

Space-Time Hierarchical Radiosity

Cyrille Damez and François Sillion
iMAGIS – GRAVIR/IMAG INRIA

Abstract. This paper presents a new hierarchical simulation algorithm allowing the calculation of radiosity solutions for time-dependent scenes where all motion is known *a priori*. Such solutions could, for instance, be computed to simulate subtle lighting effects (indirect lighting) in animation systems, or to obtain high-quality synthetic image sequences to blend with live action video and film. We base our approach on a Space-Time hierarchy, adding a life span to hierarchical surface elements, and present an integrated formulation of Hierarchical Radiosity with this extended hierarchy. We discuss the expected benefits of the technique, review the challenges posed by the approach, and propose first solutions for these issues, most notably for the space-time refinement strategy. We show that a short animation sequence can be computed rapidly at the price of a sizeable memory cost. These results confirm the potential of the approach while helping to identify areas of promising future work.

1 Introduction

Producing animated sequences has been a natural application of image synthesis since its inception, but remains an expensive endeavor, despite the impressive progress of computer performance over the years. A lot of effort has been devoted to the reduction of resource consumption for these calculations: Considering the large number of frames necessary even for a short movie sequence, shortening the production time carries an important economic value.

Interestingly, almost all animation systems to date still operate on a frame-by-frame basis, for understandable robustness and simplicity issues. Admittedly, and as we shall see below, more elaborate techniques trying to capitalize on temporal coherence are often more expensive in terms of memory usage. However even if this memory cost has appeared too expensive in the past, it is important to revisit such technological choices periodically in a period of fast change. Indeed, as computing power becomes cheaper by the day, memory prices are also decreasing in due proportion. Furthermore, while each and every frame of an animation must naturally be computed independently to reflect the appropriate position and parameters of the view at each time step, it seems promising to try and capitalize on the seemingly large temporal coherence in *illumination*, by computing lighting effects over a finite time range in a single calculation.

This paper therefore investigates the potential benefits of computing a radiosity solution over a time interval, taking into account all object movements and changes in the scene, and refining the calculation to precisely reflect the resulting illumination changes. We provide a formal framework for the derivation of space-time radiosity algorithms, and describe a hierarchical solution algorithm. The expected benefit of such a system is the ability to compute high-quality time-dependent radiosity solutions, in cases where all animation parameters are known *a priori*. In this case, and in contrast to dynamic update solutions for interactive uses, it will be possible to use the knowledge of the trajectories and changes to ensure the absence of visually distracting artifacts.

2 Previous work

Early attempts had been made to use space-time methods for ray tracing [9], by taking into account object motion when computing ray intersections. The gain then comes from the avoidance of multiple intersection tests, one for each frame. However the illumination is still computed frame-by-frame using a simple, local illumination model.

Similarly, limiting the amount of recomputation needed for computing radiosity solutions in dynamic scenes has been a subject of active research since the introduction of radiosity. Besuievsky *et al.* [2] performed Monte Carlo radiosity calculations in dynamic scenes by computing intersections with all frame-accurate positions of dynamic objects. Orti *et al.* [14]. predicted changes in visibility in a 2D world, but computed illumination at each frame.

For interactive applications, the main avenue has been the fast calculation of *updates* to an existing radiosity solution, in response to a change in the scene. Chen [3] and George *et al.* [8] performed incremental radiosity calculation by propagating energy corrections in the framework of progressive radiosity. Forsyth *et al.*[7], Shaw [15] and Drettakis *et al.* [5] attacked the issue of geometry modification, by reducing the set of links that require updating, in a hierarchical radiosity setting.

For non-interactive applications, where the movement of all objects is known in advance, Baum *et al.* [1] computed geometric shafts around the moving objects, and computed separately static and dynamic form factors. The latter were recomputed for every frame based on the actual position of the moving objects.

Incorporating time in the hierarchical radiosity framework will let us benefit from the separation between static and dynamic interactions, while maintaining the advantages of a hierarchical representation: namely, refinement will take into account the relative movement and speed of the objects, instead of re-evaluating the energy transfer at each and every frame. Furthermore, working on finite time intervals will let us apply global knowledge of motions and illumination changes to guide refinement.

3 Motivation for space-time calculations

In this section we review the potential benefits of a space-time hierarchical radiosity calculation. We start by recognizing the need for global illumination calculations for animated sequences: global illumination effects (indirect lighting) are key components in defining the “atmosphere” of a scene. Furthermore, realistic lighting is sometimes indispensable, when synthetic imagery must be combined with live action film: for instance a radiosity solution can be used to decide how to re-light actors, or as a background, influenced by the actors. This is particularly interesting for movie applications or in virtual sets.

3.1 A simple experiment demonstrating illumination coherence

We chose two test scenes with different lighting characteristics to investigate the amount of temporal coherence in the global illumination solution. We did not consider time varying light source properties (which can be treated by other means [8]), but focused on changing the geometry by moving objects.

The ACTOR scene (see Figure 1, left) shows a room with two objects moving. The size and movement of these objects were chosen to represent possible motion of human actors in a room, anticipating on the possible use of our technique for movie lighting or synthetic set applications. The LIGHT (see Figure 1, right) scene shows a similar room,

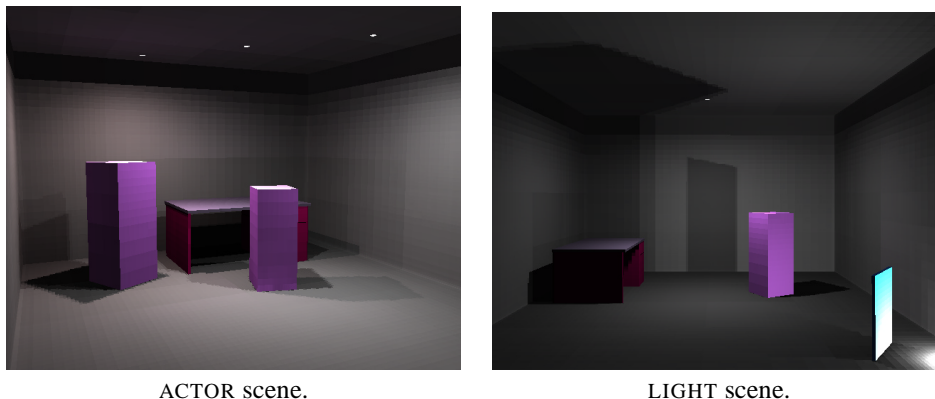


Fig. 1. Two test scenes chosen to illustrate situations that could be encountered in “real” applications. LIGHT has more obvious and large-scale illumination variations (both in space and time) due to the moving object near the right-hand light source.

but in which lighting conditions change dramatically in the entire scene over the course of the shot, because of a moving object in front of a light source. Both animations last 3 seconds or 75 frames.

In order to estimate the amount of CPU-time spent in redundant computations when computing a complete animated sequence frame by frame, we computed an independent radiosity solution for each frame, recording a complete view of the link hierarchy.

We then compared the link hierarchies and recorded the duration of validity of each link, by determining over how many frames it remained identical (we consider two links to be identical if they join the exact same elements, and they carry the same amount of energy within a threshold).

This allows us to compute, for any given duration, the amount of energy that was exchanged across links that were valid for that duration. The following table gives the proportion of energy (both for direct and indirect lighting) that was found to be constant during the *entire length* of the animation (*i.e.* traveling through links that were shared by every frame). The detail of the average amount of energy that remains valid seen as

Table 1. Proportion of energy constant during the entire animation

	Direct light	Indirect light
ACTOR	69.1%	49.5%
LIGHT	63.9%	42.7%

a function of the number of frame during which it is constant is given in Figure 2.

We can see that a large portion of light radiated in the scene needs only be computed once and for all. This is mostly due to links going from the light sources to the unoccluded parts of walls and static objects (the desk) for direct light, and between unoccluded part of the walls for indirect light. The fact that a significant number of links can be considered valid for more than one frame but not for the whole animation (especially for indirect lighting), makes a multi-resolution approach such as hierarchical radiosity particularly appealing. Note that in the LIGHT scene, significant energy is

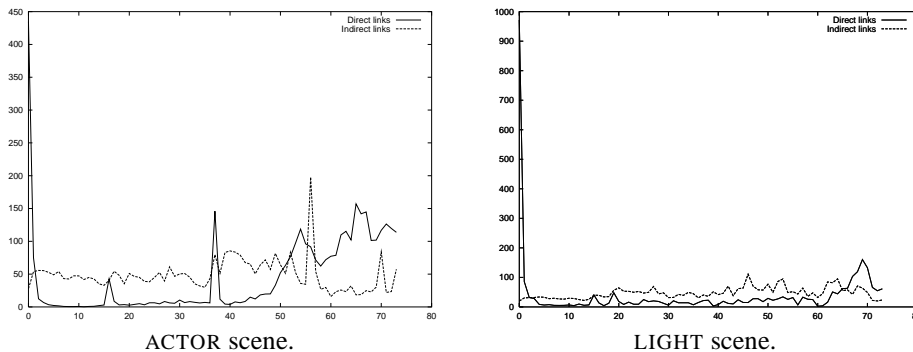


Fig. 2. Average amount of radiated energy with respect to the number of frames during which the associated link remains valid (arbitrary units). Because of scaling constraints, these figures does not include the links that remain valid for the whole animation.

carried by links that are valid for a single frame, reflecting the large amount of temporal change in direct lighting.

Another potential advantage of performing an integrated hierarchical radiosity solution over time is the ability to “play back” the animation in a virtual reality system or walkthrough application. In addition, having an integrated space-time hierarchy lets us combine coarse and fine object representations in both dimensions, without having to maintain separate structures as in the multi-resolution video application of [6].

4 A formal presentation of 4-dimensional radiosity

In this section we introduce notation and mathematical grounding for radiosity calculations over finite time intervals. Even though radiant energy exchanges are always considered instantaneous, which means that the radiosity balance equation at each time step is independent from all other time steps, space-time radiosity actually introduces temporal coupling between discrete radiating elements. This will allow us to build a hierarchical solution with elements of varying duration.

4.1 The 4D diffuse illumination equation

We want to compute the outgoing light for a set of ideally diffuse surfaces, for each moment in a given time interval. In other words, we want to find a function $B(x, t)$ giving us the radiosity of point x at time t , defined on $S \times T$, where S is the set of all points of all surfaces of the scene, and $T = [t1, t2]$ is the interval during which we want to render the scene.

We should be given two functions $M(x, t)$ and $N(x, t)$ with values over \mathbb{R}^3 giving us the position and outgoing normal of point x in space at time t . We should also be given the diffuse reflectance $\rho(x)$ of each point x , and the exitance $E(x, t)$ at point x at time t . Under those assumptions, the rendering equation [12] becomes :

$$\forall X = (x, t) \in (S \times T) \quad B(X) = E(X) + \int_{Y=(y, t') \in (S \times T)} B(Y) \mathcal{K}(X, Y) dY \quad (1)$$

where

- \mathcal{K} is a function defined over $(S \times T)^2$ by

$$\mathcal{K}((x, t), (y, t')) = \rho(x)k(x, y, t)V(x, y, t)\delta(t, t') \quad (2)$$

- $\delta(t, t')$ is equal to 1 when $t = t'$ and 0 otherwise.
- k is the kernel function defined as :

$$k(x, y, t) = \frac{\cos\theta(x, y, t)\cos\theta(y, x, t)}{\pi r(x, y, t)^2} \quad (3)$$

- $\theta(x, y, t)$ is the angle between the directions of $N(x, t)$ and $(M(y, t) - M(x, t))$
- $r(x, y, t)$ is the distance between $M(x, t)$ and $M(y, t)$
- $V(x, y, t)$ is equal to 0 whenever the ray joining $M(x, t)$ and $M(y, t)$ is intercepted by a surface of S at time t , and equal to 1 otherwise (*i.e.* visibility).

Note that equation (1) describes inter-temporal light exchanges since t can be different from t' . However the Dirac function $\delta(t, t')$ sets those exchanges to zero. Equation (1) in fact still governs the energy exchange equilibrium independently for each frame as is physically the case. However, the introduction of time will ease the development of our method.

4.2 Finite element formulation

We now derive the finite element formulation of 4D radiosity similarly to that of standard radiosity [17] [4]. Given a set of basis functions $(u_i)_{1 \leq i \leq N}$ and an inner product $\langle \cdot, \cdot \rangle$ we want to compute the approximation \tilde{B} such that the residual

$$r(X) = \tilde{B}(X) - E(X) - \int_{Y \in (S \times T)} \tilde{B}(Y)\mathcal{K}(X, Y)dY \quad (4)$$

is orthogonal to all the u_i .

Space-Time radiosity can then be expressed as a matrix equation comparable to the traditional radiosity equation :

$$MB = \epsilon \quad (5)$$

with

$$\forall i \in [1, N] \quad \epsilon_i = \langle E, u_i \rangle \quad (6)$$

$$\forall (i, j) \in [1, N]^2 \quad M_{i,j} = \langle u_i, u_j \rangle - \int_{X \in (S \times T)} u_i(X) \int_{Y \in (S \times T)} \mathcal{K}(X, Y)u_j(Y)dYdX \quad (7)$$

4.3 4D radiosity with constant elements

We now derive the radiosity equation (5) in the case of constant basis function. We assume that our scene S is composed of polygons with rigid motion. We decompose these polygons into N disjoint spatio-temporal patches P_i over which radiosity is assumed constant.

The patches P_i are defined by their time domain $[t_i^1, t_i^2] \subset T$ and by their (possibly moving) geometry $Q_i^t \subset S$. Note that since the motion of Q_i^t is rigid, its area A_i is

constant. Assume moreover that its reflectance ρ is constant in time and space. We use equation (7) to find the expression of M :

$$M_{i,j} = A_i(t_i^2 - t_i^1)\delta_{i,j} - \rho_i A_i(t_i^2 - t_i^1)\mathcal{F}_{i,j} \quad (8)$$

where $\mathcal{F}_{i,j}$, the spatio-temporal form-factor between P_i and P_j is:

$$\mathcal{F}_{i,j} = \frac{1}{A_i(t_i^2 - t_i^1)} \int_{[t_i^1, t_i^2] \cap [t_j^1, t_j^2]} \int_{P_i} \int_{P_j} \frac{\cos\theta \cos\theta'}{\pi r^2} V(x, y, t) dy dx dt \quad (9)$$

Note that the time dependency of our radiosity coefficients B_i is now implicit through the time extension of u_i 's support. Furthermore, inter-temporal interactions are now possible, since any two time intervals sharing a non-empty intersection can exchange radiosity. This is not a violation of physical laws but rather a byproduct of the underlying approximation performed in the discretization of the problem.

5 A space-time hierarchical radiosity algorithm

Let us now define an efficient approach to solve equation (5) in a time-dependent scene. As noted in section (3), a hierarchical algorithm allowing us to store multi-resolution information for both spatial and temporal variations would ensure that each interaction between objects can be represented with an appropriate level of precision. Thus we will be able to factor out redundant computation.

In order to build a hierarchical algorithm for space-time radiosity, we define :

- An adequate data structure to store the values of the radiosity function over space-time elements, allowing us to compute and obtain these values at various levels of precision both over space and time.
- A refinement method with a time specific approach.

We discuss below the issues raised by these two items, and present first solutions chosen in our current implementation. We also discuss possible improvements of these basic solutions.

5.1 Hierarchical data structure

As stated in Section 4.3, discrete elements are defined both by their geometrical description and their temporal description, the time interval of their validity, over which the radiosity is assumed constant, and referred to as their *time range*. They can therefore be split either in space (*e.g.* in four sub-patches for quads and triangles) or in time (by dividing the time range, leaving the geometry unchanged). Therefore an element in our hierarchy can have children either in space or in time. Currently, we do not allow our elements to be subdivided in time and in space simultaneously, since this would only reduce the depth of our hierarchy but not the number of hierarchical elements.

Figure 3 shows two scenarios for the possible subdivision of a space-time patch, in a simplified space with only one spatial dimension (therefore it shows two children elements created by a spatial splitting instead of four).

Considering memory consumption issues, note that we store separately the geometry mapping function M giving us the position of each vertex at a given time. Thus we do not have to store this position in our element description; the only data that needs

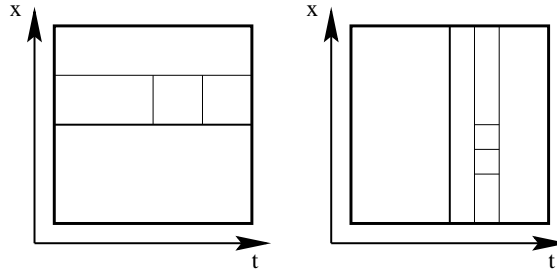


Fig. 3. Two examples of space/time hierarchy:

Left: An object is partially lit, therefore it is first subdivided spatially. At the end of our animation, a (small) shadow is briefly cast on its illuminated portion, inducing a temporal subdivision.

Right: An object is submitted to mainly uniform, though time-varying lighting conditions, therefore it is first subdivided in time. In the middle of the animation, a shadow is (briefly) cast on a small portion of its area.

to be created when time-splitting an element is its time extent and its radiosity value. Everything else can be shared between the two siblings and their parent.

We have implemented this space-time subdivision mechanism in a hierarchical radiosity simulator, using quadtree-like spatial subdivision and midpoint binary temporal subdivision. Note that more elaborate temporal subdivision schemes could easily be tried, in particular when information is available about a special event happening within the time range, to align the temporal boundaries to the event. For now, only surface patches are treated, but the notion of temporal refinement could be applied to other types of objects such as clusters.

With this hierarchical structure, the algorithm to solve equation 5 is the same as for classical hierarchical radiosity. From the original three parts composing an iteration (namely refine, push-pull, and gather), only the link refinement procedure needs specific changes in our case, which are described in section 5.2.

Once the solution computed, displaying the results for any given time t is straightforward. Leaf patches whose time interval contains t are found in the hierarchy and displayed with respect to their position (given by the function $M(t)$) and radiosity. This can be done rapidly.

5.2 Refinement

As in classical hierarchical radiosity, the refinement process is the most time-consuming part of each iteration, and inadequate refinement can produce an excessively heavy, yet inefficient hierarchy. The refinement pass requires a time specific approach be developed. We need to know whether a given link is a faithful enough representation of the light exchanged between two hierarchical elements, as in the purely spatial case. In addition, for the case where refinement is necessary, we need to know whether the elements' spatial or temporal extents are "responsible" for the inadequate representation, so as to be able to decide between time- and space-subdivision.

This decision is not easy in general, and can drastically impact the memory consumption as well as the time needed to compute the solution.

We chose a refinement criterion based on estimates of the variation of the form

factor over the patch surface and time range. We proceed by sampling the radiosity kernel function k and visibility function V (cf. Section 4.1), as well as the radiosity of the sender if it is not a leaf of the hierarchy (this allows us to take into account indirect lighting variations). To rely only on sampling can be problematic: shadow edges can be missed, causing flickering edges, or shadow movements can leave stains. On the other hand we can expect visibility to not depend on time for a large number of links (e.g. between nearby surfaces that do not move or that have the same movement). As a consequence, we decided to use shafts [11] to detect moving surfaces between the two nodes of a given link, to be able to force an appropriate subdivision.

1. If both the source and the receiver of the considered link share the same movement, we build the shaft joining their bounding boxes. Then we hierarchically test objects that do not move in the same way to see if they intersect the shaft and cause “significant” occlusion, during a limited portion of the link’s duration. When such transient blockers are found, we force time-refinement for this link.
2. In all other cases we sample the kernel function k and visibility function V . Then we compute the standard deviations of our sample set, considered with fixed time for various points and with varying time for fixed points. Finally we compare them first to a given threshold, to decide if we should refine, then one with each other, to decide which kind of refinement we apply.

Some optimizations are added to avoid unnecessary extensive sampling of both space and time:

- If both the source and receiver share the same motion, then the kernel function k does not depend on time (since we only consider rigid motions)
- If we detected that our shaft will never be crossed by a moving object then visibility does not depend on time

Unless there are very many large occluders moving around the scene, we can expect that a great number of links will benefit from these optimizations and therefore that the refinement test for these links will be as fast as if we were in a static scene.

5.3 Possible improvements

Improving the memory consumption of this algorithm could be achieved by taking advantage of the fact that computing the amount of light received by a patch only involves patches whose time extent overlap his own. Therefore it would be possible to find a hierarchy traversal method that could allow us to temporarily dump large parts of the hierarchy on disk, keeping only those needed in memory.

Another improvement that can be made addresses the fact that our hierarchy stores both time and space subdivisions, raising the following issue : it is possible that we might need to split an element in time (resp. in space) while it is already split in space (resp. in time) because of multiple or indirect light sources. For the moment we choose to push the interaction further down in the hierarchy until a leaf is found and the requested subdivision can be performed. This might lead to significant overcost as discussed in Section 6. A hierarchy maintenance pass could be added at the end of refinement that would be aimed at properly ordering our hierarchy to restrict this.

6 Results

We ran our algorithm on the two test scenes introduced in Section 3 (both animations are 3 seconds long, or 75 frames), and compared the results to the application of a static radiosity algorithm, for each frame, using the same spatial refinement parameters. Results (computed on a 250 MHz MIPS R10,000 processor) are summarized in the following table:

	Computation time		Memory usage	
	Space-Time	Static (per frame)	Space-Time	Static
ACTOR	335 s	11.5 s	112 Mb	11 Mb
LIGHT	671 s	18.1 s	237 Mb	14.3 Mb

In terms of quality, the solution obtained with space-time radiosity is almost identical to the frame-by-frame solution. Representative frames of both animations are shown in the color plates.

These results lead us to the following comments:

- The construction of a hierarchy representing the full length of the animation introduces a significant memory cost. Memory usage is greater than that of static calculation by a factor of about 10 (ACTOR) and 17 (LIGHT). Future efforts should therefore concentrate on making better use of available memory by scheduling calculations in order of increasing time, producing frames as time progresses and reclaiming memory on the fly.
- Our algorithm accelerates the calculation of a full-size animation clip, with the same illumination quality. Acceleration factors of about 2.5 (ACTOR) and 2 (LIGHT) are observed. This result is somewhat disappointing considering the increase in memory consumption and is clearly due to our intensive use of time-sampling during the evaluation of the oracle. In our first implementation attempts, we obtained acceleration factors of about 10 and 7, due to a different temporal sampling strategy which turned out to be unfair to the static version. The difference we observe shows that our current temporal sampling scheme (shooting significant numbers of rays distributed in time) heavily affects performance. Reducing the number of temporal visibility samples greatly decreases computation time, but damages the solution since occlusions can be missed, creating “holes” in moving shadows. Clearly, smarter techniques such as motion volumes should therefore allow large acceleration, similar to the use of shafts in the spatial dimensions.

Due to the hierarchical nature of the technique, the actual acceleration factor depends on the length of the sequence: in the example of the ACTOR scene, doubling the duration of the time interval (6 seconds, 150 frames) with equivalent movement resulted in a 55 % increase in computation time and 60 % increase in memory usage. This is an interesting result, similar to the well-known observation that spatial subdivision concentrates around shadow boundaries in static images, which means that additional subdivision is only performed in these limited areas when more detail is requested.

Note that this algorithm performs best for indirect lighting : an acceleration factor of 4 (ACTOR) and 3 (LIGHT) is observed when comparing only the time needed to obtain the indirect lighting in the dynamic and static calculations. This is due to the fact that indirect light exchanges vary more smoothly in time, as they do not induce sharp shadow edges movement.

We also observe that the LIGHT scene was specifically created to challenge our technique: direct illumination changes across the entire scene and during the entire duration of the cut; significant secondary illumination is due to the floor and wall areas near the lower light source, areas which are finely subdivided in time due to the shadow cast by the moving panel. Therefore this scene illustrates the difficulties encountered when these areas would warrant space-subdivision, but are already time-subdivided. Our current implementation forces the interactions to be pushed down the time hierarchy, with significant cost, but better solutions can certainly be found.

Beyond these numerical results, it is important to note that this framework opens the way for more elaborate refinement that will operate on the global knowledge of illumination variations across a time interval. Suppose the amount of energy flowing between a wall and an object is continuously increasing over a time interval, for instance because a light source is getting closer to the wall, and increases its radiosity. In the case of a frame-by-frame computation, there will be a discontinuous jump when the secondary illumination goes above the refinement threshold. If changes are computed incrementally [5], this will either be done from frame to frame, in which case the increment may be too small to be noticed, or with respect to a reference frame, which again introduces the risk of a discontinuous jump as we reach a threshold. Working over the full duration of a time interval, however, let us characterize illumination variations, and would allow us to apply knowledge of human perception to decide when additional detail or refinement is needed.

7 Conclusion and future work

In this paper, we have shown how the hierarchical radiosity algorithm can be extended to account for time-dependent scenes. The theoretical formulation of the time-dependent radiosity discretization provides a clear ground on which different approaches can be implemented. For instance, various time-space subdivision strategies can now be investigated, in particular to avoid explicit time-sampling of visibility and to address the problem of competing subdivisions in time and space.

The technique allows the calculation of animations equivalent to frame-by-frame computation, in a shorter time, at the cost of a much larger memory requirement. These results, obtained on a preliminary implementation, provide a proof that the concept is valid. From the restricted point of view of computing resources, our results are somewhat disappointing, considering the potential acceleration that should be drawn from the hierarchical formulation. However it should be stressed that a major advantage of space-time calculations is the potential ability to exploit the knowledge of temporal variations over finite time intervals to guide the subdivision. In this regard also, our refinement criterion is somewhat preliminary, and better, perceptually-based criteria are called for.

Our results also point us towards several very interesting areas for future research, which we are actively pursuing. In order of priority we can quote :

1. The derivation of good space-time refinement criteria that would avoid explicit time-sampling of visibility by taking into account object trajectories (*e.g.* using swept volumes, knowledge of the speed of the blocker, etc.). Extending our shaft-culling routine to estimate visibility between two objects with different motion. Using the shafts joining their motion volumes' bounding box, will certainly help us to do so.
2. Efficient treatment and maintenance of the hierarchy. Two directions of improve-

ment are : (a) introducing the ability to “space-subdivide” an element which is already “time-subdivided”, with graceful management of different views of the hierarchy, and (b) optimizing the hierarchy according to perceptual and energy criteria, to remove unnecessary subdivision. Note that the latter optimization could perhaps be incorporated in the refinement criterion, but requires a global knowledge of all interactions in space. Therefore it is perhaps best left to a post-processing stage (after each refinement iteration). Both improvements would reduce the number of elements and links in the hierarchy and therefore the memory consumption and computation time as well.

3. Definition of a good ordering strategy for refinement, so as to allow efficient usage of memory, by processing energy exchanges roughly in the order in which they happen. This would facilitate garbage collection mechanisms, but is a non-trivial problem because of the global solution involving links of different “duration”. A long-lived link can both influence a short-lived “early patch” and be influenced by a later patch. Such temporal influences are, of course, non-physical and should be kept minimal by appropriate refinement. A precise study of the allowable time approximations is therefore required.
4. Strategies for time dependent *clustering* [18, 16] have to be developed. As the complexity of scenes increases, this would be likely to change the pattern of hierarchical subdivision. While our subdivision trees are fairly balanced between time and space in the two tests scenes, for more complex scenes it would be more likely that time subdivision continues after space subdivision has stopped at an appropriate cluster level. Greater savings can then be expected. Amidst the problems that we are currently investigating is the fact that clusterisation is typically implemented as an *a priori* spatial subdivision of the scene. This means that we have to recursively duplicate all the objects in a cluster if we want to time-split the cluster.
5. Replacing mesh subdivision by textures [19, 10] is an effective means of avoiding the cost due to the subdivided hierarchy itself. In the case of time-dependent illumination, texture movies could be generated for this purpose, although a specific algorithm has yet to be proposed [13].
6. The use of a *shooting* approach [20, 10] would eliminate the need to store all the links in the solution, especially the links from primary light sources.
7. A precise comparison of this algorithm with Drettakis’ and Sillion [5] dynamic update technique based on the line-space hierarchy would help to investigate the potential for time-dependent line space refinement.
8. The use of more elaborate finite elements for radiosity representation should be carefully studied. For instance, linear elements in the time direction could prove especially useful to avoid discontinuities. In general, reconstruction of a frame from a space-time radiosity solution should be carried out in a precise manner.

We believe that hierarchical computation of time-dependent global illumination can be put to efficient use if sufficient attention is devoted to all these issues, resulting in practical tools that will effectively accelerate lighting simulation for real applications such as movie lighting and virtual sets.

Acknowledgments

iMAGIS is a joint research project of INRIA, CNRS, INPG and UJF. This work was supported in part by the European union under the “ARCADE” ESPRIT LTR project, #24944.

References

1. Daniel R. Baum, John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. The back-buffer algorithm : an extension of the radiosity method to dynamic environments. *The Visual Computer*, 2:298–306, 1986.
2. Gonzalo Besuievsky and Mateu Sbert. The Multi-Frame Lighting Method: A Monte Carlo Based Solution for Radiosity in Dynamic Environments. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 185–194, New York, NY, 1996. Springer-Verlag/Wien.
3. Shenchang Eric Chen. Incremental radiosity: An extension of progressive radiosity to an interactive image synthesis system. *Computer Graphics*, 24(4):135–144, August 1990. Proceedings SIGGRAPH '90 in Dallas (USA).
4. Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, Boston, 1993.
5. George Drettakis and François Sillion. Interactive update of global illumination using a line-space hierarchy. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '97* (Los Angeles, CA), pages 57–64. ACM SIGGRAPH, New York, August 1997.
6. Adam Finkelstein, Charles E. Jacobs, and David H. Salesin. Multiresolution video. In *Computer Graphics, Annual Conference Series: (ACM SIGGRAPH '96 Proceedings)*, pages 281–290, 1996.
7. David A. Forsyth, Chien Yang, and Kim Teo. Efficient Radiosity in Dynamic Environments. In *Fifth Eurographics Workshop on Rendering*, pages 313–323, Darmstadt, Germany, June 1994.
8. David W. George, François Sillion, and Donald P. Greenberg. Radiosity redistribution for dynamic environments. *IEEE Computer Graphics and Applications*, 10(4), July 1990.
9. Andrew S. Glassner. Spacetime ray tracing for animation. *IEEE Computer Graphics and Applications*, 8(2):60–70, March 1988.
10. Xavier Granier and George Drettakis. Controlling memory consumption of hierarchical radiosity with clustering. In *Graphics Interface*, June 1999.
11. Eric Haines and John Wallace. Shaft culling for efficient ray-traced radiosity. In *Eurographics Workshop on Rendering*, pages 122–138, 1991.
12. James T. Kajiya. The Rendering Equation. In *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.
13. Ignacio Martin. Personal Communication.
14. Rachel Orti, Stephane Riviere, Fredo Durand, and Claude Puech. Radiosity for Dynamic Scenes in Flatland with the Visibility Complex. In *Computer Graphics Forum, Proc. EUROGRAPHICS '96*, volume 15, pages C237–C248. Blackwell, September 1996.
15. Erin Shaw. Hierarchical radiosity for dynamic environments. *Computer Graphics Forum*, 16(2):107–118, 1997. ISSN 0167-7055.
16. François Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3), September 1995. (a preliminary version appeared in the fifth Eurographics workshop on rendering, Darmstadt, Germany, June 1994).
17. François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann publishers, San Francisco, 1994.
18. Brian Smits, James Arvo, and Donald P. Greenberg. A clustering algorithm for radiosity in complex environments. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '94* (Orlando, FL), pages 435–442. ACM SIGGRAPH, New York, July 1994.
19. Cyril Soler and François Sillion. Fast calculation of soft shadow textures using convolution. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '98* (Orlando, FL), pages 321–332. ACM SIGGRAPH, New York, July 1998.
20. Marc Stamminger, Hartmut Schirmacher, Philipp Slusallek, and Hans-Pieter Seidel. Getting rid of links in hierarchical radiosity. In *Computer Graphics Forum, Proc. EUROGRAPHICS '98*, pages 165–174. Blackwell, 1998.

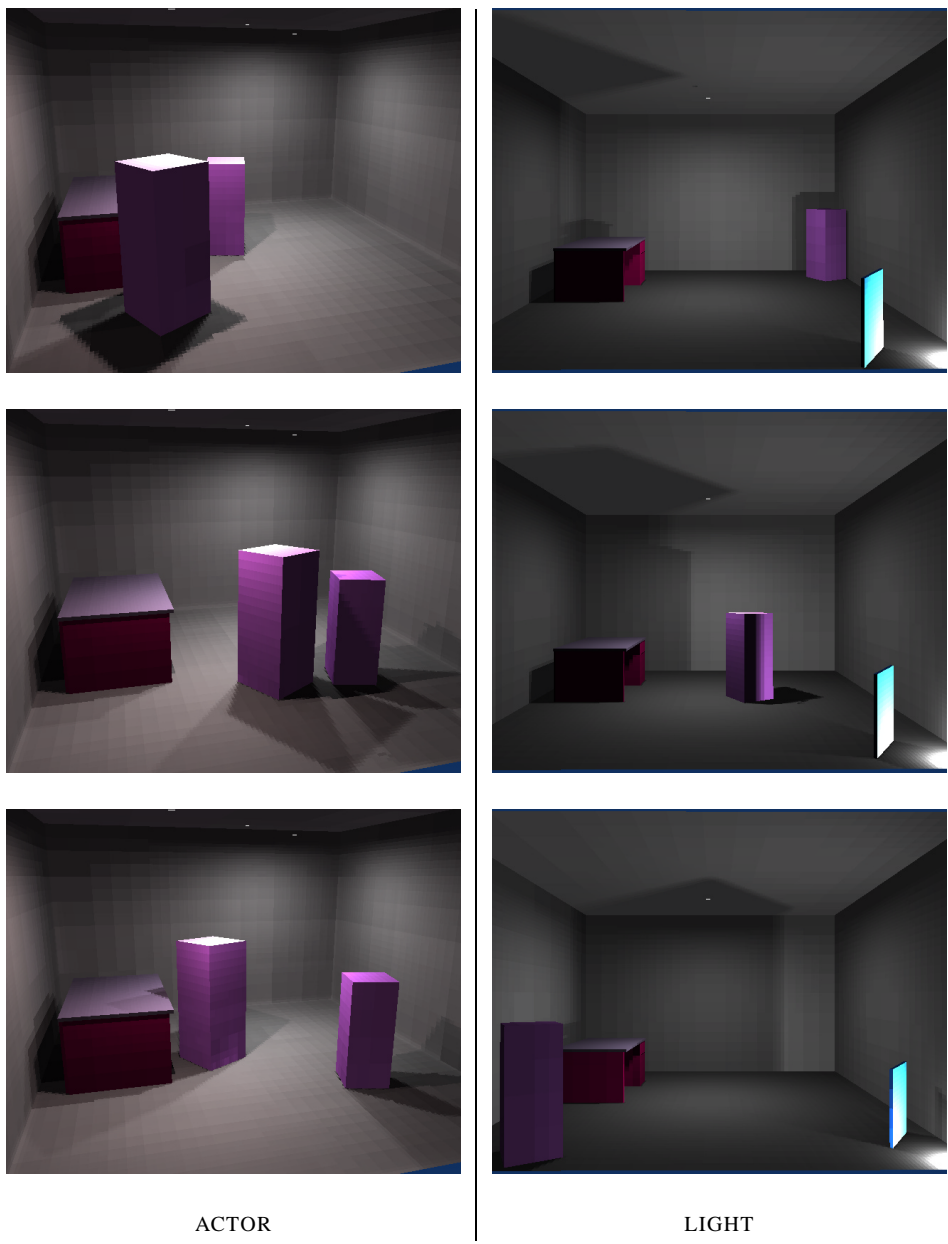


Fig. 4. Representative frames from example animations.