Direct Lighting Calculation by Monte Carlo Integration

Proceedings of the Second Eurographics Workshop on Rendering, June, 1991

Peter Shirley Changyaw Wang

Department of Computer Science
Indiana University
Bloomington, IN 47405 USA
shirley@cs.indiana.edu
wangc@cs.indiana.edu

Abstract

The details of doing a Monte Carlo direct lighting calculation are presented. For direct lighting from multiple luminaires, a method of sending one shadow ray per viewing ray is presented, and it is argued that this is preferable for scenes with many luminaires. Some issues of the design of probability densities on unions of luminaire surfaces are discussed.

1 Introduction

Many rendering algorithms separately calculate direct and indirect lighting for visible surfaces. In early rendering algorithms, the direct lighting calculation was carried out on point luminaires, and the indirect lighting was approximated by an "ambient" term [3, 10]. This idea was later used by Whitted, but shadowing was added using visibility rays [15]. Cook extended this idea to area luminaires by using Monte Carlo integration [1]. Since then many researchers have extended Cook's method to include adaptive sampling [8, 2, 11, 5, 9].

One problem with Cook's method is that a direct lighting calculation generates a shadow ray for every luminaire, which is excessive for some environments. Kajiya pointed out that it would be better to send some number of rays to luminaires chosen probabilistically, though he did not propose a specific selection method [5]. Shirley implemented a method where one shadow ray was generated regardless of the number of luminaires, but did not supply many analytical details [12, 13]. In this paper, we derive an unbiased estimator for direct lighting using Monte Carlo integration. The method builds on work from [12], but will work for arbitrary collections of area luminaires.

Expand this paragraph to include discussion of number of dimensions and sampling density. The method will work best when many samples are taken in each pixel. If few samples are to be taken, culling methods [14] would undoubtably work better.

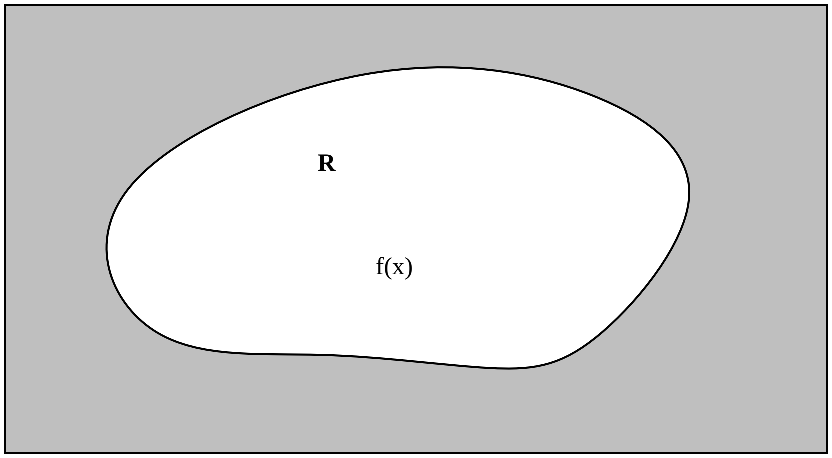

Figure 1: A function **f** defined over a region **R**.

2 Direct Lighting for One Luminaire

Suppose we want to calculate the direct lighting component at a point x viewed from direction ψ. This quantity will be a *spectral radiance* [6], and can be written as a function of wavelength, λ: $L(\mathrm{x}, \psi, \lambda)$. Given a luminaire, S, the direct lighting resulting from S can be written:

$$L(\mathrm{x}, \psi, \lambda) = \int_{\mathrm{x}' \in S} g(\mathrm{x}, \mathrm{x}') \rho(\mathrm{x}, \psi, \psi', \lambda) L_e(\mathrm{x}', \psi', \lambda) \cos\theta \frac{dA' \cos\theta'}{\|\mathrm{x}' - \mathrm{x}\|^2} \quad (1)$$

where $g(\mathrm{x}, \mathrm{x}')$ is the *geometry term*, which is zero if there is an obstruction between x and x', and one otherwise [5]; $\rho(\mathrm{x}, \psi, \psi', \lambda)$ is the BRDF [6]; ψ' is the direction from x' to x; θ is the angle between ψ' and the surface normal at x; θ' is the angle between ψ' and the surface normal at x'; dA' is the differential area of x'.

Any integral over a region **R** (see Figure 1) can be approximated using Monte Carlo methods [4]:

$$\int_{x' \in \mathbf{R}} f(x') d\mu(x') \approx \frac{f(x)}{p(x)} \quad (2)$$

where the point x is a random variable with probability density p. For this formula to be valid, p must be positive where f is nonzero. Equation 2 gives a *primary estimator* which might have a high variance. A lower variance *secondary estimator* can be generated by averaging several primary estimators (each with a different x).

Equation 2 can be applied to the direct lighting integral (Equation 1) if we have a density p with which to sample, and a method to choose x_i with density p on the surface of the luminaire:

$$L(\mathrm{x}, \psi, \lambda) = g(\mathrm{x}, \mathrm{x}') \rho(\mathrm{x}, \psi, \psi', \lambda) L_e(\mathrm{x}', \psi', \lambda) \cos\theta \frac{\cos\theta'}{p(\mathrm{x}') \|\mathrm{x}' - \mathrm{x}\|^2} \quad (3)$$

Once a random point x' has been chosen on S, evaluating this expression is straightforward except for the geometry term $g(\mathrm{x}, \mathrm{x}')$, where a visibility ray must be sent.

An important issue is the selection of p on the luminaire. At a minimum, p must be a valid

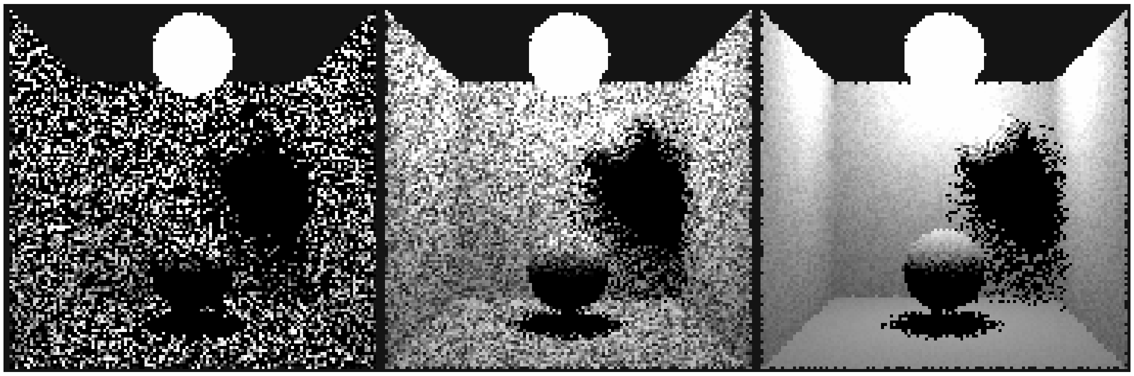

Figure 2: Spheres.

probability density on S, and p must be non-zero for all points on S that are both visible to $\mathbf{x}$ and have non-zero L_e.

If S is a finite polygon, choosing p(x) is straightforward. We can use a simple uniform density ($p = 1/A$) function defined on the polygon surface. If the polygon is large, then we might be better making p drop with the square of the distance from $\mathbf{x}$ (essentially integrating with respect to the solid angle).

rewrite slightly and include a figure?. In our implementation, we break a polygon into triangles. Since it's very difficult to compute the exact solid angle of a triangle, we evenly sample points on an imaginary triangle, which covers the same solid angle as the triangle light source and whose vertexes have distance 1 to x. So, this imaginary triangle is very close to the solid angle and the uniform samples on the imaginary triangle yields a better approximation, to uniform samples via solid angle, than the simple uniform sample scheme. The improvement is more obvious if x is at a grazing angle to a trianlge light source.

If S is a sphere, making p a uniform density would yield an unbiased estimator, but that estimator would have an unnecessarily large variance. This high variance arises because there is at least a one half chance that we will pick a sample point not facing $\mathbf{x}$. The closer the object is to a spherical luminaire, the larger the chance that a point invisible to $\mathbf{x}$ will be chosen. A better way is to uniformly select a sample point from the part of the sphere that is visible to $\mathbf{x}$. Better still, we could select points uniformly with respect to the solid angle as seen from $\mathbf{x}$. *rewrite and reference figs 2 and 3.*

3 Direct Lighting for Multiple Luminaires

If there are N luminaires S_1 to S_N, then the direct lighting will be given by the same integral as Equation 1, but the domain of integration will have to be extended to the union of the areas of every luminaire. Assuming we can construct a probability density function that covers all luminaires, then we can use a Monte Carlo estimator for direct light, and thus use only one or a few shadow rays.

We usually view the direct lighting as I, a sum of N integrals, where each integral represents the radiance contribution from a single luminaire. In an abstraction of this problem we have N integrals over N domains $\mathbf{R}_1$ through $\mathbf{R}_N$ (see Figure 4):

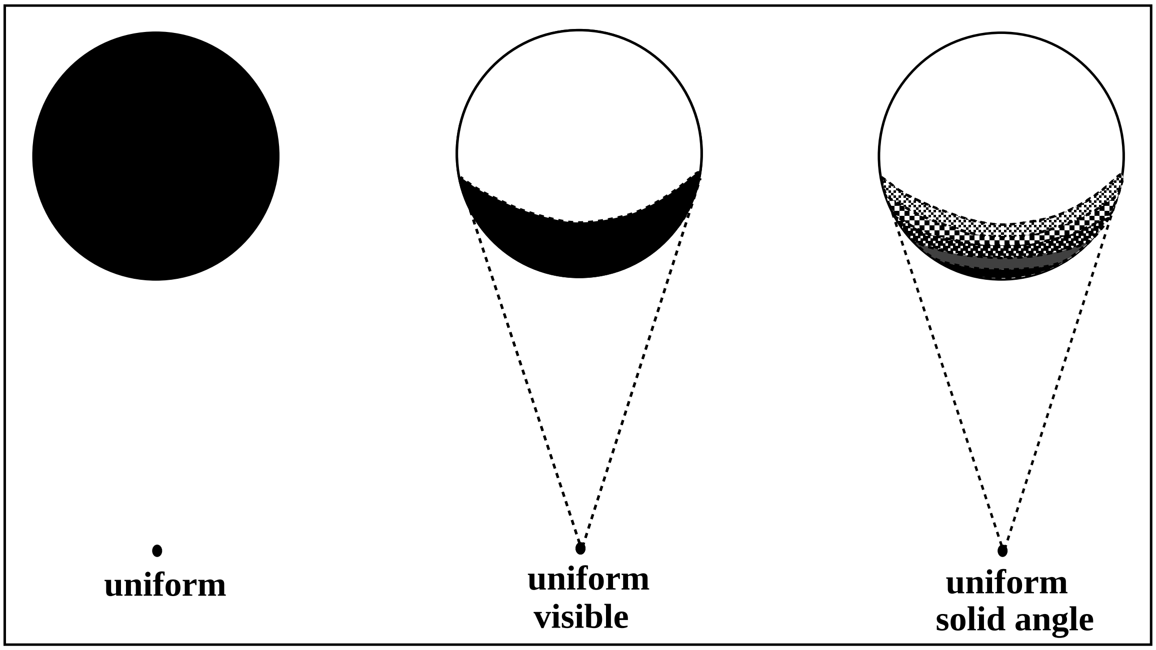

Figure 3: Spheres.

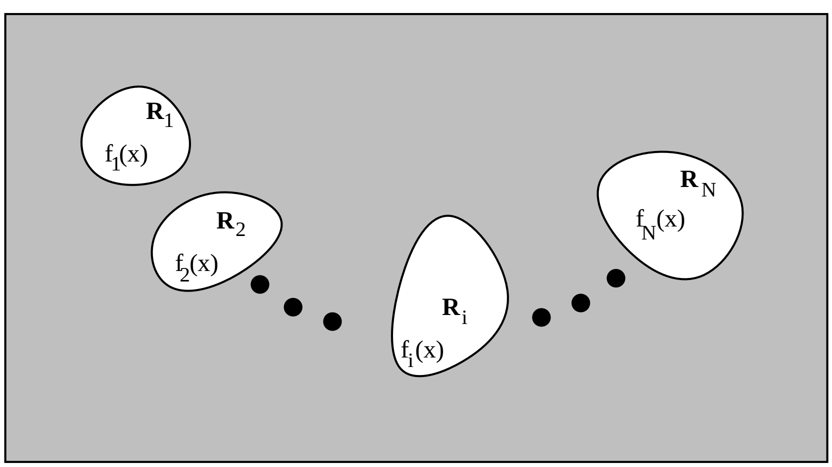

Figure 4: Several functions **f** defined over several disjoint regions **R**.

$$I = I_1 + I_2 + \cdots + I_N \tag{4}$$

where each integral I_i is defined by:

$$I_i = \int_{x' \in \mathbf{R}_i} f(x') d\mu(x') \approx \frac{f_i(x)}{p_i(x)} \tag{5}$$

where $p_i(x)$ is a probability density on $\mathbf{R}_i$

This can be extended using N separate Monte Carlo integrations:

$$I \approx \frac{f_1(x)}{p_1(x)} + \frac{f_2(x)}{p_2(x)} + \cdots \frac{f_N(x)}{p_N(x)} \tag{6}$$

This corresponds to sending N shadow rays to N luminaires. Instead, we can define a region $\mathbf{R}$ to be the union of all $\mathbf{R}_i$, and define a function $f(x)$ to be whatever f_i is appropriate for the point being evaluated. This comes from the simple observation that Equation 2 can be applied even if $\mathbf{R}$ has holes or is not fully connected. We have a estimator as soon as we can develop a valid density function p on $\mathbf{R}$. An easy way to do this is to combine the known p_i:

$$p(x) = \begin{cases} \alpha_1 p_1(x) & \text{if } x \in \mathbf{R}_1 \\ \alpha_2 p_2(x) & \text{if } x \in \mathbf{R}_2 \\ \vdots & \vdots \\ \alpha_N p_N(x) & \text{if } x \in \mathbf{R}_N \end{cases} \tag{7}$$

where the α_i sum to one, and where each α_i is non-zero if I_i is non-zero. We can define f in a similar manner:

$$f(x) = \begin{cases} f_1(x) & \text{if } x \in \mathbf{R}_1 \\ f_2(x) & \text{if } x \in \mathbf{R}_2 \\ \vdots & \vdots \\ f_N(x) & \text{if } x \in \mathbf{R}_N \end{cases} \tag{8}$$

We can use the same types of p_i for luminaires as used in the last section. The question remaining is what to use for α_i. We could just make $\alpha_i = 1/N$ for all i, but this would produce a high variance. Instead, we should make α_i large for bright or nearby luminaires. To obtain better α_i we can get an estimated contribution L_i at $\mathbf{x}$ by evaluating Equation 3 for S_i (with a random $\mathbf{x}'$) with the geometry term set to one. These L_is can be directly converted to α_i by scaling them so their sum is one. Sometimes, an estimated L_i can be a false zero, because we happen to select an invisible point, which is blocked by the object or the luminaire. To get an unbiased estimation, the α_i *must* be assigned a positive value in this case. We check to make sure no point on the ith luminaire is above the tangent plane of the object at $\mathbf{x}$ before we allow α_i to be set to zero.

4 Examples

A simple system with both conventional and one ray shadow testing was implemented in C++. All figures we run at 128 by 128 pixel resolution to allow noise to be clearly visible.

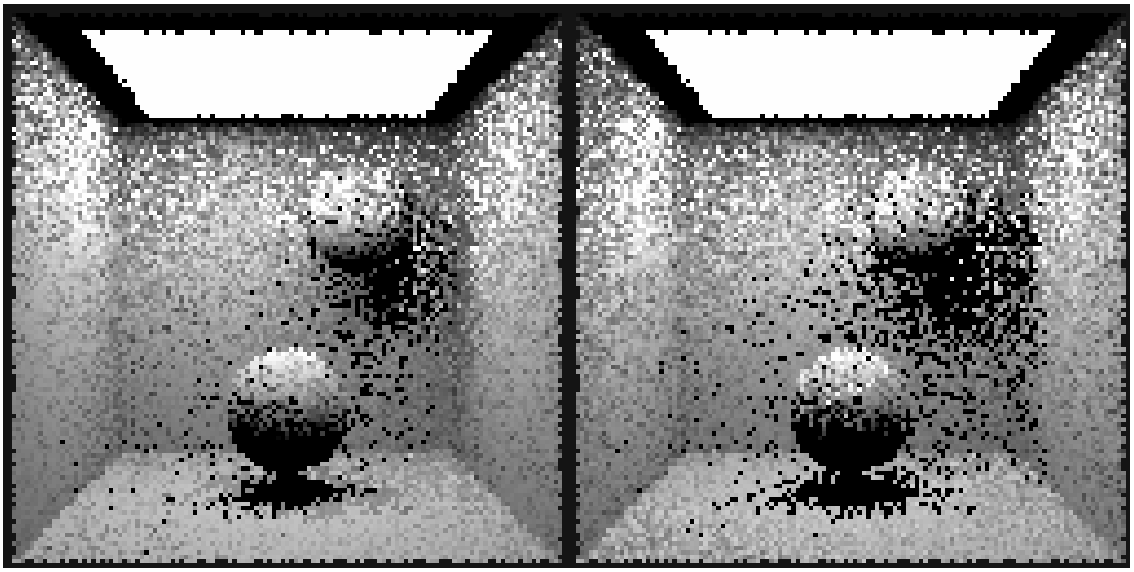

Figure 5: One ray per pixel with two triangular luminaires on the ceiling. On the left, one shadow ray is sent to each source, and on the right one shadow ray is sent to the union of the two luminaires.

Figures 5 and 6 show a two luminaire scene with one viewing ray per pixel and 100 viewing rays per pixel. On the left of each figure, conventional shadow testing is used, and on the right, only one shadow ray is fired for each viewing ray. In this case, the number of shadow rays is only halved, so the optimization makes little difference.

In Figures 7 and 8 a scene with 32 luminaires is shown. In the one ray per pixel images (Figure 7), the conventional shadow ray test generates 32 shadow rays at each pixel, so the variance of the estimate of the shadow is much lower than the variance of pixels containing edges. In the 100 viewing ray images (Figure 8), both scenes end up with low variance shadows and edges. Because conventional shadow testing generates 3200 shadow rays, that shadow is probably overly sampled.

In the left image of Figure 9, the α_i of Equation 7 are all set to be the same (1/32). This is considerably noiser than the image on the right, where α_i is made proportional to the potential influence of the ith luminaire.

On the left image in Figure 10, the α_i are chosen based on the potential contribution to luminaire centers. The incorrect banding on the sphere results from cases where the center of a luminaire is not visible, but off-center regions of the luminaire are visible. This emphasizes the rule that α_i must not be set to zero unless the ith luminaire contributes no light to the point being tested.

5 Conclusion

We have presented a method of direct lighting calculation that uses only one shadow ray for any number of area luminaires with arbitrary directional properties. Any number of shadow

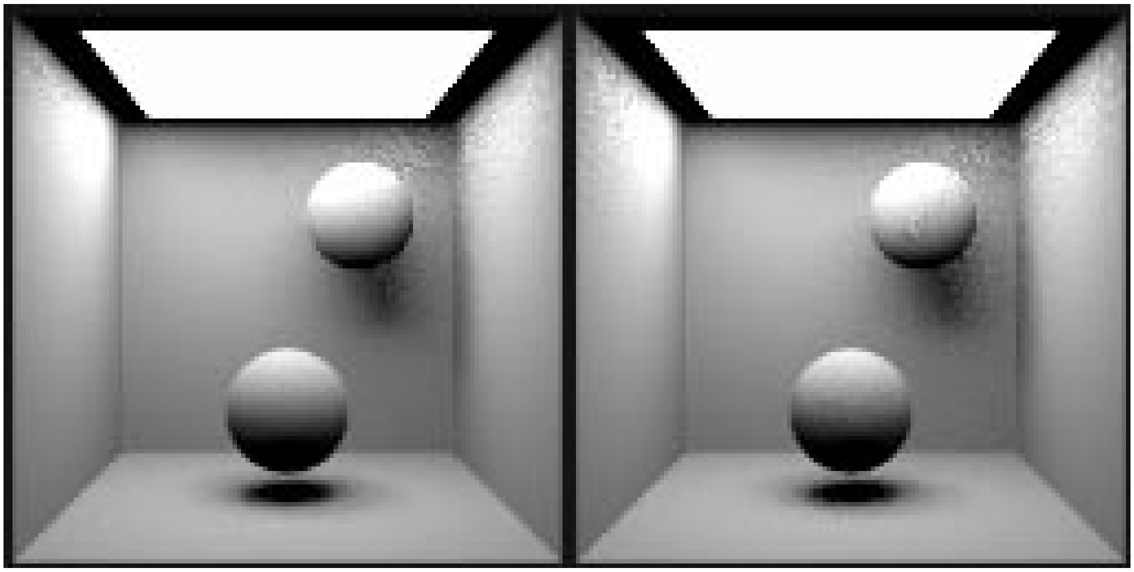

Figure 6: Simple scene with 100 rays per pixel.

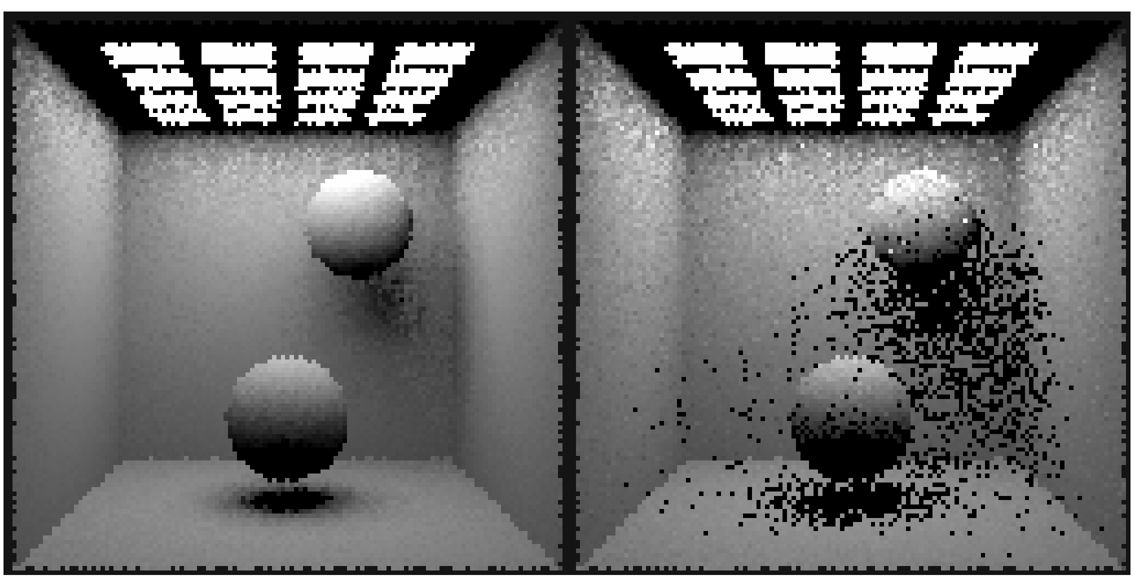

Figure 7: One ray per pixel with 32 triangular luminaires on the ceiling. On the left, one shadow ray is sent to each source, and on the right one shadow ray is sent to the union of the 32 luminaires.

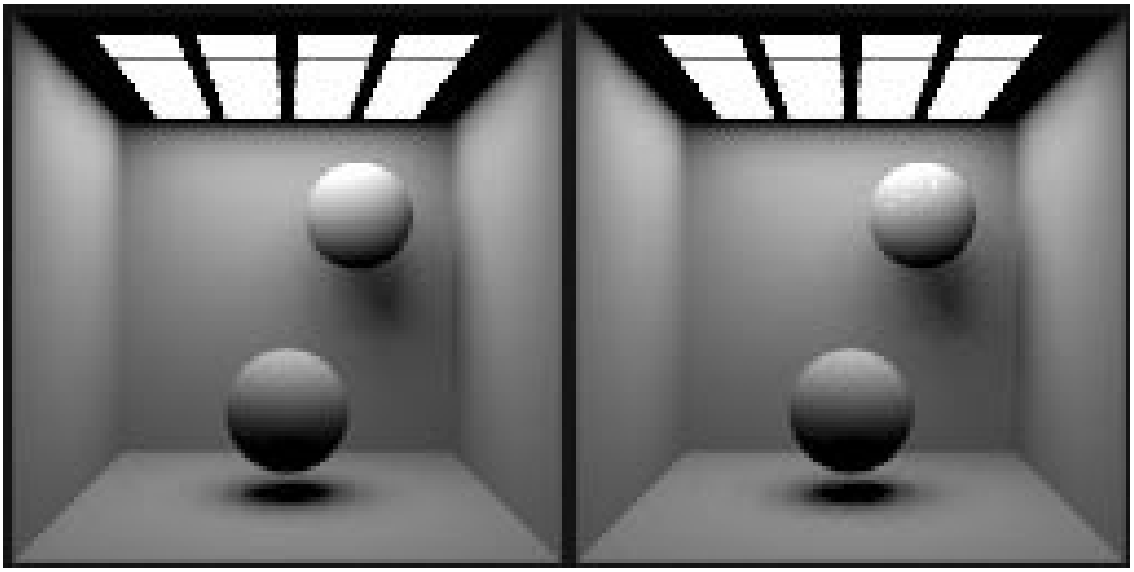

Figure 8: 32 luminaires with 100 rays per pixel. On the left, each viewing ray generates 32 shadow rays. On the right, each viewing ray generates one shadow ray.

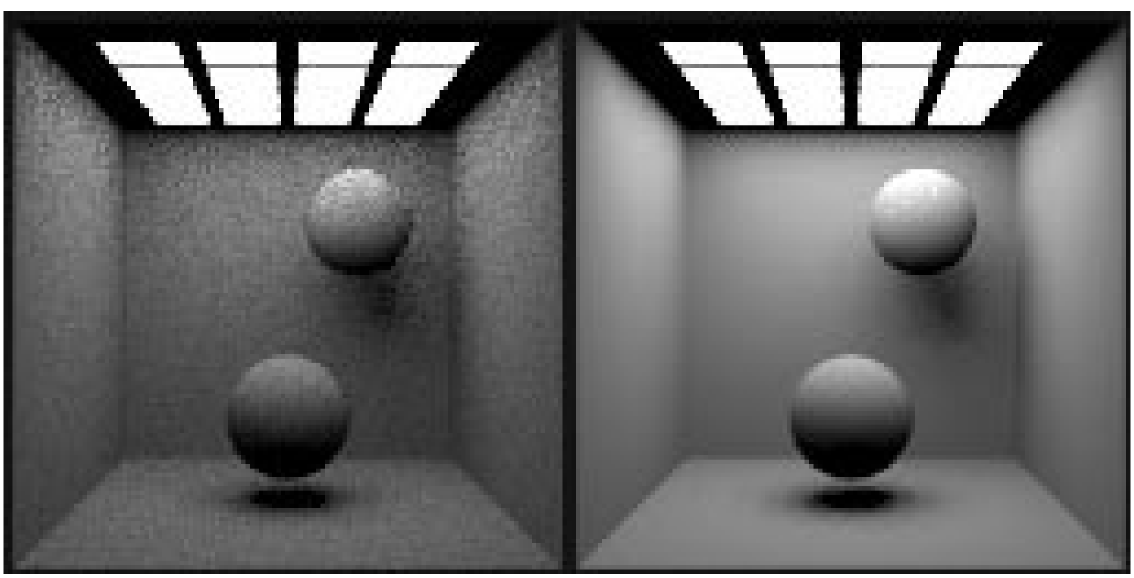

Figure 9: On the left, α_i is set to the same value for all i, and on the right, α_i is made proportional to the potential influence of the ith luminaire.

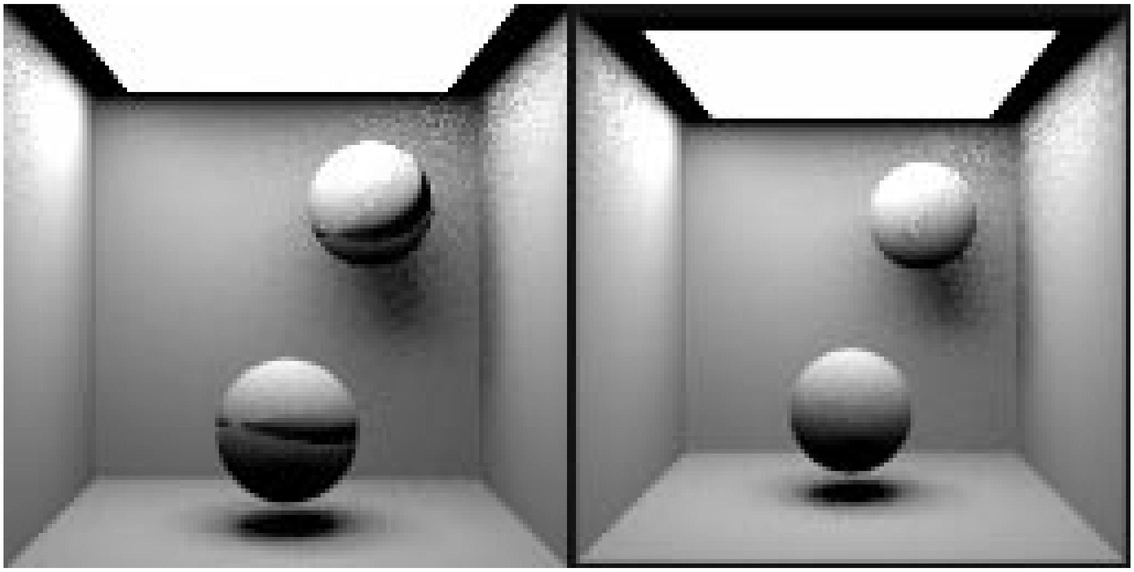

Figure 10: If the α_i are chosen based on the potential contribution to luminaire centers, false zero contribution estimates can cause artifacts such as those shown on the left. The viewpoints for the two images are slightly different.

rays can be sent if more accuracy is required by using secondary estimators. The method will work for planar and non-planar luminaire surfaces, as long as a method for selecting uniform random points from the luminaire surface is known.

The method could behave poorly in the presence of very bright luminaires that do not contribute radiance to the visible points in the scene. An example of when this could happen is a room at noon with the window shades fully closed. In this case it might be wise to send one shadow ray to the sun, and one to the union of interior luminaires.

Future work should include more sophisticated ways to construct probability densities on luminaires, and fast estimates of luminaire contributions for the assignment of α_i.

Use for sources that are not luminaires [7].

The basic rationale for this method is that direct lighting should not be calculated to a higher accuracy than necessary. This is very similar in concept to Kajiya's argument that we should not expend much work for deep parts of the ray tree [5].

6 Acknowledgements

Thanks to William Brown, Jean Buckley, Greg Rogers, Kelvin Sung, and Greg Ward for their help and input, and to William Kubitz, the thesis advisor for the project in which this work began.

References

[1] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *Computer Graphics*, 18(4):165–174, July 1984. ACM Siggraph '84 Conference Proceedings.

[2] Mark A. Z. Dippe and Erling Henry Wold. Antialiasing through stochastic sampling. *Computer Graphics*, 19(3):69–78, July 1985. ACM Siggraph '85 Conference Proceedings.

[3] H. Gouraud. Continuous shading of curved surfaces. *Communications of the ACM*, 18(6):623–629, June 1971.

[4] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods.* Wiley, New York, N.Y., 1964.

[5] James T. Kajiya. The rendering equation. *Computer Graphics*, 20(4):143–150, August 1986. ACM Siggraph '86 Conference Proceedings.

[6] John E. Kaufman, editor. *The Illumination Engineering Society Lighting Handbook, Reference Volume.* Waverly Press, Baltimore, MD, 1984.

[7] A. Kok and F. Jansen. Source selection for the direct lighting calculation in global illumination. In *Proceedings of the Second Eurographics Workshop on Rendering*, 1991. This Proceedings.

[8] Mark E. Lee, Richard A. Redner, and Samuel P. Uselton. Statistically optimized sampling for distributed ray tracing. *Computer Graphics*, 19(3):61–68, July 1985. ACM Siggraph '85 Conference Proceedings.

[9] James Painter and Kenneth Sloan. Antialiased ray tracing by adaptive progressive refinement. *Computer Graphics*, 23(3):281–288, July 1989. ACM Siggraph '89 Conference Proceedings.

[10] Bui-Tuong Phong. Illumination for computer generated images. *Communications of the ACM*, 18(6):311–317, June 1975.

[11] Werner Purgathofer. A statistical method for adaptive stochastic sampling. *Computers & Graphics*, 11(2):157–162, 1987.

[12] Peter Shirley. *Physically Based Lighting Calculations for Computer Graphics.* PhD thesis, University of Illinois at Urbana-Champaign, November 1990.

[13] Peter Shirley. A ray tracing algorithm for global illumination. *Graphics Interface '90*, May 1990.

[14] Greg Ward. Adaptive shadow testing for ray tracing. In *Proceedings of the Second Eurographics Workshop on Rendering*, 1991. This Proceedings.

[15] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.