A Monte Carlo Method for Accelerating the Computation of Animated Radiosity Sequences

Gonzalo Besuievsky
Xavier Pueyo
Institut d'Informàtica i Aplicacions
Universitat de Girona
Campus de Montilivi, E 17071 Girona, Spain
(gonzalo|xavier)@ima.udg.es

Abstract

Realistic rendering animation is known to be an expensive processing task when physically-based global illumination methods are used in order to improve illumination details. This paper presents an acceleration technique to compute animations in radiosity environments. The technique is based on an interpolated approach that exploits temporal coherence in radiosity. A fast global Monte Carlo pre-processing step is introduced to the whole computation of the animated sequence to select important frames. These are fully computed and used as a base for the interpolation of all the sequence. The approach is completely view-independent. Once the illumination is computed, it can be visualized by any animated camera. Results present significant high speed-ups showing that the technique could be an interesting alternative to deterministic methods for computing non-interactive radiosity animations for moderately complex scenarios.

Keywords: Temporal Coherence, Radiosity, Realistic Animation, Monte Carlo.

1. Introduction

Realistic animation is a major goal in Image Synthesis. One of the most important topics involving the generation of realism in an animated sequence is the accurate computation of the illumination. Although physically-based global illumination models were successfully developed to compute accurate realistic scenes they are still time consuming for dealing with animated sequences. Efficient solution are searched in order to accelerate this computation task. Considering that changes in illumination between consecutive frames in an animation are in general smooth, a natural possibility for an efficient approach is to improve global illumination models with the aim of making use of this source of coherence. A significant computation time can be saved by reusing the illumination already computed that is invariant in an interval of time within the animation.

The radiosity method, introduced in [8], is often preferred as the global illumination model because in addition to its accurate results, it brings a view-independent solution that allows interactive walkthroughs, a desired feature in many applications. Works focusing on both interactive and non-interactive application have been presented using different strategies like progressive refinements methods [3, 7, 11], hierarchical radiosity [6, 13, 5, 4, 10], range-image based techniques [12] or Monte Carlo approaches [1]. When all time-dependent parameters on an animated environment can be known a priori, temporal coherence can be exploited better than interactive cases, where no previous knowledge of the modification is available. Surprisingly, only a few techniques of previous work make use of the knowledge of the motion to improve lighting computations.

In this paper we propose an acceleration technique for computing animated radiosity sequences based on the use of temporal coherence in illumination. The technique uses the Multi-Frame Lighting method [1, 2], a Monte Carlo approach originally proposed to improve processing in radiosity animated sequences, to perform a fast estimation of the illumination over time along the whole sequence. This estimation is used to build a base of key-frames timing points which are full computed and used to reconstruct the whole animation by interpolation.

In section 2 we describe previous work on animated environments and review the Multi-Frame Lighting Method. We investigate temporal coherence in radiosity through a sample sequence in section 3. Sections 4, 5 and 6 describe all steps of our method. An analysis of the proposal is presented in section 7 and results are shown in section 8. Finally, conclusions and future work are presented in sec-

0-7695-1007-8/01 $10.00© 2001 IEEE

tion 9.

2. Background

2.1. Previous Work

Most of previous techniques developed to deal with animated environments focus on interactive purposes where a fast illumination update is expected after a given modification. In general in such cases there is a high compromise in the time response. Also, the tradeoff between quality and speed is frequently desired to be tunable. Progressive refinement strategies are widely used to attend these requirements both using classic radiosity [3, 7, 11] or hierarchical radiosity [6, 13, 5].

In animation where all time-dependent parameters are available before rendering, invariants in illumination can be detected more easily than in the interactive case and, as a consequence, temporal coherence can be exploited better. Although interactive techniques could be adapted for animation, a specific approach can give more benefits. However, only a few techniques of previous work make real use of the knowledge of the motion to improve lighting computations. Nimeroff et al. [12] presented a range-image based framework consisting on computing a set of base view images from which the whole animation is obtained by interpolation. For the lighting computation, direct and indirect illumination are split in different independent processes in order to improve spatial and temporal coherence. Recently, a frame-to-frame coherence approach based on a two-pass radiosity solution was proposed in [9]. In [1] we propose a Monte Carlo based approach called the Multi-Frame Lighting method that computes all frames of a sequence in a single process. In this approach visibility changes are handled through the use of global lines and time-independent visibility lists. The method is reviewed in the next section.

2.2. The Multi-Frame Lighting Method (MFLM)

This method works in a multi-processing task strategy: given an animated sequence description in an environment it computes the radiosity solution of all frames in a single process. The lighting computation it based on Monte Carlo radiosity using a global line generation approach. In contrast to local lines, global lines are originated irrespective of the patches in the scene. A single global line is not cast from any particular patch, but in a way that can cross all the scene maintaining the cosine distribution for all patches traversed by the line. An interesting feature of global lines is that they are useful to compute visibility relationships between surfaces at different instants of time. The Multi-Frame method consist in the following steps:

- **Pre-processing step:**
  In order to prepare the animated model for the lighting computation the animation is processed storing dynamic object's information for all frames. The aim of this pre-processing step is to obtain an animation model where we can have quickly access to any dynamic information, like the position of an object, at any time.

- **Lighting computation step:**
  Compute the illumination for all frames independently in the same simulation process. Cast global lines and intersect the scene at all instants of time. All objects, static and dynamic at all frame position of the animation must be intersected (figure 1). Intersections obtained for a given global line are stored in two temporary lists: one for the static objects and the other for the dynamic objects (figure 2). Each instance of the dynamic list is labeled with the number of the frame it belongs to. From each global line different visibility lists, each one for each frame, are built. This is done by sorting and merging the static and the dynamic lists of intersections obtained for the corresponding frame (figure 2). Identification of the frames is done using the label number attached to the dynamic items. The visibility lists are used to compute light energy transport between visible pairs of patches. An independent radiosity solution for each frame of the sequence is maintained.

- **Visualization step:**
  For a given camera description the animation can be played by projecting the radiosity solution for each frame. The radiosity solution obtained in the previous step is completely view-independent, thus, any animated camera description can be used to visualize and obtain the final images of the animation.

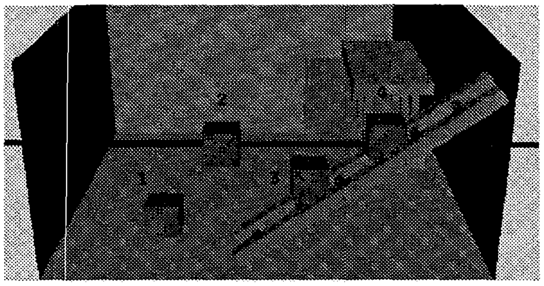

Figure 1. Global lines are cast to intersect static objects and dynamic objects (the cube in this example) at all their instances.

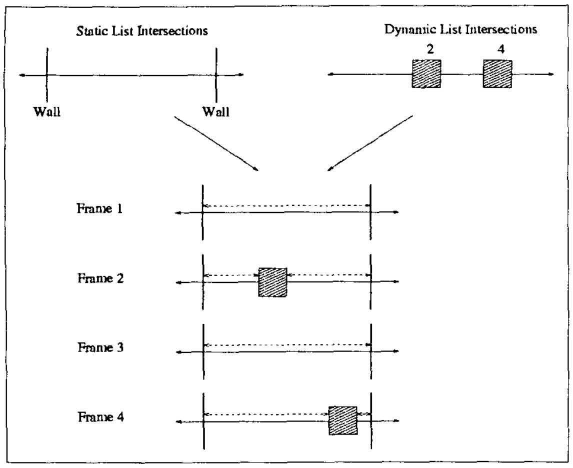

Figure 2. Following the global line example of figure 1 a visibility list for each frame is built by merging and sorting static and dynamic intersections. Light energy exchanges are computed using these lists.

3. Temporal Coherence in Radiosity Animation

Some previous work described in section 2.1 make use of temporal coherence in illumination to compute animated sequences. Nimeroff et al. [12] used it by assuming that indirect illumination varies in general smoothly across the scene. In space, they simplify objects under some refinement criteria and for time they compute the illumination for the animated scene at sparse instants of time. In order to observe the nature of temporal coherence in radiosity sequences we plotted the radiosity versus time for an animated sequence test for different kinds of patches. Figure 3 shows four frames of the sequence tested and figure 4 the curves plots for the selected patches. We chose three patches with different features in the sequence: a static patch that becomes partially in shadow in some interval (top of the red cube in figure 3), a static patch that is never in shadow (front face of the green cube in figure 3) and a patch belonging to a dynamic object (on the wing of the plane). Comparing the graphs we can see a very different behavior of the radiosity variation for each of the chosen patches. For the static patch in shadow important changes happens when the visibility discontinuity of the shadows occurs. For the dynamic patch the radiosity changes continuously, but we can also note that the radiosity values varies in a short range (0.1 to 0.3 approximately). For the other static patch the radiosity variation is smooth in the whole sequence.

We conclude from our observations that temporal radiosity coherence can be used as an improvement source for lighting animation algorithms by estimating the radiosity in intervals where variations are smooth. But we also can note that changes in the radiosity function may happen at different times for each patch, so it may be difficult in some cases to identify unchanged radiosity intervals for all patches.

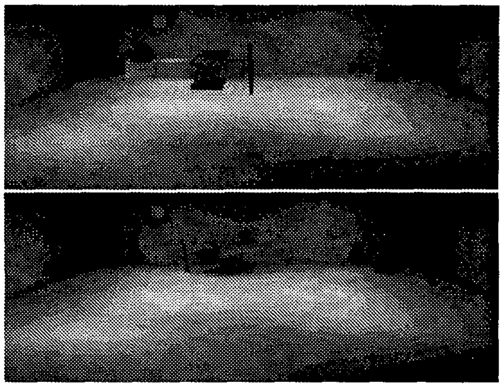

Figure 3. Two frames of a sequence where a plane flies around the room that is used to analyze temporal coherence in radiosity. We chose three different kinds of patches to observe the radiosity over time: a patch on the top of the red cube, a patch on the front side of the green cube and a patch on the top of the wing of the plane.

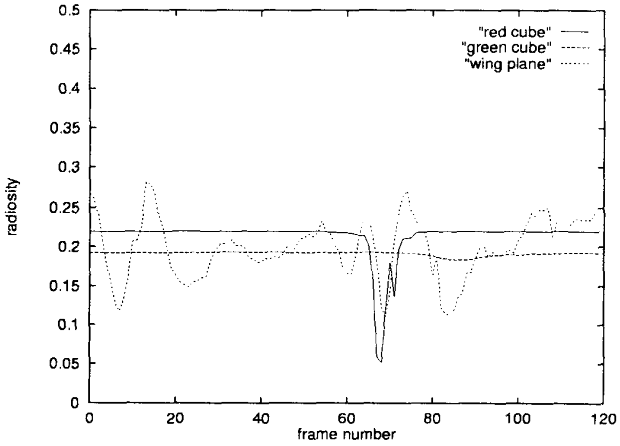

Figure 4. Radiosity vs. time comparison for the selected patches in the sequence of figure 3.

4. Proposal Overview

With the aim of exploiting temporal coherence in illumination as much as possible we propose a two-pass approach. First, the radiosity function over time, for the whole animation, is estimated. From this function we can examine the gradient of the radiosity for all patches determining time points where important radiosity changes happen. These points are named as radiosity key-frames. Then, the radiosity is computed only for those key-frames, and in-between frames are obtained by interpolation. For both the estimation of the radiosity function over time and the computation of the radiosity animation the MFLM is used.

Comparing the above described approach with the MFLM, both processing time and memory requirements are improved. For a given sequence, the cost of processing decreases because a less number of objects needs to be intersected and also less memory storage is required as only the full computed frames are stored.

5. Radiosity Estimation over Time

The key of the method is to identify important changes in the illumination during the animation. The problem is that we do not known a-priori the behavior of this function. In fact, its full computation is our final goal. We propose to perform a coarse and rapid radiosity computation over time in order to obtain a preview of the general behavior of the illumination function. From this estimation we can extract information to exploit coherence.

A coarse radiosity solution, produced in a short time, can be obtained setting a high minimum area threshold for patches subdivision. For this case, using both a hierarchical radiosity approach or a non-hierarchical solution with regular patches subdivision, the scene discretization will result in surfaces subdivided in large patches. A static scene for this kind of solution, in general, could be obtained fast using any radiosity computation technique, but if we have to compute an animated sequence composed by several frames, the processing cost for the estimation would increase enormously. An efficient animated radiosity technique must be used. The MFLM is particularly suitable for such purpose because it can process together, in an efficient way, several frames of a sequence.

5.1. Setting Key-Radiosity-Frames

We compute then, in a first step, the animated radiosity solution discretizing the scene into large patches with the Multi-Frame method. From this solution we must identify important time-points that are going to be set as key-frames for the final computation of the radiosity animation. For this goal, the gradient of the radiosity function over time is analyzed. The radiosity can vary smoothly for many patches of the scene but can have important changes in others (compare the radiosity variation of the red cube and green cube in figure 4). For this reason we decided to use a conservative algorithm for determining the radiosity key-frames: a single patch power variation between two consecutive frames over a defined threshold is enough to set a key-frame at this time point. The pseudo-code of the algorithm is presented at table 1. The algorithm iterates over all patches computing the power difference between a key-frame and successive frames.

```
1. Initialize $k \leftarrow 1$
2. Set key-frames at the beginning and at the end:
   $SetKeyframe(k)$, $SetKeyframe(totalFrames)$
3. $endAll \leftarrow false$
4. While not $endAll$, do
   (a) $actualFr \leftarrow k$
   (b) $followingFr \leftarrow actualFr + 1$
   (c) $end \leftarrow false$
   (d) While not $end$ and $followingFr < totalFrames$, do
       i. For each patch $i$, do
          A. if $|Pow(i, actualFr) - Pow(i, followingFr)| > \varepsilon$,
             then $end \leftarrow$ true
       ii. if not $end$, then $followingFr \leftarrow followingFr + 1$
   (e) if $end$, then
       i. $SetKeyframe(k)$
       ii. $k \leftarrow followingFr$
   else $endAll \leftarrow true$
```

Table 1. Algorithm for determining key-frames from an estimated radiosity animation. Key-frames are stored via the function $SetKeyframe(k)$. The function $Pow(i,k)$ returns the power of a patch i at frame k.

6. Computing the Radiosity Animation

The radiosity animation is finally computed using the MFLM but only for the radiosity key-frames obtained in the previous pass. The method proceeds in the same way as in the original proposition, the only difference is that the frames now are not sequentially but spread through the sequence (see figure 5). Only intersections of objects at key-frames positions are processed.

In-between frames of the animation are obtained by interpolation of full computed ones. Interpolation is performed in object space, that is, for the radiosity at each patch of the scene. We used linear interpolation, but of course, any higher degree interpolation, as for example cubic Hermite interpolation, can be used.

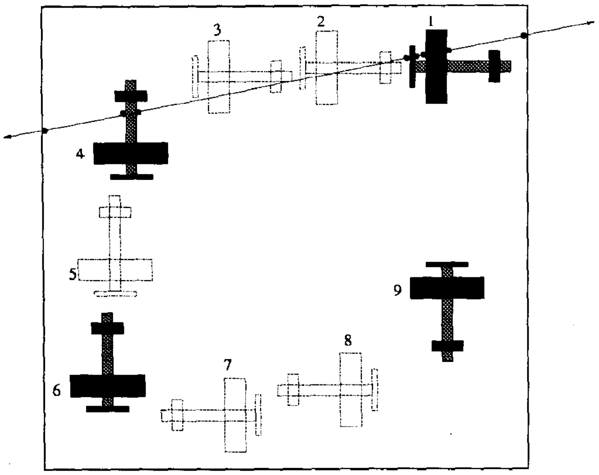

Figure 5. Multi-Frame computation with key-frames. Only objects at key-frames (shown shaded in the draw) are considered for intersection in the method.

7. Analysis

In previous work temporal coherence for radiosity animation was used in different ways by reusing both unchanged visibility relationships or links in hierarchical radiosity approaches. Within the present approach we use coherence in a different way at the final step of the radiosity computation using an estimation of the illumination. If we know that the radiosity is almost constant in an interval we can use directly this information without any visibility test for checking changes.

Although the advantages of using this strategy to improve coherence, some limitations arise because of the use of a coarse solution in the estimation of the illumination. This estimation is done for moderate large patches, thus, a variation in a small patch could not be detected by the estimation and an important key-frame could be skipped. One possibility to overcome this limitation is to increase the threshold area subdivision in the presence of a small dynamic object. But that drives to an increasing in the time processing of the estimation.

To evaluate the speed-up of the acceleration approach we must consider the radiosity estimation pre-processing in the total processing cost. Let us write $C(k,N)$ as the cost of the MFLM for an animated sequence of length k simulated with N lines. Thus, the total cost of a simulation of K frames, where k key-frames were detected, can be expressed as:

$$C_{tot} = C(K,N_{est}) + C(k,N_{full}) \qquad (1)$$

where the subscripts *est* and *full* stand for the estimation processing and for the full computation, respectively. The expression for the total cost per frame of the MFLM was analyzed in [2]:

$$C_{TotPerFr} = N\frac{C_1 \log n_s + C_2 \log k \log n_d + C_3 n_{is} \log n_{is}}{k}$$
$$+ \frac{2A_d}{A} N(C_3 n_{id} \log n_{id} + C_4(n_{is} + n_{id})) \qquad (2)$$

where N is the number of lines cast, n_s and n_d the number of static and dynamic surfaces respectively, k the number of frames, n_{is} and n_{id} the average number of static and dynamic intersections, A the bounded area of the scene and A_d the area of the volume bounding the motion. The total cost per frame does not depend on the discretization factor for the patches subdivision, it depends mainly on the number of frames processed and the number of lines needed to cast in the simulation.

For the estimation processing we process K frames, the total number of frames, which is bigger than k, the total number of key-frames that are processed in the full computation. It seems that the estimation processing could cost more than the full computation, but as we use a greater area threshold for the patches discretization in the estimation processing, less lines are needed for the convergence of the simulation. From our tests (see results in next section) the estimated processing took about one hundred times less than the full computation. Thus, it adds a relatively small increment in the total processing. Mostly, considering that the final number of frames (k) to compute is reduced in a factor between one half to one fourth of the total number of frames (K).

The memory storage is also optimized in comparison to the MFLM. For a given sequence of length K the method stores K independent radiosity solutions with the required power counters. With the improving approach the number of radiosity solutions used for the computation is reduced to only the number of key-frames.

In the final stage of the present approach we reconstruct the in-betweening radiosity frames by interpolation. The error of the approximation must be analyzed. The magnitude of the error depends on the threshold used in the estimation, a high threshold power leads to a low number of key-frames and therefore to a higher error in the reconstruction of the radiosity. When playing the movie this error could be perceptible as artifacts, which in general are more evident in shadows due to direct illumination.

8. Results

We tested the acceleration approach with different animated sequences analyzing time executions and the error.

Executions were performed on a PC-Linux Pentium III at 450 Mhz. The resulting movies can be seen in MPEG-1 format at http://ima.udg.es/~gonzalo/animations/mflm.html.

Power threshold	0.1	0.2	Full Computation
Number of key-frames	75	39	120
Radiosity Estimation time (sec)	30	30	0
Radiosity Computation time (min)	52	34	73
Time per frame (sec)	26	17	35
Speed-up	36	55	26
Relative Speed-up	1.38	2.11	1
Memory required (Mb)	150	80	250
RMSE average	1.3e-4	3.9e-4	0

Table 2. Results with two different approximation threshold and comparison with the full computation of the sequence.

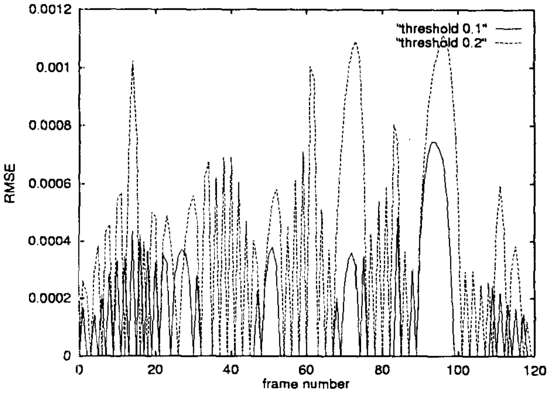

Figure 7. Root Mean Square Error for each frame for the the approximated sequences of table 2. We use a full computed sequence as a reference.

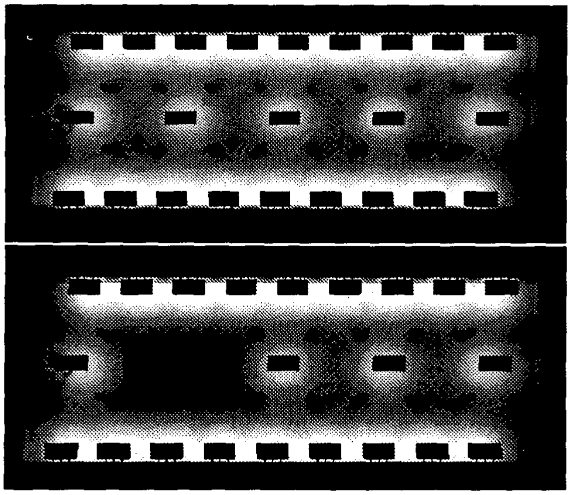

Figure 6. Global view of the garage environment used in our tests. In the second test (down) a light of the middle of the garage has been turned off

For the first test we use a 120-frames sequence of a plane flying inside an illuminated empty garage. Figure 6 shows a top view of the scenario for our tests. We perform the key-frame setting process described in section 5 for two different power thresholds (0.1 and 0.2). Figure 8 shows four frames of both sequences that were computed by interpolation from the selected key-frames of each execution and shows also the full computation without approximation; table 2 shows the results. There, we compare the two approximated sequences with the full computed sequence without approximation. An absolute speed-up of 55 and 36 was obtained with the 0.1 and 0.2 thresholds, respectively. However, some artifacts could be appreciated in the reconstruction for the highest threshold (compare frames in figure 8). With threshold 0.1 an almost inappreciable error is obtained for a relative speed-up of 1.38. We can also note the memory reduction of the technique; a reduction of about 40% and 68% is obtained with the 0.1 and 0.2 threshold approximation, respectively. The RMSE (root mean square error) for each frame, for each approximated sequence, is plotted in figure 7. We use the full computation of the sequence as a reference solution, this is the reason why the error is zero at key-frames. For the second test we turned off one of the lights of the middle row (see figure 6) in the garage of the previous test performing the setting key-frames estimation operation with threshold power 0.1. Results are shown in table 3. Comparing the result with the execution of the first test we can see that less number of key-frames results for the new scenario. This is a consequence that the dynamic object is passing through a dark region of the scene and almost no variation in radiosity is produced in that interval We observe that the presented approach deals well with sequences where changes in geometry do not have significant influence during an interval because invariant radiosity intervals can be identified relatively easily. We compute a final sequence

Power threshold	0.1	Full Computation
Number of key-frames	61	120
Radiosity Estimation time (sec)	30	0
Radiosity Computation time (min)	44	71
Time per frame	22	35
Speed-up	42	26
Relative Speed-up	1.6	1
Memory required approx. (Mb)	125	250
RMSE average	1.7e-4	0

Table 3. Results for the second test. An approximated sequence with threshold 0.1 is compared with the full computation of the Multi-Frame.

in a different environment where there are many visibility changes due to both direct and indirect illumination. Figure 9 show four frames of the sequence computed with the acceleration approach. The environment is composed by 4200 polygons where 500 are dynamic and it is discretized into about 50000 patches. We process this 120-frames sequence with the illumination estimation method described in section 5 with a power threshold of 0.05 obtaining 34 key-frames. An absolute speed-up of 23.3 was obtained for the whole computation of the sequence and a relative 2.5 speed-up to the full computation of the MFLM. The resulting time per frame for the sequence was 50 seconds.

9. Conclusions and Future Work

We presented an acceleration technique for computing animated sequences in radiosity environments. The technique, based on a pre-processing of the animated sequence for determining smooth illumination intervals followed by a spread use of the MFLM, brings a relative speed-up of between about 1.3 and 3 comparing to the direct use of the MFLM and an absolute speed-up of about 40, depending on the parameters of the animation. These results present significant improvements in comparison with previous work. In addition, a reduction (between 40% and 60%) in the memory required is also achieved.

A novel issue of the acceleration technique is the way the illumination function over time is investigated in order to detect almost-invariant illumination intervals. For this purpose we used the Multi-Frame method with a low resolution for the scene discretization to obtain a fast coarse illumination estimation. The Multi-Frame method used as a tool for the estimation of the illumination function over time could be useful also for other radiosity animation techniques. It can be used as a first step to detect illumination invariants in the same way we used, but to provide information for being processed with another animated radiosity method.

Our approach is completely based on Monte Carlo methods. No explicit visibility relationships are needed to be stored in these methods and, as consequence, it can deal better with more complex environment than other deterministic approaches. The amount of memory required is in general an important drawback in most of previous work when dealing with complex environments. Monte Carlo approaches, in addition, guarantee accurate results since provide physically-based solutions.

Future work includes a view-dependent driven analysis of the error due to interpolation. We also believe that the combination of our method with the use of an importance driven radiosity algorithm could benefit our approach for applications like walkthroughs with dynamic objects on large environments.

10. Acknowledgments

This work has been partly supported by project TIC 98-0586-03-02 of the CICYT, and grant 1999SGR-00162 of the DGR-Generalitat de Catalunya.

References

[1] G. Beseuievsky and M. Sbert. The Multi-Frame Lighting Method: A Monte Carlo Based Solution for Radiosity in Dynamic Environments. In X. Pueyo and P. Schröeder, editors, *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 185–194, New York, NY, 1996. Springer-Wien.

[2] G. Besuievsky and X. Pueyo. Animating Radiosity Environments Through the Multi-Frame Lighting Method. *Journal of Visualization and Computer Animation (to appear)*, 2000. Available from http://ima.udg.es/~gonzalo/papers/multiFrame.ps.gz.

[3] S. E. Chen. Incremental Radiosity: An Extension of Progressive Radiosity to an Interactive Image Synthesis System. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24, pages 135–144, August 1990.

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

[5] G. Drettakis and F. X. Sillion. Interactive Update of Global Illumination Using a Line-Space. 1997.

[6] D. A. Forsyth, C. Yang, and K. Teo. Efficient Radiosity in Dynamic Environments. In *Fifth Eurographics Workshop on Rendering*, pages 313–323, Darmstadt, Germany, June 1994.

[7] D. W. George, F. X. Sillion, and D. P. Greenberg. Radiosity Redistribution for Dynamic Environments. *IEEE Computer Graphics and Applications*, 10(4):26–34, July 1990.

[8] 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)*, volume 18, pages 212–222, July 1984.

[9] I. Martín. *Temporal Coherence in Animation of Global Illumination Environments*. Ph.D. thesis, Universitat Politecnica de Catalunya, Barcelona, Spain, 2000.

[10] I. Martín, X. Pueyo, and D. Tost. Frame-to-frame Coherent Animation with two-pass Radiosity. Technical Report 99-08-RR, IIiA-UdG, Jun 1999.

[11] S. Müller and F. Schoeffel. Fast Radiosity Repropagation for Interactive Virtual Environments Using a Shadow-Form-Factor-List. In *Fifth Eurographics Workshop on Rendering*, pages 325–342, Darmstadt, Germany, June 1994.

[12] J. Nimeroff, J. Dorsey, and H. Rushmeier. A Framework for Global Illumination in Animated Environments. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 92–103, New York, NY, 1995. Springer-Verlag.

[13] E. Shaw. Hierarchical radiosity for dynamic environments. *Computer Graphics Forum*, 16(2):107–118, June 1997.

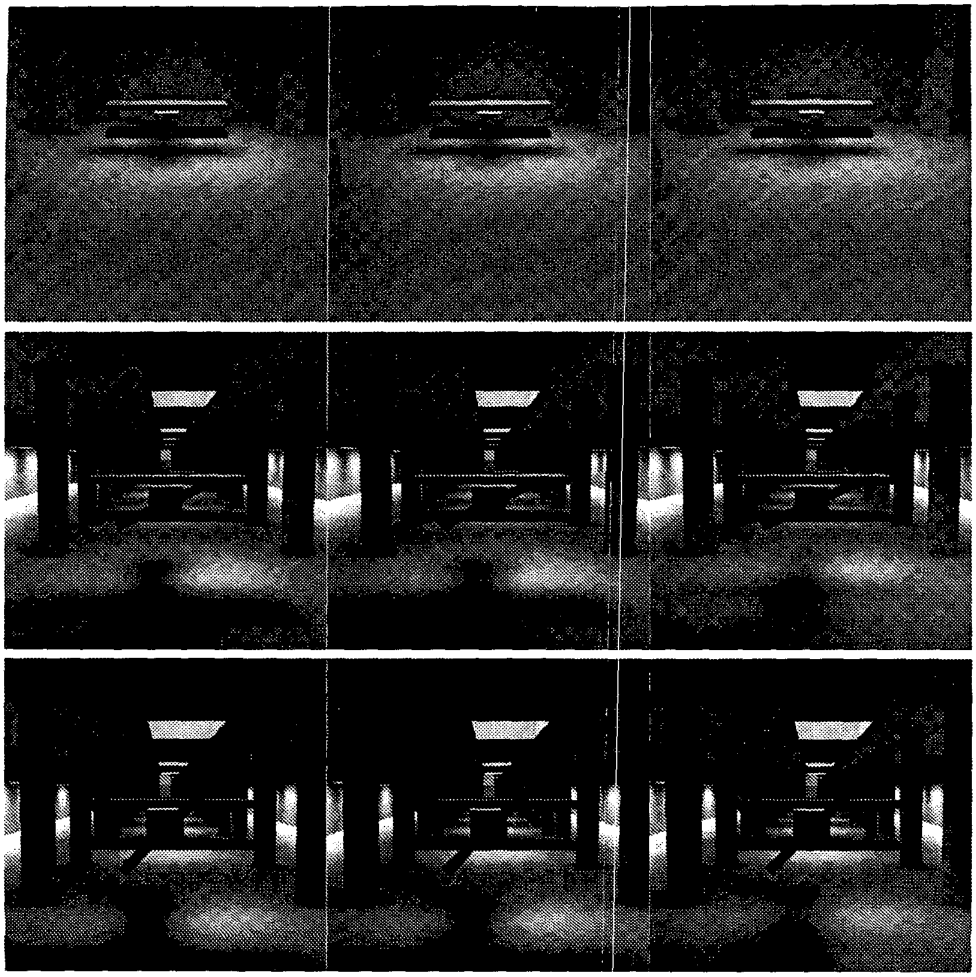

Figure 8. Four frames selected (number 10, 31, 84 and 111) of the first sequence tested. In the left column frames are computed without approximation, in the center column, with key-frames estimated with threshold power 0.1 and in the right with key-frames estimated with threshold power 0.2.

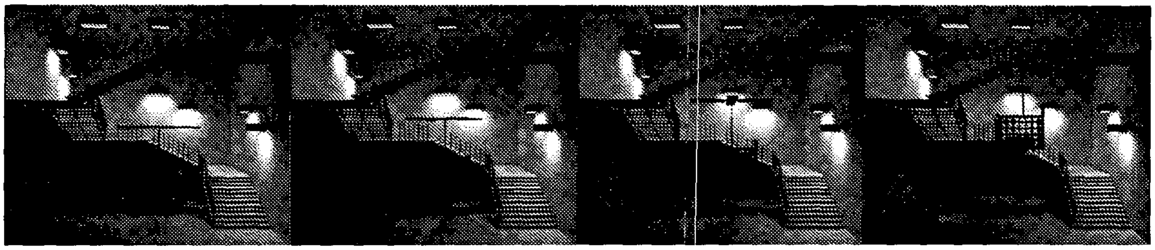

Figure 9. Four frames of a animated radiosity sequence computed using the interpolated approach. The sequence is composed by 120 frames where only 34 are full computed.