity meshing might reduce the link error most.

When there is full visibility between two patches in an interaction, an analytical expression can be used for the inner integral $K_{j,1}(x)$ [2]. To our knowledge, no such analytical expressions exist for $K_{j,\beta}(x)$ with $\beta > 1$, for non-constant approximation on the source patch j. Numerical integration, as used in step 3 of algorithm 4.1 will never yield an exact result. If a realistic estimate for the error due to imprecise numerical integration is available, a good approach will be to add another term to expression (5) and the expressions derived from it. If the error due to imprecise numerical integration dominates in an interaction, the error should be reduced by using a more precise numerical integration rule (cfr. [6]). Another strategy might be to subdivide one of the elements so that the integration error is reduced to the largest extent. More precise numerical integration might also be possible by using irradiance gradients, such as described in [1, 11].

Even without explicitly dealing with visibility error and error due to imprecise numerical integration, the result of the methods described in this paper already seem to be quite reliable, as is illustrated by figure 2. In general, the discretisation error dominates over other sources of error. Dealing more directly with visibility error and errors due to imprecise numerical integration will however lead to even more efficient and reliable algorithms.

8 Conclusion and future directions

We have presented an analysis of discretisation error in radiosity methods. This analysis was used to develop an algorithm for the a-posteriori error estimation and a good refinement indicator and strategy for hierarchical radiosity methods. Combining the error estimation techniques and the refinement strategy, a new hierarchical radiosity method was presented where a user specifies the maximum allowable absolute radiance error rather than a link error threshold in order to control the accuracy of the computed radiance solution.

One weak point of the methods described in this paper is the point-sampling based approach for finding the maximum discretisation error on a receiving patch. Although the technique proposed in §4.2 in general already yields trustworthy results, finding more reliable but still efficient techniques for non-constant approximations is a first area for future research.

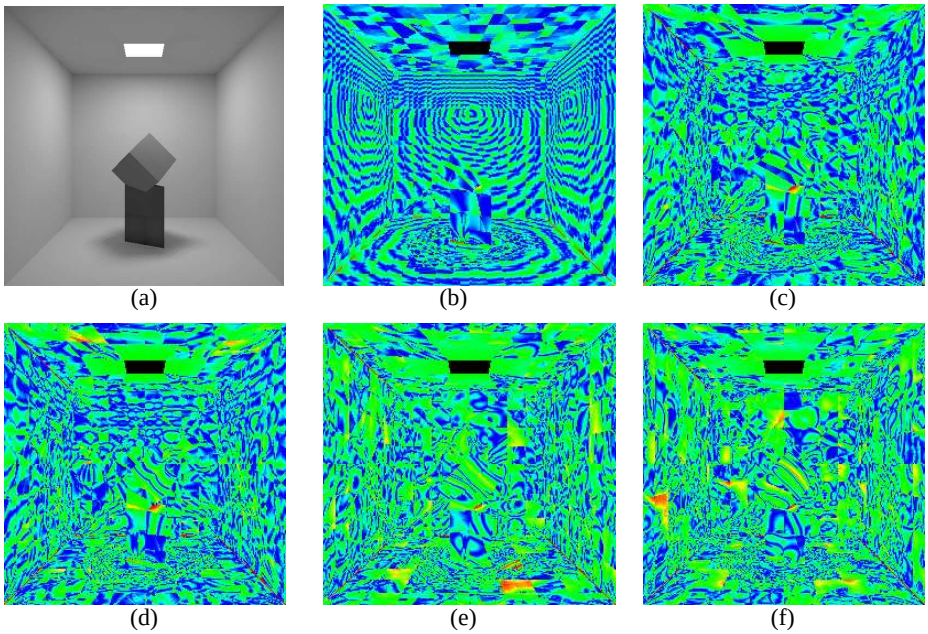
Another area for future research is the extension of the methods in this paper to deal correctly with partial occlusion and imprecise numerical integration as was explained in §7.

A mandatory extension of the methods in this paper is also to integrate them with a clustering algorithm [15, 14]. Without clustering, the methods in this paper, like all gathering radiosity methods, suffer from an expensive initial linking step for environments of realistic complexity.

Other areas for future research include human-perception based error control, the extension to non-diffuse reflection and validation by comparing the computed radiance solutions and error estimates with reality.

References

1. J. Arvo, K. Torrance, and B. Smits. A framework for the analysis of error in global illumination algorithms. In *SIGGRAPH '94 Proceedings*, pages 75–84, July 1994.
2. D. R. Baum, H. E. Rushmeier, and J. M. Winget. Improving radiosity solutions through the use of analytically determined form-factors. In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 325–334, July 1989.
3. Ph. Bekaert and Y. D. Willems. Hirad: A hierarchical higher-order radiosity implementation. In *12 th Spring Conference on Computer Graphics, Bratislava, Slovakia*, June 1996. http://www.cs.kuleuven.ac.be/cwis/research/graphics/.
4. M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, San Diego, CA, 1993.
5. R. Cools and Ph. Rabinowitz. Monomial cubature rules since "Stroud": a compilation. *Journal of Computational and Applied Mathematics*, 48:309–326, 1993.
6. R. Gershbein. A study of integration methods for couplings of galerkin radiosity systems. In *6 th Eurographics Rendering Workshop, Dublin, Ireland*, June 1995.
7. S. J. Gortler, P. Schröder, M. F. Cohen, and P. Hanrahan. Wavelet radiosity. In *SIGGRAPH '93 Proceedings*, pages 221–230, 1993.
8. E. Haines and J. Wallace. Shaft culling for efficient ray-traced radiosity. In *2 nd Eurographics Workshop on Rendering, Barcelona, Spain*, May 1991.
9. P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.
10. N. Holzschuch. *Le contrôle de l'erreur dans la méthode de radiosité hiérarchique*. Ph.D. thesis, Université Joseph Fourier — Grenoble I, France, March 1996.
11. N. Holzschuch and F. Sillion. Accurate computation of the radiosity gradient for constant and linear emitters. In *6 th Eurographics Rendering Workshop, Dublin, Ireland*, June 1995.
12. D. Lischinski, B. Smits, and D. P. Greenberg. Bounds and error estimates for radiosity. In *SIGGRAPH '94 Proceedings*, pages 67–74, July 1994.
13. S. N. Pattanaik and K. Bouatouch. Linear radiosity with error estimation. In *6 th Eurographics Rendering Workshop, Dublin, Ireland*, June 1995.
14. F. Sillion and G. Drettakis. Feature-based control of visibility error: A multiresolution clustering algorithm for global illumination. In *SIGGRAPH '95 Proceedings*, pages 145–152, August 1995.
15. B. Smits, J. Arvo, and D. Greenberg. A clustering algorithm for radiosity in complex environments. In *SIGGRAPH '94 Proceedings*, pages 435–442, July 1994.



(a) (b) (c)

(d) (e) (f)

Fig. 1. (a) shows a simple scene. (b)-(f) show false colour images indicating the ratio of the estimated maximum approximation error on a patch and the measured error for various approximations: (b) constant basis, (c) linear, (d) bilinear, (e) quadratic and (f) cubic. The error was measured by comparing with a solution obtained with a final gather step with per pixel form factors. A green colour indicates that the measured approximation error is about equal to the estimated maximum for the patch. A blue colour indicates that the measured error is smaller (by a factor of 10 for deep blue), a yellow or red colour that the error was underestimated (by a factor 3 and 10 respectively). The approximation error estimate is quite realistic, except in a few places.

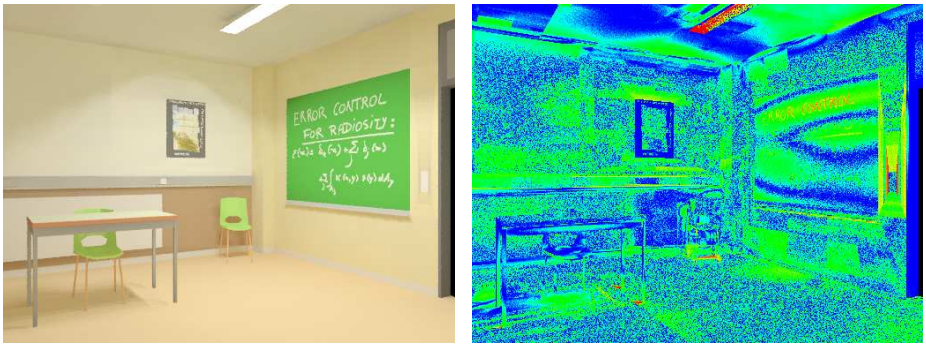


Fig. 2. The left figure shows a part of an office, computed up to 5 lux precision with quadratic approximations. The average luminosity is about 100 lux. The false color image on the right indicates the ratio of measured error and the 5 lux precision. A green colour indicates that the measured error and the precision are of the same magnitude. A blue colour indicates that the measured error was smaller and a yellow or red colour that the error was larger then 5 lux. A Monte Carlo final gather step was used in order to obtain a better unbiased reference solution.