HIRAD: A Hierarchical Higher Order Radiosity Implementation

Philippe Bekaert and Yves D. Willems

Paper submitted for the 12 th Spring Conference on Computer graphics
Bratislava, Slovakia, 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

HIRAD: A Hierarchical Higher Order Radiosity Implementation

Philippe Bekaert 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 describe our hierarchical higher order radiosity implementation, called HIRAD. We present a number of improvements to the original wavelet radiosity method: We propose new basis functions for (convex) quadrilaterals and for triangles offering finer granularity than the usual product-1D bases, a new refinement criterion for hierarchical refinement and a very general strategy for maintaining a consistent hierarchical representation of radiance.

1 Introduction

The radiosity method, first proposed in 1984 [7, 3], is the method of first choice to compute the radiance[1] in an environment exhibiting only diffuse reflection and emission. Two recent breakthroughs are hierarchical radiosity [9] and the use of higher order approximations of the radiance function over each surface patch [15, 16].

In hierarchical radiosity [9, 11], the surface patches describing the geometry of a scene are automatically subdivided into elements over which the radiance can be assumed to be constant within given error bounds. A representation of radiance at multiple levels of detail allows a $\mathcal{O}(k^2 + n)$ time complexity instead of $\mathcal{O}(n^2)$ with k the number of input surface patches and n the total number of surface elements.

Non-constant, or higher-order, radiance approximations [16, 15] often lead to accurate radiance solutions using less surface patches, less computer memory and with less computational effort than when supposing that the radiance is constant on each patch.

These two techniques do not exclude each other: wavelet radiosity [8, 13, 12] is a hierarchical radiosity method with higher-order approximations. In this paper we describe our implementation presenting a number of improvements to the original wavelet radiosity method [8]. In §2 we first present the main ideas behind radiosity computations with non-constant approximations. We present non-product basis functions in §2.2 and discuss in detail how the necessary basis overlap and coupling coefficients can be computed in §2.3 and §2.5. In §3 we present a new solution for some problems arising in hierarchical radiosity computations with non-constant bases. We conclude in §4 and give some directions for future research.

[1] Radiosity $[W/m^2]$ is π $[sr]$ times radiance $[W/m^2 sr]$ in a diffuse environment.

2 Non-constant radiance approximations

2.1 The equations and how to solve them

In order to compute the radiance distribution $L(x)$ in a diffuse environment, an integral equation (see e.g. [3])

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

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. $L_e(x)$ is the self-emitted radiance at a point x. The integral is over the surfaces A of all objects in the scene. $\rho(x)$ is the reflectivity at point x. $\cos\theta_x$ and $\cos\theta_y$ are the cosine of the angle between the line connecting 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 visible and 0 if occluded from each other.

This equation can be solved approximately by discretising the surfaces of the objects in the scene into polygonal patches (we have used convex quadrilaterals as well as triangles) and by assuming that the radiance $L(x)$ on patch i can be well approximated by a linear combination

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

of some set of basis functions $\phi_{i,\alpha}$ defined on the patch. If the basis functions are non-overlapping ($\phi_{i,\alpha}(x) = 0$ if $x \notin$ patch i) and the self-emitted radiance is well approximated by a similar linear combination $L_e(x) = \sum_\alpha E_{i,\alpha}\phi_{i,\alpha}(x)$, the integral equation (1) is well approximated by

$$\begin{aligned}\sum_{i,\alpha} L_{i,\alpha}\phi_{i,\alpha} &= \sum_{i,\alpha} E_{i,\alpha}\phi_{i,\alpha} \\ &+ \sum_{j,\beta}\int_{A_j} K(x,y)L_{j,\beta}\phi_{j,\beta}(y)dA_y\end{aligned}$$

Projection (integration over each patch i of the product of both sides of this equation with each basis function $\phi_{i,\alpha'}(x)$) yields a set of linear equations from which the unknown radiance coefficients $L_{i,\alpha}$ can be solved:

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

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

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

There are as many equations in (2) and unknowns $L_{i,\alpha}$ as there are basis functions $\phi_{i,\alpha}$. The computation of the overlap coefficients and coupling coefficients is discussed in §2.3 and §2.5.

When the basis functions are orthogonal, the basis overlap coefficients will be nonzero only if $\alpha = \alpha'$:

$$O_{i,\alpha,\alpha'} = \delta_{\alpha,\alpha'} N_{i,\alpha}$$

with norms

$$N_{i,\alpha} = \int_{A_i} \phi^2_{i,\alpha}(x) dA_x.$$

In this case (2) reduces to

$$L_{i,\alpha} = E_{i,\alpha} + \frac{1}{N_{i,\alpha}} \sum_{j,\beta} K_{i,\alpha;j,\beta} L_{j,\beta}. \tag{5}$$

It is easy to verify that for a constant basis

$$\phi_{i,\alpha}(x) = \phi_i(x) = 1 \text{ if } x \in A_i \text{ and } 0 \text{ if } x \notin A_i$$

the norm $N_i = A_i$ is the area of patch i and $K_{i,j} = \rho_i A_i F_{ij}$ where F_{ij} is the classical constant-to-constant form factor.

Algorithm 2.1 shows how the equations (2) can be solved using a variant of the Gauss-Seidel iterative method. The progressive refinement radiosity method [2] can be extended to higher order approximations in a similar way. The coupling coefficients $K_{i,\alpha;j,\beta}$ are only computed when they are first needed. The overlap coefficients $O_{i,\alpha,\alpha'}$ are computed once when loading a scene into HIRAD. In order to solve the small linear systems in step 2(a)ii efficiently, the inverse of the basis overlap matrix is precomputed and stored in memory.

Algorithm 2.1 *Gauss-Seidel like solution of (2)*

1. Initialise $L_{i,\alpha} \leftarrow E_{i,\alpha}$;
2. Until convergence
   (a) for each patch i
      i. for each $\alpha' = 1 \ldots a$, with a the size of basis on i, compute
      $$T_{i,\alpha'} = \sum_{j,\beta} K_{i,\alpha';j,\beta} L_{j,\beta}$$
      ii. solve the set of linear equations with unknowns the $R_{i,\alpha}$ (dimension a)
      $$\sum_{\alpha} R_{i,\alpha} O_{i,\alpha,\alpha'} = T_{i,\alpha'}$$
      iii. for $\alpha = 1 \ldots a$
      $$L_{i,\alpha} = E_{i,\alpha} + R_{i,\alpha}$$

2.2 Non-product basis functions

In literature (e.g. [8, 12, 15, 16]), use has been made of products of 1D basis functions, such as Legendre polynomials. The bases, built in HIRAD, have been constructed directly on a quadrilateral or triangle however.

	quadrilaterals
ϕ_1	1
ϕ_2	$\sqrt{3}(2u-1)$
ϕ_3	$\sqrt{3}(2v-1)$
ϕ_4	$3(1-2u-2v+4uv)$
ϕ_5	$\sqrt{5}(1-6u+6u^2)$
ϕ_6	$\sqrt{5}(1-6v+6v^2)$
ϕ_7	$\sqrt{7}(-1+12u-30u^2+20u^3)$
ϕ_8	$\sqrt{15}(-1+6u+2v-12uv-6u^2+12u^2v)$
ϕ_9	$\sqrt{15}(-1+2u+6v-12uv-6v^2+12uv^2)$
ϕ_{10}	$\sqrt{7}(-1+12v-30v^2+20v^3)$

	triangles
ϕ_1	1
ϕ_2	$\sqrt{2}(3u-1)$
ϕ_3	$\sqrt{6}(-1+u+2v)$
ϕ_4	$1.133893419027696(1-4u-4v+20uv)$
ϕ_5	$3.273268353539930(1-6.8u-1.2v+6uv+7u^2)$
ϕ_6	$\sqrt{15}(1-2u-6v+6uv+u^2+6v^2)$
ϕ_7	$-2+30u-90u^2+70u^3$
ϕ_8	$\sqrt{12}(-1+13u+2v-24uv-33u^2+21u^3+42u^2v)$
ϕ_9	$\sqrt{20}(-1+9u+6v-48uv-15u^2-6v^2+7u^3+42u^2v+42uv^2)$
ϕ_{10}	$\sqrt{28}(-1+3u+12v-24uv-3u^2-30v^2+u^3+12u^2v+30uv^2+20v^3)$

Table 1: Basis functions used in HIRAD. The bases are: constant = $\{\phi_1\}$, linear = constant $\cup$ $\{\phi_2, \phi_3\}$, bilinear = linear $\cup$ $\{\phi_4\}$, quadratic = bilinear $\cup$ $\{\phi_5, \phi_6\}$, cubic = quadratic $\cup$ $\{\phi_7, \phi_8, \phi_9, \phi_{10}\}$. No bilinear basis is offered for triangular patches since it seems not be be suited for hierarchical refinement (see §3.4).

Table 1 shows the basis functions $\phi_\alpha(u, v)$ as defined on the unit square $[0, 1]^2$ for quadrilaterals and on the standard triangle $(0, 0), (1, 0), (0, 1)$ for triangular patches. They were obtained by orthogonalising the functions $1, u, v, uv, u^2, v^2, u^3, u^2v, uv^2, v^3$ using the Gram-Schmidt orthogonalisation procedure known from basic algebra. In HIRAD, only bases that are suited for hierarchical refinement (cfr. §3.4) are offered to the user.

Basisfunctions for an arbitrary convex quadrilateral or triangular patch are then obtained by relating points x on the patch to coordinates (u, v) on the unit square or the standard triangle using bilinear parametrisation for quadrilaterals and barycentric parametrisation for triangles (see e.g. [3]).

The main advantage of using non-product basisfunctions is the resulting finer granularity: when using product-Legendre basisfunctions on a quadrilateral, there are only bases with 1, 4, 9, 16, ... basisfunctions. We also have bases of size 3, 6, 10, ... This is an important advantage since the number of coupling coefficients for each pair of (interacting) patches is the product of the number of basis functions on each patch: with e.g. our linear basis, there are 9 coefficients per pair of interacting patches, with a bilinear basis, having one more basis function per patch, 16. Experiments (cfr. §3.5) have however shown that the performance of a linear basis is not much worse than a bilinear basis. The construction is easy to use for other (convex) domains than triangles or quadrilaterals as well.

2.3 Computation of the overlap coefficients $O_{i,\alpha,\alpha'}$

In order to evaluate (4), integrals of the form

$$\int_{A_i} f(x) dA_x$$

on a patch i need to be computed. We first transform to the standard domain for the patch:

$$\int_{A_i} f(x) dA_x = \int_S f(u,v) J_i(u,v) du dv \tag{6}$$

where S is the unit square for quadrilaterals and the standard triangle for triangular patches. $J_i(u,v)$ is the jacobian of the parametrisation relating the standard domain to the patch. For barycentric parametrisation, the jacobian is the area A_i of the triangular patch. For quadrilateral patches with bilinear parametrisation, the jacobian is of the form

$$J_i(u,v) = a_i + b_i u + c_i v$$

where

$$\begin{aligned} a_i &= ((\vec{p_2} - \vec{p_1}) \times (\vec{p_4} - \vec{p_1})) \cdot \vec{n} \\ b_i &= ((\vec{p_2} - \vec{p_1}) \times (\vec{p_4} - \vec{p_3})) \cdot \vec{n} \\ c_i &= ((\vec{p_4} - \vec{p_1}) \times (\vec{p_3} - \vec{p_2})) \cdot \vec{n} \end{aligned}$$

with $\vec{p_1}$, $\vec{p_2}$, $\vec{p_3}$, $\vec{p_4}$ the vertices of the quadrilateral patch (counterclockwise) and $\vec{n}$ the normal to the patch. Note that the area of a quadrilateral patch is given by

$$A_i = \int_{A_i} J_i(u,v) du dv = a_i + \frac{1}{2}(b_i + c_i).$$

Note also that the constants b_i and c_i are 0 for parallelograms, so that the surface area $A_i = a_i$ in this case. The constants a_i, b_i and c_i are computed once when loading a scene in HIRAD and stored for (frequent) later use.

Having basis functions $\phi_\alpha(u,v)$ on the standard domain, it is now easy to compute the norms and overlaps of the corresponding basis functions on the patches. For triangles with barycentric parameterisation, the norms and overlap coefficients of the basis functions $\phi_{i,\alpha}(x)$ are simply the norms and overlap coefficients of the basis functions on the standard domain, multiplied by the patch surface area A_i.

On quadrilaterals however, the overlap coefficients are given by

$$\begin{aligned} O_{i,\alpha,\alpha'} &= \int_{A_i} \phi_{i,\alpha}(x) \phi_{i,\alpha'}(x) dA_x \\ &= a_i \int_{[0,1]^2} \phi_\alpha(u,v) \phi_{\alpha'}(u,v) du dv \\ &+ b_i \int_{[0,1]^2} \phi_\alpha(u,v) \phi_{\alpha'}(u,v) u du dv \\ &+ c_i \int_{[0,1]^2} \phi_\alpha(u,v) \phi_{\alpha'}(u,v) v du dv. \end{aligned} \tag{7}$$

One implication is that the basis functions $\phi_{i,\alpha}(x)$ on a patch i will only be orthogonal if the standard basis functions $\phi_\alpha(u,v)$ are orthogonal and if the patch is a triangle or a parallelogram. Although the basis functions used in HIRAD are orthogonal on the standard domain, they will not be orthogonal after transformation to an irregular quadrilateral and algorithm 2.1 will be needed to compute the radiosity on such patches.

2.4 Cubature rules

Integrals on the standard domain are computed using a suitable numerical cubature formula for the domain:

$$\int_S f(u,v)dudv = \sum_{k=1}^{n} w_k f(u_k, v_k)$$

where w_k are the n weights of the cubature formula and (u_k, v_k) the nodes. Such cubature rules are often constructed to integrate exactly a small class of functions such as a function space spanned by low-order bivariate monomials $u^a v^b$ [5]. In literature [16, 8, 6], product-rules constructed from e.g. 1D Gauß-Legendre quadrature rules have been used. Such product rules are best suited with a product basis on regular quadrilaterals. In this case the monomials that should be integrated exactly in (7) or similar expressions for computing coupling coefficients, are $u^a v^b$ with $0 \le a, b \le k$ with e.g. $k = 1$ for a bilinear basis.

For non-product bases $\{\phi_\alpha(u,v)\}$, more efficient non-product cubature rules exist: the class of functions that should be integrated exactly is the function space spanned by all monomials $u^a v^b$ with a and b satisfying $0 \le a + b \le p$. The precision p of the rule should be at least two times the highest degree $k + l$ of a monomial $u^k v^l$ in the basis functions plus one. The highest degree $k + l$ in a quadratic basis is two, the same as for a bilinear basis. More efficient means that less function evaluations are needed to compute the integral. This is important when computing coupling coefficients (see §2.5) since evaluation of the kernel $K(x,y)$ in (3) is a very expensive operation, involving visibility determination.

An recent overview of cubature formulae known for the unit square and the standard triangle can be found in [4]. For each precision, a rule was chosen with nodes exclusively in the interior of the domain. If there was choice, a rule with only positive weights was chosen. If there were several such rules, the one for which the convex hull of the nodes was largest was used. The C-code for the cubature formulae that were used in HIRAD is available from the first author.

2.5 Computation of the coupling coefficients $K_{i,\alpha;j,\beta}$

Different cubature formulae may be used on the patches i and j, e.g. because i is a quadrilateral and j a triangle. Suppose that we are using a cubature rule

$$\int_{A_i} f(x)dA_x = \sum_{k=1}^{n} w_k f(u_k, v_k) J_i(u_k, v_k) \tag{8}$$

with n nodes on patch i and

$$\int_{A_j} g(y)dA_y = \sum_{l=1}^{m} w'_l g(u_l, v_l) J_j(u_l, v_l)$$

with m nodes on patch j.

When there are a basis functions $\phi_{i,\alpha}(x)$ on a patch i and b basis functions $\phi_{j,\beta}(x)$ on a patch j, the $a \times b$ coupling coefficients

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

between i and j can then be computed simultaneously using algorithm 2.2. The visibility vis(x_k, y_l) in step 1 between the point x_k on patch i and y_l on patch j is determined using ray-casting. The function $K_{j,\beta}(x)$ in step 2 is a function such that $K_j(x) = \sum_{\beta=1}^{b} L_{j,\beta} K_{j,\beta}(x)$ yields the illumination at point x on patch i due to patch j.

Algorithm 2.2 *Computation of the coupling coefficients $K_{i,\alpha;j,\beta}$ between two patches i and j:*

1. For all $k = 1 \ldots n$, $l = 1 \ldots m$, compute
$$K(x_k, y_l) = \rho(x_k)\text{vis}(x_k, y_l)\frac{\cos\theta_{x_k}\cos\theta_{y_l}}{\pi\|x_k - y_l\|^2}.$$

2. For all $k = 1 \ldots n$, $\beta = 1 \ldots b$, compute
$$\begin{aligned} K_{j,\beta}(x_k) &= \int_{A_j} K(x_k, y)\phi_{j,\beta}(y)dA_y \\ &= \sum_{l=1}^{m} w'_l K(x_k, y_l)\phi_\beta(u_l, v_l)J_j(u_l, v_l) \end{aligned}$$

3. For all $\alpha = 1 \ldots a$, for all $\beta = 1 \ldots b$, compute
$$\begin{aligned} K_{i,\alpha;j,\beta} &= \int_{A_i} \phi_{i,\alpha}(x)K_{j,\beta}(x)dA_x \\ &= \sum_{k=1}^{n} w_k \phi_\alpha(u_k, v_k)K_{j,\beta}(x_k)J_i(u_k, v_k) \end{aligned}$$

3 Hierarchical refinement

3.1 Motivation

Hierarchical refinement [9, 8] is a key technique leading to efficient radiosity computations. Consider the situation sketched in figure 1. The radiosity on patch A due to patch C may be well approximated by assuming that it is constant. This is however not true for the radiosity received from patch B (relative distance between the patches is small) and D (shadow boundary). While non-constant bases, as discussed in §2, also lead to better radiosity approximations, the same problems will however still occur.

One primitive solution to these problems would be to construct a mesh which does allow to assume that the radiosity due to all other patches is constant (or well approximated by a linear combination of the basis functions) on the patch. E.g. patch A could be subdivided into smaller patches near the edge it shared with patch B and near the shadow boundary in order to better approximate the radiosity due to patch D. Fine subdivision of patch A near the edge with B is however not needed for the interaction with patch C or D. Neither is fine subdivision near the shadow necessary for the interaction with patch B and C. Such a solution would lead to a lot of unnecessary storage and computation time. A better solution is to automatically construct a mesh in which information about the radiosity on the patches is kept at multiple levels of detail and choosing the right level of detail for each interaction. Wavelet-theory provides the mathematical basis for hierarchical radiosity methods [13].

Such a hierarchical radiosity representation can be obtained by examining each interaction with another patch. If the energy transport from a patch j to a patch i can be computed accurately enough, the interaction between i and j is used. If not, one of both patches i

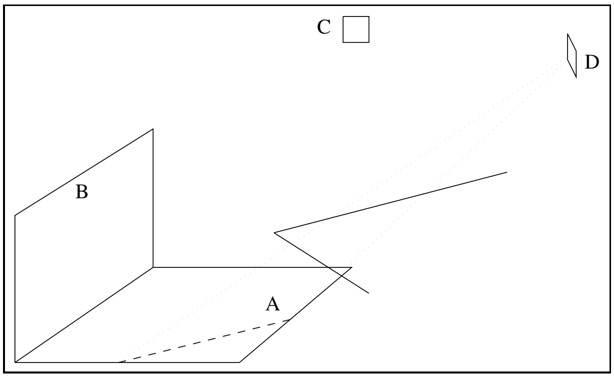

Figure 1: The radiosity on patch A due to patch C can be approximated well by assuming that it is constant. This is however not true for the radiosity received from patch B (relative distance between the patches is small) and D (shadow boundary).

of j needs to be subdivided and a representation of radiosity on the sub-patches much be used. The interaction between i and j is replaced by new interactions with the sub-patches. For each of these new interactions, the question whether energy transport can be computed accurately enough is posed again.

Three questions need to be solved in order to implement such a technique: When is refinement necessary? How to carry out refinement? How to keep the hierarchical representation of radiosity consistent? By "consistent" we mean that the same information (radiance on a patch) is represented at all levels of the hierarchy, the only difference being the resolution at which the information is represented. Our answer to these questions is given in the next three subsections.

3.2 Oracle function

Suppose a patch i receives light from a patch j, on which the radiance is accurately approximated by a linear combination of basis functions $L(y) = \sum_\beta L_{j,\beta}\phi_{j,\beta}(y)$. Refinement of the interaction between i and j will be necessary if the radiance $L_j(x)$ i receives from j cannot be approximated accurately enough by a linear combination of the basis functions $\phi_{i,\alpha}(x)$ defined on the patch i:

The radiance $L_j(x)$ received in a point x on patch i from patch j is given by

$$\begin{aligned} L_j(x) &= \int_{A_j} K(x,y)L(y)dA_y \\ &= \sum_\beta L_{j,\beta} \int_{A_j} K(x,y)\phi_{j,\beta}(y)dA_y \\ &= \sum_\beta L_{j,\beta} K_{j,\beta}(x). \end{aligned} \tag{9}$$

It is approximated by a linear combination

$$\tilde{L}_j(x) = \sum_\alpha \left(\sum_\beta R_{i,\alpha;j,\beta} L_{j,\beta} \right) \phi_{i,\alpha}(x) \tag{10}$$

where the coefficients $R_{i,\alpha;j,\beta}$ for a given β are solved from

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

or, for orthogonal basis functions,

$$R_{i,\alpha;j,\beta} = \frac{K_{i,\alpha;j,\beta}}{N_{i,\alpha}}.$$

The L_1-norm of the difference between (9) and (10) is a good refinement-indicator. Bounding the L_1-norm by the product of area and maximum, an interaction will need to be refined if

$$A_i \max_{x \in A_i} \left| \sum_\beta L_{j,\beta} \left(\sum_\alpha R_{i,\alpha;j,\beta} \phi_{i,\alpha}(x) - K_{j,\beta}(x) \right) \right| > \delta. \tag{11}$$

with δ a predefined link error threshold. In HIRAD, $\delta = 10^{-4} P_{max}$ is used as a default link error threshold with P_{max} the maximum power emitted by a patch in the environment. In the constant radiosity case, the refinement criterium is

$$\rho_i A_i L_j \max_{x \in A_i} |F_{ij} - F_{dA_x,j}| > \delta \tag{12}$$

where $F_{dA_x,j}$ is the point-to-patch form-factor from a point x on patch i to patch j. A link will be refined if there are points x on patch i for which the difference between the average point-to-patch form-factor F_{ij} with the actual point-to-patch form-factor $F_{dA_x,j}$, weighted by reflectivity ρ_i and area A_i of the receiving patch i and radiance L_j of the source patch j, is larger than the threshold δ. Expression (11) is a generalisation for arbitrary basis functions.

In general, the maximum $\max_{x \in A_i}$ should be determined or bounded using some numerical optimisation algorithm, possibly based on irradiance gradients and Hessians as in [10]. In our implementation however, we have only computed the maximum over the nodes x_k of the integration rule (8) used on patch i and used this as an approximate interaction error estimate.

A further simplification of (11) is to keep only one term of the sum over β, e.g. the first ($\beta = 1$). This makes sense only if $L_{j,1}\phi_{j,1}(y)$ can be considered a good approximation for the radiance on patch j, or, i.o.w. the terms for $\beta > 1$ can be considered correction terms to the first. With this simplification, only one number

$$\Delta K_{ij} = \max_{x_k \in A_i} \left| \sum_\alpha R_{i,\alpha;j,1} \phi_{i,\alpha}(x_k) - K_{j,1}(x_k) \right|$$

needs to be stored in order to compute an interaction error estimate between i and j whenever needed. This is needed at least once during each Gauss-Seidel or Jacobi iteration. It is our experience that such a crude simplification still leads to acceptable error estimates. Algorithm 3.1 shows the extension to algorithm 2.2 how an interaction error estimate for deciding whether or not to refine can be computed with little extra work.

The main advantage of this new interaction error estimate (11) is that it is tighter than when estimating the kernel variation with respect to some interpolating polynomial as in [8, 12]. For constant radiosity approximations, estimating kernel variation yields

$$\Delta K_{ij} = \max_{x \in A_i, y \in A_j} K(x,y) \quad - \quad \min_{x \in A_i, y \in A_j} K(x,y)$$

as was done in e.g. [14]. The variation of the kernel $K(x,y)$ over points $y \in A_j$ however doesn't affect the error on patch i, which is rather determined by the variation of the point-x-to-patch-j form-factors for points $x \in A_i$ as in (12), leading to a similar result as in [11]. Equation (11) is a generalisation for non-constant basis functions. It can also be used for realistic, although non-conservative, estimation of total radiance error as will be shown in a forthcoming paper by the authors.

Algorithm 3.1 *Computation of the coupling coefficients $K_{i,\alpha;j,\beta}$ and an error estimate between two patches i and j:*

1,2,3 As in algorithm 2.2;

4 Compute $R_{i,\alpha;j,1}$ by solving

$$\sum_{\alpha} R_{i,\alpha;j,1} O_{i,\alpha,\alpha'} = K_{i,\alpha';j,1};$$

5 Compute and store

$$\Delta K_{ij} = \max_{x_k} \left| \sum_{\alpha} R_{i,\alpha;j,1} \phi_{i,\alpha}(x_k) - K_{j,1}(x_k) \right|$$

Return $\rho_i A_i L_{j,1} \Delta K_{ij}$ as interaction error estimate which can be used to decide whether the interaction between i and j needs to be refined.

3.3 Subdivision strategy

At this time, the refinement strategy built in HIRAD is quite primitive: the largest of two patches i and j is subdivided using regular subdivision as shown in figure 2 and the interaction between i and j replaced by the four new interactions between the smallest of the two patches i and j and the four sub-patches of the largest.

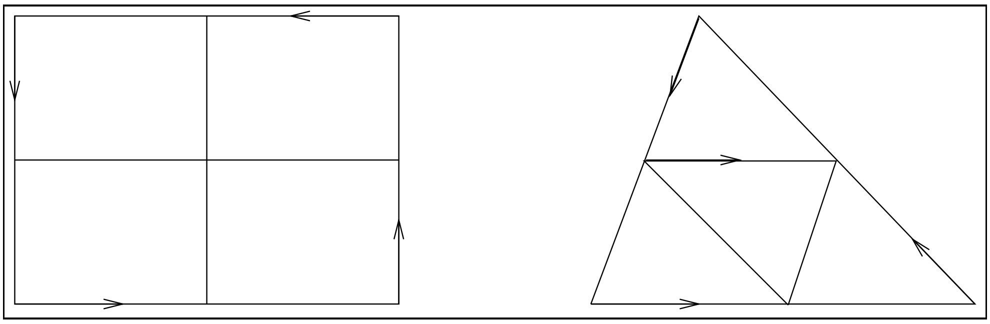

Figure 2: Subdivision of quadrilaterals and triangles in HIRAD. The arrows point in u-direction from the origin $(u, v) = (0, 0)$ chosen on the sub-patch.

In theory, subdivision of the receiving patch i is already sufficient in order to reduce the error over the interaction $i \leftarrow j$ when the error on the source patch j is ignored as is the case here. In practice however, this may lead to very small sub-patches of i interacting with a much larger patch j. In such situations, it might be difficult to perform the integrations for computing the coupling coefficients accurately. We therefore preferred always to subdivide the larger of the two patches i and j. A better strategy requires to compute how much error is propagated from j to i. If the propagated error is more important than the discretisation error as estimated in (11), the source patch j must be subdivided. If the discretisation error dominates, the receiving patch i should be subdivided [11].

A better refinement strategy is not always to use regular subdivision. Subdivision along shadow boundaries [1] will often lead to significant savings in both memory and computation time.

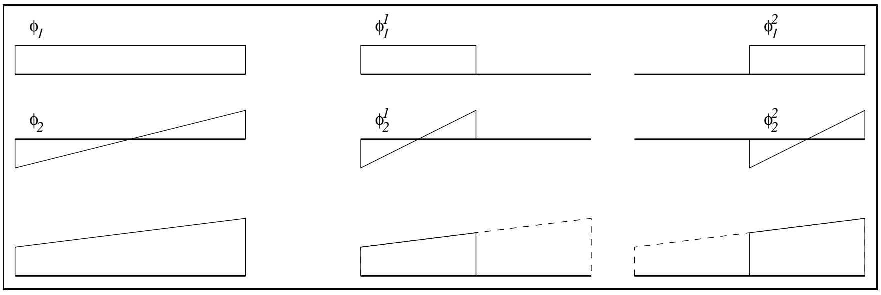

Figure 3: Pushing: On the left hand side of the figure are shown two basis functions defined on an interval. On the right hand side, the scaled and translated basis functions on two subintervals are shown. We required in HIRAD that an arbitrary linear combination of the basis functions on the whole interval must be representable exactly as a linear combination of the basis functions on the subintervals.

3.4 Keeping the hierarchy consistent

Radiance is gathered at virtually every level of the radiance hierarchy on each patch. In order to make the representation of radiance consistent, received radiance will be pushed down to the leaf nodes of the hierarchy (the smallest elements) and then pulled up to the top node. We describe the two operations, push and pull, in the next subsections.

3.4.1 Push: finding a representation on sub-patches of the radiance on a parent patch

When subdividing a patch, a number of sub-patches with index σ is created. Each of this sub-patches is related to the same standard domain as the parent patch. This way, a scaled, translated and possibly rotated version $\phi^{\sigma}_{i,\alpha}$ of the basis functions $\phi_{i,\alpha}$ on the parent patch is defined on each sub-patch σ. By subdividing a patch, a new set of basis functions is defined over the patch that allow radiance to be represented at a finer level of detail. Figure 3 shows a simple 1D analogy.

The question is now how radiance $L(x) = \sum_{\alpha} L_{i,\alpha}\phi_{i,\alpha}(x)$ on the parent patch can be represented in the larger basis defined by the basis functions $\phi^{\sigma}_{i,\alpha}$ on the sub-patches, i.o.w. how can the coefficients $L^{\sigma}_{i,\beta}$ in the expression below be obtained from the given $L_{i,\alpha}$?

$$L(x) = \sum_{\alpha} L_{i,\alpha}\phi_{i,\alpha}(x) = \sum_{\sigma,\beta} L^{\sigma}_{i,\beta}\phi^{\sigma}_{i,\beta}(x) \tag{13}$$

In order to solve this problem, it is required that $\phi_{i,\alpha}$ is a linear combination of the basis functions $\phi^{\sigma}_{i,\beta}$ on the sub-patches

$$\phi_{i,\alpha}(x) = \sum_{\sigma,\beta} h^{\sigma}_{\alpha,\beta}\phi^{\sigma}_{i,\beta}(x) \tag{14}$$

when an arbitrary patch i is subdivided into four sub-patches with index σ using regular subdivision as shown in figure 2. This is not always possible, e.g. for a bilinear basis on triangles. For this reason, no bilinear or other "product" basis on triangles is offered to the user in HIRAD. By substituting (14) in (13) it is easy to prove that then

$$L^{\sigma}_{i,\beta} = \sum_{\alpha} h^{\sigma}_{\alpha,\beta} L_{i,\alpha}.$$

We have computed the coefficients $h^{\sigma}_{\alpha,\beta}$ by writing down expression (14) for a sufficient number of randomly chosen points x on each sub-patch σ and solving the resulting set of

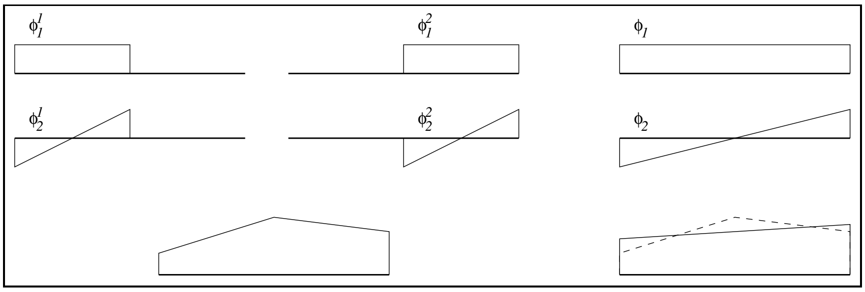

Figure 4: Pull: A linear combination of the basis functions on the sub-patches can in general not be written exactly as a linear combination of the basis functions on the parent patch. We will approximate it by projection on the basis on the parent patch.

linear equations. The advantage of this approach is its generality: there are no assumptions about orthogonality of the basis functions neither about the domain. The strategy can also be used for other then regular subdivision, e.g. to do subdivision along shadow boundaries as in [1]. It can also be used to define hierarchical basis functions on other (convex) domains than triangles or quadrilaterals. although this is an area for further research.

3.4.2 Pull: finding a good approximation on a parent patch for the radiance represented on sub-patches.

It is in general not possible to write a function $L(x) = \sum_{\sigma,\beta} L^{\sigma}_{i,\beta}\phi^{\sigma}_{i,\beta}(x)$, defined on the sub-patches, exactly as a linear combination $\sum_{\alpha} L_{i,\alpha}\phi_{i,\alpha}(x)$ of the basis functions $\phi_{i,\alpha}$ on the parent patch (see figure 4). We will therefore find the best approximation of $L(x)$ on the parent patch by projection: In order to obtain the coefficients $L_{i,\alpha}$ from the given $L^{\sigma}_{i,\beta}$ in

$$L(x) = \sum_{\sigma,\beta} L^{\sigma}_{i,\beta}\phi^{\sigma}_{i,\beta}(x) \approx \sum_{\alpha} L_{i,\alpha}\phi_{i,\alpha}(x)$$

we multiply with each basis function $\phi_{i,\alpha'}$ and integrate over the parent patch:

$$\sum_{\alpha} L_{i,\alpha} \int_{A_i} \phi_{i,\alpha}(x)\phi_{i,\alpha'}(x)dA_x = \sum_{\sigma,\beta} L^{\sigma}_{i,\beta} \int_{A_i} \phi^{\sigma}_{i,\beta}(x)\phi_{i,\alpha'}(x)dA_x$$

The integrals in the left hand side of this expression are the basis overlap coefficients $O_{i,\alpha,\alpha'}$ defined in §2.1. The integrals in the right hand side can be precomputed once for each set of basis functions and standard domain in a similar way the basis overlap coefficients are computed (cfr. §2.3). What results is a small set of linear equations with unknowns $L_{i,\alpha}$. In the case of orthogonal basis functions, the set reduces to

$$L_{i,\alpha} = \sum_{\sigma,\beta} L^{\sigma}_{i,\beta} \cdot \frac{1}{N_{i,\alpha}} \int_{A_i} \phi^{\sigma}_{i,\beta}(x)\phi_{i,\alpha'}(x)dA_x.$$

It is desirable that information on a parent patch remains unchanged after pushing down and pulling back up. This was explicitly verified when computing the necessary coefficients in HIRAD.

3.5 Results

It takes a lot of effort to implement a hierarchical higher-order radiosity code. It is a good question whether it is really worth the effort? We have found that this is indeed the case.

basis	elements	interactions
constant	2570	18653
linear	826	5093
bilinear	826	4361
quadratic	746	3209
cubic	266	845

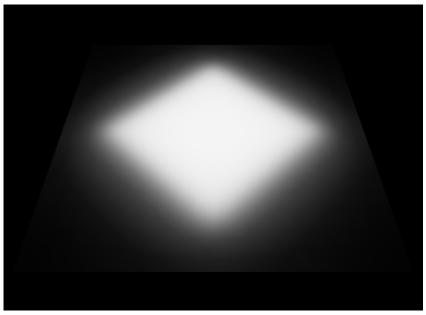

Table 2: Two parallel faces: the user-experienced work is roughly proportional to the number of interactions. A same cubature rule was used for all the experiments. Even when using a cheaper cubature rule for lower order bases, a speedup by a factor 10 can be observed. The needed amount of storage is proportional to both the number of elements and the number of interactions. While the user experienced work for a linear and bilinear basis is about the same, significantly less coupling coefficients need to be stored for the former.

basis	elements	interactions
constant	634	4469
linear	426	2042
bilinear	394	1709
quadratic	338	1181
cubic	306	839

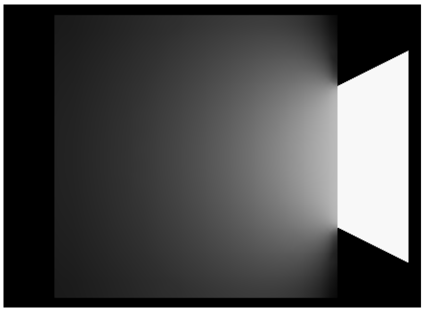

Table 3: Results for two perpendicular patches.

In table 2 and 3 we show the results of an experiment with two simple scenes found in [8]. The first scene consists of a $1\,\mathrm{m}^2$ pure emittor 0.1 m. above a $2\,\mathrm{m}^2$ pure reflector. The second scene consists of a perpendicular emittor and reflector, sharing an edge along which the integral kernel $K(x, y)$ is singular. Especially the last case occurs frequently in practice, e.g. where a wall of a room touches the ceiling or the floor. The tables show the number of elements and interactions resulting for various sets of basis functions and the default choice of δ (see §3.2).

The user-experienced work is roughly proportional to the number of kernel evaluations, which is proportional to the number of interactions. The proportionality factor is related to the product of the number of nodes in the cubature rules used on the receiving and source patch. A more efficient cubature rule (see §2.4) directly affects the computation time this way. Note that although the same cubature rules were used for all the experiments, a rule of lower precision is sufficient for a lower order basis. Even so, a speedup by a factor of about 10 can be observed.

The storage required for the computations is proportional to both the number of elements and the number of interactions. There are asymptotically $k^2 + n$ interactions with k the number of input surface patches and n the total number of resulting elements. The number of coefficients to store for each interaction is the product of the number of basis functions on the receiving and on the source patch (see §2.5). Taking into account that a cheaper cubature rule can be used for a linear basis than for a bilinear basis, the user-experienced work is about equal while significantly less coefficients need to be stored: 9 instead of 16 per interaction.

On the average, for more complex scenes, we often obtained a reduction of computation time by a factor of 2 when using a linear of bilinear basis. These results depend on the required accuracy. For high accuracies, much higher speedups may be obtained. For low accuracies, constant approximations are easy and almost as good as linear bases. We have however not specified what we mean exactly by “accuracy”, although it should be clear that accuracy is controllable by the interaction error threshold δ introduced in §3.2. A more complete discussion of accuracy and error control does unfortunately not fit in this paper, but will be presented in a forthcoming paper by the authors.

4 Conclusions and future directions

We have explained how we have solved a number of problems that arise when doing hierarchical higher order radiosity computations. It takes quite some effort to develop a hierarchical higher order code, but we have found that non constant radiosity approximations lead to a significant speedup, justifying the effort.

While developing this code, we came across a number of unsolved, or only partially solved, problems:

- Discontinuity-driven subdivision instead of regular subdivision. This problem has been addressed in [1] and the results there, for linear multiwavelets, can be generalised for the bases presented in this paper;
- Error control: what is the influence of the interaction error threshold δ in §3.2 on the error on computed radiance? Is there an algorithm to control total radiosity error? This issues will be dealt with in a forthcoming paper by the authors;
- Higher order bases lead to an increase in number of needed coupling coefficients. While a smaller number of patches may be sufficient to compute radiosity up to a given accuracy, the effect may be undone by the increased number of coupling coefficients. In future, we will try to solve this problem by not only using subdivision as a way to increase accuracy, but also by switching to a higher order basis without subdividing;
- The theory in this paper is not restricted to triangles and convex quadrilaterals. Constructing bases for curved surfaces, e.g. NURBS, is a promising an interesting direction for future research;
- Finally, it might be interesting to extend the theory in this paper and the implementation to the case of non-diffuse reflection.

Acknowledgements

The first author would like to acknowledge the financial support by a grant from the Flemish 'Institute for the Promotion of Scientific and Technological Research in the Industry' (IWT#941120).

References

[1] Kadi Bouatouch and S. N. Pattanaik. Discontinuity meshing and hierarchical multiwavelet radiosity. In *Graphics Interface '95*, 1995.

[2] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 75–84, August 1988.

[3] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, San Diego, CA, 1993.

[4] Ronald Cools and Philip Rabinowitz. Monomial cubature rules since "Stroud": a compilation. *Journal of Computational and Applied Mathematics*, 48:309–326, 1993.

[5] P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Academic Press, New York, 1975.

[6] R. Gershbein. A study of integration methods for couplings of galerkin radiosity systems. In *Eurographics Rendering Workshop 1995*, June 1995.

[7] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 212–22, July 1984.

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

[9] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, July 1991.

[10] Nicolas Holzschuch. *Le contrôle de l'erreur dans la méthode de radiosité hiérarchique*. Ph.D. thesis, Université Joseph Fourier — Grenoble I, France, March 1996.

[11] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. In *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 67–74, July 1994.

[12] Sumanta N. Pattanaik and Kadi Bouatouch. Linear radiosity with error estimation. In *Eurographics Rendering Workshop 1995*. Eurographics, June 1995.

[13] Peter Schröder. *Wavelet Algorithms for illumination computations*. Ph.D. thesis, Princeton University, November 1994.

[14] Brian E. Smits, James R. Arvo, and David H. Salesin. An importance-driven radiosity algorithm. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 273–282, July 1992.

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

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