TECHNICAL REPORT NO. 343

# Luminaire Sampling in Distribution Ray Tracing

by

Peter Shirley and Changyaw Wang

January 1992

COMPUTER SCIENCE DEPARTMENT
INDIANA UNIVERSITY
Bloomington, Indiana 47405-4101

# Luminaire Sampling in Distribution Ray Tracing

Peter Shirley [*]
Department of Computer Science
Indiana University

Changyaw Wang [†]
Department of Computer Science
Indiana University

## Abstract

In a distribution ray tracer, the crucial part in the direct lighting calculation is the sampling strategy for shadow ray testing. It is demonstrated that the naive strategies used in traditional distribution ray tracers are not adequate for rendering scenes that include a large number of luminaires. A new family of sampling strategies is introduced, and is shown to solve many of the problems associated with naive sampling strategies.

**CR Categories and Subject Descriptors:** I.3.0 [Computer Graphics]: General; I.3.6 [Computer Graphics]: Methodology and Techniques.

**Additional Key Words and Phrases:** luminaires, direct lighting, Monte Carlo integration, probability density function, ray tracing, rendering equation.

---

[*] Department of Computer Science, Lindley Hall, Indiana University, Bloomington, IN 47405. Email: shirley@cs.indiana.edu.

[†] Email: wangc@cs.indiana.edu.

# 1 Introduction

In recent years the advantages of Monte Carlo solutions to rendering problems have become increasingly clear. These advantages exist primarily because Monte Carlo methods lend themselves to problems of high dimension or high complexity. The chief problem of Monte Carlo methods is that the expected solution error tends to be relatively high. Fortunately, rendering problems usually require results within no more than about one percent of the correct value, so the high error of Monte Carlo methods is not a fatal flaw.

Monte Carlo methods have been used in ray tracing programs[4, 13, 33], and for radiosity programs[20, 1, 25, 10]. An unfortunate oversight in the graphics literature is that it is usually assumed that once the decision to apply Monte Carlo integration is made, that application is mechanical. It is true that it is mechanical to generate a *valid* estimate, but generating a *good* estimate (an estimate with low variance) can be difficult. Kirk and Arvo have demonstrated that choosing ray paths and reflected ray directions probabilisticly is also more difficult than is usually implied by the literature[17, 16].

In this paper, we focus on the design of Monte Carlo estimators for direct lighting from areal luminaires[1]. We phrase our solutions in terms of the probability density functions used to guide the shadow rays used for visibility tests. These methods are especially appropriate when many (tens or hundreds) samples will be taken in each pixel. If only a few sample will be taken in each pixel, then culling methods such as that of Ward[32] or Kok and Jansen[18] would certainly work better.

In Section 2 we describe the direct lighting integral (a special case of the rendering equation[13, 11]), and the fundamentals of Monte Carlo integration. In Section 3, we show the specifics of a Monte Carlo estimator for the color of a point illuminated by a single luminaire. In Section 4, we argue that applying Monte Carlo integration to a scene with many luminaires should *not* be done by sending a shadow ray to each luminaire. Instead, we construct a probability density function over the union of all luminaires and send a single shadow ray to that union. That section contains the most important new ideas in this paper. Section 5 discusses some time complexity issues and what kinds of programs would benefit from the techniques described in the previous sections. Finally, Section 6 discusses the strengths and weaknesses of the techniques presented in this paper, and suggests some possible directions for research.

---

[1] A *luminaire* is an object that produces light, such as a light bulb filament. An *areal* luminaire has a non-zero surface area, such as a fluorescent light panel.

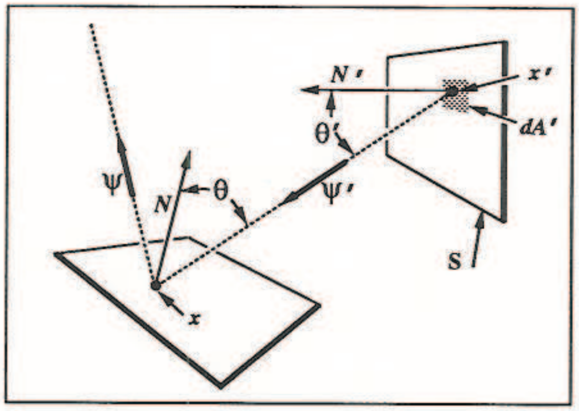

Figure 1: The geometry for Equation 1.

# 2 Theory

Many researchers have expressed the lighting at a point in terms of an integral equation, usually referred to as "the rendering equation"[4, 13, 11, 7]. Cook et al. argued that Monte Carlo integration was a good way to get an approximate solution to this integral equation. Kajiya extended the approach of Cook et al. into the first fully unbiased Monte Carlo solution to the rendering equation. Cook et al.'s approach is now usually called "distribution ray tracing" and Kajiya's method is called "path tracing". Both methods do a Monte Carlo integration for each luminaire by choosing sample points on each luminaire. Each sample point will have a "shadow ray" sent to it from the point being illuminated. Path tracing makes two changes to distribution ray tracing. The first change is that ray branching is not allowed, so transparent surfaces will generate either a reflected or refracted ray, but not both. The second change is that all surfaces, including diffuse surfaces, generate a reflected ray.

It is also necessary to generate a good estimate for the color of a point being illuminated by a single luminaire. Suppose we want to calculate the direct lighting component at a point x viewed from direction $\psi$. This quantity will be a *spectral radiance*[2], and can be written as a function of wavelength, $\lambda$: $L(\mathrm{x}, \psi, \lambda)$. We will assume that the wavelength dependencies are handled by calculation at a set of discrete wavelengths $\lambda_i$ and will drop the wavelength term from our notation. Given a luminaire, $S$, the direct lighting resulting from $S$

---

[2] In the heat transfer literature, and and some of the graphics literature, the word *intensity* is used instead of radiance. Because radiance is part of an ANSI standard for Illumination Engineering[12], and is used in Physics[5] and Colorimitry[34], we think it, along with the other standard terms of [12] should be adopted by the graphics community.

(see Figure 1) can be written:

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

where $g(\mathbf{x}, \mathbf{x}')$ is the *geometry term*, which is zero if there is an obstruction between $\mathbf{x}$ and $\mathbf{x}'$, and one otherwise [13]; $\rho(\mathbf{x}, \psi, \psi', \lambda)$ is the BRDF [15]; $\psi'$ is the direction from $\mathbf{x}'$ to $\mathbf{x}$; $\theta$ is the angle between $\psi'$ and the surface normal at $\mathbf{x}$; $\theta'$ is the angle between $\psi'$ and the surface normal at $\mathbf{x}'$; $dA'$ is the differential area of $\mathbf{x}'$.

Any integral over a region $\mathrm{R}$ can be approximated using Monte Carlo methods [8]:

$$I = \int_{x' \in \mathrm{R}} f(x')d\mu(x') \approx \frac{f(x)}{p(x)} \tag{2}$$

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 $x$).

Though it is easy to devise a valid estimator, a low-variance estimator requires either a very simple $f$ (which almost never occurs in graphics problems) or a careful design of the density function $p$. Careful design is usually called importance sampling[8, 13, 2]. A estimator a variance of zero if we use a $p$ that is similar to $f$:

$$p(x) = \frac{f(x)}{I}$$

As has been pointed out many times, this is not possible in any interesting cases because we must know $I$ (the answer we seek) to construct this $p$. Instead, we usually try to approximate this perfect $p$, or to approximate it for some parts of $\mathrm{R}$.

## 3 Direct Lighting for One Luminaire

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:

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

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

An important issue is the strategy for designing $p$ on the luminaire. At a minimum, $p$ must be a valid 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 sphere, making $p$ a uniform density (so the every point on the surface of the sphere is equally likely to be chosen) 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 approach 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 $\mathbf{x}$. This is the sampling strategy suggested by Kirk and Arvo in the context of directional sampling[17].

It is worth noting that sampling evenly within the solid angle subtended by the luminaire is *not* optimal, even if the geometry term is always one. We can see from Equation 3 that we will have a perfect estimator if $p(\mathbf{x})$ is proportional to the integrand:

$$p(\mathbf{x}) \propto g(\mathbf{x}, \mathbf{x}')\rho(\mathbf{x}, \psi, \psi')L_e(\mathbf{x}', \psi')\cos\theta \frac{\cos\theta'}{||\mathbf{x}' - \mathbf{x}||^2} \tag{4}$$

If we sample evenly within the solid angle subtended by the luminaire, then $p$ is:

$$p(\mathbf{x}) \propto \frac{\cos\theta'}{||\mathbf{x}' - \mathbf{x}||^2} \tag{5}$$

The "missing" terms in Equation 5 (those that are in Equation 4 and not in Equation 5) should cause noise whenever they are not constant. These missing terms are:

$$g(\mathbf{x}, \mathbf{x}')\rho(\mathbf{x}, \psi, \psi')L_e(\mathbf{x}', \psi')\cos\theta \tag{6}$$

The geometry term $g$ is not constant when the luminaire is partially visible. This causes the familiar noisy shadow edges in distribution ray tracing.

The $\rho$ term is constant for diffuse surfaces. For non-diffuse surfaces we should expect more noise. For specular surfaces such as mirrors, reflection rays are used and no explicit direct lighting calculations are performed, so there is no variance problem for specular surfaces. The $L_e$ term is constant for simple diffuse luminaires, but is not otherwise. For example, an ideal television screen will be diffuse at every point, but the radiance will vary across the screen. A more

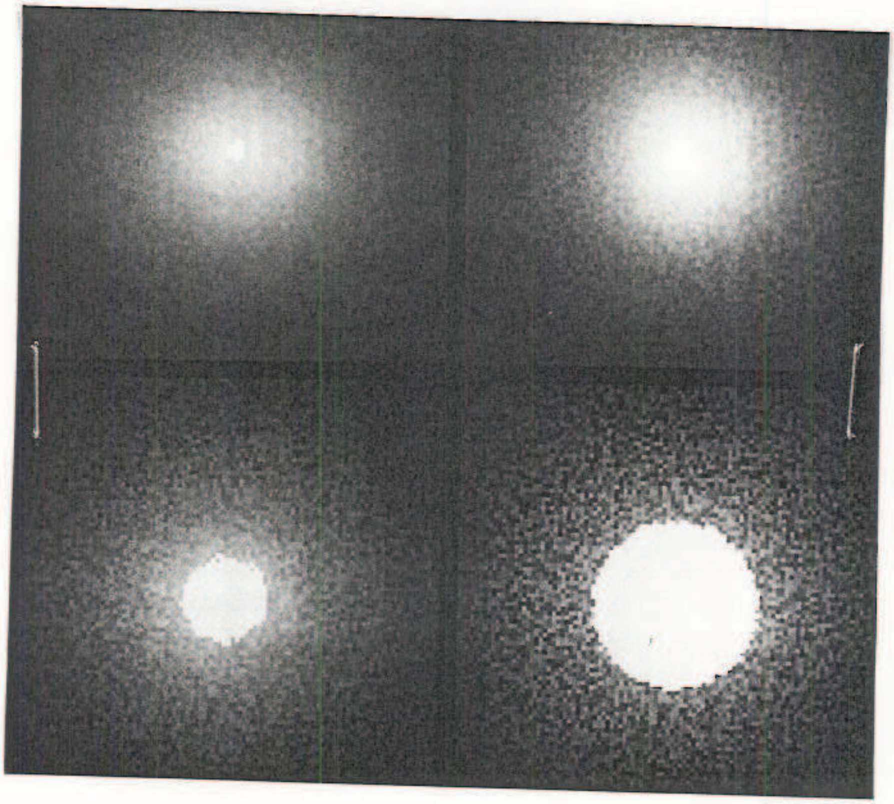

Figure 2: Noise caused by variation in the cosine term. These four lights have equal power and illuminate a diffuse plane seen from above.

intelligent sampling strategy might build a probability distribution function on $S$ and thus more heavily sample the brightest parts of the screen. If $S$ is not diffuse, it should require either more sampling or more selective sampling than diffuse luminaires. The $cos\theta$ term is constant only for point luminaires. It will cause more noise for luminaires that subtend a large solid angle. An example of this effect is shown in Figure 2, where three spherical luminaires are shown from above. Notice that the noise on the ground plane increases as the size of the luminaire (and thus the variation in the $\cos\theta$ term) increases.

An important topic of research is how to design $p(\mathbf{x})$ for non-diffuse surfaces and luminaires. As material and luminaire models become more complex, we need to strive to make densities behave more like the density in Equation 4. We do not believe this will be trivial. For triangular luminaires, we could only approximate constant sampling within the solid angle subtended by the triangle, so implementation of even the simple density of Equation 4 was difficult. The details of our implementations of $p(\mathbf{x})$ on triangles and spheres can be found in [31].

# 4 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. Such an approach makes much more sense when making a picture of an environment with thousands or millions of luminaires. In this section, we show several ways to make a density function that covers all luminaires, and show the results of our implementation.

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 $\mathbf{R}_N$.

$$I = I_1 + I_2 + \cdots + I_N \tag{7}$$

where each integral $I_i$ is defined by:

$$I_i = \int_{x' \in \mathbf{R}_i} f(x') d\mu(x') \approx \frac{f_i(x)}{p_i(x)} \tag{8}$$

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

This can be extended using $N$ separate Monte Carlo integrations:

$$I \approx \frac{f_1(x)}{p_1(x)} + \frac{f_2(x)}{p_2(x)} + \cdots \frac{f_N(x)}{p_N(x)} \tag{9}$$

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:

$$f(x) = \begin{cases} f_1(x) & \text{if } x \in \mathbf{R}_1 \\ f_2(x) & \text{if } x \in \mathbf{R}_2 \\ \vdots & \vdots \\ f_N(x) & \text{if } x \in \mathbf{R}_N \end{cases} \tag{10}$$

Note that Equation 2 can be applied even if $\mathbf{R}$ has holes or is not fully connected. We simply need to estimate the integral:

$$I = \int_{x' \in (\mathbf{R} = \cup \mathbf{R}_i)} f(x') d\mu(x') \approx \frac{f(x)}{p(x)} \tag{11}$$

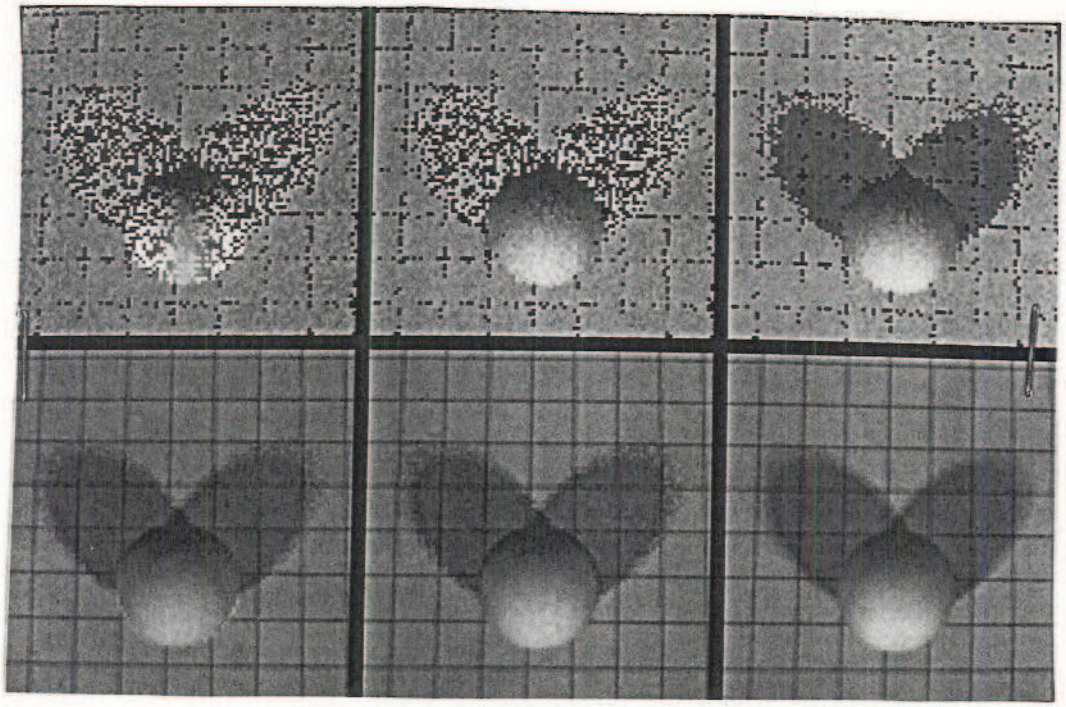

Figure 3: Top row: 1 Sample per pixel. Bottom row: 20 samples per pixel. Left: constant $\alpha_i$. Middle: linear $\alpha_i$. Right: conventional sampling.

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 $p_i$:

$$p(x) = \begin{cases} \alpha_1 p_1(x) & \text{if } x \in \mathbf{R}_1 \\ \alpha_2 p_2(x) & \text{if } x \in \mathbf{R}_2 \\ \vdots & \vdots \\ \alpha_N p_N(x) & \text{if } x \in \mathbf{R}_N \end{cases} \tag{12}$$

where the $\alpha_i$ sum to one, and where each $\alpha_i$ is non-zero if $I_i$ is non-zero. The value of $\alpha_i$ is the probability of selecting a point in $\mathbf{R}_i$. The $p_i$ is then used to determine which point in $\mathbf{R}_i$ is chosen.

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 $\alpha_i$.

## 4.1 Constant $\alpha_i$

The simplest way to choose values for $\alpha_i$ would be to make them all equal: $\alpha_i = 1/N$ for all $i$. This would definitely make a valid estimator because the $\alpha_i$

sum to one and none of them is zero. This approach was proposed by Lange[19]. Unfortunately, in many scenes this would not be a good estimator because it would produce a high variance.

## 4.2 Linear $\alpha_i$

Suppose we had perfect $p_i$ (see Equation 4) defined for all the luminaires. A zero variance solution would then result if we could set $\alpha_i = I_i$. Recall that $I_i$ in this case is the radiance of x due specifically to $S_i$, which we will denote $L_i$. If we can make $\alpha_i$ approximately proportional to $I_i$, then we should have a fairly good estimator. We call this the *linear* method of setting $\alpha_i$ because its execution time is linearly proportional to $N$, the number of luminaires.

To obtain such $\alpha_i$ we get an estimated contribution $L_i$ at x by approximating Equation 1 for $S_i$ with the geometry term set to one. These $L_i$s (from all luminaires) can be directly converted to $\alpha_i$ by scaling them so their sum is one. This method of choosing $\alpha_i$ will be valid because all potentially visible luminaires will end up with positive $\alpha_i$. We should expect the highest variance in areas where shadowing occurs, because this is where setting the geometry term to one causes $\alpha_i$ to be a poor estimate of $\alpha_i$. Figure 3 shows a sphere illuminated by two luminaires, calculated using constant $\alpha_i$, linear $\alpha_i$, and traditional sampling (a shadow ray for each luminaire). This case is simple enough that the behavior of the constant $\alpha_i$ method is not too bad. If the luminaires were less symmetrically placed, the constant method would produce more noise.

This method of setting $\alpha_i$ was first used in [25], and was first theoreticly justified in [28].

Implementing the linear $\alpha_i$ code was trickier than we expected. We implemented a method for each type of luminaire that estimated $L_i$ for a particular x and $\rho$. If the entire luminaire is below the tangent plane at x, then the estimate for $L_i$ should be zero. An easy mistake to make (which we made), is to set $L_i$ to zero if the center of the luminaire is below the horizon. This will make $\alpha_i$ take the one value that is not allowed: an incorrect zero. Such a bug will become obvious in pictures of spheres illuminated by luminaires that subtend large solid angles, but for many scenes such errors are not noticeable (the figures in [25] had this bug, but it was not noticeable). An example of one sphere illuminated by a large luminaire is shown with and without the bug in Figure 4.

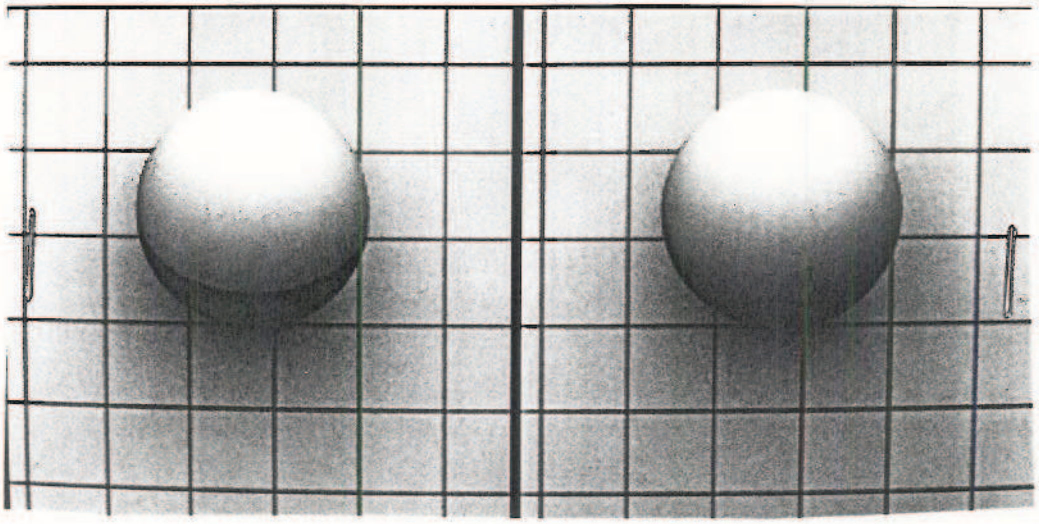

Figure 4: Left: incorrect $\alpha_i$. Right: correct $\alpha_i$.

### 4.3 Spatially Subdivided $\alpha_i$

In the linear method, choosing $\alpha_i$ based on estimated contribution requires querying every luminaire in the scene. This is acceptable for many scenes, but if $N$ is large (thousands or millions), even that might be too slow. In such scenes at most a few hundred luminaires (and usually at most tens) will contribute significantly to the radiance at any particular point. Suppose we can partition $S = \{S_1, \ldots, S_N\}$ into two subsets $S_{bright}$ and $S_{dim}$, where $S_{bright}$ is the set of luminaires that are "important" to x (i.e. they contribute significantly to the radiance of x), and $S_{dim}$ is simply $S - S_{bright}$. With these two subsets of $S$ we can construct low cost $\alpha_i$.

Suppose that the size of $S_{bright}$ is $N_b$. If we have partioned $S$ correctly, then $N_b$ should be much less than $N$. For each $S_i$ in $S_{bright}$, we estimate $L_i$ in the same way we did in the linear method. We choose one $S_r$ randomly from $S_{dim}$, and estimate $L_r$ accurately. If $L_r$ is zero, we repeatedly choose $S_r$ until a non-zero $L_r$ is found. We now assume that all members of $S_{dim}$ contribute approximately the same amount as $S_r$. This gives estimates of $L_i$ for all $S_i$, and will be reasonably accurate for the important luminaires. We can now construct a probability space by normalizing the $\alpha_i$ so that they sum to one. If we want more accuracy for the $\alpha_i$ of members of $S_{dim}$, we can average $L_r$ over several

$S_r$.

The difficult part of this method is deciding which luminaires are in $S_{bright}$ for a particular x. As pointed out by Kok and Jansen[18], a luminaire that is important to the color of x is likely to be important to the neighboring points of x. This implies we can use a spatial subdivision scheme to precompute a $S_{bright}$ for each spatial cell in the spatial subdivision structure. For a particular cell (a box aligned to the Cartesian axes), a luminaire is put in $S_{bright}$ if it might contribute more than a threshold average spectral radiance to a diffuse surface within the cell. A simple way to determine whether the maximum potential contribution of a luminaire is above the threshold is to evaluate Equation 1 with $g$, $\rho$, and $\cos\theta$ all set to one for all points (implementationally a large number of points) on the boundary of the spatial subdivision cell. For diffuse spherical luminaires, we need only evaluate the potential contribution at the point on the cell nearest the luminaire. For more complicated luminaires, methods for avoiding a brute force search need further study.

An easy way to choose the subdivision cells is simply to use the leaf cells of the conventional subdivision structure (e.g. the octree leaves of a Glassner style octree used for ray intersection acceleration[6]), and maintain a separate $S_{bright}$ list at each leaf. This is a finer than needed subdivision for scenes where $P$, the number of objects, is much greater than $N$, the number of luminaires. An advantage of this is that no luminaire lists need to be constructed for empty cells, and the characteristics of reflective objects can be used to construct the lists. Instead, we have implemented a separate *light octree* that recursively subdivides itself until each leaf is at a maximum allowed depth or when the size of $S_{bright}$ for that cell is below a specified limit. The depth and size limits are similar to those in a conventional octree, and their values are even less well understood by us so far. To avoid excess subdivision, we check to see if the minimum contribution of an important luminaire to a cell is above the threshold. If all members of $S_{bright}$ are thus determined to be in $S_{bright}$ for any possible descendant of that cell, we do not subdivide.

Another difficulty in building the light octree is what to use for the average radiance threshold. For our program we assume we know what spectral radiance distribution, $L_w$, would map to white (an rgb triple of [255,255,255] on our CRT) on the display device. We set the radiance threshold to be some fraction of $L_w$[3]. Because our display device has eight bits per channel, we usually make the threshold a few percent of $L_w$. Such a threshold will ensure that any luminaire that can change the pixel color more than a few steps will be included in $S_{bright}$. If $L_w$ is chosen correctly, than the sum of contributions of luminaires should be

---

[3] $L_w$ should be chosen according to a perceptual viewer model, such as the model implemented by Tumblin and Rushmeier[30]. Such models will become increasingly important as physically based rendering becomes more popular.

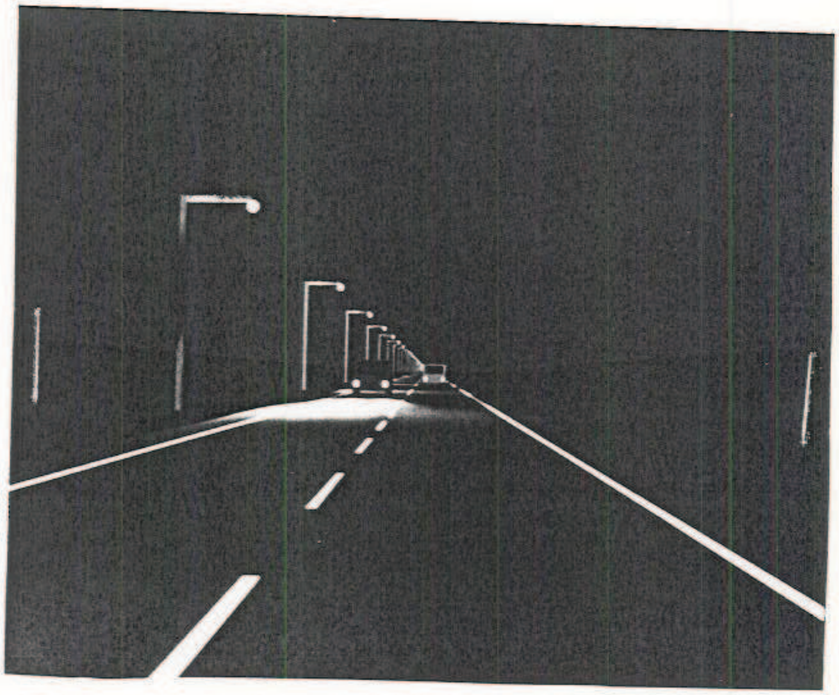

Figure 5: A scene rendered using the light octree with 9 rays per pixel, and 30 rays sent in pixels with high variance (fewer than ten percent).

no more than $L_w$, and thus the number of important luminaires should be less than one hundred for most scenes.

Figure 5 shows a scene with 108 luminaires renderered using the light octree. This runs only a little faster than the linear method on this scene because $N$ is only 108, but the two methods produce very similar results. Figure 6 shows four pictures of a pair of sphereflakes. Each sphereflake is composed of 7381 spheres. In the three images with 7381 luminaires, the light octree was used. As expected, the one sample case has a large variance, and the forty sample case is fairly smooth. Notice that both 10 sample cases (with 1 and 7381 luminaires) have visible noise. This implies that both cosine and visibility errors cause noise in the 7381 luminaire case. The 7381 luminaire figure with ten samples took less than twice as long as the one luminaire case with ten samples. This means the overhead of choosing the ray using the light octree does not swamp the cost of sending the shadow ray, even in this extreme case where half of the objects are luminaires. In less pathological scenes, the light octree should be smaller relative to the number of primatives, and performance should be even better.

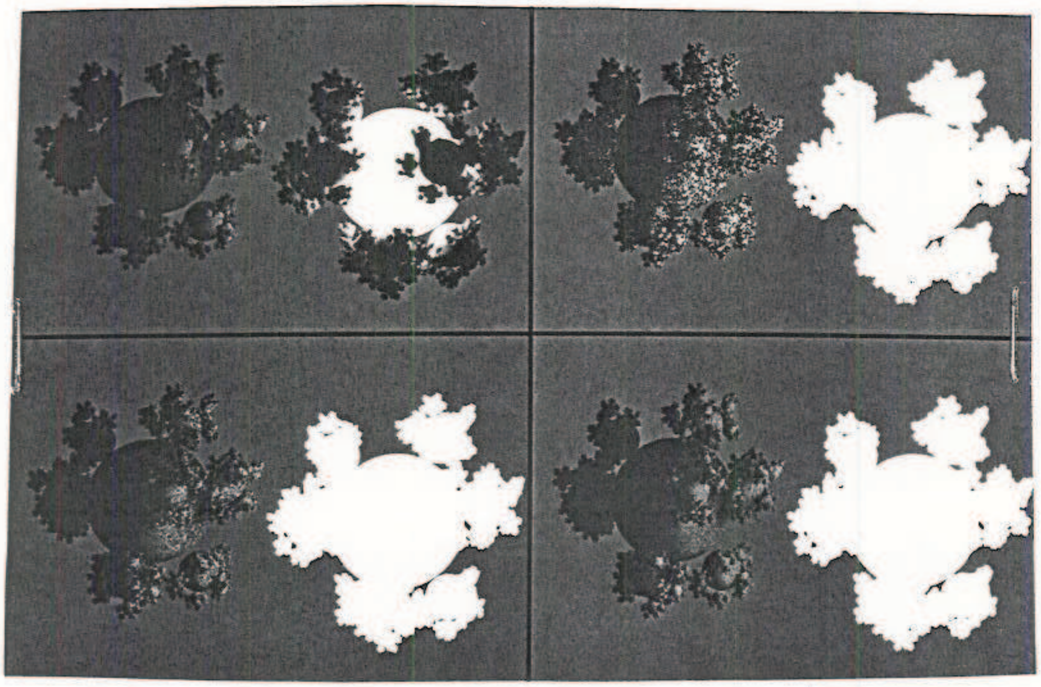

Figure 6: Top left: One luminaire, 10 samples per pixel. Top right: 7381 luminaires, 1 sample. Bottom left: 7381 luminaires, 10 samples. Bottom right: 7381 luminaires, 40 samples.

## 5 Discussion

In Section 3 we showed that it is relatively easy to develop a valid sampling scheme for any luminaire on whose surface we can devise a probability density function $p(\mathrm{x})$. This means we can add even complicated luminaires such as toroidal luminaires to a ray tracing system without adding any bias to the results. For a torus, and for most explicit surfaces, we can sample with a constant $p$. More intelligent $p$ could be substituted later when they are found, without changing anything but this component of the code. This is consistent with our experience that there is a direct mapping from the mathematics of Section 3 to the lighting code, then the code is easy to prototype and debug with a uniform $p$, and can later be improved cleanly by changing to a better $p$.

In Section 4 we showed three ways to sample a space of many luminaires. The first method, where $\alpha_i$ are all the same, has the advantage that it is $O(1)$ time to choose a shadow ray, and is easy to implement, but it is not useful for complex scenes (except as a debugging case). The second method we have found to work quite well in practice, is reasonably easy to implement, and is $O(N)$, where $N$ is the number of luminaires, to generate a shadow ray. The light octree method is much harder to implement, but is approximately $O(\log N)$ (this assumes the depth of the octree is approximately $\log N$ and that the cells contain an approximately constant size $S_{bright}$). For scenes with hundreds of thousands of luminaires, this complexity is much better than the $O(N)$ method. Ward's method[22], which he has shown to work quite well in scenes with reasonably large $N$, should probably not be used for *very* large $N$ because he sorts the entire luminaire list and thus takes $O(N \log N)$ time. After selecting a shadow ray, there will be a time cost for the intersection calculation of of at best $O(\log P)$[4] where $P$ is the number of geometric primitives in the scene. Because $P \geq N$, and sending a ray costs $O(\log P)$ time, a time cost to select the ray of $O(\log P)$ should be acceptable.

Clearly, the linear and light octree methods are only useful for scenes with a large number of luminaires. Such scenes are becoming increasingly important. In outdoor scenes, especially in urban settings, scenes with thousands and even hundreds of thousands of luminaires are commonplace. In opera and theater applications, hundreds or thousands of luminaires are common[29]. In infrared scenes, almost all surfaces are luminaires, so something such as the light octree is crucial. These methods also make it easy to use luminaires defined by many polygons or parametric patches. No matter how many patches define a light bulb surface, it will receive only one shadow ray.

---

[4] This is a loose use of the big-oh notation. We mean average time complexity for environments encountered in practice. If hierarchical spatial subdivision is used, the a $O(\log P)$ time will happen for well-behaved environments. Such behavior has been assumed for time complexity analysis in radiosity[9, 27], and has much empirical evidence behind it, but has not yet been proven even for special cases.

Even in scenes with only a few luminaires, the linear or light octree techniques can be useful. Many of the recent rendering techniques can be viewed in terms of replacing surfaces with what we call *imposters*. An example of an imposter is a luminaire with zero reflectivity that is shaped like a reflecting patch and *emits* light in exactly the same distribution and intensity as the reflecting patch *reflects* light. There is no way visually to tell an imposter from the original surface. Adding an imposter reduces the number of patches whose reflected light needs to be calculated and increases the number of luminaires. Using imposters becomes more attractive if the cost of direct lighting does not increase linearly with $N$. Imposters have been used effectively by Kok and Jansen[18] and by Chen et al.[3]. Radiosity programs can be viewed as a preprocess that replace all diffuse surfaces with imposters, so only viewing is required.

The methods 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 view the lighting calculation as more than one integral, where especially bright luminaires are sampled separately from the dimmer luminaires. to the union of interior luminaires. The overall reason this case is a problem is because the $\alpha_i$ are bad estimates. Bad $\alpha_i$ estimation can happen because the geometry term, $g(\mathbf{x}, \mathbf{x}')$, is not accounted for, or because the radiance estimate for the luminaire is bad, as might happen when using a complicated directional luminaire. Perhaps some inclusion of shadowing in the selection of $S_{bright}$ would help.

The careful reader may have noted that two important sampling issues have been ignored in this paper. The first issue is that, in practice, *Quasi-Monte Carlo* integration is used instead of true Monte Carlo. The second issue is that we must be able to choose sample points that are distributed according to density $p$. These two issues are related. If $p$ is a density on a d-dimensional space, and we have a random d-dimensional point $\xi$ that is distributed evenly on the d-dimensional cube $[0,1]^d$, than the $t(\xi)$ will be distributed according to $p$ for some transformation function $t$. To perform Monte Carlo integration we use a set of input random points $\{\xi_1, \xi_2, \ldots, \xi_n\}$ and use $t$ to generate $n$ sample points with the proper density: $\{t(\xi_1), t(\xi_2), \ldots, t(\xi_n)\}$. To produce a lower variance result we can use Quasi-Monte Carlo integration, where a quasirandom[5] set of points is used instead of the random $\xi_i$[35]. The most up to date discussion of quasirandom points for ray tracing is by Mitchell[21]. A description of how to generate a transformation function $t$ is described in some classic Monte Carlo books[8, 24, 14], and several $t$ useful in ray tracing are given in [26].

---

[5] quasirandom objects have some of the desired distribution quantities of random numbers without being truly random. See [35] for further details.

It should be noted that we have phrased our discussion in terms of an integration over visible and nonvisible surfaces. It is also possible to integrate over all visible surfaces. This method is usually phrased in terms over an integration over all solid angles (see the form of the rendering equation given by Immel et al.[11]). This gives rise to sending rays toward certain directions, rather than toward certain points. Kirk and Arvo proposed such a method[17], and it has been used for the indirect illumination component in many programs[4, 13, 33, 23].

## 6 Conclusion

The methods we have presented illustrate that the crucial step in a Monte Carlo integration is the design of the probability density function the samples are chosen from. We have shown that careful design of the function on luminaires can significantly decrease the computation time of the direct lighting calculation in complex scenes.

Future work should include more sophisticated ways to construct probability densities on luminaires, and fast estimates of luminaire contributions for the assignment of $\alpha_i$. The most unfinished business is how to design and use subdivision structures for lighting. We are confident that our octree approach, though able to produce pictures not practical by other means we know of, can be greatly improved upon.

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[13]. It is certainly not true that one shadow ray per viewing ray is optimal, however. For the 100 luminaire case, one shadow ray is better than 100 shadow rays, but two or three might be better still. This issue requires further investigation.

## 7 Acknowledgements

Thanks to William Brown, Jean Buckley, Frederick Jansen, William Kubitz, Don Mitchell, Greg Rogers, Kelvin Sung, Minru Wang, and Greg Ward for their help and input, and to Eric Haines for making his very useful object databases publicly available.

# References

[1] John M. Airey, John H. Rohlf, and Frederick P. Brooks. Towards image realism with interactive update rates in complex virtual building environments. *Computer Graphics*, 24(1):41–50, 1990. ACM Workshop on Interactive Graphics Proceedings.

[2] James Arvo and David Kirk. Particle transport and image synthesis. *Computer Graphics*, 24(3):63–66, August 1990. ACM Siggraph '90 Conference Proceedings.

[3] Shenchang Eric Chen, Holly Rushmeier, Gavin Miller, and Douglass Turner. A progressive multi-pass method for global illumination. *Computer Graphics*, 25(4):165–174, July 1991. ACM Siggraph '91 Conference Proceedings.

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

[5] Walter G. Driscoll. *Handbook of Optics*. McGraw-Hill, New York, N.Y., 1978.

[6] Andrew S. Glassner. Space subdivision for fast ray tracing. *IEEE Computer Graphics and Applications*, 4(10):15–22, 1984.

[7] Roy Hall. *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York, N.Y., 1988.

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

[9] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, July 1991. ACM Siggraph '91 Conference Proceedings.

[10] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics*, 24(3):145–154, August 1990. ACM Siggraph '90 Conference Proceedings.

[11] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics*, 20(4):133–142, August 1986. ACM Siggraph '86 Conference Proceedings.

[12] American National Standard Institute. Nomenclature and definitions for illumination engineering. ANSI Report, 1986. ANSI/IES RP-16-1986.

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

[14] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*. John Wiley and Sons, New York, N.Y., 1986.

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

[16] David Kirk and James Arvo. Unbiased sampling techniques for image sysnthesis. *Computer Graphics*, 25(4):153–156, July 1991. ACM Siggraph '91 Conference Proceedings.

[17] David Kirk and James Arvo. Unbiased variance reduction for global illumination. In *Proceedings of the Second Eurographics Workshop on Rendering*, 1991.

[18] 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.

[19] Brigitta Lange. The simulation of radiant light transfer with stochastic ray-tracing. In *Proceedings of the Second Eurographics Workshop on Rendering*, 1991.

[20] Thomas J. V. Malley. A shading method for computer generated images. Master's thesis, University of Utah, June 1988.

[21] Don P. Mitchell. Spectrally optimal sampling for distribution ray tracing. *Computer Graphics*, 25(4), July 1991. To appear in ACM Siggraph '88 Conference Proceedings.

[22] Holly Rushmeier and Greg Ward. Experimental comparison and evaluation. *Radiosity*, 1990. ACM Siggraph '90 Course Notes.

[23] Holly E. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. PhD thesis, Cornell University, May 1988.

[24] Y. A. Screider. *The Monte Carlo Method*. Pergamon Press, New York, N.Y., 1966.

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

[26] Peter Shirley. Discrepancy as a quality measure for sampling distributions. In *Eurographics '91*, September 1991.

[27] Peter Shirley. Time complexity of monte carlo radiosity. In *Eurographics '91*, September 1991.

[28] Peter Shirley and Changyaw Wang. Direct lighting by monte carlo integration. In *Proceedings of the Second Eurographics Workshop on Rendering*, 1991.

[29] Francois Sillion, James Arvo, Stephen Westin, and Donald Greenberg. A global illumination algorithm for general reflection distributions. *Computer Graphics*, 25(4):187–196, July 1991. ACM Siggraph '91 Conference Proceedings.

[30] Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic computer generated images. Technical Report GIT-GVU-91-13, Graphics, Visualization, and Usuability Center, Georgia Institute of Technology, 1991.

[31] Changyaw Wang. Physically correct direct lighting for distribution ray tracing. In David Kirk, editor, *Graphics Gems 3*. Academic Press, New York, NY, 1992. (to appear).

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

[33] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics*, 22(4):85–92, August 1988. ACM Siggraph '88 Conference Proceedings.

[34] Gunter Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, New York, N.Y., 1982.

[35] S. K. Zeremba. The mathematical basis of monte carlo and quasi-monte carlo methods. *SIAM Review*, 10(3):303–314, July 1968.