

# Reuse of Paths in Light Source Animation

Mateu Sbert,

Francesc Castro

Institut d'Informàtica i Aplicacions. Universitat de Girona

mateu—castro@ima.udg.es

John Halton

University of North Carolina at Chapel Hill

halton@cs.unc.edu

## Abstract

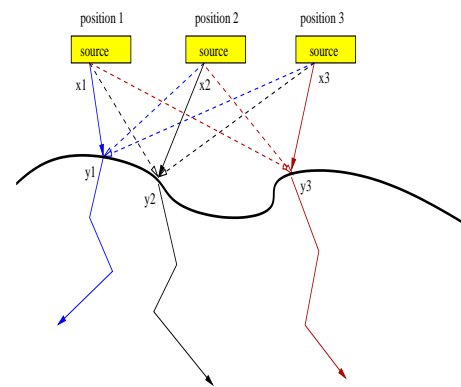
*In this paper we extend the reuse of paths to the shot from a moving light source. In the classical algorithm new paths have to be cast from each new position of a light source. We show that we can reuse all paths for all positions, obtaining in this way a theoretical maximum speed-up equal to the average length of the shooting path.*

## 1. Introduction

Shooting random walks are used in rendering, and particularly in radiosity, to simulate the distribution of power from the light sources [1]. The main drawback of these Monte Carlo techniques [5] is the high number of paths needed to obtain an acceptable result. The quadratic error obtained is known to be proportional to  $\frac{1}{N}$ ,  $N$  being the number of paths. On the other hand, the cost of a path depends on its length, namely the number of hits on the surfaces of the scene.

We propose in this paper a strategy for accelerating the computation of frames in light source animation, where each frame corresponds to one light source position. This strategy is based on the reuse of shooting paths, and it will be applied in the context of Radiosity. Paths leaving a light source position will be reused from the rest of positions (see Fig. 1), resulting in a notable reduction of the cost needed to obtain an acceptable result for each of the frames. The gain will depend on the number of frames, the average length of the paths, and the positions of the light source.

This paper is organized as follows. In next section we will refer to previous work. The description of our method is in section 3. Section 4 presents the results, with both error graphs and images. Finally, in last section we present the conclusions and future work.



**Figure 1.** 2D representation of a Shooting Random Walk with reuse of paths. Path from light source in position 1 is reused to distribute power from positions 2 and 3 of the light source, and the same with paths from positions 2 and 3.

## 2. Previous work

The cost of a random walk simulation is mainly the cost of computing the next hit point in the walk, that is, the point visible from the old hit point in the sampled direction. Many walks have to be cast to obtain a noiseless image. Thus achieving some sort of path reusing can reduce the computational cost. This is still more dramatic in an animation computation, due to the high number of frames to be computed.

The bidirectional path-tracing technique [8, 9] can be considered as a first attempt to reduce the cost by reusing the paths. They join sub-paths from the same pixel or from the same source point. The idea of the reuse of full paths and for different states (i.e., pixels, patches or light sources) was first presented by John Halton in [4], in the context of the random walk solution of an equation system. The idea was applied by Bekaert et al. to the context of path-

tracing in [2], combined with multiple importance sampling [9] to avoid biasness. Pixels were grouped in tiles and paths from one pixel in the tile were reused for the other pixels in the tile from the second hit point of the path. A speed-up of one order of magnitude was reported for fairly complex scenes. Havran et al. [6] present the reuse of paths in a walk-through, this is, when the observer changes position. Paths cast from one observer position are reused for other neighbor positions. Although obtaining a high speed-up, the method remained biased as the samples were not weighted with the respective probability. In the Radiosity context Besuievsky [3] used the same set of lines to expand direct illumination from different light source positions. The source positions were packed in a bounding box and lines crossing this box expanded the power of all intersected positions. The drawback of this method is that lines are wasted if the source positions are not tightly packed. Moreover it is only valid for diffuse sources.

### 3. Reuse of shooting paths in light source animation

The main idea of the new algorithm is based on the reuse of paths: a path exiting from a point of the light source can be used to distribute power from every light source position, or in other words, to calculate the illumination of every frame. It means that each path will be used to build  $n$  paths,  $n$  being the number of light source positions (see Fig. 1).

#### 3.1. The new estimator

Suppose we have a source with power  $\Phi$  taking  $n$  positions (1.. $n$ ). We take  $N$  points in each position, which by coherence reasons should be the same points. Let  $x$  be a point on the light source repeated in each different position  $x_1, x_2, \dots, x_n$ . Let  $j$  be the position taken into account. The power to be distributed from  $x_j$  to the scene  $S$  is given by

$$\int_S F(x_j, y) \frac{\Phi}{N} dy = \frac{\Phi}{N} \quad (1)$$

where  $F(x_j, y)$  is the form factor from point  $x_j$  to a point  $y$  (we consider  $y$  being any point to receive direct illumination from  $x_j$ ). Following Veach [9] we define a mixture probability density function  $p(y)$ ,

$$p(y) = \frac{F(x_1, y) + F(x_2, y) + \dots + F(x_n, y)}{n} \quad (2)$$

Taking  $n$  samples from  $p(y)$  corresponds to generating point  $y$  in  $n$  different ways, using probabilities  $F(x_1, y), F(x_2, y), \dots, F(x_n, y)$ . The unbiased estimator for the power arriving to point  $y$  from  $x_j$  will be then

$$\frac{F(x_j, y)}{p(y)} \times \frac{\Phi}{N \times n} \quad (3)$$

as there are  $N$  points and from each point there would start  $n$  paths, this is,  $N \times n$  samples. Substituting  $p(y)$ ,

$$\begin{aligned} & \frac{F(x_j, y)}{\frac{F(x_1, y) + F(x_2, y) + \dots + F(x_n, y)}{n}} \times \frac{\Phi}{N \times n} \\ = & \frac{F(x_j, y)}{F(x_1, y) + F(x_2, y) + \dots + F(x_n, y)} \times \frac{\Phi}{N} \quad (4) \end{aligned}$$

#### 3.2. Expected cost and speed up factor

Let us assume to be a constant  $K$  the cost of generating a ray from a point in a surface and finding the nearest intersection. Let  $\phi$  be the total power of the source. Let  $n$  be the number of positions of the source, and  $N$  the number of rays to cast from each position.

The computational cost of a path will be  $K$  multiplied by the length of the path. So,  $K \times l$  is the average cost of a path, where  $l$  is the average length of a path.

In our algorithm, each path is used  $n$  times, involving an additional cost corresponding to the computation of  $n - 1$  point-to-point form factors. A point-to-point form factor is given by the formula

$$F(x, y) = \frac{\cos \theta_x \cos \theta_y V(x, y)}{\pi r^2} \quad (5)$$

where  $\theta_x, \theta_y$  are the angles between the line that joins both points and the respective normals,  $r$  is the distance between both points and  $V(x, y)$  is the visibility function between  $x$  and  $y$  (0 or 1). Most of the cost of computing the form factors corresponds to this visibility evaluation. This cost is assumed to be similar to  $K$  and, for the sake of simplicity, we take  $K$  as this cost. Thus, the cost of a path in the new algorithm is given by

$$lK + (n - 1)K = (l + n - 1)K \quad (6)$$

Since we use  $N$  paths for each of the  $n$  positions, the total cost is given by (7). Due to the reuse of paths, this corresponds to a total of  $N \times n$  paths per light source position.

$$(l + n - 1)KNn \quad (7)$$

The cost of computing these  $N \times n$  paths per position using the classical random walk algorithm is  $lKNn \times n = lKNn^2$ . Dividing this value by (7) we have the ratio

$$\frac{lKNn^2}{(l + n - 1)KNn} = \frac{ln}{l + n - 1} \quad (8)$$

which is the expected acceleration factor using the new algorithm (to obtain the same number of paths per position). When the number of light source positions (frames) grows, we have

$$\lim_{n \rightarrow \infty} \frac{ln}{l + n - 1} = l \quad (9)$$

that is, the theoretical acceleration factor is bounded by  $l$ , the average length of the paths. Observe that  $l$  can be computed as  $\frac{1}{1-\bar{\rho}}$ ,  $\bar{\rho}$  being the mean reflectance of the scene.

In the best case, using the same number of paths per position in both classical and new approach will produce the same error. This corresponds to the limiting case in which all the light source positions were the same. In general, the more distant the positions, the bigger the error. This is due to the fact that when positions get distant,  $p(y)$  goes away from the importance sampling function corresponding to each source position  $j$ .

This means that the new algorithm presents a theoretical speed-up factor bounded by the average length of paths  $l$  (corresponding to the theoretical acceleration factor bound). The speed-up factor approaches this value the more the positions are closer.

#### 4. Implementation and results

The idea presented in this paper can be applied to any shooting random walk algorithm (see section 1). In this case we have implemented a shooting random walk collision algorithm [1], in which we decide at each photon hit if the photon either dies or survives, taking the reflectance of the hit surface as the surviving probability. Another feature of this algorithm is that we take the hit point as the next exiting point of the photon, that is, we only have to sample the exiting point on the light source.

The new algorithm implies some additional storage: for each patch, the accumulated power for each frame must be stored. However, no storage for the paths is required, due to the immediate update of power for each position when dealing with a path (note that form factor is computed when dealing with each path).

In section 3.2 we have assumed the cost of a visibility computation to be equal to the cost of finding the nearest intersection. However, we have to note that the computation of the nearest intersection has a higher cost than the visibility computation between  $x_i$  and  $y$ , since in this last case it is enough to make a search for an intersected polygon closer to  $x_i$  than  $y$ . This fact has been considered in our implementation.

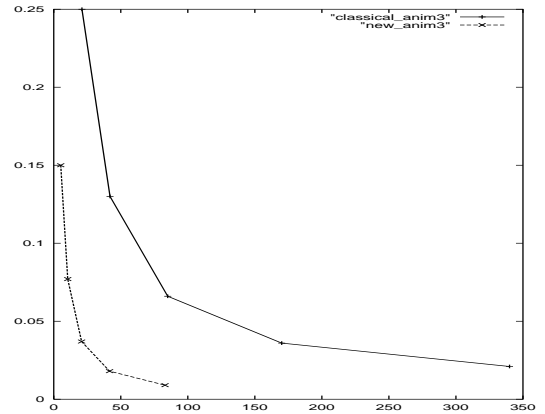
All the executions presented in this paper have been done on a Pentium IV at 1.6 Ghz.

The test scene is discretized in 11311 patches. We have considered an animation involving 30 very close and even

overlapping light source positions<sup>1</sup>. Using (8), and considering an average path length of 5, the expected acceleration factor is 150/34 (about 4.4). Our results show an acceleration factor of about 7.3, larger than the expected value. This is due to the acceleration in the visibility computations previously indicated.

We also note that the Mean Square Error (MSE) using the new method is lower for frames placed in the middle of the frame sequence than for frames in the extremes. This behavior can be theoretically justified by the fact that central positions have an importance sampling function closer to the pdf we have used to obtain the  $n$  samples (2) than the extreme positions.

The speed up factor obtained in the central frames is very remarkable, since even surpasses the theoretically expected one (that is the average length of the paths, see section 3.2), as can be seen in the time vs. MSE graph corresponding to the frame 17 (see Fig. 2). This fact is due to the small increase in MSE produced by the reuse of paths together with the high acceleration factor.



**Figure 2.** Graph of time in seconds (horizontal axis) vs. MSE (vertical axis). This graph corresponds to frame 17. Note the clear advantage of new method in front of the classical one. Speed up about 6.6.

#### 5. Conclusions and future work

We have presented in this paper an application of the reuse of paths in shooting random walk for the case of moving light sources. The expected acceleration factor is bounded by the average length of the path when the number of frames tends to infinity, see (9). Also, the more close the source positions, the higher the speed up for all frames. This

<sup>1</sup>Available at [ima.udg.es/~castro/animCGI.html](http://ima.udg.es/~castro/animCGI.html), referred to as Animation n.2

is due to the closeness of pdf (2) to the importance sampling function corresponding to the source positions when these positions are close, as explained in section 3.2. Again, as the importance sampling function corresponding to central positions is more similar to the used pdf (2) than the one corresponding to extreme positions, the more central the position of the light source in relation with the rest, the higher the speed up. Also, the coherence between consecutive frames in the animations can notably improve the quality of videos by reducing the aliasing between consecutive frames.

The idea presented in this paper is not limited to radiosity. It can be applied to different techniques in global illumination, like bidirectional path-tracing [8, 9], virtual light sources [7], distribution of photons in photon maps [10], etc., being extended in this way to non-diffuse global illumination environments.

## 6. Acknowledgments

This project has been funded in part with a Catalan-Flemish Joint Action n. ACI2002-52 from the Catalan Government, with grant number TIC2001-2416-C03-01 from the Spanish Government, and with grant number 2001/SGR/00296 from the Catalan Government.

## References

- [1] P. Bekaert. Hierarchical and stochastic algorithms for radiosity. *Ph.D. thesis. Catholic Univ. of Leuven*, 1999.
- [2] P. Bekaert, M. Sbert, and J. Halton. Accelerating path tracing by re-using paths. *Proceedings of Workshop on Rendering*, pages 125–134, 2002.
- [3] G. Besuievsky. A monte carlo approach for animated radiosity environments. *Ph.D. thesis. Universitat Politècnica de Catalunya, Barcelona*, 2001.
- [4] J. Halton. Sequential monte carlo techniques for the solution of linear systems. *Journal of Scientific Computing*, 9(2), pages 213–257, 1994.
- [5] J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods*. Methuen and Co. Ltd., 1975.
- [6] V. Havran, C. Domez, K. Myszkowsky, and H.P. Seidel. An efficient spatio-temporal architecture for animation rendering. *Rendering techniques '03*, pages 106–117, 2003.
- [7] A. Keller. Instant radiosity. *Computer Graphics Proceedings, Siggraph'97*, pages 49–56, 1997.

- [8] E.P. Lafortune and Y.D. Willems. Bi-directional path tracing. *Compugraphics Conference Proceedings*, pages 145–153, 1993.
- [9] E. Veach. Robust monte carlo methods for light transport simulation. *Ph.D. thesis. Stanford University*, 1997.
- [10] H. Wann Jensen. Global illumination using photon maps. *Rendering Techniques*, pages 21–30, 1996.



**Figure 3.** Frame n.17. (top) Classical algorithm. Time=340 sec. N.paths= 1920000. (bottom) New algorithm. Time= 41.4 sec. N.paths = 1920000 (including reuse). The error is practically the same in both cases, resulting in a speed up about 6.6.