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

Cyrille Damez, Nicolas Holzschuch and François X. Sillion

*i*MAGIS/GRAVIR-IMAG[†] INRIA

Abstract

We address in this paper the issue of computing 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 dangerous and 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 changing the wavelet basis used for the temporal dimension. The resulting algorithm is capable of creating time-dependent radiosity solutions efficiently.

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, no global illumination technique to date delivers the desired effects in a simple and easy-to-use manner. In fact, these techniques remain typically expensive to use, even more so in the case of frame-by-frame lighting calculations.

The space-time hierarchical radiosity method[5] is a promising method to compute global illumination simulations for animated scenes. It works with scenes containing moving solid objects, whose itinerary is known beforehand, by computing 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. Areas with little variations of the illumination in time will be little refined in time, while possibly being refined in space if there is a variation of the illumination 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.

Space-time hierarchical radiosity is an interesting method for scenes which contain lots of changing indirect illumination. It efficiently computes the change in the illumination of the scene due to a moving object or light source. Thus it is particularly interesting for applications where it is important to have a physically accurate representation of the scene illumination, including indirect illumination.

In a way similar to the original hierarchical radiosity algorithm[7], the efficiency of the space-time hierarchical radiosity algorithm depends on the levels 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 very complex scenes.

In this paper, we present two major improvements of the

[†] GRAVIR is a joint research laboratory of CNRS, INRIA, Institut National Polytechnique de Grenoble and Université Joseph Fourier.

space-time hierarchical radiosity method : first, we have modified the algorithm so that it uses higher order 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 review previous work in hierarchical radiosity and time-dependent illumination. Then, in section 3, we present our algorithm for using higher-order wavelets for the time dimension. In section 4, we present our lazy-clustering algorithm for space-time hierarchical radiosity. Finally, in section 5 we draw our conclusions and trace directions for future work.

2. Previous Work and Motivation

Hierarchical radiosity was introduced in 1991[7]. Using a multi-level representation of the radiosity function, hierarchical radiosity achieves great speed-ups of illumination simulations. The acceleration introduced by the hierarchical formulation is directly linked to the depth of the hierarchy.

This depth is limited by the original geometry of the scene. If it consists of many small polygons, refining these polygons does not improve the precision of the computations. As a consequence, the original scene will be little refined, and we will not benefit from the hierarchical formulation.

This limitation can be removed if the hierarchy is extended above the polygons in the original scene[10, 11]. Grouping together spatially close polygons into clusters results in a considerable acceleration of radiosity computations. It also reduces the numerical complexity of the algorithm, enabling computations on larger scenes (more than a million input polygons).

In space-time hierarchical radiosity[5], the hierarchical formulation is extended along the time dimension. Interactions between objects can be refined either in space, the usual way, or in time. This results in a 4 dimensional space-time hierarchy that exploits temporal coherence. In its original form[5], space-time hierarchical radiosity did not work with clustering, which limited computations to smaller scenes, below ten thousand input polygons.

It also introduced temporal discontinuities in the radiosity function. These discontinuities appeared more clearly in indirect illumination, at times equal to $\frac{k}{2^p}$ times the length of the animation, with k and p integers (see figure 2). These discontinuities are clearly noticeable on the accompanying video, at $t = \frac{1}{2}l$, $t = \frac{1}{4}l$ and $t = \frac{3}{4}l$, where l is the length of the animation. Similar discontinuities also appeared in space with hierarchical radiosity (see figure 1). They are more disturbing when they occur in the temporal dimension because they cause sudden "jumps" in the illumination of numerous patches of the scene.

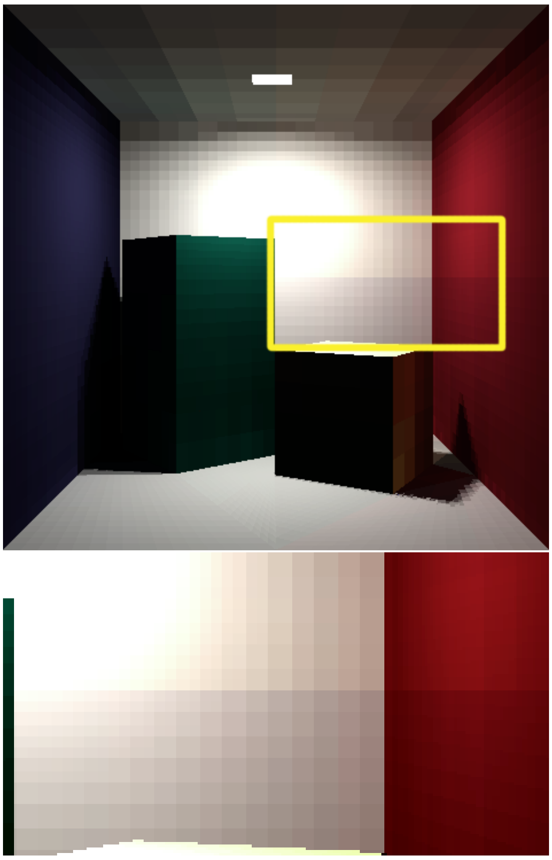

Figure 1: *Example of spatial discontinuities.*

Such spatial discontinuities can be treated either by a post-processing or eliminated during the computations, using higher-order functions:

- post-processing requires knowing the neighbors of any element, which is difficult with space-time hierarchy. It doesn't improve the accuracy of the solution but only correct its aspect.
- higher-order functions are, at the time of writing, not compatible with clustering.

Given our specific requirements we have elected to use higher-order functions for the time dimension, and constant basis functions for the spatial dimensions to allow the use of clusters, with post-processing to remove spatial discontinuities.

Higher-order wavelets have been previously used as function basis for hierarchical radiosity[6, 9]. If not carefully implemented, they were proved impractical, slower than hierarchical radiosity and giving poorer results[13]. Recent research[4] has shown that using new implementation methods[1, 2, 4, 12]

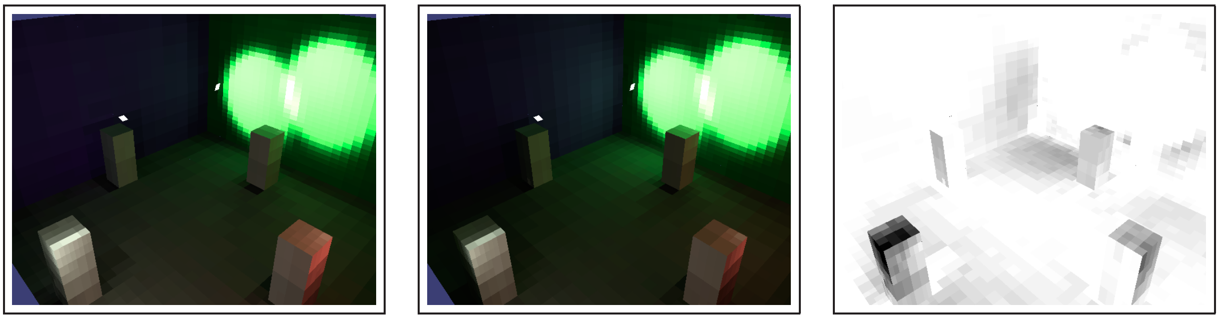

Figure 2: *Example of temporal discontinuities. Visualizations of the scene at $t = \frac{1}{2} - \varepsilon$, at $t = \frac{1}{2} + \varepsilon$, and the difference. The illumination of the entire scene has been modified in a single frame update.*

higher order wavelets were providing a better approximation of the illumination function, requiring less memory and computation time. The radiosity function produced looks continuous without post-processing thanks to the refinement oracle.

3. Higher-Order Wavelets for Space-Time Hierarchical Radiosity

We introduce higher-order wavelets in the time dimension of space-time hierarchical radiosity to reduce temporal discontinuities in radiosity. We use constant basis function in the spatial dimensions, in order to use clustering.

Since we are introducing wavelets only along one dimension, time, we will not encounter the implementation problems caused by multi-dimensional wavelets[4], such as memory costs or tedious computation of push-pull coefficients. We only need a specific refinement oracle; the rest of the implementation is straightforward.

3.1. The Refinement Oracle

A refinement oracle created for constant basis functions[5] would not exploit the power of higher order wavelets to interpolate rapidly varying functions. We have extended for space-time radiosity an oracle that works for wavelet radiosity[1, 2, 4], estimating the error on the propagated energy instead of the error on the transfer coefficients.

We use a grid of control points located on the receiving element, and a set of control dates. On these control points, at the control dates, we estimate the radiosity value using two methods: first, by interpolating the radiosity coefficients already computed for this interaction; second, by direct integration of the radiosity on the emitter, using a quadrature. The difference between these two values is an indication of the error in the representation of the interaction at the current level at this point. 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.

The control points and dates must be carefully chosen so that they provide meaningful information. They must be different from the quadrature points and dates used for form factor computations. The number of control points and dates must be higher for large receivers so that we do not miss important features. We have control dates at the beginning and at the end of the time interval, whose weight we can changer in order to ensure 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 our error on the interaction:

- A *spatial variance*, computed at a fixed date with varying control points.
- A *temporal variance*, computed at a fixed point, for varying control dates.

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

3.2. Experimental Results

We have tested space-time hierarchical radiosity with higher order wavelets on a simple test scene, to estimate the improvement in temporal continuity.

Our test scene (see accompanying videos and figure 2) is simple, but it is mainly lit by indirect lighting. It contains four white boxes, illuminated by four moving spotlights. While the boxes, floor and ceiling are white, each wall of the room has a different color (light red, green, blue, and grey). Indirect illumination is varying rapidly, and the scene can be seen as a worst-case scenario for space-time hierarchical radiosity. All our animations are 24*s* long, which corresponds to 600 frames.

Our first video shows a computation of the animation using standard hierarchical radiosity. The radiosity function has several noticeable temporal discontinuities. At first

	Computation time			Memory
	Direct	Indirect	Per frame	
Constant	254s	1,492s	2.9s	587 MB
$\mathcal{M}2$	271s	1,172s	2.4s	464 MB
1 frame	1s	15s	16.0s	5 MB

Table 1: *Computation Time, first test scene*

glance, these discontinuities can be confused with temporal discontinuities occurring from video decompression. A careful examination of the video will show that the direct illumination component is continuous, while the indirect illumination has discontinuities. These discontinuities are located at $t = \frac{k}{2^p} \times AnimationLength$, mainly at $t = 12s$ and $t = 6s$.

The second video shows the same animation computed using $\mathcal{M}2$ wavelets for time. The entire animation is much smoother. The temporal discontinuities still appear on careful examination, but they are much less prominent and can be ignored at first glance.

An example of these temporal discontinuities is shown in figure 3. The radiosity function is plotted for the entire animation, at the point defined by the center of the highlighted element, for both basis. The $\mathcal{M}2$ wavelets are not giving a continuous representation of the radiosity function, but the discontinuities are much smaller than with the constant basis.

The largest temporal discontinuities usually occur at the middle of the animation. Figure 4 shows the difference between the two frames just before and just after the middle of the animation, for constant basis and $\mathcal{M}2$ wavelets. In the first case, the discontinuities in indirect illumination are clearly visible. With $\mathcal{M}2$ wavelets, the differences are much smaller.

Table 1 summarizes the computation time and memory usage of our implementation for the constant and $\mathcal{M}2$ function basis (computed on a 300Mhz MIPS R12000 processor). We also computed a single frame using a similar refinement oracle performing only spatial refinement.

The following comments can be made about these results:

- Acceleration factor of about 5.5 and 6.5 are obtained for the constant and $\mathcal{M}2$ basis respectively.
- Though the memory requirements are still high (keep in mind that we are computing a complete view independent solution for 600 frames of animation), we can see that the use of the $\mathcal{M}2$ function basis only introduces a 15 % memory overcost : despite their higher storage cost for each interaction, they reduce the total number of interactions. This also explains the faster computation time.

4. Using clusters in space-time hierarchical radiosity

To apply space-time hierarchical radiosity on complex scenes, we need to combine the space-time approach with clustering. This raises two key issues:

- we need to build a hierarchy of clusters over the scene, compatible with the space-time approach. We introduce a new, *lazy clustering* approach, described in section 4.1.
- we need appropriate computation of light exchanges involving clusters. Our method is described in section 4.2.

Introducing clusters in a wavelet space-time radiosity method also requires some minor adjustments to the refinement oracle, which we shall describe in section 4.3.

4.1. Lazy Clustering

Space-time hierarchical radiosity is an extension of hierarchical radiosity, in which a fourth dimension, time, is added to the three dimensions of space. This algorithm produces a radiosity solution defined on a 4-dimensional hierarchical mesh. The main difference in the algorithm is that at each refining step, we must not only decide if refinement of each link is necessary, but also whether to refine this link in time or in space. The algorithm produces a single hierarchy merging together both spatial and temporal hierarchical representations.

The construction of the cluster hierarchy is performed as a pre-computation step in classical hierarchical radiosity. It is impossible to use such an approach in the space-time hierarchical radiosity algorithm, since the resulting spatial hierarchy would pre-exist the refinement, preventing any temporal refinement of a link until we reach the surface level. The construction of the cluster hierarchy needs to be performed gradually along the refinement process, in the same way the spatial hierarchy of patches is produced on surfaces. We refer to this approach as *lazy clustering*.

In space-time radiosity, an unrefined cluster is a group of surfaces, sharing the same movement and defined over the same time interval. We start our computations with a very small set of clusters: the root cluster, plus one cluster for each different trajectory in the animation. The hierarchy of clusters will be produced as a byproduct of the refinement procedure.

As for surface elements in the space-time mesh, we can split unrefined clusters either in space or in time:

- Time-refinement is performed in defining two clusters as children of the original, each one defined over half the original time interval. Every surface inside the original cluster must be duplicated and one copy must be assigned to each of the two children clusters. It might be desirable to prevent time-splitting of clusters containing a large number of surfaces for evident memory efficiency reasons; in our implementation, we chose to forbid time re-

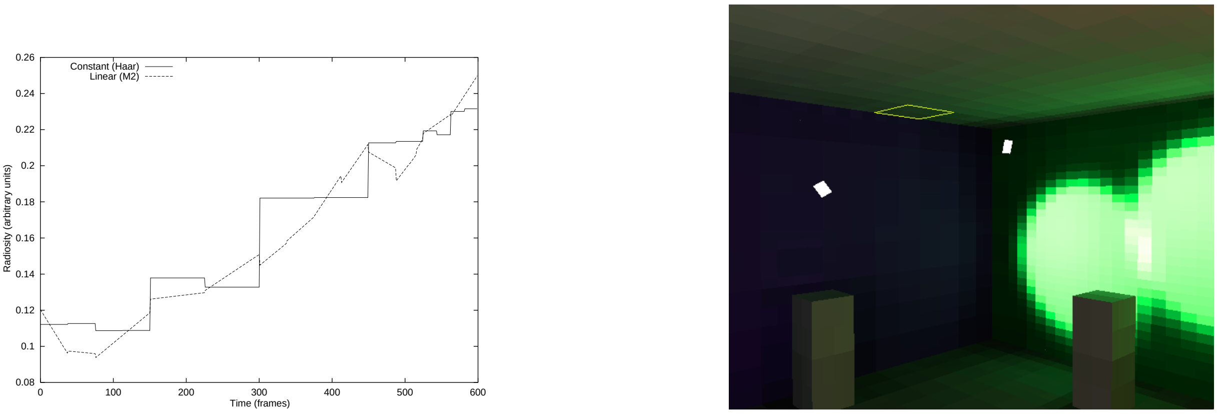

Figure 3: *Variation of the radiosity function at the center of the highlighted element during the animation*

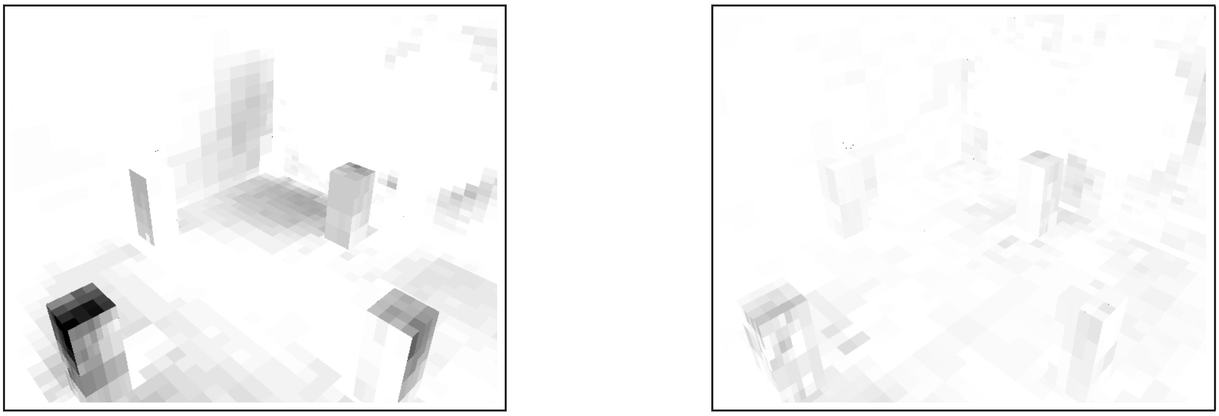

Figure 4: *Comparison of temporal discontinuities at $t = \frac{1}{2}$. Darker colors indicate a higher discontinuity. Left: hierarchical radiosity, right: $\mathcal{M}2$ wavelets.*

finement of clusters containing more than a few hundreds of polygons.

- Space-refinement of a cluster is the act of grouping together some of the surfaces contained in the said cluster, forming new children clusters, eventually leaving some surfaces as direct children. This is achieved in applying only one step of any classical top-down recursive clustering method (without the recursive call). We chose to adapt the “tight-fitting octree” method[3], which is straightforward to implement, top-down, recursive and produces a rather well-formed spatial hierarchy[8].

4.2. Computing energy transfers involving clusters

In this section, we focus on the description of the computation of light exchanges involving clusters in space-time hierarchical radiosity.

Our approach is based on the one exposed by Sillion for standard clusters[10]. Roughly, anisotropic emission from a cluster is approximated by going down to the surface 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.

4.2.1. 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 Q and the receiver P. 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 all the surfaces

Q_i contained in the cluster Q[10]. At a given time it is given by:

$$I_{\text{received}}(y,t) = \sum_i \int_{Q_i} B_{Q_i}(x,t)K(x,y,t)V(x,y,t)dx \qquad (1)$$

where the kernel function is defined by:

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

and the $\mathcal{R}$ function is the receiver factor defined in[10] as $\cos\theta$ if the receiver is a surface and 1 if the receiver is a cluster (orientation of the surfaces in the receiver cluster will be taken into account in the immediate push).

We need to compute the projection of equation 1 on our function basis. This amounts to compute the coefficients α_j such that:

$$I_{\text{received}}(t) = \sum_j \alpha_j \phi_j(t)$$

where the ϕ_j are our wavelet basis functions at the current time interval.

Since the ϕ_j form an orthonormal basis, we can express the α_j as scalar products:

$$\begin{aligned}\alpha_j &= < I_{\text{received}}|\phi_j > \\ \alpha_j &= \sum_i \int_t \left(\int_P \int_{Q_i} K(x,y,t)V(x,y,t)dxdy\right) B_{Q_i}(t)\phi_j(t)dt\end{aligned}$$

Computing the aforementioned 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:

$$\alpha_j = \sum_i \int_t \left(\int_P \int_{Q_i} K(x,y,t)dxdy\right) B_{Q_i}(t)V(t)\phi_j(t)dt \quad (2)$$

where

$$V(t) = \int_P \int_Q V(x,y,t)dxdy$$

We compute the α_j and $V(t)$ using a Gaussian quadrature(quadrature points for the cluster Q are chosen in the bounding box of all surfaces Q_i).

4.2.2. Reception Inside a Cluster: Immediate Push

The reception inside a cluster obeys the *immediate push* principle[10]: the irradiance received at the cluster level is immediately dispatched to all surfaces inside the cluster, where it is multiplied by the *receiver factor* – the angle between the normal of the surface and the direction of the incoming radiance. The direction of the incoming radiance is assumed to be the center of the emitter, whether a cluster or a surface.

As the cluster and the sender might be mobile, the receiver factor is time-dependent. We need to project it on our wavelet base. Let us assume a cluster P has received an irradiance I_{received}. This irradiance is now computed as a set of wavelet coefficients:

$$I_{\text{received}}(t) = \sum_j \alpha_j \phi_j(t)$$

For every surface P_i in the cluster P, this irradiance is then distributed to each P_i according to its receiver factor:

$$I_{P_i} = I_{\text{received}}(t)\mathcal{R}_i(t) = I_q(t)\cos\theta_i(t)$$

I_{P_i} is then reprojected on the wavelet base for the time interval of P_i:

$$I_{P_i} = \sum_j \beta_j \phi_j(t)$$

Since the ϕ_j form an orthonormal base, each wavelet coefficient is expressed as a scalar product:

$$\begin{aligned}\beta_j &= < I_{P_i}|\phi_j > \\ &= \int_t I_{P_i}(t)\cos\theta_i(t)\phi_j(t)dt \\ &= \sum_j \alpha_j \int_t \cos\theta(t)\phi_j(t)\phi_i(t)dt\end{aligned}$$

The problem therefore resolves to computing n integrals, where n is the order of the wavelet base. These integrals are 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 base, then dispatched to the surfaces, taking into account the surface movement, and re-projected on the function basis for the receiving surface. This double approximation is consistent with the clustering approach. If the refinement oracle decide that we must 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 performances.

4.3. Refinement oracle using clusters

Our refinement criterion (cf. 3.1) requires us to select control points on the receiving object. This is straightforward if the object is a surface. If the target object is a cluster, taking control points at spatial locations inside the cluster, then look for the closest surface would be too costly.

We have elected to use a surface-based sampling method: if the oracle wants N sample points in the cluster, and the cluster contains M surfaces:

- if $N > M$, we choose $\lceil \frac{N}{M} \rceil$ control points on each surface.
- if $N < M$, there is less than one control point on each surface. We take one surface out of $\frac{M}{N}$, and take one control point on each of these surfaces.

This algorithm usually ensures an homogeneous repartition of the control points inside the cluster.

	Computation time			Memory
	Direct	Indirect	Per frame	
$\mathcal{M}2$	146s	4,445s	8s	415MB
1 frame	6s	103s	109s	17 MB

Table 2: *Experimental results, clusterized scene*

4.4. Experimental Results

We have tested our clustering approach on complex test scenes (see accompanying videos): the four cubes in our room were replaced by more complex models. We kept the same moving spotlights. The resulting scene contains about 30,000 polygons, and exhibits complex time-dependent lighting conditions. Experimental results are summarized in table 2.

We obtain similar quality frames from both the time dependent and the classical hierarchical radiosity algorithm. The acceleration factor is about 14. However, this good result must be moderated by the fact that the approximations implied by clustering have brought back some temporal discontinuities in our solution.

5. Conclusion and Future Work

In this paper, we have shown how space-time hierarchical radiosity can be made to work with complex scenes, by introducing the notion of lazy clustering and allowing a flexible tradeoff in the organization of the space-time hierarchy.

We have also proposed a modification of the space-time radiosity method using higher-order wavelets for the specific representation of temporal variations of the radiosity function, resulting in smoother animations, with fewer discontinuities and slightly better computation times thanks to reduced hierarchical subdivision.

The resulting algorithm brings us closer to our ultimate goal, which is to provide a tool to generate high-quality lighting solutions for animated sequences known in advance, where all parameters of the animation can be employed to optimize the usage of resources and minimize visual artifacts.

This algorithm is still in progress. Areas for future research includes the definition of better refinement criteria ensuring even better temporal continuity, especially when using clustering. We are also investigating ordering strategies for refinement to allow disk caching and therefore reduce the amount of memory required by the space-time radiosity algorithm.

References

1. P. Bekaert and Y. Willems. Error Control for Radiosity. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 153–164, New York, NY, 1996. Springer-Verlag/Wien. 2, 3

2. 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, June 1996. Comenius University Press. 2, 3

3. P. H. Christensen, D. Lischinski, E. J. Stollnitz, and D. H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1):3–33, January 1997. 5

4. F. Cuny, L. Alonso, and N. Holzschuch. A novel approach makes higher order wavelets really efficient for radiosity. *Computer Graphics Forum (Proc. Eurographics 2000)*, 19(3):C–99–C–108, 2000. 2, 3

5. C. Damez and F. Sillion. Space-time hierarchical radiosity. In *Rendering Techniques '99*, pages 235–246, New York, NY, 1999. Springer Wien. 1, 2, 3

6. S. J. Gortler, P. Schroder, M. F. Cohen, and P. Hanrahan. Wavelet Radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH '93 Proceedings)*, pages 221–230, 1993. 2

7. P. Hanrahan, D. Salzman, and L. Aupperle. A Rapid Hierarchical Radiosity Algorithm. *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, 25(4):197–206, July 1991. 1, 2

8. J. M. Hasenfratz, C. Damez, F. Sillion, and G. Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. *Computer Graphics Forum (Proc. Eurographics '99)*, 18(3):C–221–C–232, September 1999. 5

9. P. Schroder, S. J. Gortler, M. F. Cohen, and P. Hanrahan. Wavelet Projections for Radiosity. In *Fourth Eurographics Workshop on Rendering*, pages 105–114, Paris, France, June 1993. 2

10. F. Sillion. A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3), Sept. 1995. 2, 5, 6

11. B. Smits, J. Arvo, and D. Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. In *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pages 435–442, 1994. 2

12. M. Stamminger, H. Schirmacher, P. Slusallek, and H.-P. Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Forum (Proc. Eurographics '98)*, 17(3):C165–C174, September 1998. 2

13. A. Willmott and P. Heckbert. An empirical comparison of progressive and wavelet radiosity. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 175–186, New York, NY, 1997. Springer Wien. 2

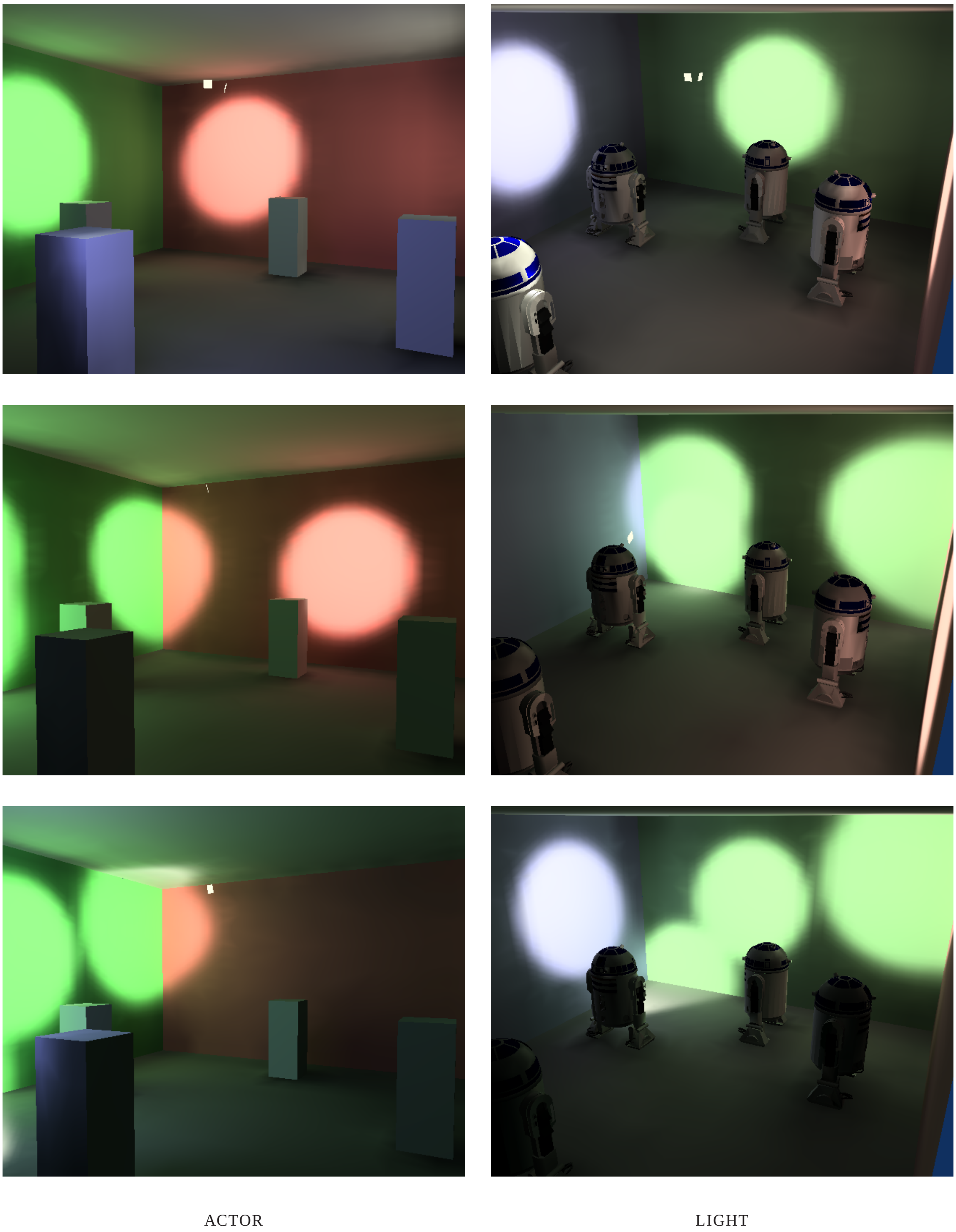

Figure 5: *Representative frames from example animations.*