A MOTION BLUR METHOD FOR ANIMATED RADIOSITY ENVIRONMENTS

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

ABSTRACT

A method to generate motion blur images for radiosity scenes is presented. The method, based on Global Monte Carlo techniques, is intended to perform an accurate lighting simulation of the effect. Once the illumination is computed a rendering step based on an accumulation buffer approach is used to obtain the motion blur images. Due to the fact that radiosity solutions are view independent, motion blur images can be obtained from different points of view only with this rendering step. Results show that the illumination computed with the method is correct.
The proposed method is well suited for being included in the whole computation of animations in a radiosity environment.
Keywords: Radiosity, Motion Blur, Monte Carlo, Accumulation Buffer, Animation

1 INTRODUCTION

Motion blur images are frequently used in realistic rendering and animations with the goal of increasing realism. When part of a scene is in motion, this effect is produced by capturing the light arrived to a camera, in a finite interval of time. Thus, to keep a realistic simulation of such effect, a physically based lighting method must be improved to compute all light energy interactions between surfaces in the time interval.

Several motion blur algorithms have been proposed for different rendering methods. Distributed ray tracing [CPC84] presents an image based solution, from stochastic point sampling techniques. Although the method can improve a global illumination simulation of the effect it is not an efficient method for scenes containing diffuse surfaces like in radiosity environments. Other methods are based on post-processing approaches [PC83, Max90]. They blur a previously rendered image of an object in the direction of that object's motion on the screen. No global illumination is considered in all instants of time. In accumulation buffer methods, [HA90], hardware graphics z-buffer polygon render is applied at each sampled instant and the resulting images combined. To achieve a global illumination simulation of blurring, each image at each instant time, must be computed using a global illumination algorithm. As these algorithms are in general expensive, if we need many time samples, the processing of the whole solution would be dramatically increased.

In this paper we propose a Monte Carlo method for computing motion blur effects in radiosity scenes in an efficient way. The illumination of all surfaces, both static and dynamic, is guaranteed to converge to the radiosity values accumulated in the time interval of the motion. Once the lighting simulation is performed, we use an accumulation buffer technique to render the motion blur image. We can obtain motion blur images for different points of view, by only processing this rendering pass, once radiosity solutions are view independent. Our approach is based on the Multi-Frame Lighting Method [BS96], a radiosity proposition for animated environments, where the main feature is that many frames of an animation are processed at once within a single simulation, in an efficient way. The method we are proposing for motion blur can also be included in the multi-frame lighting method in order to obtain motion blur effects in animated radiosity scenes.

In section 2 we review the basis of the multi-frame lighting method and in section 3 we formulate the illumination problem for motion blur in radiosity. In section 4 our approach is presented. Results obtained are shown in section 6 and conclusions in section 7.

2 REVIEW OF THE MULTI-FRAME LIGHTING METHOD (MFLM)

This method efficiently computes radiosity solutions for animated environments. It is based on Global Monte Carlo techniques [Sbe97] which have been proposed as an alternative to Monte Carlo radiosity methods. While classic Monte Carlo cast rays from surfaces, Global Monte Carlo cast lines that are independent from the surfaces position. These lines are called global lines and can be built by taking two random points in a bounding sphere of the whole scene. The fact that these global lines remain valid to compute light energy if surfaces are moved, was used to develop the MFLM.

For a given scene and a certain animation description, the method acts in two steps. In a preprocessing step, all geometric and light information of each frame is merged into a unique structure description of a single environment (Fig. 1.a). Each dynamic object then has as many instances as frames of the animation, whereas static objects have only one. In a second step the lighting simulation is performed. Global lines are cast to compute the light energy exchange between visible surfaces in each frame (Fig. 1.b). To achieve this step we use the multi-path algorithm [SPNP96].

3 MOTION BLUR IN RADIOSITY

Motion blur effect is caused when a picture is taken from a finite time exposure camera, in a scene where part of the objects are in motion. If we are interested in a realistic rendering of such effect, we must concern with the global illumination problem. To compute the total amount of light arriving to a camera in a finite time interval, we must integrate the rendering equation [Kaj86], over the whole time interval. If we consider the case of radiosity environments, where all surfaces are assumed to be purely diffuse and using the classical radiosity equation [CW93], the radiosity of a patch i in an interval Δt can be expressed as:

$$B_i^{\Delta t} = \frac{1}{\Delta t} \int_{\Delta t} (E_i(t) + \rho_i(t) \sum_{j=1}^{N} B_j(t) F_{ij}(t)) dt \tag{1}$$

where all the patches emission (E), surfaces reflectance (ρ) and form factors (F_{ij}) may vary over time. Considering discrete functions in time, this integral can be approximated by an average sum, taking samples over time:

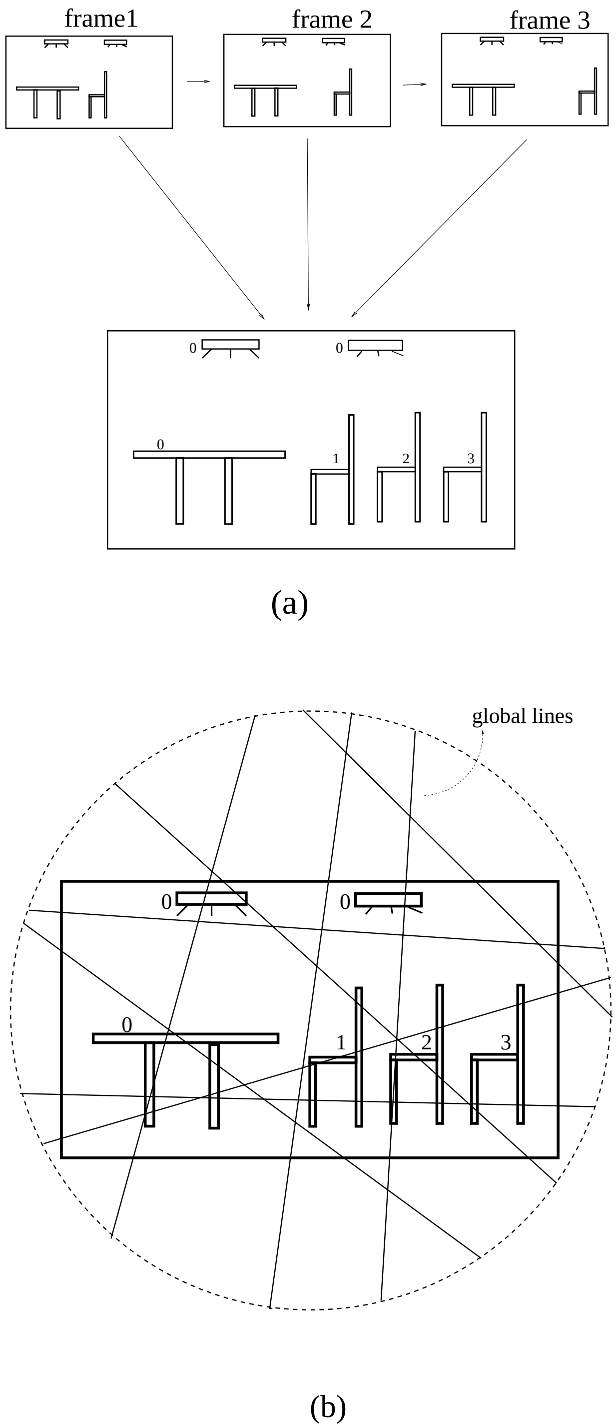

Figure 1: (a) First Step of the multi-frame lighting method: all frames are merged into a single structure description of the environment. (b) The lighting computation is performed with a global Monte Carlo method.

$$B_i^{\Delta t} = \frac{1}{M}\sum_{k=1}^{M}(E_i^k + \rho_i^k \sum_{j=1}^{N} B_j^k F_{ij}^k) \qquad (2)$$

where M is the number of time samples being taken. To achieve an accurate approximation of integral 1, several samples should been taken. Computing all complete radiosity solutions, one for each sample, would be computationally expensive. Since the changes in geometry and illumination from consecutive samples are smooth, algorithms that improve coherence can be used to save processing. Approaches like the Incremental radiosity method [Che90, GSG90] could be adapted for this purpose. The most difficult part of approaches that exploit coherence in this way is to identify the form factors that have been affected as a result of the change in visibility between surfaces.

4 A MONTE CARLO RADIOSITY METHOD FOR MOTION BLUR

Stochastic methods were successfully used in previous works on motion blur. By casting rays distributed in random samples over time, Cook et. al [CPC84] developed a general method to compute not only motion blur but other blurring effects, such as depth of field and glossy reflections.

Our approach is based on Global Monte Carlo techniques for radiosity and uses the same principles of the MFLM described in section 2: The same global lines are used to compute exchanges of light energy between surfaces in many different instants of time.

The method splits the motion blur image generation into a lighting simulation step and a rendering step. The rest of this section describes both steps.

4.1 Lighting Simulation Step

For this step, which is the main contribution of this paper, we present two algorithms that solve equation 2 in different ways: a straightforward solution and a more efficient approach. The goal of this first step is to obtain the illumination for all surfaces for the time interval the camera shutter is open.

4.1.1 Multi-frame Average Algorithm

A straightforward solution for computing equation 2 is to average all radiosity solutions, each one for each frame, obtained with the MFLM. Within this strategy we guarantee that both static and dynamic patches converge to the radiosity solution since we are explicitly computing equation 2. Although all solutions for each frame are computed this is done in an efficient way within a single process. The complexity of adding more time sampling to the simulation is studied in [BS96].

4.1.2 Global Line Average Algorithm

In the algorithm previously described we need to store all radiosities, each one for each sample time, for all patches of the scene. We introduce here a better algorithm, in terms of memory storage, that computes integral 1 in a different way. The algorithm also uses global lines for lighting computation. The exchange of light energy between surfaces in all single global lines cast is averaged over time. When a global line intersects the moving part of the scene (Fig. 2.a), we construct a list of intersections of patches for all time samples of the motion objects (Fig. 2.b). We use this list to compute the amount of exchange of light energy between patches in the time interval. This operation requires the following steps: First we separate the list by samples obtaining one different list for each time sample, then in each resultant list we sort intersections by distances to build lists of visible pairs of patches, and finally we compute the light exchange between this visible patches (Fig. 2.c). To compute the illumination we also use the multi-path algorithm [SPNP96].

To improve the intersection operation of the algorithm we use a motion volume bounding box structure. This volume cover the space where the motion takes place (see Fig. 2.a). Lines that do not intersect the volume are used in the same way as in a static scene.

The idea behind the global line average algorithm is that equation 1, the integral over time, is computed for each patch, integrating in time for all single directions. The main difference between this approach and the multi-frame average is that in the global line average we store only the radiosity solution of one scene, whereas in the multi-frame average we must store the radiosity solution for each sample time, and this could be a drawback if we deal with large scenes.

4.2 Rendering Step

The rendering and final step is performed using an accumulation buffer technique. The computed scenes obtained, are projected and combined at each instant time using a z-buffer hardware graphics.

Although an accurate global illumination solution is required to achieve realistic results, most of the visual impact of the blurring effect is derived from the quality of the image render. The number of samples must be enough to avoid strobing artifacts. The appropriate number of samples needed depends on the size of the view projection of the moving object onto the image plane.

Noting the importance of taking several samples to produce good results we propose an approxi-

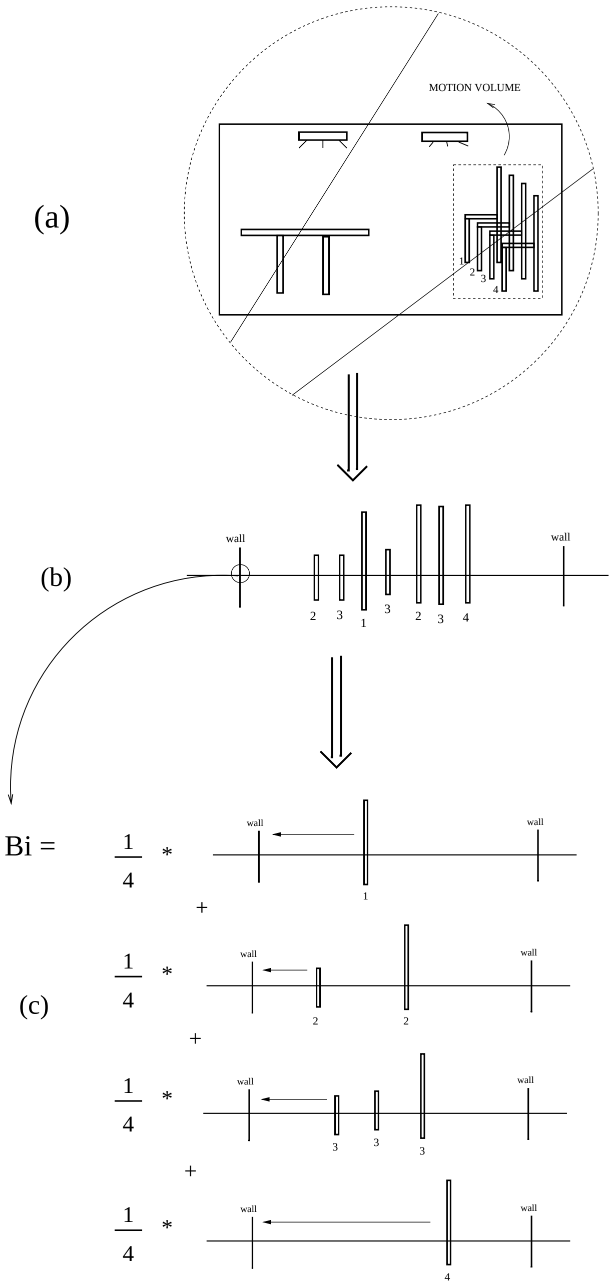

Figure 2: Global line average algorithm. A global line intersects moving objects at all instants time (a) and generate a list (b). This list is separated by samples resulting in visibility lists (c). The radiosity of a patch i is computed as the average of all the energy arrived to it in all instants.

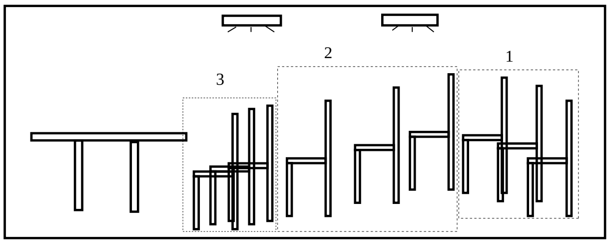

Figure 3: Motion blur in animation. The scene is structured in groups of samples for each frame, then the global line average algorithm is applied to compute independent radiosity solutions for each frame.

mated approach by taking more samples in the rendering step, than in the lighting step. In this way, we do not increase the processing of the first step, which could be affected with the increase in the number of instance objects to intersect, and we improve the rendering step. This approximation is valid since the illumination function we are computing has a smooth variance over time, and therefore we do not need many time samples to compute it.

5 INCLUDING MOTION BLUR IN ANIMATION

The motion blur method described in the previous section can be included in a natural way into the MFLM in order to produce realistic animations. Given an animation description to be displayed in k frames, where we can obtain the position of dynamic objects at any time, we first decide to take a set of samples (subframes) for each frame. These subframes are used to compute the illumination over time in the interval of a frame. Second, we perform the lighting calculations for each frame independently using the global line average algorithm described in section 4. Each frame has a radiosity solution based on the samples being taken for that frame.

Fig. 3 shows an example with three frames and three subframes per frame of how samples are grouped to obtain the animated scene structured.

For each frame we only store a radiosity solution that represents the illumination of all surfaces in the time interval the camera shutter is open. Again, using an accumulation buffer based rendering technique, we render each frame for the viewing position of the camera and store the resulting motion blur image to compile the final animation. As described in section 4 more samples should be taken in the rendering step to get best results.

Another possibility for improving an animation with blurring effects is to apply the strategy described in [HA90]. If we are generating a motion blur image by accumulating N projections for one frame of the animation, we can obtain the next

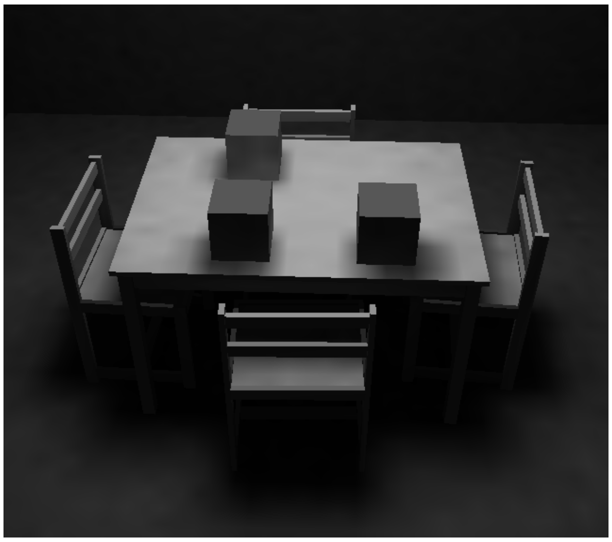

Figure 4: Radiosity scene used to test the method, with all objects static.

frame by drawing and subtracting the subframe that was rendered $N-1$ projections ago, and then add the projection of the next subframe. In this way we obtain a new frame with only two rendering operations. The radiosity solutions can be computed by both algorithm presented in section 4, the multi-frame average and the global line average.

6 RESULTS

In this section we present results of the proposed method. Fig. 4 shows the scene which was used to test our method and Fig. 5 shows different views of the same scene with two blurred objects. For the lighting step, using the global line average algorithm, takes about 20 minutes on a SGI Indigo2 workstation, for four time samples for the moving objects. For the rendering step, 32 accumulated projections were used, which take less than half a minute of time processing on the same computer. Typical details of radiosity scenes, as soft shadows and color bleeding can be observed in the surface of the table. Color images of this results are located at http://ima.udg.es/~gonzalo/images.html.

6.1 Convergence of the Method

In order to study the convergence of the global line algorithm (see section 4.1.2) used in the lighting step, we compute the mean square error (MSE) using a reference solution obtained with the multi-frame average algorithm (see section 4.1.1). We follow this strategy because we know that the multi-frame average algorithm converges to the correct solution, since it explicitly computes equation 2. In Fig. 6 we plot the MSE versus execution

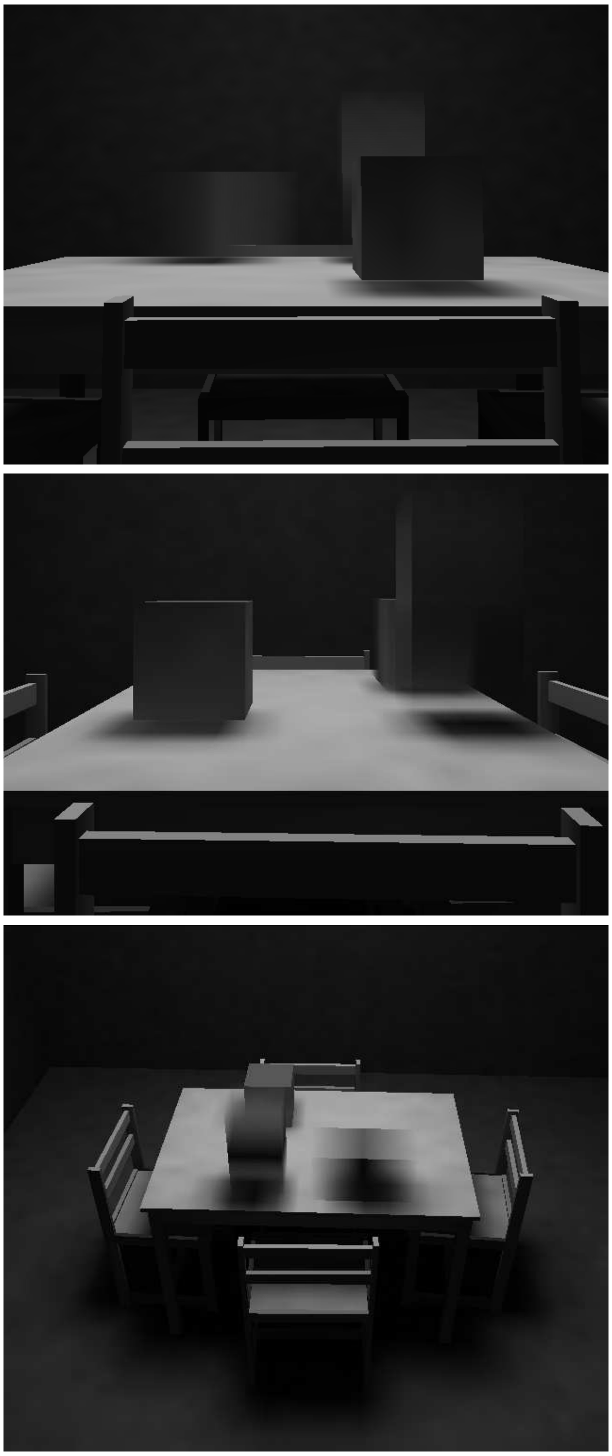

Figure 5: Motion blur images from different view positions. All of them were rendered with 32 accumulated projections.

time with a growing number of lines in the scene shown in Fig. 5. Only the surface of the table has been taken to compute the error because it has more significant changes in the illumination. The graph shows that the algorithm converge to the correct solution.

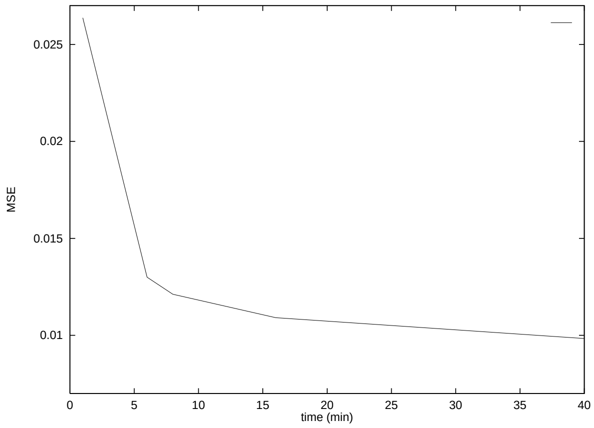

Figure 6: Error of the global line average algorithm used in the lighting step. MSE vs. time (in minutes).

7 CONCLUSIONS

We have presented a method for computing motion blur images in radiosity environments. With the goal of producing accurate solutions, the integral illumination equation for such effect is established and computed using a global Monte Carlo approach. Two different algorithms that integrate the equation over time are proposed: the multi-frame average and the global line average. While the first explicitly averages radiosities over time, the second performs the same simulation but optimizing the amount of memory needed. Once the illumination is computed, motion blur images from different view position can be rendered using an accumulation buffer approach. Because of the importance of the image quality, more samples are used in the rendering step.

This motion blur method can also be included into the MFLM in order to produce realistic animations in radiosity environments.

8 ACKNOWLEDGMENT

The authors wish to thank Mateu Sbert and Ignacio Martín for helpful comments, and Jorge Diz and Leo Pini Magalhães for discussing early ideas of this work. This project has been funded in part with CICYT grant number TIC 95-614-C03-03 and Austrian-Spanish joint action number HU 96-37.

REFERENCES

[BS96] Gonzalo Besuievsky and Mateu Sbert. The Multi-Frame Lighting Method: A Monte Carlo Based Solution for Radiosity in Dynamic Environments. *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 185–194, 1996.

[Che90] Shenchang Eric Chen. Incremental Radiosity: An Extension of Progressive Radiosity to an Interactive Image Synthesis System. *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, 24(4):135–144, August 1990.

[CPC84] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed Ray Tracing. *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, 18(3):137–145, July 1984.

[CW93] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press Professional, Boston, MA, 1993.

[GSG90] David W. George, Francois X. Sillion, and Donald P. Greenberg. Radiosity Redistribution for Dynamic Environments. *IEEE Computer Graphics and Applications*, 10(4):26–34, July 1990.

[HA90] P. Haeberky and K. Akeley. The accumultaion buffer. *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, 23(3):309–318, August 1990.

[Kaj86] James T. Kajiya. The Rendering Equation. *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, 20(4):143–150, August 1986.

[Max90] Nelson Max. Polygon based post-process motion blur. *The Visual Computer*, 6(2):308–314, 1990.

[PC83] M. Potmesil and I. Chakravarty. Modelling motion blur in computed-generated images. *Computer Graphics (ACM SIGGRAPH '83 Proceedings)*, 17(3):389–399, August 1983.

[Sbe97] Mateu Sbert. *The Use of Global Random Directions to Compute Radiosity: Global Monte Carlo Techniques.* PhD thesis, Universitat Politecnica de Catalunya, Barcelona, Spain, 1997.

[SPNP96] Mateu Sbert, Xavier Pueyo, Lazlo Neumann, and Werner Purgathofer.

Global Multipath Monte Carlo Algorithms for Radiosity. *The Visual Computer*, 12(2):47–61, 1996.