Optimally Combining Sampling Techniques for Monte Carlo Rendering

Eric Veach   *Leonidas J. Guibas*

Computer Science Department
Stanford University

Abstract

Monte Carlo integration is a powerful technique for the evaluation of difficult integrals. Applications in rendering include distribution ray tracing, Monte Carlo path tracing, and form-factor computation for radiosity methods. In these cases variance can often be significantly reduced by drawing samples from several distributions, each designed to sample well some difficult aspect of the integrand. Normally this is done by explicitly partitioning the integration domain into regions that are sampled differently. We present a powerful alternative for constructing robust Monte Carlo estimators, by combining samples from several distributions in a way that is provably good. These estimators are unbiased, and can reduce variance significantly at little additional cost. We present experiments and measurements from several areas in rendering: calculation of glossy highlights from area light sources, the "final gather" pass of some radiosity algorithms, and direct solution of the rendering equation using bidirectional path tracing.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.3.3 [Computer Graphics]: Picture/Image Generation; G.1.9 [Numerical Analysis]: Integral Equations—Fredholm equations.

Additional Keywords: Monte Carlo, variance reduction, rendering, distribution ray tracing, global illumination, lighting simulation.

1 Introduction

Technically, rendering is all about clever ways to approximate integrals. For example, the pixel values in an "ideal" image usually involve integration over the image plane, lens position, and so on. Furthermore, the quality of a rendering algorithm is frequently measured by the accuracy and efficiency with which these integrals are approximated. In this paper, we focus on Monte Carlo (MC) methods for evaluating such integrals. These methods use random sampling to simplify the integration problem, by expressing the integral as the expected value of a random variable. The major drawback of MC integration is that the resulting estimates can have high variance; this is perceived as noise in a rendered image.

Address: Computer Science Department, Robotics Laboratory
Stanford University, Stanford, CA 94305-2140
E-mail: ericv@cs.stanford.edu, guibas@cs.stanford.edu
Web: http://www–graphics.stanford.edu/

Unfortunately, the functions that we need to integrate in computer graphics are often ill-behaved. They are almost always discontinuous, and often have singularities or very large values over small portions of their domain. Because of this, we often need more than one sampling technique to estimate an integral with low variance. Normally this is accomplished by explicitly partitioning the domain of integration into several regions, and designing a sampling technique for each region. For example, a simple distribution ray tracer may use one technique to evaluate direct lighting, another to estimate glossy reflections, and a third for ideal specular contributions.

In this paper, we explore the general problem of constructing low-variance estimators by combining samples from several techniques. We do not construct new sampling methods—all the samples we use come from one of the given distributions. Instead, we look for better ways to combine the samples; in particular, strategies that compute *weighted combinations*. We show that there is a large class of unbiased estimators of this type, parameterized by a set of weighting functions. We then seek weighting functions within this class that minimize variance. In a sense, we are asking the inverse problem: given several sampling techniques, how should the domain be partitioned among them? (Or more generally, how should the samples be weighted?)

A good solution to this problem turns out to be surprisingly simple. We show how to combine samples from several distributions in a way that is provably good, both theoretically and practically. This allows us to construct MC estimators that have low variance for a broad class of integrands—we call such estimators *robust*. The significance of our methods is not that we can take several bad sampling techniques and concoct a good one out of them, but rather that we can take several potentially good techniques and combine them so that the strengths of each are preserved.

In Sec. 2, we review the fundamentals of MC integration for rendering, and give an example to motivate our variance reduction framework. Sec. 3 explains our ideas on combining samples from several distributions, and gives theoretical justification under several models (proofs can be found in App. A). In Sec. 4 we present computed images and numerical results for several application areas: glossy highlights from area light sources, the "final gather" pass of some radiosity algorithms, and direct solution of the rendering equation using bidirectional path tracing. Finally, Sec. 5 discusses of a number of tradeoffs and open issues related to our work.

2 Monte Carlo rendering

2.1 Integrals for radiance

We have chosen two basic problems in rendering to illustrate our techniques: evaluation of the radiance leaving a surface given a description of the incoming illumination (as in distribution ray tracing or some "final gather" approaches), and direct solution of the rendering equation[5]. For further details and background see [3].

Given the incident radiance distribution $L_i(\mathbf{x}', \vec{\omega}_i')$ at a point $\mathbf{x}'$,

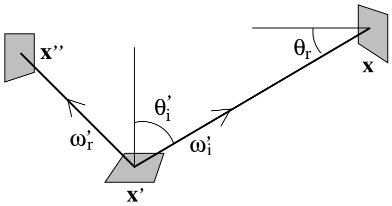


Figure 1: Geometry for the reflectance equation.

the reflected radiance $L_{\mathrm{r}}(\mathbf{x}', \vec{\omega}'_{\mathrm{r}})$ is given by the *reflectance equation*

$$L_{\mathrm{r}}(\mathbf{x}', \vec{\omega}'_{\mathrm{r}}) = \int_{\mathcal{S}^2} f_{\mathrm{r}}(\mathbf{x}', \vec{\omega}'_{\mathrm{i}} \leftrightarrow \vec{\omega}'_{\mathrm{r}}) L_{\mathrm{i}}(x, \vec{\omega}'_{\mathrm{i}}) \left|\cos(\theta'_{\mathrm{i}})\right| \, d\sigma(\vec{\omega}'_{\mathrm{i}}) \quad (1)$$

where f_{r} is the bidirectional reflectance distribution function (BRDF), $\mathcal{S}^2$ is the set of all unit direction vectors, σ is the usual solid angle measure, and θ'_{i} is the angle between $\vec{\omega}'_{\mathrm{i}}$ and the surface normal at $\mathbf{x}'$ (see Fig. 1). We allow f_{r} to model transmission as well (in this case f_{r} is the bidirectional scattering distribution function).

Sometimes it is preferable to express the reflectance equation as an integral over the domain $\mathcal{M}$ of scene surfaces (e.g. for direct lighting calculations). This form is given by

$$L_{\mathrm{r}}(\mathbf{x}' \!\rightarrow\! \mathbf{x}'') = \int_{\mathcal{M}} f_{\mathrm{r}}(\mathbf{x} \!\leftrightarrow\! \mathbf{x}' \!\leftrightarrow\! \mathbf{x}'') G(\mathbf{x} \!\leftrightarrow\! \mathbf{x}') L_{\mathrm{i}}(\mathbf{x} \!\rightarrow\! \mathbf{x}') \, dA(\mathbf{x}) \quad (2)$$

$$\text{where} \qquad G(\mathbf{x} \!\leftrightarrow\! \mathbf{x}') = V(\mathbf{x} \!\leftrightarrow\! \mathbf{x}') \cdot \frac{\cos(\theta_{\mathrm{r}}) \cos(\theta'_{\mathrm{i}})}{\|\mathbf{x} - \mathbf{x}'\|^2} \ .$$

Here A is the usual measure of surface area, θ_{r} and θ'_{i} measure the angle between $\mathbf{x} \leftrightarrow \mathbf{x}'$ and the surface normals at $\mathbf{x}$ and $\mathbf{x}'$ respectively, while $V(\mathbf{x} \!\leftrightarrow\! \mathbf{x}')$ is 1 if $\mathbf{x}$ and $\mathbf{x}'$ are mutually visible and 0 otherwise. The term $G(\mathbf{x} \leftrightarrow \mathbf{x}')$ measures the *differential throughput of a beam*[3] from $\mathbf{x}$ to $\mathbf{x}'$.

Often the incident radiance distribution is unknown, and we must solve for it. This leads to the global illumination problem: given an emitted radiance distribution L_{e}, find the equilibrium radiance distribution L satisfying

$$\begin{aligned} L(\mathbf{x}' \!\rightarrow\! \mathbf{x}'') \;&=\; L_{\mathrm{e}}(\mathbf{x}' \!\rightarrow\! \mathbf{x}'') \qquad (3) \\ &\quad + \int_{\mathcal{M}} f_{\mathrm{r}}(\mathbf{x} \!\leftrightarrow\! \mathbf{x}' \!\leftrightarrow\! \mathbf{x}'') G(\mathbf{x} \!\leftrightarrow\! \mathbf{x}') L(\mathbf{x} \!\rightarrow\! \mathbf{x}') \, dA(\mathbf{x}). \end{aligned}$$

This is known as the three-point *rendering* or *light transport equation*[5]. Equation (3) can be written concisely in operator form as $L = L_{\mathrm{e}} + \mathcal{T} L$, where $\mathcal{T}$ is the *light transport operator*. Under weak assumptions, the solution is given formally by the Neumann series

$$L = \sum_{i=0}^{\infty} \mathcal{T}^i L_{\mathrm{e}} \ . \quad (4)$$

This says that the equilibrium radiance L is the sum of emitted light, plus light that bounces once, twice, etc.

Our goal is to compute a finite set of measurements that approximately represent L. Each measurement I_p is expressed as an inner product or "weighted average" of the radiance distribution L, as modeled by the *measurement equation*:

$$I_p = \langle W_p, L \rangle = \int_{\mathcal{M} \times \mathcal{M}} W_p(\mathbf{x} \!\rightarrow\! \mathbf{x}') L(\mathbf{x} \!\rightarrow\! \mathbf{x}') G(\mathbf{x} \!\leftrightarrow\! \mathbf{x}') \, dA(\mathbf{x}) \, dA(\mathbf{x}') \quad (5)$$

where $W_p(\mathbf{x} \rightarrow \mathbf{x}')$ is the weighting function corresponding to a particular measurement I_p.

For example, the value of each pixel p in an image can be expressed in the form (5), using a weighting function W_p that is non-zero on the set of rays mapped to pixel p by the virtual lens. W_p can model arbitrary lens systems used to form the image, as well as any linear filters used for anti-aliasing.

2.2 Monte Carlo integration

We review the basic principle of MC integration, and establish some notation for the following sections. Our goal is to estimate

$$\mathcal{F} = \int_{\Omega} f(x) \, d\mu(x)$$

where $f : \Omega \rightarrow \mathcal{R}$ and μ is a measure function.

We define a *sampling technique* as an algorithm for choosing random points in the domain Ω. Let $p(x)\, d\mu(x)$ be the probability distribution of the points generated. The idea of MC integration is to generate a sample X, and then use $f(X)/p(X)$ as an estimate of $\mathcal{F}$. As long as the *sample value* $f(X)/p(X)$ is finite for all samples X, it is easy to show that this estimate is unbiased:

$$E\left[\frac{f(X)}{p(X)}\right] = \int_{\Omega} \frac{f(x)}{p(x)} p(x) \, d\mu(x) = \int_{\Omega} f(x) \, d\mu(x) = \mathcal{F} \quad (6)$$

where $E[Z]$ denotes the expected value of Z. In practice, we estimate $\mathcal{F}$ by taking several samples $X_1, \ldots, X_n$ distributed according to p, and computing

$$\mathcal{F} \approx \frac{1}{n} \sum_{i=1}^{n} \frac{f(X_i)}{p(X_i)} \ . \quad (7)$$

MC integration has one inherent drawback, which manifests itself as a tradeoff between variance and running time. Letting F be the sample value $f(X)/p(X)$, the variance of F is

$$V[F] = E[F^2] - E[F]^2 = \int_{\Omega} \frac{f^2(x)}{p(x)} \, d\mu(x) - \mathcal{F}^2 \ . \quad (8)$$

If we take n independent samples according to (7), variance is reduced by a factor of n, while running time is increased by a factor of n. This tradeoff is summarized by the *efficiency* [1, 6] of a Monte Carlo estimator,

$$\epsilon[F] = \frac{1}{V[F] \cdot T[F]}$$

where $T[F]$ is the time required to take a sample from F. The higher the efficiency, the less time required to achieve a given variance. The design of efficient estimators, often simply called *variance reduction*, is a fundamental goal of MC research.

Notice that the variance in (8) is strongly affected by the sampling distribution p—e.g. if p is proportional to f (assuming $f \geq 0$), the variance $V[F]$ is zero. Unfortunately the normalization $p = f/\mathcal{F}$ requires knowledge of $\mathcal{F}$, so this is not practical. However, by choosing a distribution p whose shape is similar to f, variance can be reduced. This idea is known as *importance sampling*[6].

On the other hand, suppose that we sample f inadequately in some region U where its value is large (i.e. $p \ll f/\mathcal{F}$). By (8) we see that samples from U can make a large contribution to the variance, even if U is relatively small. This effect is a major cause of noise in Monte Carlo images. Our primary goal is to show how this problem may be avoided, by combining samples from several distributions designed to sample well each significant region of f.

2.3 An example: glossy highlights

Consider how a distribution ray tracer might render the highlight produced by an area light source S on a nearby glossy surface (see Fig. 2). Given a viewing ray that strikes the glossy surface, there are two obvious strategies for MC evaluation of the reflected radiance, corresponding to forms (1) and (2) of the reflectance equation.

With *area* sampling, we randomly sample points on S to evaluate the integral (2). To compute the estimate (7), we must know the distribution $p(\mathbf{x})\, dA(\mathbf{x})$ of the samples—for example, they may be chosen uniformly on S with respect to surface area or emitted power. Since there is considerable freedom in choosing p, area sampling

(a) Sampling the light sources

(b) Sampling the BRDF

(c) A combination of samples from (a) and (b).

Figure 2: Sampling of glossy highlights from area light sources (Sec. 2.3, 4.1). There are four spherical light sources of varying radii and color, plus a spotlight overhead. All spherical light sources emit the same total power. There are also four shiny rectangular plates of varying surface roughness, each one tilted so that we see the reflected light sources.

Given a viewing ray that strikes a glossy surface, images (a), (b), (c) use different techniques for the highlight calculation. All images are 500 by 450 pixels.

(a) A sample direction $\vec{\omega}_{\mathrm{i}}'$ is chosen uniformly (with respect to solid angle) within the cone of directions subtended by each light source, using $n_1 = 4$ samples per pixel.

(b) $\vec{\omega}_{\mathrm{i}}'$ is chosen with probability proportional to the BRDF $f_{\mathrm{r}}(\mathbf{x}', \vec{\omega}_{\mathrm{i}}' \leftrightarrow \vec{\omega}_{\mathrm{r}}')\, d\sigma(\vec{\omega}_{\mathrm{i}}')$, using $n_2 = 4$ samples per pixel.

(c) A weighted combination of the samples from (a) and (b) is computed, using the power heuristic with $\beta = 2$.

The glossy BRDF is a symmetric, energy-conserving variation of the Phong model. The Phong exponent is $n = 1/r - 1$, where r is a surface roughness parameter, $0 < r < 1$. The glossy surfaces also have a small diffuse component. Similar results could be obtained with other glossy BRDF's.

is really a family of techniques. The glossy highlights in Fig. 2(a) were computed with an area sampling strategy.

With *directional* sampling, we estimate the integral (1) by random sampling of the incident direction $\vec{\omega}_{\mathrm{i}}'$. Evaluation of L_{i} requires casting a ray; only the rays that strike S contribute to the highlight calculation. Typically the distribution $p(\vec{\omega}_{\mathrm{i}}')\, d\sigma(\vec{\omega}_{\mathrm{i}}')$ is chosen to be proportional to $f_{\mathrm{r}}(\mathbf{x}', \vec{\omega}_{\mathrm{i}}' \leftrightarrow \vec{\omega}_{\mathrm{r}}')$ or to $f_{\mathrm{r}}(\mathbf{x}', \vec{\omega}_{\mathrm{i}}' \leftrightarrow \vec{\omega}_{\mathrm{r}}')|\cos(\theta_{\mathrm{i}}')|$. Fig. 2(b) was computed with a directional sampling strategy.

One of these strategies can have a much lower variance than the other (see Fig. 2). For example, if the light source is very small, we are unlikely to hit it with rays chosen by randomly sampling the BRDF. On the other hand, if the BRDF is nearly specular, randomly chosen points on the light source will probably not contribute significantly to the radiance reflected along the viewing ray.

In both these cases, noise is caused by inadequate sampling where the integrand is large. To understand this, notice that the integrand in the reflectance equation (2) is the product of various unrelated factors—the BRDF, the emitted radiance L_{e}, and several geometric quantities. However, the area sampling distribution used in Fig. 2(a) does not take into account the BRDF for example, while the directional sampling in Fig. 2(b) does not depend on the emitted radiance. When an unconsidered factor is dominant (e.g. a small bright light, or a shiny surface), that sampling technique will do poorly.

It is important to realize that both strategies are importance sampling techniques aimed at generating sample points on the same domain (in this case, the light source S). Area sampling chooses a point $\mathbf{x} \in S$ directly, while directional sampling chooses $\mathbf{x}$ by casting a ray in the chosen direction $\vec{\omega}_{\mathrm{i}}'$. Given a directional distribution $p(\vec{\omega}_{\mathrm{i}}')\, d\sigma(\vec{\omega}_{\mathrm{i}}')$, the corresponding area distribution $p(\mathbf{x})\, dA(\mathbf{x})$ is

$$p(\mathbf{x}) = p(\vec{\omega}_{\mathrm{i}}') \cdot \frac{d\sigma(\vec{\omega}_{\mathrm{i}}')}{dA(\mathbf{x})} = p(\vec{\omega}_{\mathrm{i}}') \cdot \frac{\cos(\theta_{\mathrm{r}})}{\|\mathbf{x} - \mathbf{x}'\|^2} \quad . \tag{9}$$

(see Fig. 1)[1]. This lets us compute the probability densities assigned by area and directional methods to the same point $\mathbf{x}$.

2.4 Our framework for variance reduction

When choosing a Monte Carlo sampling technique, we rarely know exactly what the integrand is. Instead, we have some model for the integrand, defined by a set of parameters (e.g. the BRDF, the scene geometry, etc). Given several sampling techniques to choose from, the variance of each one can change dramatically as these parameters vary.

Our main goal is to show how Monte Carlo integration can be made more robust, by constructing estimators that have low variance for a broad class of integrands. To achieve this, we must avoid

[1] One could argue that $V(\mathbf{x} \leftrightarrow \mathbf{x}')$ should appear in (9). But if $V(\mathbf{x} \leftrightarrow \mathbf{x}') = 0$, the integrand (2) is also zero, which makes $p(\mathbf{x})$ irrelevant.

insufficient sampling of each candidate integrand f where its value is large. Our approach to this problem has three steps.

First, we design a set of importance sampling distributions $p_1, \ldots, p_n$. For each region where f has the potential to be large, we try to construct a sampling distribution that approximates f well over that portion of the domain. An excellent source of these distributions is the situation in the example above, where f is a product of several unrelated functions, and each p_i is proportional to the product of a subset of these.

Next, we determine how many samples to take from each p_i. We assume this is fixed in advance, based on knowledge of f and p_i.

Finally, the integral is estimated as a weighted combination of all the sample values. The main subject of this paper is how to do this, such that the estimate is unbiased and has low variance.

3 Combining sampling techniques

We are given an integrand $f : \Omega \rightarrow \mathcal{R}$, and several importance sampling distributions $p_1, \ldots, p_n$. Our goal is to estimate $\int_\Omega f(x)\, d\mu(x)$. We assume that only two operations are available: we can take a sample from any of the distributions p_i, and we can evaluate $f(x)$, and $p_i(x)$ for any $x \in \Omega$. Each sample is assumed to be independent, i.e. we generate new random bits to control its selection.

As mentioned above, we must also decide how many samples to take from each p_i. We define c_i as the relative number of samples taken from p_i, where $\sum_i c_i = 1$. In this paper, we assume that the c_i are fixed in advance, i.e. before any samples are taken. The choice of the c_i is an interesting problem that we discuss further in Sec. 5.2.

The key ideas in this section are simple. First, notice that by drawing a fraction c_i of the samples from each p_i, the resulting group of samples has the distribution $\overline{p}(x) = \sum_i c_i p_i(x)$. We propose that the natural way to combine importance sampling techniques is to consider this *combined sample distribution* when computing the unbiased estimate $f(X)/p(X)$.

Second, we show that this method of combining samples is provably good (compared to partitioning, simple weighted combinations, etc). To justify this claim, we explore a much larger class of unbiased combination strategies, parameterized by a set of weighting functions. We then look for weighting functions that minimize the variance of the combined estimator, and show that the combination strategy above is close to optimal. This gives us confidence that our methods compare favorably with other possible techniques.

Third, we use our framework of unbiased estimators to reduce variance further in an important special case. Specifically, it is common in practice that for the particular integrand f we are given, one of the given sampling distributions is far superior to the rest (e.g. a small bright light or shiny surface in Fig. 2). We study two families of weighting functions that perform significantly better in this situation, while retaining provably good behavior in general.

3.1 The combined sample distribution

Suppose that $n_i = c_i N$ independent samples $X_{i,j}$ are taken from distribution p_i, for a total of N samples. As a group, the samples have the distribution

$$\overline{p}(x) = \sum_{i=1}^{n} c_i p_i(x) \ .$$

More precisely, $\overline{p}(x)$ is the distribution of a random variable X which is equal to each $X_{i,j}$ with probability $1/N$. We call this the *combined sample distribution*. From this point of view, the standard estimator (7) gives

$$F = \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{\overline{p}(X_{i,j})} \ . \tag{10}$$

As we will show, this is a provably good way to combine samples from several distributions. Within the framework described below, this strategy is called the *balance heuristic* (Sec. 3.3).

3.2 The multi-sample model

In this section we consider unbiased estimators that allow samples to be weighted differently, depending on which underlying distribution p_i they were chosen from. Each estimator is parameterized by a set of *weighting functions* $w_1, \ldots, w_n$, where $w_i(x)$ gives the weight associated with a sample x drawn from p_i. The *combined estimator* is given by

$$F = \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})} \tag{11}$$

where the $X_{i,j}$ are independent samples from distribution p_i, as before. For this estimator to be unbiased, we require that $\sum_i w_i(x) = 1$ for all x, since this gives

$$E[F] = \sum_{i=1}^{n} \frac{1}{n_i} n_i \int_\Omega \frac{w_i(x) f(x)}{p_i(x)} p_i(x)\, d\mu(x) = \int_\Omega f(x)\, d\mu(x) \ .$$

Think of this as a weighted sum of the estimators $f(X_{i,j})/p_i(X_{i,j})$. The weights are allowed to vary with position, but must always sum to one. For example, if at every point x all but one of the w_i are zero, we get a *simple partitioning* of the domain into n regions. This represents a heuristic such as dividing the visible hemisphere into light source regions and non-light-source regions, which are then sampled using different methods.

3.3 The balance heuristic

We now have a large parameter space over which to optimize (the space of allowable weighting functions w_i). Our goal is to minimize the variance of F by choosing the w_i appropriately. Consider the weighting functions

$$\hat{w}_i(x) = \frac{c_i p_i(x)}{\sum_j c_j p_j(x)} \ . \tag{12}$$

These $\hat{w}_i$ have the unique property that the sample value $\{\hat{w}_i(x) f(x)\} / \{n_i p_i(x)\}$ from (11) does not depend on i. Because the sample value at a particular x is the same for all underlying distributions, we call this strategy the *balance heuristic*. Substituting $\hat{w}_i$ into (11), this is simply a reformulation of the estimator (10) we obtained using the combined probability distribution.

The following theorem gives evidence that these weighting functions are good:

Theorem 1. *Let $w_1, \ldots, w_n$ be any non-negative functions with $\sum_i w_i = 1$, and let $\hat{w}_1, \ldots, \hat{w}_n$ be the weighting functions above (the balance heuristic). Let F and $\hat{F}$ be the corresponding combined estimators (11). Then*

$$V[\hat{F}] \leq V[F] + \left(\frac{1}{\min_i n_i} - \frac{1}{\sum_i n_i} \right) \mathcal{F}^2 \ .$$

See App. A for a proof. This theorem says that no choice of the w_i can improve upon the variance of the balance heuristic by more than $(1/\min_i n_i - 1/N)\mathcal{F}^2$ (recall that $\mathcal{F}$ is the quantity we are trying to estimate). This "variance gap" is very small relative to the variance caused by a poorly chosen sampling distribution, as we saw in Fig. 2. Also, the variance gap goes to zero as the number of samples increases (assuming all n_i are increased).

Furthermore, these weighting functions are practical to evaluate. The key requirement is that given a sample X_i from p_i, we must be able to evaluate $p_j(X_i)$ for all j. Any unbiased Monte Carlo algorithm must be able to evaluate $p_i(X_i)$, so this is often just a matter of reorganizing the routines that compute probabilities. The time to evaluate these probabilities is generally insignificant compared to other rendering calculations, as we show in Sec. 4.

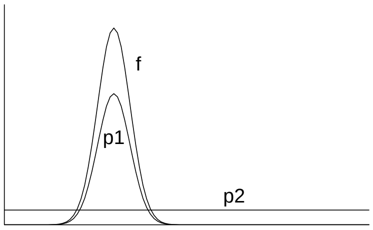


Figure 3: Two distributions for sampling the integrand.

3.4 Other weighting heuristics

Theorem 1 implies that although the balance heuristic is good, there is still room for improvement. In this section we discuss two families of heuristics that in practice often have lower variance than the balance heuristic. These heuristics satisfy $\sum_i w_i(x) = 1$ and thus give unbiased estimates.

We are motivated by the common situation where one of the p_i is an almost perfect match for f (e.g. BRDF sampling with the mirror-like surface in Fig. 2). To develop our ideas, consider the situation in Fig. 3, where f is a very peaked distribution, p_1 is proportional to f, and p_2 is the uniform distribution. Assume that we take an equal number of samples from both p_i, and form a weighted combination using the multi-sample model (11).

Since p_1 is a zero-variance importance sampling distribution ($f(X_1)/p_1(X_1) = \mathcal{F}$ is constant), the optimal weighting functions are obviously $w_1(x) \equiv 1$, $w_2(x) \equiv 0$. We cannot expect to guess this using only pointwise evaluation of the p_i and f; however, we would like to get as close to this ideal as possible.

How well does the balance heuristic perform in this situation, and how can we improve it? Consider the contributions of samples from p_1 and p_2 separately. Most samples from p_1 occur near the peak, where the weighted sample value (see (12)) is approximately equal to $\mathcal{F}$. Similarly, most samples from p_2 occur away from the peak, where their sample value is zero (because f is zero there).

So far, this is very close to optimal. However there are two effects that lead to additional variance. Occasionally a sample from p_1 occurs away from the peak (i.e. where $p_1 \gg p_2$ does not hold). In this case the weight $p_1/(p_1+p_2)$ produces a sample value smaller than $\mathcal{F}$; in an image, this shows up as dark spots. On the other hand, sometimes a sample X_2 from p_2 occurs near the peak of f. These have a weighted sample value slightly smaller than $\mathcal{F}$ (see Sec. 3.3). In an image, this shows up as occasional bright spots. However, these "spikes" are relatively small in magnitude, because a sample from p_2 contributes the same as an equivalent sample from p_1.

We present two families of heuristics that reduce variance in this important limiting case. They are variations on the balance heuristic, where the weighting functions have been "sharpened" by making large weights closer to one and small weights closer to zero. This is effective at reducing both types of noise above.

The *cutoff heuristic* modifies the weighting functions by discarding samples with low weight:[2]

$$w_i = \begin{cases} 0 & \text{if } p_i < \alpha\, p_{\max} \\ \dfrac{p_i}{\sum_j \{p_j \mid p_j \geq \alpha\, p_{\max}\}} & \text{otherwise} \end{cases} \qquad (13)$$

where $p_{\max} = \max_j p_j$. The constant α determines how small p_i must be compared to $p_{\max}$ before we assign it a zero weight.

The *power heuristic* raises all weights to a power β, and then normalizes:

$$w_i = \frac{p_i^\beta}{\sum_j p_j^\beta} \quad . \qquad (14)$$

[2] All p_i and w_i are implicitly functions of x. For simplicity we have assumed all n_i are equal; otherwise replace p_i by $n_i p_i$ everywhere.

Notice that when $\alpha = 0$ or $\beta = 1$, we get the balance heuristic. When $\alpha = 1$ or $\beta = \infty$, we get the *maximum heuristic*:

$$w_i = \begin{cases} 1 & \text{if } p_i = p_{\max} \\ 0 & \text{otherwise} \end{cases} \quad . \qquad (15)$$

This heuristic simply partitions the domain according to which distribution p_i generates samples there with the highest probability.

The advantage of these heuristics is reduced variance when one of the p_i is much better than the rest. Their performance is otherwise similar to the balance heuristic; it is possible to show they are never much worse (we give bounds in App. A, measurements in Sec. 4.1).

3.5 The one-sample model: optimality

In this section, we consider a sampling model where we our combination methods are optimal. Under this *one-sample model*, each sample is taken from a randomly selected distribution p_i. Distribution p_i is chosen with probability c_i. This idea is used in path tracing for example, where at each bounce we choose randomly between the diffuse, specular, or transmitted distributions.

Again, each estimator is parameterized by a set of weighting functions $\{w_i(x)\}$. The process of choosing a distribution, taking a sample, and computing the weighted sample value is described mathematically by the combined estimator

$$F = \frac{w_I(X_I) f(X_I)}{c_I p_I(X_I)} \ , \quad \text{where } I = \min\{i \mid U < \sum_{j=1}^{i} c_j\} \ . \qquad (16)$$

Here U is a uniformly distributed random variable on $[0,1)$, I is the index of the randomly chosen distribution, and X_I is a sample from distribution I. This estimator is unbiased as long as $\sum_i w_i = 1$.

In this case, the balance weighting strategy is optimal:

Theorem 2. *Let $w_1, \ldots, w_n$ be any non-negative functions with $\sum_i w_i = 1$, and let $\hat{w}_1, \ldots, \hat{w}_n$ be the weighting functions (12). Let F and $\hat{F}$ be the corresponding combined estimators (16). Then $V[\hat{F}] \leq V[F]$.*

4 Experiments

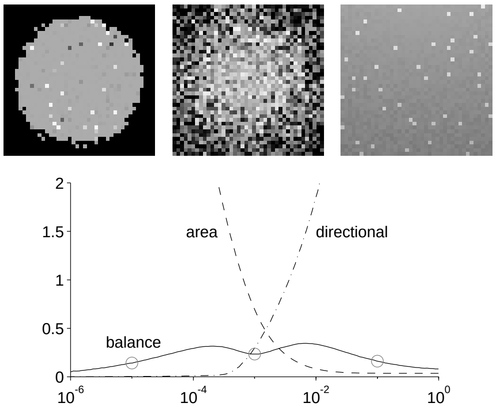
4.1 Distribution ray tracing

Our first test is the computation of glossy highlights from area light sources (see also Sec. 2.3 and Fig. 2). The area sampling technique[3] used in Fig. 2(a) works well for small light sources and rough surfaces. The directional sampling technique in (b) does well for large light sources and smooth surfaces. In (c), the power heuristic with $\beta = 2$ is used to combine both kinds of samples. This method works very well for all light source/surface combinations.

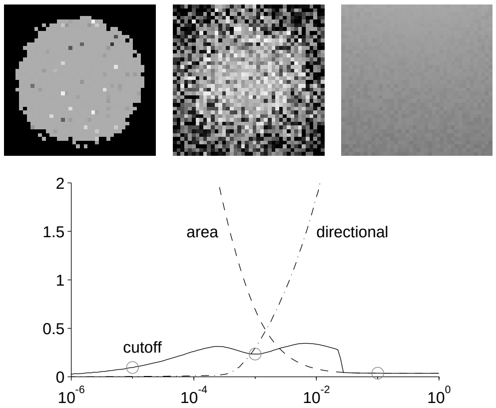
We have also measured variance numerically as a function of roughness. Fig. 4 shows the test setup, and the results are summarized in Fig. 5. Notice that all four weighting heuristics yield a variance that is close (on an absolute scale) to the minimum variance when either sampling technique is used alone. In particular, Thm. 1 guarantees that the variance σ^2 of the balance heuristic is within $\mu^2/2$ of the best input technique. The plots in Fig. 5(a) are well within that bound.

At the extremes of the roughness axis there are significant differences among the heuristics. As expected, the balance heuristic (a) performs worst at the extremes, since the other heuristics were specifically designed for the case when one sampling technique is much better than the rest. The power heuristic (c) with $\beta = 2$ works especially well over the whole range of roughness values.

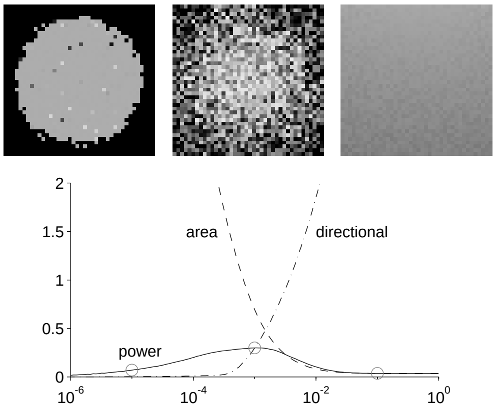
[3] Direction $\vec{\omega}_i'$ is used to compute a point x on the light source directly, rather than casting a ray to find the first visible point. Thus form (2) of the reflectance equation is used, making this an area sampling technique.



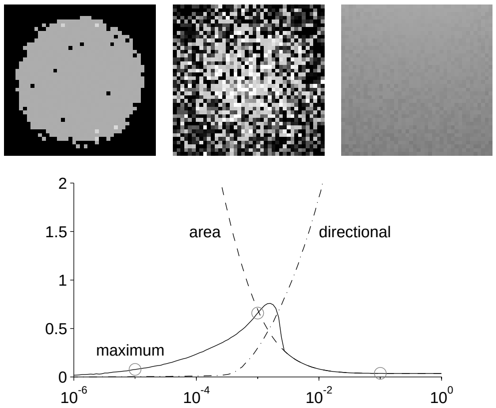
(a) The balance heuristic.



(b) The cutoff heuristic ($\alpha = 0.1$).



(c) The power heuristic ($\beta = 2$).



(d) The maximum heuristic.

Figure 5: Variance measurements for the test case in Fig. 4. Each graph plots σ / μ vs. surface roughness, where σ^2 is the variance of a single sample and μ is the mean. Three curves are shown, corresponding to the area sampling technique from Fig. 2(a), the directional sampling technique from Fig. 2(b), and a weighted combination of both sample types using the (a) balance, (b) cutoff, (c) power, and (d) maximum heuristics. The images above each graph are computed with the corresponding heuristic, for the three roughness values circled (one sample per pixel, box filter). The center pixel of these images corresponds to the viewing ray used for the variance measurements.

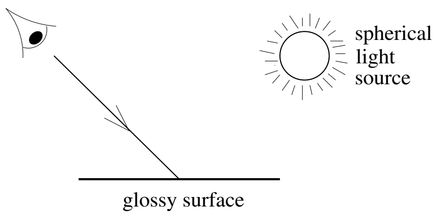


Figure 4: A scale diagram of the setup used to measure the variance of the highlight calculation. The light source occupies a solid angle of 0.063 radians. The variance for each roughness value was measured by taking 100,000 samples using the viewing ray shown.

Above the graphs we show how the variance of each method appears in an image, for three circled roughness values. Notice how the cutoff, power, and maximum heuristics reduce the "bright spot" and "dark spot" noise (Sec. 3.4) at the extremes.

Recall that to evaluate the weights at a point x, we must compute the probabilities with which *both* methods generate x. For example, if x is a point on the light source generated by (a), we find the probability $p_2(\vec{\omega}_i') \, d\sigma(\vec{\omega}_i')$ that (b) generates the direction $\vec{\omega}_i'$ pointing toward x, and convert this probability to the measure $p_2(x) \, dA(x)$ using (9). The total time spent evaluating probabilities and weighting functions in our tests was less than 5%.

4.2 Final gather

In this section we consider a simple test case motivated by *multi-pass global illumination algorithms*. These algorithms typically compute an approximate solution using the finite element method, followed by one or more ray tracing passes to replace parts of the solution that are poorly approximated or missing. For example, some radiosity algorithms use a *local pass* or *final gather* to recompute certain coefficients more accurately.

We examine a variation called *per-pixel final gather*. The idea is to compute an approximate radiosity solution, and then use it to illuminate the visible surfaces during a ray tracing pass[11, 2]. Essentially, this type of final gather is equivalent to ray tracing with many area light sources (one for each patch, or one for each link in a hierarchical solution). As with the glossy highlight example, there are two common sampling techniques. The brightest patches are classified as "light sources"[2], and are handled with an area sampling technique (e.g. samples are distributed on the light sources according to emitted power). The remaining patches are sampled by casting rays randomly into the scene (i.e. directional sampling from the point intersected by the viewing ray). If one of these rays hits a light source patch, the sample value is zero (to avoid counting those patches twice). Within our framework for combining sampling techniques, this is clearly a partitioning of the integration domain into two regions.

Given some classification of patches into light sources and non-

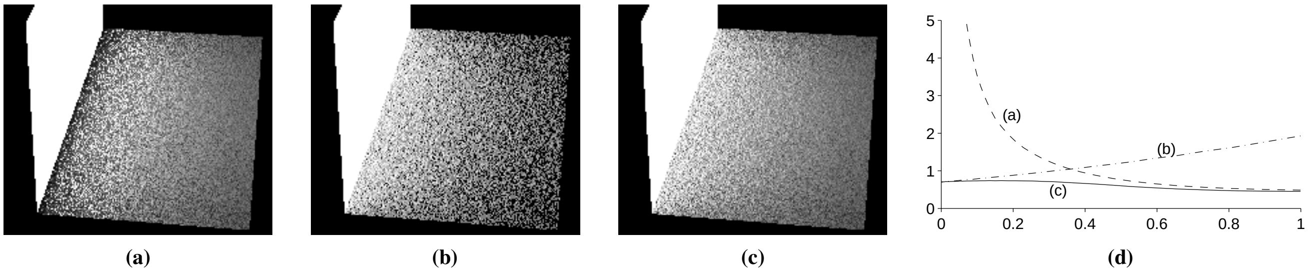


Figure 6: A simple test scene consisting of one area light source (i.e. a bright patch) and an adjacent diffuse surface. The images were computed by **(a)** sampling the light source according to emitted power, with $n_1 = 3$ samples per pixel, **(b)** sampling the hemisphere according to the *projected solid angle*[3] $\cos(\theta_i') \, d\sigma(\vec{\omega}_i')$ (see Fig. 1), with $n_2 = 6$ samples per pixel, and **(c)** a weighted combination of samples from (a) and (b) using the power heuristic with $\beta = 2$. **(d)** A plot of σ / μ (standard deviation divided by mean) as a function of distance from the light source, for $n_1 = 1$ and $n_2 = 2$.

light sources, we consider alternative ways of combining the two types of samples. To test our weighting strategies, we used the extremely simple test scene shown in Fig. 6. Twice as many samples are taken in (b) than (a); in practice this ratio would be substantially higher (i.e. the number of directional samples vs. the number of samples for any one light source).

Notice that Fig. 6(a) does poorly for points near the light source, because the sample distribution does not take into account the $1/r^2$ distance term of the reflectance equation (2). On the other hand (b) does poorly far away from the light source, when the light subtends a small solid angle. In Fig. 6(c), the power heuristic is used to combine samples from (a) and (b). As expected, this method performs well at all distances. Although (c) uses more samples (the sum of (a) and (b)), this is a valid comparison with the partitioning approach (which also uses both kinds of samples). Variance measurements are plotted in Fig. 6(d).

4.3 Bidirectional path tracing

The basic goal of Monte Carlo path tracing is to estimate the value of each pixel in an image by direct sampling of the rendering and measurement equations (Sec. 2.1). In this section, we show that by combining samples from several importance sampling techniques, this process can be made more efficient. As a source of sampling distributions, we use *bidirectional path tracing* (introduced independently in [14] and [8, 9]). We briefly overview the theory below.

To apply our methods, we must first express the value I_p of a pixel p in the standard form $\int_\Omega f(x) \, d\mu(x)$. To do this, we write out equations (3), (4), and (5) explicitly:

$$\begin{aligned} I_p &= \langle W_p, L \rangle = \langle W_p, \textstyle\sum_i \mathcal{T}^i L_e \rangle \qquad (17) \\ &= \int_{\mathcal{M}^2} L_e(\mathbf{x}_0 \!\to\! \mathbf{x}_1) G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1) W_p(\mathbf{x}_0 \!\to\! \mathbf{x}_1) \, dA(\mathbf{x}_0) \, dA(\mathbf{x}_1) \\ &+ \int_{\mathcal{M}^3} L_e(\mathbf{x}_0 \!\to\! \mathbf{x}_1) G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1) f_r(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1 \!\leftrightarrow\! \mathbf{x}_2) \\ &\qquad G(\mathbf{x}_1 \!\leftrightarrow\! \mathbf{x}_2) W_p(\mathbf{x}_1 \!\to\! \mathbf{x}_2) \, dA(\mathbf{x}_0) \, dA(\mathbf{x}_1) \, dA(\mathbf{x}_2) \\ &+ \cdots \end{aligned}$$

To write this as a single integral $\int_\Omega f(x) \, d\mu(x)$, let Ω be the set of *transport paths* of all lengths. Each transport path π of length k is a sequence $\mathbf{x}_0 \mathbf{x}_1 \ldots \mathbf{x}_k$ of points $\mathbf{x}_i \in \mathcal{M}$. The measure $d\mu(\pi)$ on Ω is defined by $d\mu(\pi) = dA(\mathbf{x}_0) \ldots dA(\mathbf{x}_k)$.[4] Finally, the integrand $f(\pi)$ is simply the appropriate term from the expansion above, for example $f(\mathbf{x}_0 \mathbf{x}_1) = L_e(\mathbf{x}_0 \!\to\! \mathbf{x}_1) G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1) W_p(\mathbf{x}_0 \!\to\! \mathbf{x}_1)$.

Path tracing algorithms can be interpreted as methods for sampling this integral directly, by generating transport paths π randomly and using the standard estimate $f(\pi)/p(\pi)$. Observe that paths where $f(\pi)$ is large satisfy two conditions: they carry a relatively large amount of light, and they have a relatively large weight in the measurement process that generates the final image. Bidirectional path tracing uses this idea to construct a family of importance-sampling techniques that trade off one property against the other.

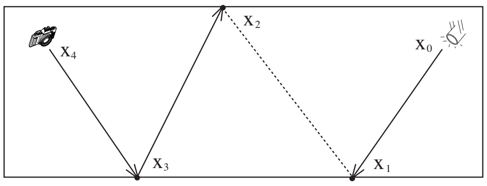


Figure 7: A transport path from a light source to the camera lens, created by concatenating two separately generated pieces.

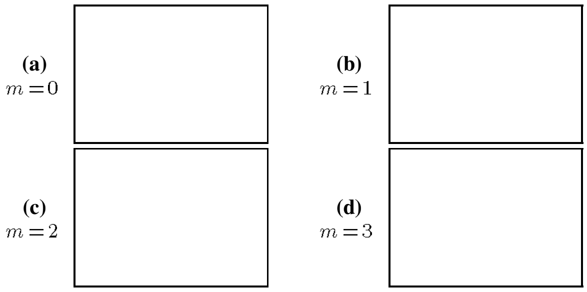


Figure 8: The four bidirectional sampling strategies for paths of length two (direct lighting). Intuitively, they can be described as **(a)** Monte Carlo path tracing with no special handling of light sources, **(b)** standard MC path tracing with direct lighting, **(c)** depositing (*splatting*) light on the image when a "photon" hits a visible surface, and **(d)** depositing light when a photon hits the camera lens.

Unlike standard path tracing, which generates transport paths by starting from the eye and following random bounces backward to the light sources, the bidirectional approach builds a path by connecting two independently generated pieces, one starting from the light sources and the other from the eye. For example, in Fig. 7 the *light subpath* $\mathbf{x}_0 \mathbf{x}_1$ is constructed by choosing a random point $\mathbf{x}_0$ on a light source (area sampling), followed by casting a random ray (directional sampling) to find $\mathbf{x}_1$. The *eye subpath* $\mathbf{x}_2 \mathbf{x}_3 \mathbf{x}_4$ is constructed by a similar process starting from a random point $\mathbf{x}_4$ on the camera lens. A complete transport path is formed by concatenating these two pieces. (This path may carry no light, for example if $\mathbf{x}_1$ and $\mathbf{x}_2$ are not mutually visible.)

This idea leads to a set of sampling techniques for transport paths. Each technique generates paths of a specific length k, by randomly generating a light subpath with m vertices, randomly generating an eye subpath with $k{+}1{-}m$ vertices, and concatenating them. In total there are $k+2$ distinct bidirectional sampling techniques for paths of length k (letting $m = 0, \ldots, k{+}1$, see Fig. 8). Each of these is really a framework for sampling rather than a specific technique,

[4] $d\mu(\pi) = dA(\mathbf{x}_0) G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1) dA(\mathbf{x}_1) \ldots G(\mathbf{x}_{k-1} \!\leftrightarrow\! \mathbf{x}_k) dA(\mathbf{x}_k)$ is another possibility—this measures the *differential throughput of a path*.

(a) The weighted contribution that each bidirectional sampling technique makes to image (b)

(b) Combines samples from all the bidirectional techniques

(c) Standard path tracing using the same amount of work

Figure 9: The scene contains a spot light, a floor lamp, a table, and a big glass egg. Image **(b)** uses the power heuristic (with $\beta = 2$) to combine samples from a family of bidirectional path tracing techniques, whose weighted contributions are shown in **(a)**. Row i shows techniques that sample transport paths of length $i+1$; the m-th image uses the distribution $p_{i+1,m}$ (see Sec. 4.3). Images in row i have been over-exposed by i f-stops so that details can be seen.

since the paths generated depend on the distributions used to choose each vertex (area sampling for the first vertex of each subpath, usually directional sampling for the rest). These methods can be very diverse, e.g. sophisticated direct lighting techniques can be used to choose the first vertex of the light subpath.

Each technique defines a probability distribution $p_{k,m}(\pi)\,d\mu(\pi)$ on paths of length k. We can compute $p_{k,m}(\pi)$ explicitly by multiplying the probabilities $p(\mathbf{x}_i)\,dA(\mathbf{x}_i)$ with which the individual vertices were generated. Vertices that were chosen using a directional distribution $p(\vec{\omega})\,d\sigma(\vec{\omega})$ can be converted to the area measure using (9). To see why these distributions are good candidates for importance sampling, consider the integrand (17) for paths of length k. It is a product of many unrelated functions: L_{e}, W_p, k different G factors, and $k-1$ different f_{r} factors. Each bidirectional technique includes a different subset of these factors in its sampling distribution; among them, we are more likely to generate paths that contribute significantly to the image.

We now have all the tools to combine samples from these techniques using the methods of Sec. 3: we can take a sample from any of the distributions $p_{k,m}$, and given any path π of length k we can evaluate $f(\pi)$ and $p_{k,m}(\pi)$.

Fig. 9 shows a scene that we used to test these ideas. Diffuse, glossy, and pure specular surfaces are present. Transport paths of lengths up to $k=5$ were sampled using the bidirectional distributions $p_{k,m}$ described above. For efficiency, we randomly generate maximum-length eye and light subpaths in pairs. We then take samples from all $p_{k,m}$ by joining each prefix of the light subpath to each suffix of the eye subpath. For example, to sample $p_{2,1}$ we concatenate the first vertex of the light subpath and the last two vertices of the eye subpath. Each such group of samples is dependent, but this does not appear to significantly affect our results. Another important optimization reduces the number of visibility tests between the eye and light subpaths, by using Russian roulette [6] to randomly suppress small potential contributions without adding bias.

The final image in Fig. 9(b) was created by combining samples from all distributions using the power heuristic (with $\beta=2$). The image is 500 by 500 with 25 samples per pixel. The weighted contribution from each technique is shown in the pyramid in Fig. 9(a). The pyramid does not show the complete set of sampling techniques; paths of length one are not shown because the light sources are not directly visible, and one column has been stripped from the left and right sides of each row because these images are virtually black (i.e. the weighted contributions are very small).

Observe the caustics on the table, both directly from the spotlight and indirectly from reflected light on the ceiling. The unusual caustic pattern to the left is caused by the square shape of the spotlight's emitting surface. Notice that some effects, such as caustics and specular reflections, get their contributions almost entirely from one sampling technique. This says that the other techniques are very poor estimators of these contributions.

For comparison, Fig. 9(c) shows standard MC path tracing with 56 samples per pixel (the same computation time as Fig. 9(b)). Direct lighting was used on all paths except for caustics, which were rendered by following paths right back to the light sources (the caustics would otherwise not be visible).

5 Discussion

5.1 Conclusions

As we have shown, our methods for combining sampling techniques can substantially reduce the variance of Monte Carlo rendering calculations. These techniques are practical, and the additional cost is small—less than 10% of the time in our tests was spent evaluating probabilities and weighting functions. We also have strong bounds on their performance relative to other combination strategies.

Overall, we found that the power heuristic (with $\beta=2$) gave the best results. It is similar to the balance heuristic in general, but has significantly lower variance when one of the p_i is a good match for f. When none of the given sampling distributions is a good match for f (e.g. Fig. 6), the differences among the various weighting strategies are small.

5.2 Choosing the number of samples

First, observe that no strategy is greatly superior to that of simply setting all c_i equal. If we are allocating N samples among n sampling techniques, it is easy to show that

$$V[\hat{F}] \le nV[F] + \frac{n-1}{N}\mathcal{F}^2$$

where $\hat{F}$ uses the balance heuristic with all c_i equal, and F uses any unbiased weighting functions and c_i (satisfying $\sum_i w_i = 1$ and $w_i \equiv 0$ if $c_i = 0$). Thus, changing the c_i can improve the variance by at most a factor of n, plus a small additive term. On the other hand, a poor choice of the w_i (e.g. a poor partitioning of the integration domain) can increase variance by an arbitrary amount.

Also, there are situations where the c_i are naturally constrained. For example, in bidirectional path tracing it is more efficient to take one sample from all distributions at once (Sec. 4.3). In the glossy highlights example, the c_i are constrained because the samples are used for other purposes (direct lighting samples for the diffuse component, and directional samples for glossy reflections of objects other than light sources). Often these other purposes will dictate the number of samples taken. In this case, by taking a weighted combination of both types of samples we can reduce the variance of the highlight calculation essentially for free.

5.3 Comments on direct lighting

The examples of Sec. 4.1, 4.2 are essentially direct lighting problems. They differ only in the terms of the reflectance equation that cause high variance—the BRDF, the $1/r^2$ distance term, or the emitted radiance distribution L_e.

In Sec. 4.2, we used a simple light source sampling technique. Although there are more sophisticated techniques for direct lighting[13], it can still be useful to combine several kinds of samples. Observe that *any* strategy for sampling a group of patches as light sources induces some probability distribution on the patch surfaces. Since these strategies are always approximations, some factors of the reflectance equation (2) will not be approximated well. In parts of the scene where these omitted factors become dominant, simple directional sampling can be more efficient. By combining both kinds of samples, we can make such strategies more robust.

Shirley and Wang[12] also compare directional and area sampling techniques for glossy highlights (Sec. 4.1). They analyze a specific Phong-like BRDF and light source sampling method, and derive an expression for when to switch from one to the other (as a function of surface roughness and light source solid angle). In contrast, our methods work for general BRDF's and sampling techniques, and can combine samples from any number of distributions.

5.4 Approximating the weighting functions

The models in Sec. 3 assume that given a sample X_i from distribution p_i, we can compute $p_j(x)$ exactly for all other j. Sometimes this is problematic—e.g. $p_j(x)$ may be expensive or complicated to evaluate. More difficulties arise when a sampling technique p_j uses random numbers that cannot be determined from the resulting sample point x. For example, some direct lighting strategies[13] generate several candidate sample points x_i, and then choose one randomly. Given an arbitrary point x, it is difficult to evaluate

$p_j(x)$ because this probability depends on information other than the sample location x itself.

The easiest way to handle these problems is to recall that the results are unbiased as long as $\sum_i w_i(x) = 1$. When computing the w_i, it is perfectly reasonable to use an approximation p'_j of the true probabilities p_j. This will give unbiased results even if the approximations p'_j are poor, as long as they are consistently used (i.e. $p'_j(X_i)$ does not depend on i). Of course, poor approximations may lead to increased variance. Note that $p_i(X_i)$ must always be evaluated exactly in (11) to avoid bias; however this is required of any unbiased Monte Carlo algorithm.

5.5 Future work

We would like to explore other applications where it makes sense to use several sampling distributions. Even within the framework of global illumination, there are many such problems. For example, bidirectional path tracing can be used to estimate the coefficients of basis functions defined on scene surfaces (let W_p in (5) be the dual basis function). This is an unexplored alternative to particle tracing models for Monte Carlo radiosity, and may be an effective solution to the problem of patches that do not receive enough particles.

We think that there is great potential for designing better sampling distributions—we hope that the existence of good methods to combine the samples will spur further work in this area. Again, global illumination provides a rich framework, because of the complexity of the domain and the integrand.

Another interesting problem is how to choose the c_i. One research area is the derivation of *a priori* rules for specific applications (similar to [12]). Another goal is to find strategies for the general case; adaptive methods seem promising here. Note that adaptive methods can introduce bias, unless two-stage sampling is used [7].

Acknowledgments

Thanks to Pat Hanrahan, Marc Levoy, Luanne Lemmer, and the anonymous reviewers for helpful comments that improved the presentation. Discussions with John Tukey were also useful. Thanks to Bill Kalsow for answering lots of questions about Modula-3 [10], the language we used for our rendering system. This research was supported by the National Science Foundation (CCR-9215219), the Digital Systems Research Center, and the Digital External Research Program.

References

[1] J. Arvo and D. Kirk. Particle transport and image synthesis. *Computer Graphics (SIGGRAPH '90 Proceedings)*, **24**, 63–66 (1990).

[2] S. Chen, H. Rushmeier, G. Miller, and D. Turner. A progressive multipass method for global illumination. *Computer Graphics (SIGGRAPH '91 Proceedings)*, **25**, 165–174 (1991).

[3] M. Cohen and J. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, 1993.

[4] R. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (SIGGRAPH '84 Proceedings)*, **18**, 137–146 (1984).

[5] J. Kajiya. The rendering equation. *Computer Graphics (SIGGRAPH '86 Proceedings)*, **20**, 143–150 (1986).

[6] M. Kalos and P. Whitlock. *Monte Carlo Methods, Volume I: Basics*. J. Wiley, New York, 1986.

[7] D. Kirk and J. Arvo. Unbiased sampling techniques for image synthesis. *Computer Graphics (SIGGRAPH '91)*, **25**, 153–156 (1991).

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

[9] E. Lafortune, Y. Willems. A theoretical framework for physically based rendering. *Computer Graphics Forum*, **13**(2), 97–108 (1994).

[10] G. Nelson, editor. *Systems Programming with Modula-3*. Prentice Hall, 1991. An implementation of Modula-3 is available at http://www.research.digital.com/SRC/.

[11] H. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. Doctoral Thesis, Cornell University, May 1988.

[12] P. Shirley and C. Wang. Distribution ray tracing: theory and practice. *Proceedings of the Third Eurographics Workshop on Rendering*, Bristol, England, 33–44 (1992).

[13] P. Shirley, C. Wang, and K. Zimmerman. Monte Carlo Techniques for Direct Lighting Calculations. *ACM Transactions on Graphics*, to appear.

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

Appendix A Proofs

Proof of Thm. 1: The variance is

$$\begin{aligned} V[F] &= V\left[\sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} F_{i,j}\right] \quad \text{where } F_{i,j} = \frac{w_i(X_{i,j}) f(X_{i,j})}{p_i(X_{i,j})} \\ &= \sum_{i=1}^{n} \frac{1}{n_i^2} \sum_{j=1}^{n_i} E[F_{i,j}^2] - \sum_{i=1}^{n} \frac{1}{n_i^2} \sum_{j=1}^{n_i} E[F_{i,j}]^2 \end{aligned}$$

where the covariance terms are zero because the $X_{i,j}$ are sampled independently. We bound the two terms separately. For the first term, we get

$$\sum_{i=1}^{n} \frac{1}{n_i^2} \sum_{j=1}^{n_i} E[F_{i,j}^2] = \int_\Omega \sum_{i=1}^{n} \frac{w_i^2(x) f^2(x)}{n_i p_i(x)}\, d\mu(x) \ .$$

Using the method of Lagrange multipliers, we minimize the integrand independently at each point x subject to the condition $\sum_i w_i = 1$. Noting that $f^2(x)$ is a constant and dropping x from our notation, we must minimize

$$\sum_i \frac{w_i^2}{n_i p_i} + \lambda \left(\sum_i w_i - 1 \right) \ .$$

Setting all $n+1$ partial derivatives to zero, we obtain $w_i = \hat{w}_i$ (12). Thus no other weighting strategy can reduce this term further.

The second term makes a negative contribution to the variance, so we will prove an upper bound $\mathcal{F}^2/\min_i n_i$ for the w_i and a lower bound $\mathcal{F}^2/\sum_i n_i$ for the $\hat{w}_i$. Letting $\mu_i = E[F_{i,j}]$ (this is independent of j), for the upper bound we have

$$\sum_{i=1}^{n} \frac{1}{n_i^2} \sum_{j=1}^{n_i} \mu_i^2 = \sum_{i=1}^{n} \frac{1}{n_i} \mu_i^2 \le \frac{1}{\min_i n_i} \sum_{i=1}^{n} \mu_i^2 \, .$$

Since $\sum_i \mu_i = \mathcal{F}$, we have $\max_i \mu_i \le \mathcal{F}$, and thus $\sum_i \mu_i^2 \le \mathcal{F}^2$ which proves the upper bound. The lower bound $\sum_i \hat{\mu}_i^2/n_i \ge \mathcal{F}^2/\sum_i n_i$ is easily proven with Lagrange multipliers. ∎

Proof of Thm. 2: Because $\mathcal{F}^2$ is fixed in (8), it is enough to minimize the second moment $E[F^2]$. We have

$$E[F^2] = \int_\Omega \sum_{i=1}^{n} \frac{w_i^2(x) f^2(x)}{c_i p_i(x)}\, d\mu(x) \ ,$$

which is virtually identical to the second moment term that we minimized in the proof of Thm. 1. ∎

We also present worst-case bounds for the weighting heuristics from Sec. 3.4. The bounds have the form

$$V[\hat{F}] \le c V[F^*] + \left(\frac{1}{\min_i n_i} - \frac{1}{\sum_i n_i} \right) \mathcal{F}^2 \ ,$$

where $\hat{F}$ uses the indicated heuristic, and F^* uses the (unknown) optimal weighting functions. For the cutoff heuristic, we can show $c = 1 + \alpha(n-1)$, while for the power heuristic we can show

$$c = 1 + \frac{1}{\beta}((n-1)(\beta-1))^{1-1/\beta} \ .$$

When $\beta = 2$, we can prove the stronger bound $c = \frac{1}{2}(1 + \sqrt{n})$.