Reducing the Number of Shadow Rays in Bidirectional Path Tracing

Eric P. Lafortune   Yves D. Willems

Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
Eric.Lafortune@cs.kuleuven.ac.be

Abstract

Bidirectional path tracing is a fairly new Monte Carlo algorithm for physically based rendering. Introduced as a generalisation of path tracing, the algorithm traces paths for each pixel, not only from the eye point but also from the light sources. It subsequently links the intersection points on the respective paths by means of shadow rays. Each shadow ray determines if a lighting contribution has to be added to the estimate of the radiance of the pixel.

In this paper we present a technique which reduces the number of shadow rays that are traced effectively. The resulting slight increase in the variance of the stochastic process can be undone by taking more samples. Practical tests show that the approach yields a reduction of the variance for the same amount of work.

1 Introduction

Physically based rendering consists of simulating the transfer of light throughout a given scene, in order to compute a realistically looking image of that scene. The input data are the geometry and optical properties of the surfaces of the model and some viewing parameters. The well-known ray tracing algorithm by Whitted [1] was an early attempt at rendering scenes with some basic lighting effects such as specular reflection and direct diffuse illumination. Cook *et al.* [2] extended the algorithm by tracing several randomly reflected rays at each intersection point. Kajiya introduced path tracing [3], a variant of stochastic ray tracing which only traces a single reflected ray at each intersection point. He was the first to give stochastic ray tracing a physical basis in the shape of the rendering equation. This integral equation and its more recently presented dual and generalised counterparts describe the global illumination problem in a mathematically precise way. They therefore allow known stochastic techniques and optimisations to be applied [4, 5].

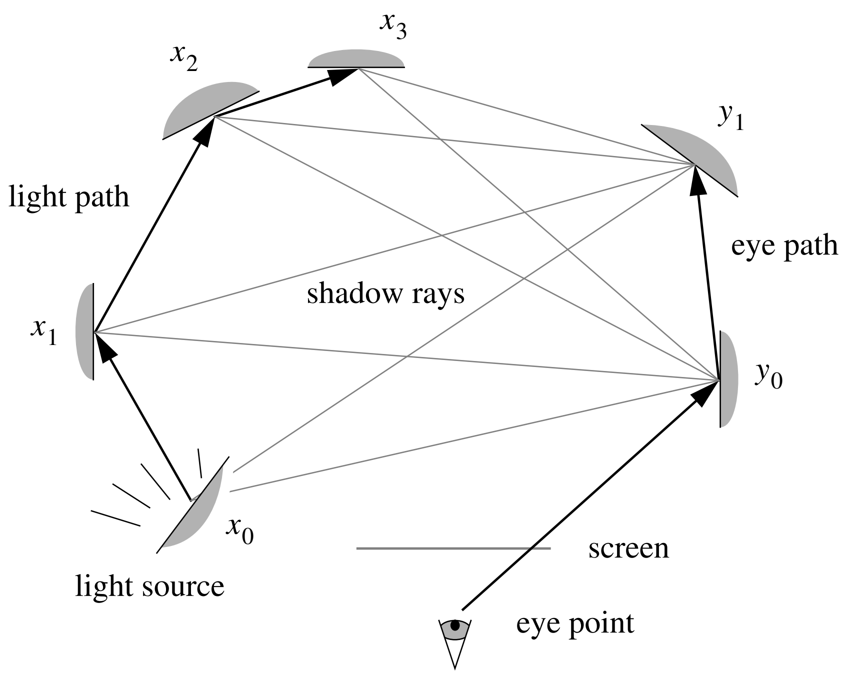

Figure 1: A schematic representation of the bidirectional path tracing algorithm.

The main problem with Monte Carlo rendering algorithms – and Monte Carlo techniques in general – is their slow convergence. Due to the stochastic nature of the algorithms their results are subject to some variance. In stochastic ray tracing this variance is typically visible as uncorrelated noise in the image. The essential approach of Monte Carlo algorithms to this problem is to take a large amount of samples and to average the results. The reduction of the variance can easily be shown to be proportional to the number of samples. The standard deviation therefore only decreases proportionally to the square root of the number of samples. Obtaining accurate results for any but the simplest problems can be computationally extremely expensive as a result. Most research on Monte Carlo rendering is devoted to reducing the variance using optimised sampling strategies. Common sampling strategies such as stratified sampling, importance sampling, Russian roulette and control variates have been successfully applied to the rendering equation [6, 7, 8, 9, 10].

Lafortune and Willems [11, 12] and Veach and Guibas [13] have recently presented a further generalisation of path tracing which also traces random paths from the light sources. This bidirectional path tracing yields a lower variance for scenes which are indirectly illuminated. Interior scenes lighted indirectly by spot lights are a typical example.

Bidirectional path tracing has the property that more shadow rays are traced than in classical path tracing. This paper presents a technique which reduces the number of shadow rays that are traced effectively. The first section briefly outlines the principles of the original algorithm. The next section presents the optimisation in some alternative guises. Some results then demonstrate the kind of improvements that may be expected in practice.

2 Bidirectional path tracing

Classical path tracing is a stochastic ray tracing algorithm which computes a radiance value (from which a colour can be derived) for each of the pixels in the image. It starts by tracing a primary ray from the eye point through the pixel that is being considered. The origin and direction of the ray are sampled on the basis of a filter which may also account for depth of field and motion blur for instance. At the first intersection point one or more shadow rays are cast to the light sources to compute a direct lighting contribution. The contribution is only added if the shadow rays do not hit an intervening object. A stochastic test known as Russian roulette usually determines if the ray is reflected or absorbed. If it is reflected a single reflection direction is sampled based on the reflective properties of the surface. The process of finding the closest intersection point, computing the direct illumination and sampling a reflection is then repeated recursively. The recursion ends if a ray is absorbed. As the primary estimate obtained from this single random walk still has a large variance a more accurate secondary estimate is computed by averaging the results of a large amount of random walks.

If the scene to be rendered is illuminated indirectly only few shadow rays may reach a light source, resulting in a large variance for the primary and secondary estimators. Bidirectional path tracing tries to alleviate this problem by not only sampling a random walk from the eye (an eye path), but also from a light source (a light path). For this purpose a starting point and direction are sampled on the light sources on the basis of their intensities. Bright light sources and directions have a higher probability of being selected than weaker light sources. The random walk then continues in exactly the same fashion as for the eye path. After both random walks have been constructed all intersection points on the light path (including the starting point) are connected to all intersection points on the eye path (usually excluding the eye point) by means of shadow rays. The shadow rays again determine the lighting contributions which have to be added to the estimate. Figure 1 gives a schematic overview of the algorithm. Once more a more accurate secondary estimate is obtained by averaging several primary estimates.

For a complete overview of the algorithm and the exact expressions for the lighting contributions we refer to [11, 12]. Suffice to say that if we call the points on the light path $x_0, x_1, \ldots, x_{N_l-1}$ and the point on the eye path $y_0, y_1, \ldots, y_{N_e-1}$ we can write the primary estimate for the radiance as:

$$\langle L \rangle = \sum_{i=0}^{N_l-1} \sum_{j=0}^{N_e-1} C_{ij} v_{ij}$$

where:

- C_{ij} is the lighting contribution which results from connecting point x_i on the light path to point y_j on the eye path, without taking visibility into account,
- v_{ij} is 0 or 1 depending on whether the shadow ray between x_i and y_j hits an object in the scene or not.

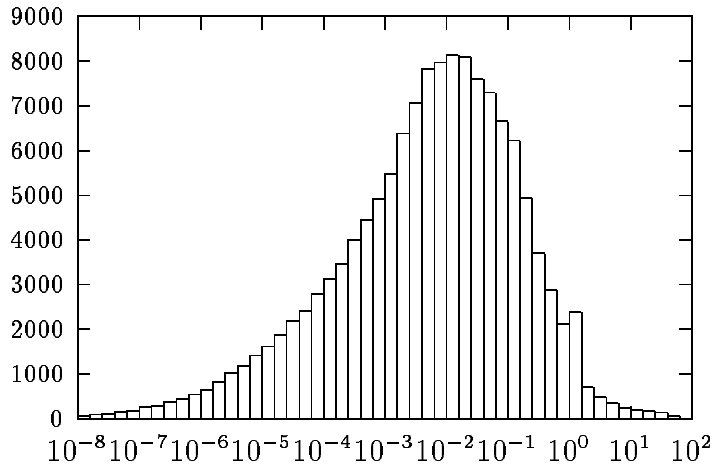

Figure 2: Distribution of all the lighting contributions computed while rendering one of the test images.

3 Optimisation

Shirley *et al.* have extensively investigated optimisations of the the path tracing algorithm which only trace a single shadow ray from each point on the eye path [14, 15, 16]. This means that N_e shadow rays have to be traced. N_e is a stochastic variable on its own. If the absorption with Russian roulette is based on the reflectivities of the surfaces then N_e can be estimated by summing all possible path lengths multiplied by the probability for these path lengths:

$$E(N_e) = \sum_{j=1}^{\infty} j\,\rho_{av}^{j-1}(1-\rho_{av}) = \frac{1}{1-\rho_{av}}$$

where ρ_{av} is the average reflectivity of the surfaces in the scene ($0 < \rho_{av} < 1$).

The bidirectional path tracing algorithm on the other hand traces $N_l \times N_e$ shadow rays. The expected number of shadow rays then becomes:

$$E(N_l \times N_e) = E(N_l) \times E(N_e) = (1 + \frac{1}{1-\rho_{av}})\frac{1}{1-\rho_{av}} = \frac{2-\rho_{av}}{(1-\rho_{av})^2}$$

There will be a significantly larger number of shadow rays for scenes with a high average reflectivity. Another observation in practice is that the range of lighting contributions corresponding to these shadow rays is very large. Figure 2 shows the distribution of the contributions for a test scene which is also used later on. Note that the ordinate axis has a logarithmic scale. Due to the range spanning several orders of magnitude the larger contributions can easily overwhelm the smaller contributions. Yet the basic algorithm puts an equal amount of work into each of them. We therefore propose three stochastic procedures which only effectively trace shadow rays selected on the basis of the importance of their contributions (Fig. 3):

- One can group the lighting contributions per point x_i on the light path and trace only a single shadow ray for each group. The shadow ray is selected stochastically. It can be easily shown that the expected value for the estimator remains identical if the probabilities for selecting the shadow rays are proportional to the contributions:

$$\langle L \rangle = \sum_{i=0}^{N_l-1} (\sum_{j=0}^{N_e-1} C_{ij}) v_{il_i}$$

  where $P(l_i = l) = \frac{C_{il}}{\sum_{j=0}^{N_e-1} C_{ij}}$ for $0 \leq l < N_e$.

- One can group the lighting contributions per point y_j on the eye path and trace only a single shadow ray for each group. Similarly to the previous approach this grouping yields:

$$\langle L \rangle = \sum_{j=0}^{N_e-1} (\sum_{i=0}^{N_l-1} C_{ij}) v_{k_j j}$$

  where $P(k_j = k) = \frac{C_{kj}}{\sum_{i=0}^{N_l-1} C_{ij}}$ for $0 \leq k < N_l$.

- One can group all lighting contributions and trace only a single shadow ray which should be representative. Again similarly this yields:

$$\langle L \rangle = (\sum_{i=0}^{N_l-1} \sum_{j=0}^{N_e-1} C_{ij}) v_{kl}$$

  where $P(k, l) = \frac{C_{kl}}{\sum_{i=0}^{N_l-1} \sum_{j=0}^{N_e-1} C_{ij}}$ for $0 \leq k < N_l$ and $0 \leq l < N_e$.

Of course ignoring the bulk of the shadow rays can be expected to increase the variance. But as less shadow rays are traced more primary samples can be taken for globally the same amount of work. Test results in the next section endorse this hypothesis.

4 Results

We have tested the reduction of shadow rays on the two scenes which are shown in Fig. 4. Table 1 presents the results. Images rendered by the basic algorithm and by the three alternative optimisations were compared against reference images which were computed at high accuracy. The numbers of samples per pixel were chosen so as to obtain approximately the same total numbers of rays, since the total number of rays can be considered as a basic unit for the amount of work. The RMS errors of the pixels were normalised to get 1 for the basic algorithm. The results show that the RMS error decreases by 5 to 20% with better results for the latter optimisations. In practice the computation times did increase

Optimisation 1:

$$\begin{array}{ll} \boxed{C_{00} \quad C_{01} \quad \dots \quad C_{0,N_e}} & \rightarrow \; v_{0,l_0} \\ \boxed{C_{10} \quad C_{11} \quad \dots \quad C_{1,N_e}} & \rightarrow \; v_{1,l_1} \\ \vdots & \quad\;\; \vdots \\ \boxed{C_{N_l,0} \quad C_{N_l,1} \quad \dots \quad C_{N_l,N_e}} & \rightarrow \; v_{N_l,l_{N_l}} \end{array}$$

Optimisation 2:

$$\begin{array}{cccc} \boxed{\begin{array}{l} C_{00} \\ C_{10} \\ \vdots \\ C_{N_l,0} \end{array}} & \boxed{\begin{array}{l} C_{01} \\ C_{11} \\ \vdots \\ C_{N_l,1} \end{array}} & \dots & \boxed{\begin{array}{l} C_{0,N_e} \\ C_{1,N_e} \\ \vdots \\ C_{N_l,N_e} \end{array}} \\ \downarrow & \downarrow & & \downarrow \\ v_{k_0,0} & v_{k_1,1} & \dots & v_{k_{N_e},N_e} \end{array}$$

Optimisation 3:

$$\boxed{\begin{array}{cccc} C_{00} & C_{01} & \dots & C_{0,N_e} \\ C_{10} & C_{11} & \dots & C_{1,N_e} \\ \vdots & \vdots & & \vdots \\ C_{N_l,0} & C_{N_l,1} & \dots & C_{N_l,N_e} \end{array}} \rightarrow v_{kl}$$

Figure 3: Three alternative optimisations grouping the lighting contributions in different ways: per point x_i on the light path (1), per point y_j on the eye path (2), and all grouped together (3). A single representative shadow ray is traced per group to compute a visibility factor v_{kl}.

Scene	Optimisation	Number of samples per pixel	Total number of rays	Normalised RMS error
A	none	10	925.942	1.
	1	14	946.131	0.92
	2	15	917.904	0.88
	3	18	932.344	0.80
B	none	10	992.630	1.
	1	14	966.236	0.95
	2	15	945.364	0.94
	3	18	966.746	0.92

Table 1: Overview of the test results.

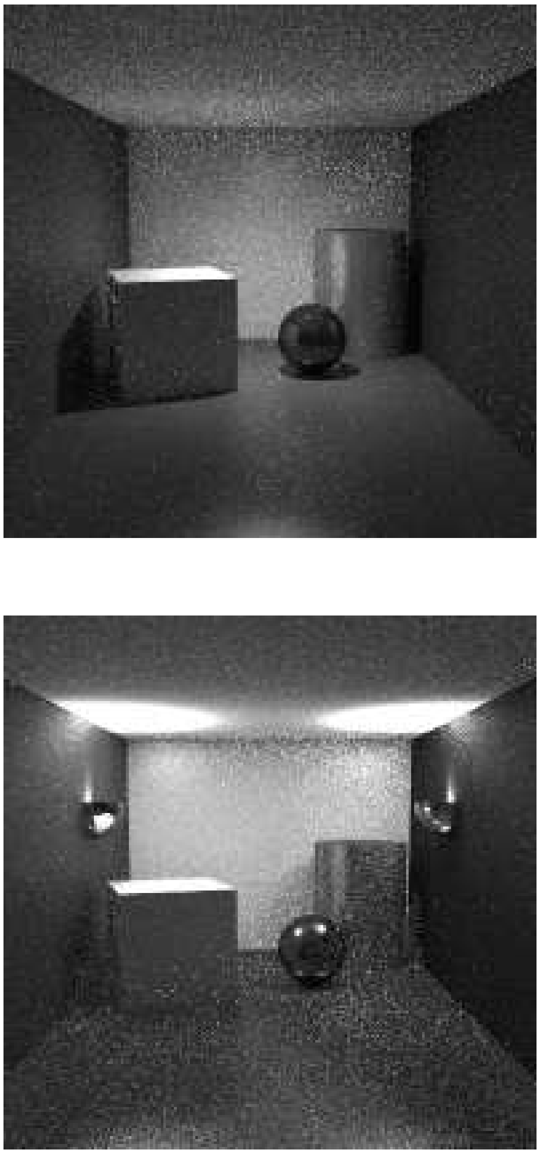

Figure 4: Test scenes A and B, respectively directly and indirectly illuminated.

though. This effect can be attributed to the overhead which is important compared to the time spent on shadow ray intersection tests, for small scenes like these.

For complex scenes the benefit of the reduction of shadow rays will be more important. Even more so if the scene contains different rooms or environments, each with its own lighting, because a single shadow ray may then be very well representative of its whole group of neighbouring shadow rays.

5 Conclusions

In this paper we have presented an optimisation for the bidirectional path tracing algorithm. The large number of shadow rays that has to be traced is reduced by stochastically selecting the most important ones. The technique can be proven not to introduce a bias in the result. The computational effort that is saved can be put into computing more primary samples. The net effect is a modest reduction of the variance. For small scenes the overhead of the computations may be significant compared to the time spent on intersection tests. The technique is therefore best-suited for complex scenes.

6 Acknowledgements

The first author would like to acknowledge the financial support by a grant from the Belgian "Instituut tot Aanmoediging van Wetenschappelijk Onderzoek in Nijverheid en Landbouw" (I.W.O.N.L. #910184).

References

[1] T. Whitted, "An improved illumination model for shaded display," *Communications of the ACM*, vol. 23, no. 6, 1980.

[2] R. Cook, T. Porter, and L. Carpenter, "Distributed ray tracing," *Computer Graphics*, vol. 18, no. 3, pp. 137–145, 1984.

[3] J. Kajiya, "The rendering equation," *Computer Graphics*, vol. 20, no. 4, pp. 143–150, 1986.

[4] M. Kalos and P. Whitlock, *Monte Carlo Methods.* Wiley & Sons, 1986.

[5] J. Hammersly and D. Handscomb, *Monte Carlo Methods.* Chapman and Hall, 1964.

[6] M. Lee, R. Redner, and S. Uselton, "Statistically optimized sampling for distributed ray tracing," *Computer Graphics*, vol. 19, no. 3, pp. 61–67, 1985.

[7] P. Shirley, *Physically Based Lighting Calculations for Computer Graphics.* PhD thesis, University of Illinois, Nov. 1990.

[8] P. Shirley and C. Wang, "Direct lighting by monte carlo integration," in *Proceedings of the Second Eurographics Workshop on Rendering*, (Barcelona, Spain), May 1991.

[9] B. Lange, "The simulation of radiant light transfer with stochastic ray-tracing," in *Proceedings of the Second Eurographics Workshop on Rendering*, (Barcelona, Spain), May 1991.

[10] E. Lafortune and Y. Willems, "The ambient term as a variance reducing technique for monte carlo ray tracing," in *Proceedings of the Fifth Eurographics Workshop on Rendering*, (Darmstadt, Germany), pp. 163–171, June 1994.

[11] E. Lafortune and Y. Willems, "Bi-directional path tracing," in *Proceedings of CompuGraphics*, (Alvor, Portugal), pp. 145–153, Dec. 1993.

[12] E. Lafortune and Y. Willems, "A theoretical framework for physically based rendering," *Computer Graphics Forum*, vol. 13, pp. 97–107, June 1994.

[13] E. Veach and L. Guibas, "Bidirectional estimators for light transport," in *Proceedings of the Fifth Eurographics Workshop on Rendering*, (Darmstadt, Germany), pp. 147–162, June 1994.

[14] P. Shirley, "A ray tracing method for illumination calculation in diffuse-specular scenes," in *Proceedings of Graphics Interface '90*, (Halifax, Nova Scotia), May 1990.

[15] P. Shirley and C. Wang, "Distribution ray tracing: Theory and practice," in *Proceedings of the Third Eurographics Workshop on Rendering*, (Bristol, UK), pp. 33–43, May 1992.

[16] P. Shirley, C. Wang, and K. Zimmerman, "Monte carlo techniques for direct lighting calculations," *ACM Transactions on Graphics*, to appear.