EUROGRAPHICS '97 / D. Fellner and L. Szirmay-Kalos
(Guest Editors)

Volume 16, (1997), Number 3

Using Subdivision on Hierarchical Data to Reconstruct Radiosity Distribution

Leif Kobbelt Marc Stamminger Hans-Peter Seidel

Computer Science Department, University of Erlangen-Nürnberg
Am Weichselgarten 9, 91058 Erlangen, Germany
`kobbelt@informatik.uni-erlangen.de`

Abstract

Computing global illumination by finite element techniques usually generates a piecewise constant approximation of the radiosity distribution on surfaces. Directly displaying such scenes generates artefacts due to discretization errors. We propose to remedy this drawback by considering the piecewise constant output to be samples of a (piecewise) smooth function in object space and reconstruct this function by applying a binary subdivision scheme. We design custom taylored subdivision schemes with quadratic precision for the efficient refinement of cell- or pixel-type data. The technique naturally allows to reconstruct functions from non-uniform samples which result from adaptive binary splitting of the original domain (quadtree). This type of output is produced, e.g., by hierarchical radiosity algorithms. The result of the subdivision process can be mapped as a texture on the respective surface patch which allows to exploit graphics hardware for considerably accelerating the display.

1. Introduction

Radiosity elements are a well-known technique to compute global illumination in a diffuse environment [10]. Let a scene be given by n patches P_i some of which emit radiosity B_i^e by themselves (light sources) and others only reflect a certain portion ρ_i of incoming light. The radiosity equation characterizes the radiosity equilibrium in the scene, i.e., the state when all n^2 patch-to-patch interactions have a constant flow of energy. This flow is controlled by *form factors* $F_{i,j}$ which describe the fraction of the radiosity emitted by patch P_j that is received by P_i. With B_i the total radiosity of P_i, the equilibrium is given by

$$[B_i]_i \;=\; [B_i^e]_i + \mathrm{diag}(\rho_i)\,[F_{i,j}]_{i,j}\,[B_j]_j, \qquad (1)$$

where we implicitly assume that B_i, B_i^e and ρ_i are constant across the patches.

However, for image synthesis this assumption is not appropriate. To achieve realistic visual appearance it is not sufficient to compute the total amount of radiosity which is transported between (large) patches but it is also necessary to obtain more detailed information about its local distribution. Hence, one has to derive an *oracle* which estimates how close to constant the true radiosity function B_i over the receiving patch P_i is. If the function varies by more than a prescribed tolerance, P_i is split and interactions between its children and the other patches yield an improved approximation of B_i which is piecewise constant on each sub-patch (element). Further recursive splits are possible.

The form factor integral $F_{i,j}$ is usually estimated by a low order cubature formula. In order to improve this approximation, the sending patch may be split as well. The ostensible asymmetry in our exposition, i.e., that the receiver is subdivided to improve the approximation of B_i (more samples on P_i) while the sender is subdivided to improve the accuracy of those samples, is balanced by the fact that the radiosity transport between patches is bidirectional and while $F_{i,j}$ describes the flow from P_j to P_i, $F_{j,i}$ accounts for the opposite direction.

Once the radiosity distributions B_i have been computed, rendering the scene is done by mapping the illumination information either directly onto the geometric model or onto its projection into image space. To accomplish this at interactive rates, a host of techniques [12, 25, 23, 18, 19, 20, 11] to maximize the visual quality with a minimum number of elements has been proposed.

We distinguish pre- and post-processing techniques, i.e., methods which modify the tesselation of the patches P_i before solving the system (1) and other methods which use the solution of (1) without interference and reconstruct the true distribution as reliable as possible. A third class of techniques improves the approximation power for individual elements by using a representation for the B_i with higher order basis functions (*wavelet radiosity*).

The discontinuity meshing approach [18, 19] tries to identify the non-smooth features of a piecewise smooth distribution B_i and subdivides patches non-uniformly to separate smooth regions. Although this approach leads to good results and improves the quality of the approximation compared to uniformly refining schemes, a post-processing stage becomes necessary to handle the non-uniform meshes.

Typical post-processing methods construct a tesselation of the patches P_i from the radiosity equation's solution B_i with a color value assigned to each node. Smooth color transitions across each patch are then obtained by piecewise linear interpolation either in the object space or in image space. While the linear interpolation is usually performed by today's graphics hardware, the tesselation still has to be done by a software post-processor.

In this paper we propose a post-processing algorithm to reconstruct smooth radiosity distributions from hierarchical, piecewise constant, illumination data. We especially consider quadtree-type data obtained through the Hierarchical Radiosity algorithm. A radiosity texture is generated which can be mapped onto the patches in real time by appropriate graphics hardware that is no longer left to expensive high-end workstations but can be found on current semi-professional PCs as well. We believe that the texture based approach is more promising than Gouraud-shading techniques since it allows to increase the level of shading detail without increasing the complexity of the scene, i.e., the number of triangles.

Besides the presented application, our method which is based on the adaption of stationary subdivision techniques to non-uniform hierarchical data, can be applied to more general reconstruction tasks. In fact, it can be expected to perform well on any set of quadtree data where deep refinement indicates regions of steep gradient and high frequency detail.

2. Hierarchical Radiosity

Hierarchical radiosity (HR) [14] is an efficient algorithm to approximate the solution of the radiosity equation. The resolution up to which it computes the interactions between each pair of patches is individually adapted to the gradient of the corresponding form factor kernel. HR generates a quadtree for every original patch P_i with each node corresponding to a sub-region of P_i (a finite element) which collects the contributions of those senders P_j (or their sub-regions) whose form factor is considered sufficiently constant across that element.

To update the radiosity functions B_i on all levels, the radiosity scattered over each quadtree is first *pushed* downwards, i.e., each node's radiosity is added to its four children. This operation is reasonable because each node's radiosity is assumed to be constant (otherwise the oracle would have decided to split). After this step has been performed only the leaves of the quadtree carry some radiosity and, patched together, they represent the currently best (piecewise constant) approximation of the true radiosity function B_i within the patch P_i.

To obtain a complete multi-resolution representation of the radiosity functions, pushing is followed by a *pull* step. This operation estimates the total radiosity of each node as a function of its children's values. This step is necessary to allow each patch to interact on all levels.

2.1. Degrees of Freedom

The general set-up for HR leaves several degrees of freedom. The most important ones are the particular way how form factors are computed and the design of the oracle.

Form Factors Let A_i be the area of patch P_i. Neglecting occlusions and assuming equal distribution, the amount of radiosity which is transported from patch P_j to P_i is determined by the form factor

$$F_{i,j} \;=\; \frac{1}{A_i}\int_{P_i}\int_{P_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2}\, dA_j\, dA_i, \tag{2}$$

where for two points $\mathbf{p}$ and $\mathbf{q}$ on P_i and P_j respectively, the angles θ_i and θ_j are spanned by the connecting ray $\overline{\mathbf{pq}}$ and the surface normals, and r is the distance $\|\mathbf{p}-\mathbf{q}\|$.

Since the evaluation of (2) is computationally quite complex one usually approximates $F_{i,j}$ by differently careful heuristics. In the simplest case $F_{i,j}$ is approximated by a one point cubature formula

$$F_{i,j} \;\approx\; A_j\,\frac{\cos\theta_i \cos\theta_j}{\pi r^2}, \tag{3}$$

where the angles and the distance are measured from the centers of both patches P_i and P_j. This approximation is accurate if the sizes of both patches are small compared to their distance and if their radiosity is approximately constant.

If one of these assumptions does not hold, we obtain a better approximation if at least one of the integrals is approximated by a higher order cubature formula, i.e.,

$$F_{i,j} \;\approx\; \frac{A_j}{A_i}\int_{P_i} \frac{\cos\theta_i \cos\theta_j}{\pi r^2}\, dA_i \tag{4}$$

accounts for variation of the kernel on the receiver and

$$F_{i,j} \;\approx\; \int_{P_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2}\, dP_j \tag{5}$$

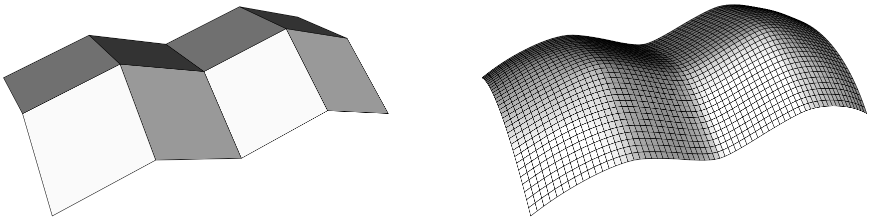


Figure 1: *Original control mesh $\mathcal{P}_0$ and the four times subdivided mesh $\mathcal{P}_4$ as an approximation of $\mathcal{P}_\infty$.*

for the sender respectively. In either case the interactions are assumed to happen between one patch and the center of the other.

Although the differential form factors (4) and (5) seem to be analogue, there is a fundamental difference in their interpretation: Speaking in terms of radiosity-*energies* $E_i = A_i B_i$, the integral (4) stands for the *collection* of incoming energy on P_i while the integral (5) is an *averaging* of emitted energy from P_j (hence the asymmetric normalization in (2)).

We further observe that while both formulae (4) and (5) were solely introduced to improve the approximation of $F_{i,j}$ and to cope with the singularity of the kernel in special configurations, formula (5) can be considered as a point sample functional of the radiosity function. The assumption of equally distributed radiosity B_j on the sending patch P_j implies in this case that $F_{i,j} B_j$ represents the *exact* radiosity received on an infinitesimal region around the center of P_i. The value obtained by this point evaluation is then taken to be the average value over P_i.

We will exploit this fact, i.e. that the radiosity values obtained by (5) are more reliable at the element's centers, later when we reconstruct a smooth radiosity function. Notice that the pull operation of HR eventually destroys this property of the values computed by (5). Hence, we have to omit the *last* pulling step if we want to use non-leaf data of HR's output for interpolatory reconstruction. The push operation does not affect the accuracy if the oracle works correctly since this guarantees that only the constantly distributed fraction of the radiosity is moved down through pushing.

Oracle This procedure decides whether the radiosity emitted by the current sender P_j is to a sufficient degree equally distributed across the current receiver P_i. If the oracle answers affirmative then no further subdivision is necessary (for this interaction) and the HR recursion stops. If the oracle advises to refine then HR splits either the receiver or the sender whichever is expected to have more impact on the approximation of the form factor integral. Formulae to compute the gradient of the form factor kernel exactly can be found in [13].

In [14] this oracle is implemented by simply testing whether the differential form factor (4) is below some threshold ε. This oracle works correctly if P_i and P_j are small compared to their distance.

A more reliable way to estimate the local variation of radiosity is to sample $F_{i,j}$ according to (5) at several locations in P_i and approximate the gradient of the radiosity function by an appropriate divided difference scheme [17]. The additional computational costs can be minimized if the samples lie on a uniform grid and can be re-used on the next refinement level.

3. Uniform Subdivision Schemes

Subdivision schemes are classically used for the efficient generation of smooth surfaces [7, 16, 15]. Starting with control points $\mathbf{p}^0_{i,j} \in \mathbb{R}^3$ which form an initial (regular) mesh of vertices $\mathcal{P}_0 = [\mathbf{p}^0_{i,j}]$, we iteratively compute refined meshes $\mathcal{P}_m = [\mathbf{p}^m_{i,j}]$ with vertices becoming more and more dense. If the rules by which the new vertices $\mathbf{p}^{m+1}_{i,j}$ are computed from $\mathbf{p}^m_{i,j}$, are chosen appropriately, the sequence $\mathcal{P}_m$ converges to a smooth limit surface $\mathcal{P}_\infty$. However, $\mathcal{P}_m$ for a moderate m usually provides a sufficient accurate approximation of that surface (cf. Figure 1 where the scheme of [15] is used).

For our application, we are not exactly interested in surface design but the subdivision paradigm of *reconstructing a smooth function by iterative refinement* is still applicable. The initial control data are radiosity samples $\mathbf{c}^0_{i,j}$ and a smooth surface corresponds to a smooth transition over the patch (which is the domain of the radiosity function).

Consider a patch P_i which is uniformly split into $2^m \times 2^m$ elements each carrying an amount of radiosity $\mathbf{c}^m_{i,j}$. A subdivision operator maps the "mesh" of elements (or cells) $\mathcal{C}_m = [\mathbf{c}^m_{i,j}]$ to a refined mesh $\mathcal{C}_{m+1}$ with $2^{m+1} \times 2^{m+1}$ cells, i.e., each cell $\mathbf{c}^m_{i,j}$ is split into $\mathbf{c}^{m+1}_{2i,2j}$, $\mathbf{c}^{m+1}_{2i+1,2j}$, $\mathbf{c}^{m+1}_{2i,2j+1}$, and $\mathbf{c}^{m+1}_{2i+1,2j+1}$ (cf. Figure 2).

We derive the refinement rules for the subdivision schemes by taking the tensor product of appropriate univariate schemes. A univariate scheme is defined by a finite sequence of coefficients $[\alpha_i]$. To subdivide the *sequence* of values $\mathcal{C}_m = [\mathbf{c}^m_i]$, the $\mathbf{c}^{m+1}_j$ are computed by

$$\mathbf{c}^{m+1}_j := \sum_i \alpha_{j-2i}\, \mathbf{c}^m_i. \tag{6}$$

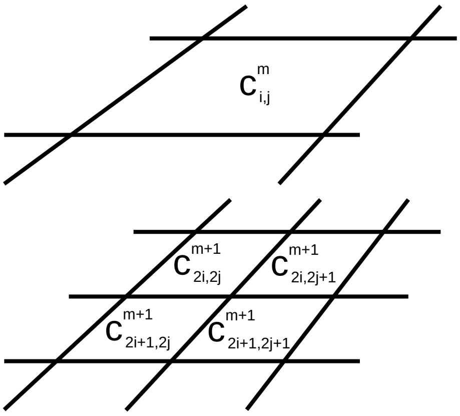


Figure 2: *Applying subdivision to cell data splits each cell into four children.*

This definition contains two individual rules triggered by the parity of j [8]. Since the sequence $[\alpha_i]$ is finite, the sum in (6) is finite as well, i.e., every new value is a linear combination of old values from a local vicinity. At the boundaries some of the neighboring $\mathbf{c}_i^m$ might be undefined. In this case we have to estimate them by extrapolation, e.g. $\mathbf{c}_{-1}^m$ can be found by evaluating the lowest order polynomial interpolating $\mathbf{c}_0^m, \ldots, \mathbf{c}_k^m$.

From a sequence $[\alpha_i]$ we easily derive the matrix $[\beta_{i,j}] = [\alpha_i]^T [\alpha_j]$ which defines a bivariate subdivision scheme by the four rules

$$\mathbf{c}_{k,l}^{m+1} := \sum_i \sum_j \beta_{k-2i,l-2j}\, \mathbf{c}_{i,j}^m \tag{7}$$

again depending on the parity of k and l.

The limit function $\mathcal{C}_\infty$ generated by iteratively applying a subdivision scheme $[\beta_{i,j}]$ can be represented as a linear combination of (shifted) *scaling functions*, each associated with one cell of the original mesh $\mathcal{C}_0$. The support of this scaling function is the region of non-vanishing values in $\mathcal{C}_\infty$ when the scheme is applied to the Dirac-type input $[\mathbf{c}_{i,j}^0] = \delta_i\,\delta_j$. The support of the scaling function describes the region of $\mathcal{C}_\infty$ which is influenced by a single cell $\mathbf{c}_{i,j}^0$.

3.1. Some Useful Schemes

The smoothness of the limit curves or surfaces of a subdivision scheme can be derived from certain conditions on the coefficients $[\alpha_i]$. This has been thoroughly investigated in [2]. We do not go into the details of the analysis of subdivision schemes but we just give three examples for schemes which are useful in the context of our application. More material about subdivision schemes can be found in [8].

Quadratic B-Splines If we choose the mask $[\alpha_i^B] = \frac{1}{4}[1,3,3,1]$, the limit function $\mathcal{C}_\infty$ will be the piecewise quadratic B-spline curve which is defined by the control vertices $\mathcal{P}_0$ [1]. The corresponding tensor product scheme is given by the matrix

$$[\beta_{i,j}^B] = \frac{1}{16}\begin{pmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{pmatrix}.$$

This notation has to be read as follows: According to (7) we select, depending on the parity of the indices k and l, the odd or even rows and columns of the matrix $[\beta_{i,j}^B]$. This will leave four non-vanishing coefficients for each subdivision rule:

$$\begin{aligned}
\mathbf{c}_{2i,2j}^{m+1} &:= \frac{1}{16}(9\mathbf{c}_{i,j}^m + 3\mathbf{c}_{i-1,j}^m + 3\mathbf{c}_{i,j-1}^m + \mathbf{c}_{i-1,j-1}^m) \\
\mathbf{c}_{2i+1,2j}^{m+1} &:= \frac{1}{16}(9\mathbf{c}_{i,j}^m + 3\mathbf{c}_{i,j-1}^m + 3\mathbf{c}_{i+1,j}^m + \mathbf{c}_{i+1,j-1}^m) \\
\mathbf{c}_{2i,2j+1}^{m+1} &:= \frac{1}{16}(9\mathbf{c}_{i,j}^m + 3\mathbf{c}_{i-1,j}^m + 3\mathbf{c}_{i,j+1}^m + \mathbf{c}_{i-1,j+1}^m) \\
\mathbf{c}_{2i+1,2j+1}^{m+1} &:= \frac{1}{16}(9\mathbf{c}_{i,j}^m + 3\mathbf{c}_{i+1,j}^m + 3\mathbf{c}_{i,j+1}^m + \mathbf{c}_{i+1,j+1}^m)
\end{aligned}$$

Due to the symmetry of $[\beta_{i,j}^B]$ the rules for the four children are obtained by rotation about $\mathbf{c}_{i,j}^m$.

The B-spline scheme has only positive coefficients and minimal support which makes it numerically very stable. No additional oscillations are introduced (variation diminishing property). However, the scheme is not interpolatory and the deviation of the limit function $\mathcal{C}_\infty$ from the initial data may be significant. Hence, this scheme should be applied in situations where the samples $\mathbf{c}_{i,j}^0$ are biased by noisy errors.

Polynomial Interpolation Better interpolatory properties provide schemes which are derived from a heuristic based on local polynomial interpolation [4]. We derive a univariate scheme where the cell $\mathbf{c}_i^m$ of the unrefined sequence $\mathcal{C}_m$ is split into $\mathbf{c}_{2i}^{m+1}$ and $\mathbf{c}_{2i+1}^{m+1}$. Cell values are represented by point samples at the center. To obtain the values for the new cells, we construct the quadratic polynomial f_i which uniformly interpolates $f_i(j) = \mathbf{c}_{i+j}^m$ for $j = -1, 0, 1$. This interpolant is then evaluated at $\mathbf{c}_{2i}^{m+1} := f_i(-\frac{1}{4})$ and $\mathbf{c}_{2i+1}^{m+1} := f_i(\frac{1}{4})$. More precisely, we get the interpolant $f_i(x) = a + bx + cx^2$ by solving the Vandermonde system

$$\begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \mathbf{c}_{i-1}^m \\ \mathbf{c}_i^m \\ \mathbf{c}_{i+1}^m \end{pmatrix}$$

and obtain $\mathbf{c}_{2i}^{m+1} := f_i(-\frac{1}{4})$ by

$$\begin{aligned}
\mathbf{c}_{2i}^{m+1} &:= (1, -\frac{1}{4}, \frac{1}{16}) \begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{c}_{i-1}^m \\ \mathbf{c}_i^m \\ \mathbf{c}_{i+1}^m \end{pmatrix} . \\
&= \frac{1}{32}(5\mathbf{c}_{i-1}^m + 30\mathbf{c}_i^m - 3\mathbf{c}_{i+1}^m)
\end{aligned}$$

The analog refinement rule for $\mathbf{c}_{2i+1}^{m+1}$ is

$$\mathbf{c}_{2i+1}^{m+1} := \frac{1}{32}\left(-3\,\mathbf{c}_{i-1}^{m} + 30\,\mathbf{c}_{i}^{m} + 5\,\mathbf{c}_{i+1}^{m}\right)$$

due to symmetry. Merging both rules, we get the coefficients

$$[\alpha_i^P] = \frac{1}{32}\,[-3, 5, 30, 30, 5, -3]$$

and the matrix

$$[\beta_{i,j}^P] = \frac{1}{1024}\begin{pmatrix} 9 & -15 & -90 & -90 & -15 & 9 \\ -15 & 25 & 150 & 150 & 25 & -15 \\ -90 & 150 & 900 & 900 & 150 & -90 \\ -90 & 150 & 900 & 900 & 150 & -90 \\ -15 & 25 & 150 & 150 & 25 & -15 \\ 9 & -15 & -90 & -90 & -15 & 9 \end{pmatrix}$$

for the bivariate tensor product scheme. Each individual rule uses a symmetric neighborhood of 3 × 3 cells around $\mathbf{c}_{i,j}^m$ to compute its four children $\mathbf{c}_{2i,2j}^{m+1}$, $\mathbf{c}_{2i+1,2j}^{m+1}$, $\mathbf{c}_{2i,2j+1}^{m+1}$, and $\mathbf{c}_{2i+1,2j+1}^{m+1}$. Again the rules are rotations of each other.

Although this scheme is derived from polynomial interpolation, the limit function obtained by iterative application of the subdivision operator, does not interpolate the original data exactly. The reason for this is that cell values are assigned to the centers of the cells and the midpoint of $\mathbf{c}_{i,j}^m$ is not a midpoint of any cell from a finer subdivision level. Nevertheless the approximation error can be bounded to about 3% of the true function value.

We can reduce the approximation error to less than 10^{-5} by enlarging the subdivision mask $[\beta_{i,j}^P]$ to 8×8 without affecting the polynomial precision of the scheme. The construction is based on the idea to add a pertubation-scheme which vanishes on data sampled from quadratic polynomials. The simplest candidate for this is the third forward difference operator. Scaling the pertubation yields an additional degree of freedom which can be exploited to minimize the interpolation error. The resulting univariate scheme is given by

$$[\alpha_i^{P+}] =$$
$$\frac{1}{32}\,[\lambda, -3-\lambda, 5-3\lambda, 30+3\lambda, 30+3\lambda, 5-3\lambda, -3-\lambda, \lambda]$$

with $\lambda \approx -0.31158$. The corresponding bivariate scheme is obtained by $[\beta_{i,j}^{P+}] = [\alpha_i^{P+}]^T [\alpha_j^{P+}]$.

Average Interpolation This scheme was first proposed in [5] and it is based on the assumption that the given values represent averages or integrals of the sampled function. Interpolatory subdivision in this context has to guarantee $\mathbf{c}_i^m = \mathbf{c}_{2i}^{m+1} + \mathbf{c}_{2i+1}^{m+1}$, i.e., the energy is distributed but its total amount is conserved. Again, we construct a quadratic polynomial f_i which interpolates the given data

$$\int_j^{j+1} f_i(x)\,dx = \mathbf{c}_{i+j}^m, \qquad j = -1, 0, 1$$

and the new values on the refined level are given by

$$\mathbf{c}_{2i}^{m+1} := \int_0^{1/2} f_i(x)\,dx, \qquad \mathbf{c}_{2i+1}^{m+1} := \int_{1/2}^{1} f_i(x)\,dx.$$

Explicitly computing the coefficients leads to

$$[\alpha_i^A] = \frac{1}{16}\,[-1, 1, 8, 8, 1, -1]$$

and

$$[\beta_{i,j}^A] = \frac{1}{256}\begin{pmatrix} 1 & -1 & -8 & -8 & -1 & 1 \\ -1 & 1 & 8 & 8 & 1 & -1 \\ -8 & 8 & 64 & 64 & 8 & -8 \\ -8 & 8 & 64 & 64 & 8 & -8 \\ -1 & 1 & 8 & 8 & 1 & -1 \\ 1 & -1 & -8 & -8 & -1 & 1 \end{pmatrix}$$

Remark There is a fundamental difference between the four proposed schemes. For $[\beta_{i,j}^B]$, $[\beta_{i,j}^P]$ and $[\beta_{i,j}^{P+}]$ the data $\mathbf{c}_{i,j}^m$ are function values and for $[\beta_{i,j}^A]$ the $\mathbf{c}_{i,j}^m$ are integrals (which is average function value *times* the size of the interval). Hence, if the input data is constant, $\mathbf{c}_{i,j}^m = 1$ for all i, j then the schemes $[\beta_{i,j}^B]$, $[\beta_{i,j}^P]$ and $[\beta_{i,j}^{P+}]$ will generate $\mathbf{c}_{i,j}^{m+1} = 1$ for all i, j while $[\beta_{i,j}^A]$ generates $\mathbf{c}_{i,j}^{m+1} = \frac{1}{4}$ since binary subdivision splits every element into four equally sized children.

3.2. Approximation power

To measure the approximation power of the subdivision schemes when reconstructing the radiosity function, we run tests on data sampled from a known reference function and analyze the asymptotic behaviour of the approximation errors. The used reference function is the illumination of a square by a point light source placed above it.

All proposed schemes have quadratic precision but only the interpolatory one, $[\beta_{i,j}^{P+}]$, can be expected to have cubic convergence $O(h^3)$. The B-spline scheme $[\beta_{i,j}^B]$ is known to have quadratic convergence $O(h^2)$ which is the rate by which a B-spline's control polygon converges to the curve under subdivision [21].

We generated a $2^9 \times 2^9$ grid of exact function values as the reference radiosity. Then we subsampled this data on a $2^k \times 2^k$ grid an performed $9-k$ refinement steps by the respective subdivision scheme. The L_2 approximation error e_k was computed by comparing the result of the subdivision scheme to the reference data. Figure 3 shows the results for the three schemes $[\beta_{i,j}^B]$, $[\beta_{i,j}^A]$, and $[\beta_{i,j}^{P+}]$. The shown values are estimates of the approximation orders, i.e., $\log_2(e_{k+1}/e_k)$.

4. Reconstructing Hierarchical Data Using Subdivision Schemes

We implement the algorithm to reconstruct the radiosity function as a post-processing step. The HR algorithm puts

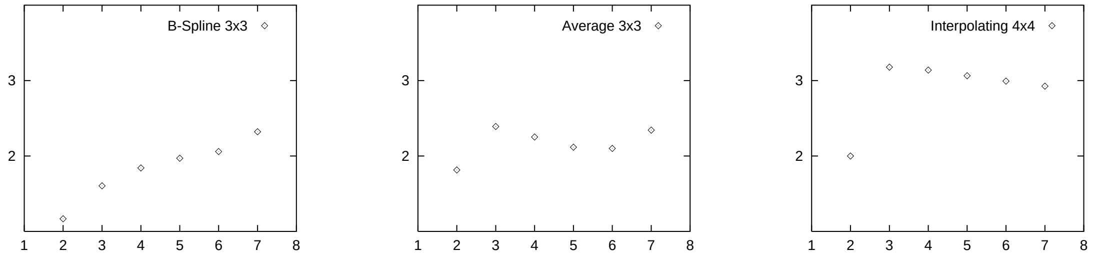


Figure 3: *Convergence rates for the approximation errors of reconstructions by the quadratic B-Spline scheme* $[\beta^B_{i,j}]$ *(left), the average interpolating scheme* $[\beta^A_{i,j}]$ *(center), and the interpolating scheme* $[\beta^{P+}_{i,j}]$ *(right).*

out a quadtree for every original patch of the scene. Depending on which formula we used to approximate the form factors, different subdivision schemes apply.

If we use (3) then the form factor has been approximated very coarsely and the scheme $[\beta^B_{i,j}]$ is appropriate since it reduces the artefacts that result from low quality sampling. Formula (4) gives good estimates of the total radiosity collected over the whole area of the receiving patch and hence the average interpolating scheme $[\beta^A_{i,j}]$ is well suited. Finally, (5) provides accurate samples of the local radiosity at the centers of the elements and therefore the interpolatory schemes $[\beta^P_{i,j}]$ and $[\beta^{P+}_{i,j}]$ are the best choice.

Subdivision schemes as described in Section 3 perform uniform subdivision on uniform meshes. Hence, we have to modify the basic algorithm to allow non-uniform input meshes as well. Within our application, we can restrict the type of non-uniformity to adaptively refined quadtree data.

An intuitive attempt for the reconstruction algorithm is the following:

We refine the quadtree level by level, i.e. we fill in the missing nodes in a breadth first order. At any time the collection of the leaves represent the currently best available reconstruction of the radiosity function. After all existing leaves at level m have been split, we proceed to the next level. The children of a level m node are computed by the subdivision rules $[\beta_{i,j}]$ if they do not already exist. The necessary information from neighboring nodes (on the same level) is always available due to the above assumption on the order by which the split operations are performed. If a node from level m has been split by HR then the data stored in its child nodes are considered to be the true values and no further computations are necessary.

It turns out that the above algorithm leads to surprisingly poor results (cf. Figure 4). To explain this effect we have to investigate the consequences of the push/pull operation in more detail.

For every quadtree representing a patch P_i, the push operation propagates the radiosity down to the leaves of the tree. This actually accumulates all the information about the radiosity distributions B_i as computed by HR. The pull step then computes a complete pyramid of down sampled approximations. This transformation implicitly assumes a decomposition of the radiosity function into box-functions (characteristic functions of the elements). Such a pre-reconstruction obviously conflicts with our post-reconstruction based on a smoothing subdivision scheme.

In order to reconstruct the radiosity function properly, we have to make sure that the influence of the input data is restricted to a fixed sized neighborhood *on the same level*. The source for the artefacts in Figure 4 is that the scaling functions resulting from the schemes of Section 3.1 have a larger support than the box-functions.

Consider a quadtree leaf $\mathbf{c}^m_{i,j}$ at level m. It should influence the reconstructed function exactly on that part of the domain which is covered by the corresponding scaling function's support. If we perform "pull" to estimate the value $\mathbf{c}^{m-1}_{i/2,j/2}$ at the $(m-1)$st level then the actual region which is affected by $\mathbf{c}^m_{i,j}$ becomes twice as big: it is the support of the scaling function associated with $\mathbf{c}^{m-1}_{i/2,j/2}$. To reconstruct the radiosity function correctly, we therefore have to omit the final pull step (earlier pull steps are necessary for HR to work correctly).

After pushing without pulling, only the leaves carry valid data. Hence, when applying the above algorithm, me may encounter situations where some of the neighbors of a cell $\mathbf{c}^m_{i,j}$ which is to be split next, do not carry valid data because they are inner nodes of the original quadtree. In that case we are not allowed to use information to be found deeper in tree because this would be a form of pulling. Instead we have to estimate the missing value by extrapolating data from the same level (just as if it was the outer boundary of the original patch). Figure 4 shows the corrected reconstruction.

Notice that the modified subdivision scheme maintains polynomial precision on non-uniform quadtree data if we use low order polynomials for the extrapolation. This is an important feature since it is crucial for the approximation order of the subdivision scheme (cf. Sect. 3.2).

Since the subdivision rules are applied only when the chil-

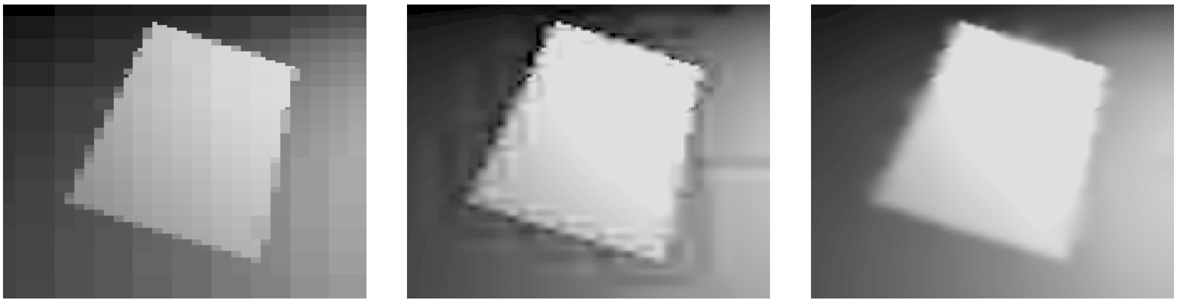


Figure 4: *Radiosity distribution with sharp shadow boundary on a quadtree partition (left). We show the corresponding radiosity function reconstructed by a smoothing subdivision scheme with pulling (middle) and without pulling (right).*

dren's values are *not* contained in HR's output, we automatically make sure that no original information on the radiosity distribution is pertubated. In fact, the algorithm only uses data from the quadtree's leaves and therefore ignores data generated by the final pull step of HR. Hence, our reconstruction works as an independent post-processing step and no modification of the HR implementation is necessary.

Remark We emphasize on the observation that HR uses adaptive splitting to refine the sampling rate of the radiosity distribution only in regions with relevant detail. Consequently, at edges where elements from different generations meet, the level of detail (of the radiosity function) changes. Our reconstruction algorithm guarantees that information propagates across these edges only on the finer of the scales while larger basis functions are chopped off. By this we are able to smoothly interpolate the data while preserving sharp detail according to the local level of detail.

5. Implementation

If we use one of the subdivision schemes proposed in Section 3.1, it is sufficient to know the values of the eight direct neighbors $\mathbf{c}^m_{i\pm1,j\pm1}$ when splitting the cell $\mathbf{c}^m_{i,j}$. We implement a single split operation by a procedure `split(N)`. If one of the nodes which are needed to compute the linear combinations (7) does not carry valid information, i.e., if it is an inner node of the HR quadtree, we substitute it by the value $\mathbf{c}^m_{i,j}$ (constant extrapolation) or by a value obtained through least squares linear extrapolation of the valid data among the $\mathbf{c}^m_{i\pm1,j\pm1}$.

The whole reconstruction algorithm can be implemented by a procedure `reconstruct(N,i)` which subdivides the nodes of a quadtree `N` down to the level `i`. This procedure has to be called for each quadtree and $\mathtt{i} = 0,1,\ldots,M-1$ (*breadth first*). Its output is a uniformly refined quadtree whose leaves describe the radiosity function in a resolution of $2^M \times 2^M$ pixels. This data can be mapped as a texture on the respective patch for display.

Compared to the computational complexity of HR, the computing time of this post-processing step can be neglected. Besides the superior quality, this algorithm is also easier to implement than reconstruction techniques based on Gouraud-shading since the tedious tesselation of the non-uniformly subdivided patches is no longer necessary. As texture hardware is meanwhile available not only for high-end graphics computers, radiosity mapping with textures could soon replace the still often applied Gouraud-shaded triangles.

6. Hierarchical vs. Wavelet Radiosity

Wavelet radiosity (WR) [22, 9] is an elegant and general approach to efficiently approximate the solution of the radiosity equation. It is a generalization of HR and uses higher order basis functions to approximate the form factors by a hierarchical decomposition. By this, WR obtains a significantly higher order of approximation which accelerates the convergence if patches are further split. However, the computational costs to evaluate a single interaction in WR are much higher than for HR and hence (in terms of approximation error per cycle) WR beats HR only if very small error tolerances are prescribed.

Another advantage of WR is that the reconstructed radiosity function is smooth which gives visually better results than the blocky appearance of HR's output. This drawback of HR is what we propose to overcome by a post-processing subdivision scheme and thus making HR competitive to WR up to moderate accuracy approximations — especially for complex scenes.

As a special case of WR, HR's output can be considered as the coefficients of the radiosity function's representation with respect to a basis of box-functions. However, the formulae (3) and (5) indicate that those coefficients are obtained by sampling the form factors at the centers of the elements and assuming (by asking the oracle) that the functions do not vary by more than a prescribed tolerance. Hence, the approximations of the function values are most reliable near the centers of the patches.

From this point of view it is reasonable to replace the original box-functions by (interpolating) smooth scaling functions when it comes to the reconstruction. The radiosity at the centers of the patches will still be as computed by HR, but the artefacts at element boundaries will be removed.

If formula (4) is used, then the same argument applies because the information contained in HR's output (the average radiosity in this case) is preserved while visual quality is improved by smoothing.

7. Example

In this section we analyze the behavior of our reconstruction scheme by applying it to data obtained from the lighting simulation of an office test scene. Figure 5 shows four versions of the scene, reconstructed from the same radiosity solution.

The upper row shows the unmodified quadtree obtained from the Hierarchical Radiosity algorithm. In the center images, radiosity is reconstructed using Gouraud-shaded triangles [12] which makes a modified tesselation necessary. Due to the linear interpolation in image space, Mach-band effects lead to clearly visible artefacts and make the underlying triangulation visible, especially along shadow boundaries.

In the lower left image, the subdivision method based on quadratic B-Splines is used to reconstruct the radiosity function on the patches. The uniform radiosity grid obtained from the algorithm is represented by a texture and mapped on the corresponding patches. Neither Machbanding effects nor the underlying topological patch structure are visible. For the lower right image, the interpolation scheme $[\beta_{i,j}^{P+}]$ was used. Since the Hierarchical Radiosity algorithm does not produce reliable data, especially for visibility, the smoothing B-Spline scheme produces better results at shadow boundaries.

In a real-time animated walk through the scene of Figure 5, we could achieve frame rates of 5 frames per second with Gouraud-shaded triangles (212 ms/frame), 10 frames with flat shading (86 ms/frame) and 20 frames with radiosity textures (48 ms/frame). The running times were measured on a Silicon Graphics O2 station. The reason for the high frame rate with radiosity textures is the minimal number of faces.

8. Conclusions and Future Work

We presented a technique to use binary subdivision schemes known from surface design for the reconstruction of the radiosity distribution on the patches of a globally illuminated scene. The proposed algorithm can process adaptively refined quadtree data as produced by hierarchical radiosity. The results are globally smooth radiosity functions while sharp detail is conserved through a proper treatment of the additional information stored in the structure of the quadtrees. Depending on the particular approximation of the form factors, different subdivision schemes lead to best results. The reconstructed radiosity function can be used for texture mapping and exploiting graphics hardware accelerates the display significantly.

Several custom taylored subdivision schemes are constructed which perform uniform refinement of cell-type data. Particularly the new scheme $[\beta_{i,j}^{P+}]$ which is interpolatory, provides an interesting contribution of its own.

The presented refinement schemes for hierarchical quadtree data can be applied to reconstruction tasks from any kind of samples provided that sharp features are indicated by adaptively refined sample grids. Especially anti-aliasing on adaptive image data, e.g., from adaptive ray-tracing, could be an area for future applications.

Future research to further improve the behavior of the smoothing scheme has to address the problem that some ringing can still be observed in regions where sharp local detail is of the same frequency as the sampling rate induced by the size of the leaves in the quadtree computed by HR. We plan to remedy this difficulty by applying low pass filters whose filter shape (e.g. the stop band) is adapted to the local level of detail.

References

1. G. Chaikin, *An Algorithm for High Speed Curve Generation*, Computer Graphics and Image Processing 3 (1974), pp. 346–349
2. A. Cavaretta / W. Dahmen / C. Micchelli, *Stationary Subdivision*, Memoirs Amer. Math. Soc. 93, 1991
3. M. F. Cohen / J. R. Wallace, *Radiosity and Realistic Image Synthesis*, Academic Press, 1993
4. G. Deslauriers / S. Dubuc, *Symmetric Iterative Interpolation Processes*, Constr. Approx. 5 (1989), pp. 49–68
5. D. Donoho, *Smooth Wavelet Decomposition with Blocky Coefficient Kernels*, Recent Advances in Wavelet Analysis, L. Schumaker / G. Webb (eds.), Academic Press, 1993, pp. 259–308
6. G. Drettakis / E. Fiume, *Accurate and Consistent Reconstruction of Illumination Functions Using Structures Sampling*, Computer Graphics Form 12 (1993), Eurographics '93 issue, pp. 273–284
7. N. Dyn / J. Gregory / D. Levin, *A Butterfly Subdivision Scheme for Surface Interpolation with Tension Controll*, ACM Trans. Graph. 9 (1990), pp. 160–169
8. N. Dyn, *Subdivision Schemes in Computer Aided Geometric Design*, Advances in Numerical Analysis II, Wavelets, Subdivision and Radial Functions, W.A. Light (ed.) Oxford University Press, 1991, pp. 36–104
9. S. Gortler / P. Schröder / M. Cohen / P. Hanrahan,

Wavelet Radiosity, Computer Graphics (Proc. Siggraph '93), pp. 221–230

10. C. M. Goral / K. E. Torrance / D. P. Greenberg, *Modeling the Interaction of Light Between Diffuse Surfaces*. Computer Graphics (Proc. Siggraph '84), pp. 212–222

11. M. Hughes / A. Lastra / E. Saxe, *Simplification of Global-Illumination Meshes*, Computer Graphics Forum 15 (1996), Eurographics '96 issue, pp. 339–345

12. B. Von Herzen / A. H. Barr, *Accurate Triangulations of deformed, intersecting surfaces* Computer Graphics (Proc. Siggraph '87), pp. 103–110

13. N. Holzschuch / F. Sillion, *Accurate Computations of the Radiosity Gradient for constant and linear emitters*, Rendering techniques '95 (Proc. of the EG Rendering Workshop '95), pp. 186–195

14. P. Hanrahan / D. Salzman / L. Aupperle, *A rapid hierarchical radiosity algorithm*, Computer Graphics (Proc. Siggraph '91), pp. 197–206

15. L. Kobbelt, *Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology*, Computer Graphics Forum 15 (1996), Eurographics '96 issue, pp. 409–420

16. C. Loop, *Smooth Subdivision Surfaces Based on Triangles*, University of Utah, 1987

17. Dani Lischinski / Brian Smits / Donald P. Greenberg, *Bounds and error estimates for radiosity*, Computer Graphics (Proc. SIGGRAPH '94), pp. 67–74

18. D. Lischinski / F. Tampieri / D. P. Greenberg, *Discontinuity Meshing for Accurate Radiosity*, IEEE Computer Graphics & Applications 12(6), November, pp. 25–39

19. D. Lischinski / F. Tampieri / D. P. Greenberg, *Combining Hierarchical Radiosity and Discontinuity Meshing* Computer Graphics (Proc. Siggraph '93), pp. 199–208

20. K. Myszkowski / T. L. Kunii, *Texture Mapping as an Alternative for Meshing During Walkthrough Animation*, Proc. Fifth Eurographics Workshop on Rendering '94, pp. 375–388

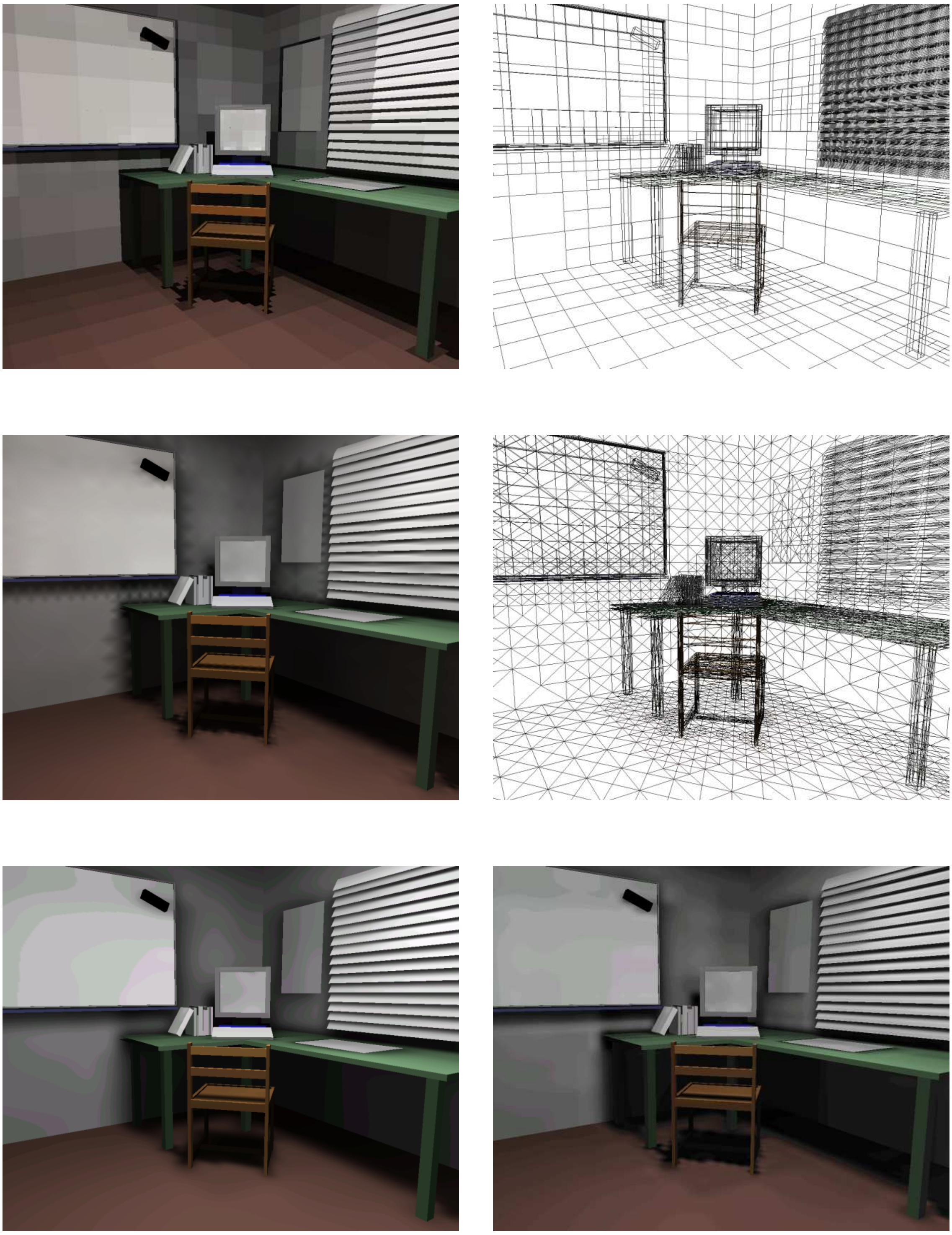
21. H. Prautzsch / L. Kobbelt, *Convergence of subdivision and degree elevation*, Adv. Comp. Math. 2 (1994), J.C. Baltzer, pp. 143–154

22. P. Schröder / S. Gortler / M. Cohen / P. Hanrahan, *Wavelet Projections for Radiosity*, Proc. Fourth Eurographics Workshop on Rendering '93, pp. 95–104

23. D. Salesin / D. Lischinski / T. DeRose, *Reconstructing illumination functions with selected discontinuites*, Proc. Third Eurographics Workshop on Rendering '92, pp. 99–112

24. F. X. Sillion / C. Puech, *Radiosity & Global Illumination*, Morgan Kaufmann, 1994

25. G. Ward / P. Heckbert, *Irradiance Gradients*, Proc. Third Eurographics Workshop on Rendering '92, pp. 85–98



Kobbelt et al., Figure 5. Upper row: Radiosity solution rendered with flat shading (left), quadriliteral mesh (right). Middle row: Same solution rendered with Gouraud shaded triangles (left) and triangle mesh (right). Bottom row: Reconstruction with B-Splines (left) and interpolatory scheme (right).