



**HAL**  
open science

# Space-Time Hierarchical Radiosity with Clustering and Higher-Order Wavelets

Cyrille Damez, Nicolas Holzschuch, François X. Sillion

► **To cite this version:**

Cyrille Damez, Nicolas Holzschuch, François X. Sillion. Space-Time Hierarchical Radiosity with Clustering and Higher-Order Wavelets. *Computer Graphics Forum*, 2004, 23 (2), pp.129 - 141. 10.1111/j.1467-8659.2004.00747.x . inria-00379367

**HAL Id: inria-00379367**

**<https://inria.hal.science/inria-00379367>**

Submitted on 28 Apr 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Space-Time Hierarchical Radiosity with Clustering and Higher Order Wavelets

Cyrille DAMEZ (a), Nicolas HOLZSCHUCH (b), François X. SILLION (b)

(a) Max-Planck-Institut für Informatik, Saarbrücken, Germany  
(b) ARTIS, GRAVIR/IMAG-INRIA, Grenoble, France

---

## Abstract

*We address in this paper the issue of computing diffuse global illumination solutions for animation sequences. The principal difficulties lie in the computational complexity of global illumination, emphasized by the movement of objects and the large number of frames to compute, as well as the potential for creating temporal discontinuities in the illumination, a particularly noticeable artifact. We demonstrate how space-time hierarchical radiosity, i.e. the application to the time dimension of a hierarchical decomposition algorithm, can be effectively used to obtain smooth animations: first by proposing the integration of spatial clustering in a space-time hierarchy; second, by using a higher order wavelet basis adapted for the temporal dimension. The resulting algorithm is capable of creating time-dependent radiosity solutions efficiently.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

---

## 1. Introduction

Global illumination techniques have reached the stage where they allow the calculation of high quality images of 3D scenes, complete with subtle lighting and inter-reflection effects. It is therefore natural to try and use them for the production of animation films, or more generally in all lighting jobs related with special effects, such as combining synthesized elements with live action film footage. Unfortunately, global illumination techniques remain typically expensive to use, even more so in the case of frame-by-frame lighting calculations.

In this paper, we present a fully developed version of the Space-Time Hierarchical Radiosity, an algorithm aimed at computing view independent global illumination simulations for animated scenes. It works with scenes containing moving solid objects, whose trajectory are known beforehand and computes a hierarchical radiosity solution for the entire animation, instead of frame-by-frame.

The hierarchical formulation for radiosity is extended by introducing a fourth dimension, time, along with the three spatial dimensions. All four dimensions are treated in the same way, meaning that we can refine an interaction either

in time or in space. This results in few computations being done in areas where there is little temporal variation of the illumination, while areas with rapid variation of the illumination will be computed to full precision. The hierarchical formulation guarantees a compact representation of the temporal variations of the radiosity function: as a result, the entire animation is computed much faster than by performing a complete radiosity solution for each frame.

In a way similar to the original Hierarchical Radiosity algorithm<sup>1</sup>, the efficiency of the Space-Time Hierarchical Radiosity algorithm depends on the depth of the space-time hierarchy built during computations: the deeper the hierarchy, the more efficient the algorithm. This suggests that it should prove especially beneficial for complex scenes.

We have previously presented a preliminary version of the Space-Time Hierarchical Radiosity algorithm<sup>2</sup>. We present here a fully developed algorithm. In particular, two major issues are addressed: first, we have modified the algorithm so that it uses linear wavelets for the time dimension, to improve the temporal continuity of the animations produced. Second, we have combined space-time radiosity with clustering, thus enabling the algorithm to work more efficiently

and on larger scenes. This allows us to use this algorithm in a range of complexity where its benefits can be fully realized.

The paper is organized as follows: in the next section, we briefly discuss previous work in time-dependent illumination of animated scenes, and review the shortcomings of our preliminary approach. Then, in section 3, we detail our fully developed algorithm. In section 4, we provide an analysis of the performances of Space-Time Hierarchical Radiosity, compared to our previous approach as well as frame-by-frame computations. Finally, in section 5 we draw our conclusions and trace directions for future work.

## 2. Background and Motivations

### 2.1. Global Illumination Algorithms for Animations

Several algorithms to compute global illumination images have been proposed since the pioneering work of Goral *et al.*<sup>3</sup>. As the performance of the said algorithms and the computing power of graphics workstations improved, several propositions have been made to extend these algorithms and reduce the overwhelming cost of computing globally illuminated animations. Two classes of applications can be distinguished:

- interactive methods, which render new solutions quickly, usually by reusing previous computation results as much as possible. They aim at offering as fast a feedback as possible in response to changes made by the user. Interactive methods have been developed for Hierarchical Radiosity<sup>4,5</sup>, Path Tracing<sup>6,7</sup> and Particle Tracing<sup>8,9,10,11</sup>.
- offline methods, where the objects movements are supposed continuous and known a priori. They aim at rendering high quality animations, and therefore should ensure a constant quality. Global Monte Carlo methods<sup>12</sup> and Particle Tracing<sup>13</sup> algorithms have been proposed to compute high-quality global illumination animations.

A study of the current state of the art for both types of animated global illumination algorithms can be found in<sup>14</sup>. In this section, we discuss briefly only the methods allowing the computation of higher quality animations.

Surprisingly, the case of high quality animations has received little attention when compared to the amount of work devoted to interactive algorithms in the literature. Indeed, most interactive algorithms could be used to compute a movie sequence. However, the quality of the resulting animation may not always be satisfying, as these methods were designed to satisfy real-time constraints instead of animation quality criteria.

In particular, the accumulated errors due to the incremental nature of most interactive algorithm may cause distracting artifacts. The resulting frames quality may seem acceptable when considered separately. Nevertheless, discontinuities in the shading of surfaces may appear between two

consecutive frames. The interactive global illumination algorithm proposed by Wald *et al.*<sup>10</sup>, though it recomputes a complete global illumination solution for each frame independently, is fast enough to converge to a good quality view-dependent solution within a couple seconds. However, in order to avoid light flickering due to its stochastic nature, it requires to use the same random seeds from one frame to the other, which only ensures temporal continuity of lighting for light paths that do not intersect moving objects.

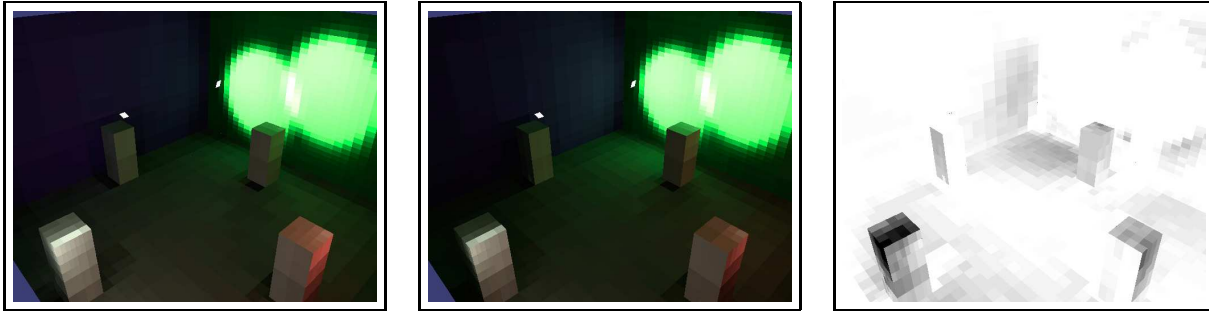
It also seems natural to try to capitalize on the knowledge of objects movement to enhance the quality of the rendered animation. Therefore, it makes sense to consider high quality animations rendering as a separate problem, and to develop algorithms specifically designed to solve it. Myszkowski *et al.*<sup>13</sup> extended the density estimation photon tracing algorithm to the case of animated scenes, allowing the use of photons for several consecutive frames. The decision to extend or contract the segment of time during which a given sample is valid is based on a perception-based Animation Quality Metric. It is used to measure the perceived difference between consecutive frames, and therefore reduce the flickering which results from the stochastic noise. However, to this date, this method is based on a fixed mesh and lack some kind of adaptive refinement scheme. Therefore, the spatial resolution of the solutions computed is limited.

Martin *et al.*<sup>15</sup> proposed a two pass algorithm based on hierarchical radiosity. During the first pass, a coarse hierarchical solution for the complete animation is computed incrementally. Then during the second pass, the resulting mesh and link structure is used to efficiently perform final gathering, assigning to each space-time mesh element a high resolution texture movie representing the radiosity of this patch during the corresponding interval of time. Since this algorithm efficiently solves the problem of high quality final gathering for animated scenes, which our approach doesn't address, both methods can be seen as complementary. In particular, the algorithm of Martin *et al.* doesn't make use of a cluster hierarchy during the first pass, which limits its application to very simple scenes. However, we show in Section 3.3 how to solve this particular issue. As a consequence, coupling both approaches seems promising.

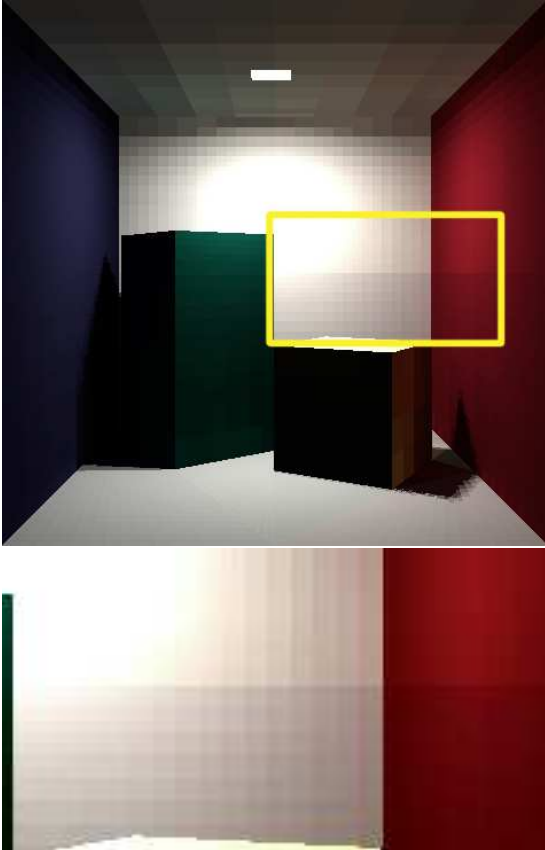
### 2.2. Previous Work on the Space-Time Hierarchical Radiosity Algorithm

In order to reduce the cost of diffuse global illumination computations for animations, we introduced in a previous publication<sup>2</sup> the Space-Time Hierarchical Radiosity algorithm. Our preliminary algorithm lacked several key features that would enable its use on scenes with complex geometry or lighting condition. In particular, it did not feature a way to extend the object hierarchy above the surfaces level (an approach that is commonly referred as *clustering*<sup>16,17</sup>).

Moreover, distracting "jumps" in the illumination could



**Figure 2:** Example of temporal discontinuities. Visualisations of the scene at  $t = \frac{1}{2} - \epsilon$ , at  $t = \frac{1}{2} + \epsilon$  and the difference image. The illumination of the entire scene has been modified in a single frame interval.



**Figure 1:** A radiosity discontinuity is clearly visible between the upper and lower half of the walls

appear in scenes where important changes in indirect lighting occur along time. Similar discontinuities can be observed in the spatial dimension for the classical static Hierarchical Radiosity algorithm (cf. Figure 1), when the refinement oracle used is based only on the global evaluation of the

error. Oracles designed to take into account the distribution of the error on the receiving elements can remove such discontinuities<sup>18, 19</sup>.

Additionally, the piecewise constant function basis we used did not allow a proper distribution of the approximation errors on the mesh elements. As a consequence, low intensity light exchanges, such as indirect bounces, were either insufficiently refined, or overly refined when the refinement threshold was reduced.

We demonstrate such discontinuities using a scene for which our preliminary algorithm performed in an obviously unsatisfying manner. This scene is composed of four boxes in a closed room with colored walls (red, blue, green and grey), illuminated by rotating spotlights. As a consequence the lighting in the scene is mostly indirect, and varies in large proportions. The resulting animation presents important lighting discontinuities in time, in particular at the main subdivisions of the animation time interval, as illustrated by Figure 2. Such discontinuities must obviously be reduced in order to make our algorithm of practical use.

### 3. The Space-Time Hierarchical Radiosity Algorithm

#### 3.1. The Space-Time Radiosity Equation

We want to compute the radiosity function  $B(p, t)$  for each point  $p$  at each time  $t$  defined over  $(\mathcal{S} \times \mathcal{T})$  where  $\mathcal{S}$  is the set of all points on all surfaces of the scene and  $\mathcal{T}$  is the time interval over which we want to compute our animation. We define the following functions:

$r : (\mathcal{S} \times \mathcal{S} \times \mathcal{T}) \rightarrow \mathbb{R}$  the distance from point  $p$  to point  $q$  at time  $t$

$\theta : (\mathcal{S} \times \mathcal{S} \times \mathcal{T}) \rightarrow [0, \pi]$  the angle between the outgoing normal at  $p$  and the direction from  $p$  to  $q$  at time  $t$

$v : (\mathcal{S} \times \mathcal{S} \times \mathcal{T}) \rightarrow \{0, 1\}$  the visibility function between  $p$  and  $q$  at  $t$

$\rho : \mathcal{S} \rightarrow [0, 1]$  the diffuse reflectance at  $p$

$E : (\mathcal{S} \times \mathcal{T}) \rightarrow \mathbb{R}$  the self emitted radiosity at  $p$  at time  $t$

Using these definitions, for every  $X = (p, t) \in (\mathcal{S} \times \mathcal{T})$  the radiosity function satisfies the following equation:

$$B(X) = E(X) + \int_{(\mathcal{S} \times \mathcal{T})} B(Y) \mathcal{K}(X, Y) dY \quad (1)$$

where :

$\mathcal{K}$  is defined on  $(\mathcal{S} \times \mathcal{T})^2$  as

$$\mathcal{K}((p, t), (q, t')) = \rho(p) k(p, q, t) \delta(t, t') \quad (2)$$

$k$  is the function defined on  $(\mathcal{S} \times \mathcal{S} \times \mathcal{T})$  by

$$k(p, q, t) = \frac{\cos \theta(p, q, t) \cos \theta(q, p, t)}{\pi r(p, q, t)^2} v(p, q, t) \quad (3)$$

$\delta$  is the Dirac distribution equal to 0 when  $t \neq t'$

Note that, though equation 1 seems to describe inter-temporal light exchanges, such non physical transfers are avoided thanks to the Dirac function in equation 2. Note also that equation 1 is formally equivalent to the classical radiosity equation in the static case. Therefore, any algorithm capable of solving the latter can probably be extended in a straightforward manner to solve the former. In particular, we shall see that we can derive a finite element formulation similar to that of standard radiosity<sup>20, 17</sup>.

### 3.2. Discretisation

Equation 1 is a Fredholm equation of the second kind, and can be discretised by the Galerkin method. We want to compute an approximation  $\tilde{B}$  of  $B$  in a finite dimensional function space spanned by an orthogonal basis of functions  $(u_i)_{1 \leq i \leq N}$ . Therefore, we can express  $\tilde{B}$  as a linear combination of the  $u_i$ :

$$\tilde{B} = \sum_{j=1}^N B_j u_j$$

The Galerkin condition<sup>17</sup> defines the approximation  $\tilde{B}$  so that the residual function

$$r(X) = \tilde{B}(X) - E(X) - \int_{Y \in (\mathcal{S} \times \mathcal{T})} \tilde{B}(Y) \mathcal{K}(X, Y) dY$$

is orthogonal to all the  $u_i$ . In such a case, the coefficients  $B_j$  that define  $\tilde{B}$  are solutions of the following linear system:

$$(I - M)B = E \quad (4)$$

$I$  is the identity matrix, the vector  $E$  is defined by

$$\forall i \in [1, N] E_i = \frac{\langle E, u_i \rangle}{\|u_i\|^2}$$

and the matrix coefficients are defined by:

$$\forall (i, j) \in [1, N]^2 M_{i,j} = \frac{\langle \int_Y \mathcal{K}(\cdot, Y) u_j(Y) dY, u_i \rangle}{\|u_i\|^2} \quad (5)$$

The simplest possible choice of a function basis is piecewise constant functions. As discussed in section 2.2, this choice proves unsatisfying in certain cases, where it causes

noticeable temporal discontinuities in indirect lighting. As a consequence, we propose instead functions that are piecewise constant in space, and piecewise *polynomial* in time. To a given element  $k$  of our mesh, defined as the cross product of polygon  $P_k$  and time interval  $T_k$ , correspond  $L$  basis functions  $u_{Lk+i}(p, t)$  with  $0 \leq i < L$ , equal to 0 when  $(p, t)$  is outside  $(P_k \times T_k)$ , and to  $\Phi_k^i(t)$  otherwise. Since the  $u_i$  have to form an orthogonal basis, the  $\Phi_k^i$  are the restriction of the first  $L$  Legendre polynomials to the time interval  $T_k = [\alpha_k, \beta_k]$ , *i.e.*

$$\begin{aligned} \Phi_k^0(t) &= 1 \\ \Phi_k^1(t) &= \sqrt{3} \left( 2 \frac{t - \alpha_k}{\beta_k - \alpha_k} - 1 \right) \\ &\dots \end{aligned}$$

Therefore, the variations of radiosity of each element  $k$  in the mesh will be described by  $L$  unknown coefficients  $B_{Lk}, \dots, B_{Lk+L-1}$ . Furthermore, from equation 5 it can be derived that each pair  $(k, l)$  of elements in our mesh corresponds to a  $L \times L$  block  $\rho_k I_{k,l}$  in the matrix  $M$ , where  $I_{k,l}$  is the following *interaction matrix*:

$$I_{k,l} = \frac{1}{\|P_k\| (\beta_k - \alpha_k)} \int_{T_k \cap T_l} G_{k,l}(t) \int_{P_k} \int_{P_l} k(p, q, t) dq dp dt \quad (6)$$

and the matrix  $G_{k,l}$  is defined as:

$$G_{k,l}(t) = \begin{pmatrix} \Phi_k^0(t) \Phi_l^0(t) & \dots & \Phi_k^0(t) \Phi_l^{L-1}(t) \\ \vdots & \ddots & \vdots \\ \Phi_k^{L-1}(t) \Phi_l^0(t) & \dots & \Phi_k^{L-1}(t) \Phi_l^{L-1}(t) \end{pmatrix}$$

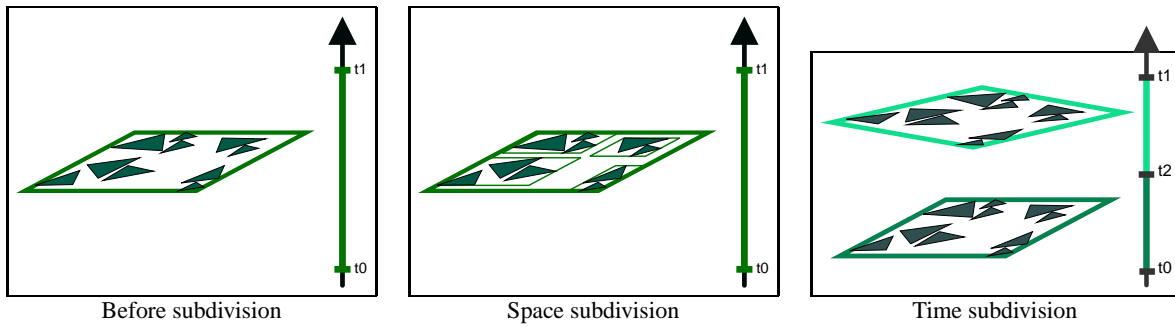
The interaction matrix extends the traditional notion of form factor used in the classical static radiosity algorithms.

### 3.3. Hierarchical Solution of the Discrete Equation

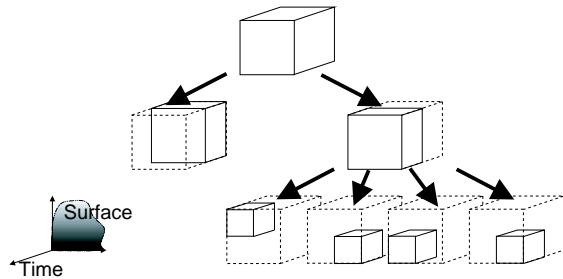
We are using piecewise polynomial functions to describe the variations of radiosity in time. Therefore the resulting algorithm is an extension of the Wavelet Radiosity algorithm<sup>21, 22</sup>, using Haar basis over the spatial dimension and Alpert's  $\mathcal{M}_L$  basis<sup>23</sup> over the time dimension.

Since our mesh elements are defined both by their geometry and their time interval, they can be subdivided either in space (partitioning their geometry, e.g. using a quadtree subdivision scheme) or in time (subdividing the time range and leaving the geometry unchanged). As illustrated by Figure 3, repeated applications of either subdivision scheme build a data structure that offers a multi-resolution representation of the radiosity function over space and time. As in the original Hierarchical Radiosity algorithm, links joining two hierarchical elements are used to specify at which level of precision the light exchanges should be actually computed.

Consequently, the Space-Time Hierarchical Radiosity algorithm, similarly to the original hierarchical algorithm of



**Figure 4:** The cluster in the left-hand image (represented in 2D for simplicity) can be subdivided either spatially (center), or temporally (on the right). Spatial subdivision builds one new hierarchical level of clusters around the surfaces. Temporal subdivision duplicates the surfaces and subdivides the corresponding time interval.



**Figure 3:** A simple example of space-time hierarchy: here the root hierarchical element (represented as a space  $\times$  time 3D volume) has been first subdivided in time. One of the resulting siblings have been subdivided in space.

Hanrahan *et al.*, is an iteration composed of the following three steps:

1. Recursive evaluation of the precision of all links to place them at the appropriate level in the hierarchy. The evaluation of the links is performed by a function named *refinement oracle*. As a side-effect, this recursive process adaptively builds the hierarchical mesh, refining the original mesh elements where more precision is needed. Our oracle is discussed in section 3.4.
2. Gathering the light through the links in the hierarchy. Light exchanges computations are detailed in section 3.5.
3. Ensuring the coherence of radiositivities between all hierarchical levels. This bidirectional traversal of the hierarchy is referred to as *Push-Pull* and is discussed in section 3.6.

To fully benefit from the strength of the hierarchical formulation, it is necessary to extend the hierarchy of elements above the initial surfaces level, by hierarchically grouping together surfaces, and eventually groups of surfaces (called *clusters*). At the top of our hierarchy will be one *root cluster* which will represent all surfaces, during the whole anima-

tion. The starting point of the algorithm will be the *root link* joining this cluster to itself, thereby representing all possible interactions between all surfaces in the scene, during the whole animation<sup>16,17</sup>.

Hierarchical Radiosity algorithms with clustering for static scenes perform the construction of the cluster hierarchy as a pre-processing step. Such an approach cannot be used in our case, since the resulting spatial hierarchy would pre-exist the recursive link refinement procedure, preventing any temporal refinement until we reach the surfaces level. We propose instead a new approach which we name *lazy clustering*. At the beginning of our algorithm, we only build the root cluster, plus one cluster for each different rigid motion in the animation. The rest of the cluster hierarchy is built as a by-product of the link refinement procedure. Therefore, as for surfaces elements in the space-time mesh, we can split clusters that have not been previously refined either in time or in space (see Figure 4):

- Time-refinement of a cluster can be performed by creating two clusters as children of the original one, each one defined over one half of the original time interval. Each surface inside the original cluster must be duplicated and one copy is assigned to each of the two children clusters.
- Space-refinement of a cluster is the act of grouping together some of the surfaces contained in the said cluster, forming new children clusters (which may be space or time split later on during the refinement process), possibly leaving some surfaces as direct children. This can be achieved in applying only one step of any classical top-down recursive clustering method (without the recursion). We chose to adapt Christensen's clustering method<sup>24</sup> which is straightforward to implement and produces a rather well-formed spatial hierarchy<sup>25</sup>.

### 3.4. Space-Time Refinement Oracle

The refinement oracle is the function in charge of evaluating the precision of a given link and decide if it is placed at the

appropriate level in the hierarchy. In the case of the Space-Time Hierarchical Radiosity algorithm, this function has two goals:

- to decide whether a given link is a precise enough representation of the corresponding light exchange (like in classical Hierarchical Radiosity).
- if it is not precise enough, to determine whether it should be split in space or time.

A simple oracle, based on a comparison of the estimated time-variance and space-variance of the irradiance gathered across the link, always produces the same mesh, regardless of the function basis used. This is obviously not satisfying since piecewise polynomial functions should offer a better approximation of the radiosity than piecewise constant functions at the same subdivision depth.

We propose to extend for Space-Time Radiosity an oracle designed for Wavelet Radiosity in the static case<sup>18, 19, 26</sup>, based on estimates of the error on the propagated energy rather than on estimates of the variation of this energy.

We use a grid of control points located on the receiving element, and a set of control points in time. On these control points, at the control times, we estimate the radiosity value using two methods:

1. by multiplying the emitters radiosity vector by the interaction matrix corresponding to the link, and then interpolating the radiosity values at the control times,
2. by direct integration of the radiosity on the emitter, using a quadrature.

The difference between these two values is an indication of the error made when evaluating the interaction at this point for this level of precision. The norm of these differences is used as the error on the current interaction. Refinement will occur if this norm is above the refinement threshold set by the user.

The control points and times must be carefully chosen so that they provide meaningful information. They must be different from the quadrature points and times used for the form factor computations. The number of control points and times must be higher for large receivers so that we do not miss important features. Also, placing control times at the beginning and at the end of the time interval greatly enhances temporal continuity.

Once we have made the decision to refine an interaction, we must choose between refinement in space or in time. We compute two variance estimates for the set of estimated error values on our grid of control points and times:

- An average *spatial variance*: for each fixed control time we compute the variance of error values at each control points, and then take the temporal average.
- An average *temporal variance*: for each fixed control point we compute the variance of error values at each control times, and then take the spatial average.

We refine the interaction in time if the average temporal variance is above the average spatial variance, and in space otherwise.

### 3.5. Light Exchanges Computations

Computing the light exchanged between two linked surfaces is straightforward. The product of the link's interaction matrix by the radiosity of the emitter is added to the radiosity of the receiver. The interaction matrix has generally been computed previously during the refinement procedure using simple gaussian quadratures.

However, interactions involving one or more clusters require a special approach, based on the one described by Sillion<sup>27</sup>. Roughly, anisotropic emission from a cluster is approximated by going down to the surfaces level to estimate the directional radiant intensity exiting the cluster (*Delayed Pull*), and the irradiance gathered by a cluster from a given hierarchical element is distributed to all the surfaces inside the cluster immediately at gathering time according to their orientation (*Immediate Push*). The specificities of the Space-Time Hierarchical Radiosity method come from the fact that the position, orientation and radiosity of the objects can change with time.

#### 3.5.1. Emission from a Cluster: Delayed Pull

In the classical hierarchical radiosity algorithm, the computation of the light emitted from an object involves the computation of the form factor between the sender  $l$  and the receiver  $k$ . It is very difficult to define what the form factor should be if the sender is an anisotropic cluster. Therefore we directly compute the irradiance emitted by the cluster to the receiver, by summing the contributions of the  $N$  surfaces contained in the cluster  $l$ . At a given time  $t$ , point  $p$  receives from the  $N$  elements  $i$  in  $l$  the total irradiance:

$$I_{received}(p, t) = \sum_{i=1}^N \int_{Q_i} B_i(q, t) g(p, q, t) v(p, q, t) dq$$

where the geometric configuration function is defined by:

$$g(x, y, t) = \frac{\mathcal{R}(t) \cos \theta'}{\pi r^2}$$

and the  $\mathcal{R}$  function is the receiver factor defined in<sup>27</sup> as  $\cos \theta$  if the receiver is a surface and 1 if the receiver is a cluster (the surfaces orientation in the receiving cluster will be taken into account by the Immediate Push mechanism described in section 3.5.2).

We approximate the received irradiance by projecting it on our function basis: The resulting approximation is a linear combination of our  $L$  basis functions:

$$\tilde{I} = \sum_{j=0}^{L-1} \lambda_j u_{Lk+j}$$

Since the  $u_i$  are orthogonal we have:

$$\begin{aligned} \lambda_j &= \frac{1}{\|u_{Lk+j}\|^2} \langle I_{received}, u_{Lk+j} \rangle \\ &= \frac{\sum_i \left[ \int_{T_k \cap T_i} \left( \int_{P_k} \int_{Q_i} g(p, q, t) v(p, q, t) dq dp \right) B_i(t) \Phi_k^j(t) dt \right]}{A_k(\beta_k - \alpha_k)} \end{aligned}$$

Computing the above integral is costly as it involves a number of visibility estimations proportional to the number of surfaces in the cluster. Therefore we approximate it by factoring out the visibility, and average it over the sending cluster:

$$\lambda_j = \frac{\sum_i \left[ \int_{T_k \cap T_i} \left( \int_{P_k} \int_{Q_i} g(p, q, t) dq dp \right) B_i(t) \Phi_k^j(t) \tilde{V}(t) dt \right]}{A_k(\beta_k - \alpha_k)}$$

where

$$\tilde{V}(t) = \frac{1}{A_k A_l} \int_{P_k} \int_{Q_l} v(x, y, t) dx dy$$

We compute the  $\alpha_j$  and  $\tilde{V}(t)$  using a Gaussian quadrature. Since the cost of evaluating the approximate visibility must not depend on the number  $N$  of surfaces inside the cluster  $l$  we place the quadrature points independently of the surfaces positions inside the cluster's bounding box.

### 3.5.2. Reception Inside a Cluster: Immediate Push

The reception inside a cluster obeys the *immediate push* principle: the irradiance received at the cluster level is immediately dispatched to all surfaces inside the cluster, where it is multiplied by the cosine of the angle between the normal of the surface and the direction of the incoming radiance. The origin of the incoming radiance is assumed to be the center of the emitter, whether a cluster or a surface.

Since both the cluster and the sender may be moving, the receiver factor is time-dependent. We need to project it on our wavelet basis. Let us assume a cluster  $k$  has received an irradiance  $I_{received}$ . This irradiance is distributed to each surface  $i$  in the cluster  $k$  according to its orientation:

$$I_i = I_{received}(t) \cos \theta_i(t)$$

$I_i$  is then reprojected on the wavelet basis for the time interval  $T_i$  over which the hierarchical element  $i$  is defined. The resulting approximate irradiance is then :

$$\tilde{I}_i = \sum_{j=0}^{L-1} \gamma_j \Phi_i^j$$

and the  $\gamma$  coefficients are:

$$\begin{aligned} \gamma_j &= \frac{1}{\|\Phi_i^j\|^2} \langle I_i | \Phi_i^j \rangle \\ &= \frac{1}{\beta_i - \alpha_i} \int_{T_i \cap T_i} I_{received}(t) \cos \theta_i(t) \Phi_i^j(t) dt \end{aligned}$$

These integrals are once again approximated using a Gaussian quadrature.

Our method contains two successive approximations: we have separately computed the irradiance received at the cluster level, which was time-dependent, projected it onto the function basis, then dispatched it to the surfaces, taking into account the surface movement, and re-projected it on the function basis for the receiving surface. This double approximation is consistent with the clustering approach. If the refinement oracle decides that we can compute an interaction at the cluster level, then this approximation should be sufficient. Spending more computation time to find a better approximation would impair the hierarchical nature of the algorithm and would reduce its performance.

### 3.6. Push-Pull Traversal

After the irradiances have been gathered across all links in the scene, a traversal of the complete hierarchy is necessary to maintain coherence between the different hierarchical levels. First, irradiance contributions computed at various level of the hierarchy have to be pushed down to the lowest level of the structure and summed along the way. There the radiosities of each leaf are computed, and these radiosities are then progressively pulled up the hierarchy and averaged to compute the correct radiosity representation corresponding to each hierarchical level.

In the case of Wavelet Radiosity<sup>21</sup>, this process is slightly more complicated than it is for static Hierarchical Radiosity since we need to define how to combine the coefficients describing the radiosity variations, to convert them from one hierarchical level to the other. Remember from section 3.2 that our multi-resolution basis functions are cross products of the scale functions of the Haar basis over space, and scale functions of the  $\mathcal{M}_L$  basis over time. Since we use a very simple midpoint subdivision scheme when subdividing elements in time, the coefficients that have to be pushed down or pulled up during this traversal can be computed using simple linear transformations, which are independent of the element or the hierarchical level. Both linear transforms are referred to as the *two scale relationship*<sup>22</sup>, and are determined by two  $L \times L$  matrices  $P$  and  $Q$ . When  $L = 2$  (linear wavelets), those matrices are:

$$P = \begin{pmatrix} 1 & 0 \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \quad Q = \begin{pmatrix} 1 & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix}$$

When pushing down the total irradiance  $I$  (remember that this is a  $L$ -dimensional vector) from a given element split in time to each of its two children, the corresponding irradiances  $I'$  and  $I''$  to be transmitted to its first and second children are given by the following linear transform:

$$I' = {}^t P I \quad \text{and} \quad I'' = {}^t Q I$$



Respectively when pulling up, the average radiosity  $B$  of an element can be computed from the radiosities of its two children  $B'$  and  $B''$ :

$$B = \frac{1}{2}(PB' + QB'')$$

### 3.7. Practical Issues and Choices

#### 3.7.1. Choice of a Space-Time Function Basis

Higher-order wavelets have been previously used as function basis for the representation of radiosity<sup>21, 22</sup>. However, the algorithms resulting from their straightforward use within the classical Hierarchical Radiosity framework were proved impractical, slower than when using classical piecewise constant function basis and giving poorer results<sup>28</sup>. Further research<sup>26</sup> has later shown that by making use of several recent advances in the field<sup>18, 19, 26, 29</sup> higher order wavelets were providing a better approximation of the illumination function, requiring less memory and computation time. In particular, the radiosity function produced looks continuous without post-processing thanks to an adequate refinement oracle.

Unfortunately, to this date, higher order wavelets in the *spatial* dimension cannot be easily applied to cluster objects. This is due to the fact that it is difficult to provide a function mapping the surfaces contained in a cluster onto the square domain where the wavelet basis is defined. Therefore, in order to be able to use higher order wavelets for surfaces and clusters for inexpensive approximations, a mechanism to use different approximation order for different hierarchical elements should be defined. Our refinement oracle would automatically adapt to the new function basis. The only point needing change would be the Push-Pull matrices we gave in section 3.6.

However, this problem does not arise in the *temporal* dimension. Moreover, the lower dimensionality makes the added cost of the use of wavelets lower in the temporal dimension than it is in the spatial dimension. Therefore, we decided to limit our use of wavelets to the description of the temporal variations of radiosity. In our implementation, our function basis was composed of linearly varying functions (the  $\mathcal{M}_2$  basis). This choice proved sufficient in practice to significantly reduce the temporal discontinuities (see Section 4).

In order to provide a smooth appearance for patches in our example animations, we applied a simple linear interpolation over the polygons as a post process, when traversing the space-time mesh to generate the images. Though it noticeably increases the visual appeal of the results, this post process doesn't improve the precision of the solution. Much better reconstruction methods have been proposed for static scenes, such as *final gathering*<sup>30, 31</sup>, and can be applied here on an image per image basis. Moreover, Martin *et al.* have proposed recently a final gathering acceleration method for

animated scenes<sup>15</sup>, whose coupling with our approach seems promising.

#### 3.7.2. Memory Management Issues and Refinement Ordering

As our experiments will show in section 4, the Space-Time Hierarchical Radiosity algorithm is quite memory intensive. This is due to the fact that we keep in memory a complete view independent solution describing the variations of radiosity for all surfaces during the whole animation time interval. Though the amount of memory available on graphic workstations is rapidly increasing, it may prove necessary to reduce our algorithm requirements when running on less powerful machines or when computing long animation sequences.

One way of reducing the memory cost is to use hard-disk space to cache parts of the hierarchy that are temporarily not required for computations. For the algorithm to remain efficient when such a caching scheme is used, accesses to the disk should be limited to a small number of large files. The order according to which we traverse the space-time hierarchy during the refinement should be chosen accordingly.

Such a desirable traversal order can be derived if we recall from equation 6 that elements whose time interval do not overlap cannot exchange energy. Since we always subdivide time intervals in two parts of equal length, for every element  $P$  that can interact with a given element  $Q$ , we know that either  $T_P \subset T_Q$  or  $T_Q \subset T_P$ . Therefore, we can easily determine which element may be needed to compute or refine a certain interaction, by bucketing hierarchical elements and links according to their time interval.

The refinement can be performed as a traversal of the hierarchy that would correspond to a depth-first order traversal of the time intervals binary tree. Only the elements whose time interval contain the one currently visited should be kept in memory. Disk access would only take place when moving from one time interval to the other.

We ran an experiment to estimate the corresponding gain in memory that could be expected from such a traversal. For the SPOT scene (see section 2.2 and 4.1), we used  $2^5 - 1 = 31$  time interval buckets to sort our elements (one for each time interval corresponding to the first 5 subdivision level). The maximum total cost of the portion of the hierarchy that needs to be kept in memory is 40 MB whereas more than 450 MB are required when we are keeping everything in RAM (*cf.* table 1). In such a case, file accesses shouldn't reduce excessively the performances of our algorithm: the added cost of reading and writing 31 files each about 15MB big should be reasonable since an iteration on this scene already requires 10 – 20 minutes.

## 4. Experimental Results

In this section we discuss the performance of our algorithm on several test scenes. In section 4.1, we demonstrate the improvement on temporal continuity that our use of piecewise linearly varying functions offers when compared to simple "box" functions. In section 4.2 we demonstrate our use of clustering and we offer comparisons with frame by frame Hierarchical Radiosity calculations to show that we obtain good acceleration factors.

### 4.1. Improvement of Temporal Continuity

To illustrate the improvement in temporal continuity obtained using the  $\mathcal{M}_2$  function basis in the time dimension, we use the test scene from section 2.2. In this scene, the sweeping movement of spotlights over walls painted in different colors cause important changes in the indirect illumination of the scene. In particular, strong color bleeding effects can be observed moving on the ceiling and the floor of the scene.

As explained in section 2.2, this scene was purposely designed as a "worst case scenario" for the Space-Time Hierarchical Radiosity algorithm in order to exhibit strong temporal discontinuities. When using a piecewise constant function basis to describe the variation of radiosity in time, the indirect lighting effects in this scene are extremely discontinuous. For example, the color bleeding patches seem to be updated only every second or so. The amplitude of these discontinuities is shown in the radiosity variation plot of Figure 5. It can be clearly seen that the greatest discontinuity is located at the middle of the animation, then at the first and third quarter. The magnitude of the largest discontinuity is about 40% of the time-average radiosity of this patch, which makes it quite noticeable. Smaller discontinuities can be observed at other even subdivisions of the time interval.

However, the same animation, computed using hierarchical elements linearly varying in time, exhibits a much more coherent indirect illumination. Without paying careful attention, it is difficult to perceive any discontinuity. We can see on Figure 5 that the indirect lighting is obviously more "continuous" than when using a piecewise constant function basis. The largest discontinuity (at  $t = \frac{1}{4}$ ) has a magnitude of about 7% of the average radiosity of the patch. Figure 6 shows that this strong reduction of discontinuities can be observed on all surfaces of the scene. Table 1 allows the comparison of computation time and memory cost when computing this animation with a frame by frame Hierarchical Radiosity algorithm, with our algorithm using the Haar basis, and with our algorithm using the  $\mathcal{M}_2$  basis respectively. All timings have been measured on a single 300Mhz MIPS R12000 processor of an SGI Onyx2 computer.

The following comments can be made about these results:

- Though this scene is geometrically quite simple, the

speedup factor obtained, when compared to a frame by frame computation, is about 6. (Note that this acceleration factor is only about 2 if we only take into account the time needed to compute the direct illumination). Our algorithm performance on such scenes where the indirect lighting is dominant and dramatically changing over time is therefore satisfying.

- The memory consumption when using the  $\mathcal{M}_2$  basis is 15% lower than when using the Haar basis, in spite of the added storage cost of the second radiosity coefficient and the interaction matrices. The animation has also been computed slightly faster. This is due to the fact that fewer subdivisions in time are needed to obtain a precise enough representation of the variations of radiosity in time, resulting in a faster refinement and a lighter mesh.

### 4.2. Validation of the Clustering Approach

We have tested our algorithm on scenes composed of several thousands of input polygons (see Figure 7). For such scenes, Hierarchical Radiosity computations without the use of clustering would have been extremely long because of the quadratic cost of the initial linking stage.

The first of our three test animations takes place in a small room with some furniture (a couple desks, chairs, pens, *etc.*). It is lit by four area light sources. The bookshelf, against the wall, falls to the floor. The animation is 4 seconds long, and is composed of 100 frames. The input geometry is composed of 7,200 polygons. The second animation takes place in a large library hall with several desks separated by rows of bookshelves. This scene is lit by numerous area light sources. A character is moving through the hall. The animation is 20 seconds long and is composed of 500 frames. There are about 35,000 input surfaces. The third animation is somehow similar to the test scene we use in section 4.1. We replaced the boxes by more complex objects. The resulting scene is composed of approximately 30,000 polygons and is 24 seconds long.

Table 2 summarizes our experimental results for our three test scenes. In this table, we compare the resources necessary to compute the animations when using the Space-Time Hierarchical Radiosity algorithm and when performing a Hierarchical Radiosity with Clustering frame by frame providing the same image quality. All timings have been observed on a 300 Mhz MIPS R12000.

The more elements the mesh is composed of, the more the hierarchical approach is advantageous. Since it makes it possible to compute more complicated animations, clustering really allows us to benefit fully from the hierarchical nature of our algorithm. The typical speedup is ranging from 6 to 18.

The memory consumption of our algorithm is quite high, since we keep in memory at the same time a complete view independent global illumination solution for all frames of

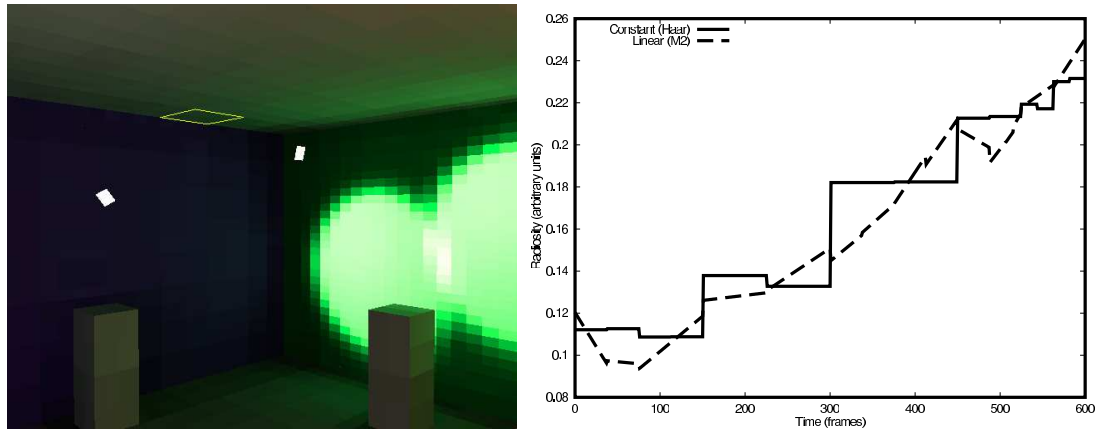


Figure 5: Variation of the radiosity function at the center of the highlighted element during the animation

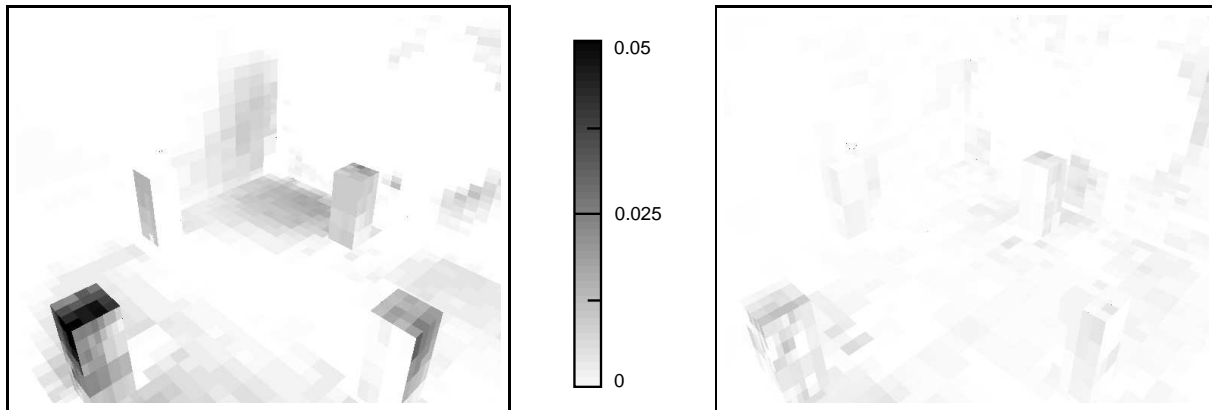


Figure 6: Comparison of temporal discontinuities at  $t = \frac{1}{2}$ . Darker colors indicate a higher discontinuity (arbitrary units). Left: hierarchical radiosity, right:  $M_2$  wavelets.

the animation (we have discussed a possible way to avoid this in section 3.7.2). However, we can note that the memory cost of our algorithm depends more on the complexity of the illumination than on the number of input polygons. The more complex the input mesh is, the smaller the polygons are on average. Therefore, they are less likely to be subdivided later, and the resulting hierarchy will not be much bigger than if it consisted initially of large unsubdivided surfaces.

## 5. Conclusion

In this paper, we proposed a new algorithm to compute global illumination in diffuse animated environment. This algorithm is based on the adaptive refinement of a hierarchical mesh defined both over time and space. It can therefore benefit from the *a priori* knowledge of objects movements to factor out a large part of redundant computations.

This technique allows computation of animations with a quality similar to frame by frame computation, in a shorter

time. Geometrically complex scenes can be dealt with thanks to the definition of a clustering approach extending the space-time mesh. The continuity of indirect lighting is improved by the simultaneous use of a piecewise-linear wavelet basis in the time dimension and of an adequate space-time refinement oracle.

Promising directions for future research include:

- the derivation of a space-time final gathering approach, adapting the one proposed by Martin *et al.*<sup>15</sup>.
- the implementation and extensive testing of disk caching schemes such as the one suggested in section 3.7.2.
- the parallelization of this algorithm. This should be straightforward on a shared memory architecture<sup>32</sup> but will certainly prove more difficult on a cluster of PC.
- experiments with alternative wavelet basis in the time dimension, for example using higher order polynomials.
- the extension of our algorithm to non-diffuse scenes, using a unified mesh based particle shooting approach<sup>9</sup>.

	Computation Time			Memory used
	Direct Lighting	Indirect Lighting	Total per image	
Frame by Frame HR	$1 \text{ s} \times 600 = 600 \text{ s}$	$15 \text{ s} \times 600 = 9,000 \text{ s}$	16.0 s	5 MB
Haar	254 s	1,492 s	2.9 s	587 MB
$\mathcal{M}_2$	271 s	1,172 s	2.4 s	464 MB

**Table 1:** Performance comparison on the SPOTS scene, between frame by frame Hierarchical Radiosity, our algorithm using the Haar basis, and our algorithm using the  $\mathcal{M}_2$  basis



**Figure 7:** Sample frames from our test animations

- the construction of a refinement criteria using human perception-based animation quality metrics<sup>13</sup>.

## References

1. P. Hanrahan, D. Salzman, and L. Aupperle, “A Rapid Hierarchical Radiosity Algorithm”, in *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, vol. 25, pp. 197–206, (1991).
2. C. Damez and F. Sillion, “Space-time hierarchical radiosity”, in *Proceedings of the 10th Eurographics Workshop on Rendering*, pp. 235–246, (1999).
3. C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, “Modelling the Interaction of Light Between Diffuse Surfaces”, in *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, vol. 18, pp. 212–222, (1984).
4. E. Shaw, “Hierarchical radiosity for dynamic environments”, *Computer Graphics Forum*, **16**(2), pp. 107–118 (1997).

	Computation time		Memory used	
	STHR	Static HRC	STHR	Static HRC
SHELF	3,335 s	184 s $\times$ 100 = 18,400 s	100 MB	16 MB
HALL	33,333 s	1.185 s $\times$ 500 = 592,500 s	842 MB	120 MB
ROBOTS	4,591 s	109 s $\times$ 600 = 65,400 s	475 MB	17 MB

**Table 2:** Comparative results for the use of clustering: we compare computation time and memory use of our algorithm to the time and memory needed to compute the same animation frame by frame with classical Hierarchical Radiosity with Clustering

5. G. Drettakis and F. Sillion, "Interactive update of global illumination using a line-space hierarchy", in *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, pp. 57–64, (1997).
6. B. Walter, G. Drettakis, and S. Parker, "Interactive rendering using the render cache", in *Proceedings of the 10th Eurographics Workshop on Rendering*, pp. 235–246, (1999).
7. P. Tole, F. Pellacini, B. Walter, and D. P. Greenberg, "Interactive global illumination in dynamic scenes", *ACM Transactions on Graphics (SIGGRAPH '02 Proceedings)*, **21**(3), pp. 537–546 (2002).
8. A. Keller, "Instant radiosity", in *Computer Graphics (ACM SIGGRAPH '97 Proceedings)*, pp. 49–56, (1997).
9. X. Granier and G. Drettakis, "Incremental updates for rapid glossy global illumination", in *Computer Graphics Forum (Proceedings of Eurographics 2001)*, vol. 20, pp. 268–277, (2001).
10. I. Wald, T. Kollig, C. Benthin, A. Keller, and P. Slusallek, "Interactive global illumination", in *Proceedings of the 13th Eurographics Workshop on Rendering*, (2002).
11. K. Dmitriev, S. Brabec, K. Myszkowski, and H.-P. Seidel, "Interactive global illumination using selective photon tracing", in *Proceedings of the 13th Eurographics Workshop on Rendering*, (2002).
12. G. Besuievsky and X. Pueyo, "Animating radiosity environments through the multi-frame lighting method", *Journal of Visualization and Computer Animation*, **12**, pp. 93–106 (2001).
13. K. Myszkowski, T. Tawara, H. Akamine, and H.-P. Seidel, "Perception-guided global illumination solution for animation rendering", in *Computer Graphics Proceedings (ACM SIGGRAPH '01 Proceedings)*, pp. 221–230, (2001).
14. C. Domez, K. Dmitriev, and K. Myszkowski, "State of the art in global illumination for interactive applications and high-quality animations", *Computer Graphics Forum*, **22**(1), pp. 55–77 (2003).
15. I. Martín, X. Pueyo, and D. Tost, "Frame-to-frame coherent animation with two-pass radiosity", *IEEE Transactions on Visualization and Computer Graphics*, **9**(1), pp. 70–84 (2003).
16. B. Smits, J. Arvo, and D. Greenberg, "A Clustering Algorithm for Radiosity in Complex Environments", in *Computer Graphics (ACM SIGGRAPH '94 Proceedings)*, pp. 435–442, (1994).
17. F. Sillion, "Clustering and Volume Scattering for Hierarchical Radiosity Calculations", in *Proceedings of the 5th Eurographics Workshop on Rendering*, pp. 105–117, (1994).
18. P. Bekaert and Y. D. Willems, "Error Control for Radiosity", in *Proceedings of the 7th Eurographics Workshop on Rendering*, pp. 153–164, (1996).
19. P. Bekaert and Y. D. Willems, "Hirad: A Hierarchical Higher Order Radiosity Implementation", in *Proceedings of the Twelfth Spring Conference on Computer Graphics (SCCG '96)*, (Bratislava, Slovakia), Comenius University Press, (June 1996).
20. M. F. Cohen and J. R. Wallace, *Radiosity and Realistic Image Synthesis*. Boston, MA: Academic Press Professional, (1993).
21. S. J. Gortler, P. Schroder, M. F. Cohen, and P. Hanrahan, "Wavelet Radiosity", in *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, pp. 221–230, (1993).
22. P. Schroder, S. J. Gortler, M. F. Cohen, and P. Hanrahan, "Wavelet Projections for Radiosity", in *Fourth Eurographics Workshop on Rendering*, pp. 105–114, (1993).
23. B. K. Alpert, "A class of bases in  $L^2$  for the sparse representation of integral operators", *SIAM Journal on Mathematical Analysis*, **24**(1), pp. 246–262 (1993).
24. P. H. Christensen, D. Lischinski, E. J. Stollnitz, and D. H. Salesin, "Clustering for Glossy Global Illumination", *ACM Transactions on Graphics*, **16**(1), pp. 3–33 (1997).

25. J.-M. Hasenfratz, C. Domez, F. Sillion, and G. Dretakis, "A practical analysis of clustering strategies for hierarchical radiosity", in *Computer Graphics Forum (Proc. Eurographics '99)*, vol. 18, pp. C-221-C-232, (Sept. 1999).
26. F. Cuny, L. Alonso, and N. Holzschuch, "A novel approach makes higher order wavelets really efficient for radiosity", in *Computer Graphics Forum (Proc. Eurographics 2000)*, vol. 19, pp. C-99-C-108, (2000).
27. F. Sillion, "A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters", *IEEE Transactions on Visualization and Computer Graphics*, 1(3), (1995).
28. A. Willmott and P. Heckbert, "An empirical comparison of progressive and wavelet radiosity", in *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)* (J. Dorsey and P. Slusallek, eds.), (New York, NY), pp. 175-186, Springer Wien, (1997). ISBN 3-211-83001-4.
29. M. Stamminger, H. Schirmacher, P. Slusallek, and H.-P. Seidel, "Getting rid of links in hierarchical radiosity", *Computer Graphics Journal (Proc. Eurographics '98)*, 17(3), pp. C165-C174 (1998).
30. M. Reichert, *A Two-Pass Radiosity Method to Transmitting and Specularly Reflecting Surfaces*. M.Sc. thesis, Cornell University, (1992).
31. A. Scheel, M. Stamminger, and H.-P. Seidel, "Grid Based Final Gather for Radiosity on Complex Clustered Scenes", *Computer Graphics Forum*, 21(3), (2002).
32. F. Sillion and J.-M. Hasenfratz, "Efficient parallel refinement for hierarchical radiosity on a DSM computer", in *Proceedings of the Third Eurographics Workshop on Parallel Graphics and Visualisation*, (2000). <http://www-imagis.imag.fr/Membres/Jean-M..Hasenfratz/PUBLI/EGWPGV00.html>.