

Bidirectional Importance Sampling for Illumination from
Environment Maps

by

David Burke

B.Sc., University of Prince Edward Island, 2002

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Department of Computer Science)

We accept this thesis as conforming
to the required standard

.....
.....
.....
.....

THE UNIVERSITY OF BRITISH COLUMBIA

October 22, 2004

© David Burke, 2004

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature) _____

Department of Computer Science

The University Of British Columbia
Vancouver, Canada

Date _____

Abstract

Image-based representations for illumination are able to capture complex real-world lighting that is difficult to represent in other forms. Current importance sampling strategies for image-based illumination have difficulties in the case where both the environment map and the surface BRDF contain important high-frequency detail, for example, when a specular surface is illuminated by an environment map containing small light sources.

We introduce the notion of *bidirectional importance sampling*, in which samples are drawn from the product distribution of both the surface reflectance and the energy in the environment map. Although this makes the sample selection process more expensive, we show significant quality improvements over traditional importance sampling strategies for the same compute time.

Contents

Abstract	ii
Contents	iii
List of Figures	v
Acknowledgements	vii
1 Introduction	1
2 Related Work	3
2.1 Environment Mapping	4
2.2 Prefiltered Environment Mapping	5
2.3 Alternate Bases and Precomputed Transport	6
2.4 Importance Sampling and Point Relaxation	7
2.5 Combined Importance Sampling	9
3 Monte Carlo Rendering	11
3.1 Environment Mapping	12
3.2 Monte Carlo Estimators and Sampling	15
3.2.1 Monte Carlo Integration	15
3.2.2 Sampling a General Distribution	19
3.3 Sampling the Environment Map	24

3.4	Improving Shadows: Area-weighted Lighting	27
3.5	Sampling from the BRDF	30
4	Sampling the Product Distribution	35
4.1	Bidirectional Importance Sampling	35
4.2	The Challenge of Bidirectional Sampling	42
4.3	Sample Generation Through Rejection	43
4.4	Sample Generation Through Sampling-Importance Resampling	47
5	Results	51
6	Conclusions	63
	Bibliography	64
A	Details on the Alias Method for Sampling	70

List of Figures

2.1	Environment mapping	4
3.1	The rendering equation illustrated	13
3.2	The transformation method: sampling via an inverted CDF .	21
3.3	Motivation for the alias method	23
3.4	The alias method	24
3.5	Sampling a 2D intensity image	27
3.6	A quantized intensity map	29
3.7	Two interpretations for the BRDF	32
3.8	The Phong BRDF	33
3.9	Sampling from the cosine term/Phong BRDF product	34
4.1	Angular maps of the EM, BRDF and product distribution . .	40
4.2	Angular maps contrasting sampling methods	41
4.3	Rejection sampling illustrated	43
4.4	Rejection sampling the product distribution	44
4.5	Our SIR method for sampling from the product distribution .	49
5.1	David in Grace Cathedral. Phong exp 10, $k_s = 1.0$, $k_d = 0.0$.	53
5.2	David in Grace Cathedral. Phong exp 50, $k_s = 1.0$, $k_d = 0.0$.	54
5.3	David in Grace Cathedral. Phong exp 50, $k_s = 0.5$, $k_d = 0.5$.	55

5.4	David in St. Peters. Phong exp 50, $k_s = 0.5$, $k_d = 0.5$	57
5.5	David in Grace Cathedral. Diffuse Phong ($k_s = 0.0$, $k_d = 1.0$)	58
5.6	David in Grace Cathedral, unzoomed	59
5.7	Comparison with Veach and Guibas(1)	60
5.8	Comparison with Veach and Guibas(2)	61
5.9	Comparison between SIR and converged image	62

Acknowledgements

Firstly, I'd like to thank everyone who has contributed ideas, discussion and proof-reading to this work; in particular, Wolfgang Heidrich, Abhijeet Ghosh, Hendrik Kueck, Simon Clavet, Nando de Freitas, and James Slack.

I also want to thank my loved ones for putting up with me: my mother Elizabeth, my father Wayne, my sister Rebecca, and my dear Amy.

Also, a big shout-out to Imager Small and everyone else who've kept me sane via games of Quake 3, UT2004, Panel de Pon, Bomberman, Netris, and general lab craziness. In no particular order: Vlad "Vlady" Kraevoy, Ben "Tron" Forsyth, Ritchie "spankbot" Argue, Hendrik "Space Cowboy" Kueck, Eric "*nipple*" Brochu, James "evil" Slack, Fred "Cheerleader" Kimberly, David "Toasty" Pritchard, Simon "Frog" Clavet, Kristian "zirkus-direktor" Hildebrand, Abhijeet Ghosh, Roger "Butt Slammer" Tam, Lisa Streit, Matt Trentacoste, Lewis "Wurmangst" Johnson, and the ever-snipey "Grumpy old man" (you know who you are).

Lastly, I'd like to thank der kommuners, past and present, for their friendship and support: Eric Brochu, Eddy Boxerman, Hendrik Kueck, David Pritchard, and Tyson Brochu.

Chapter 1

Introduction

Image-based representations for illumination, such as environment maps (EMs) and light fields, have received considerable attention in recent years. The primary reason for this attention is that images can capture complex real-world illumination which is difficult to represent in other forms.

When integrating image-based lighting, such as environment maps, into a rendering system, the employment of a good sampling strategy for illumination is an important issue. While several researchers have recently worked on this problem, the basic approach that has been taken by most work is an importance sampling strategy based on the energy distribution in the environment map. Unfortunately, such an approach performs poorly for highly specular surfaces, since samples chosen this way have a low probability of lying within the specular lobe. Similarly, if importance sampling is performed based solely on the reflectance function (BRDF) of the surface, then the sampling will not perform well for high-frequency environment maps.

This thesis introduces *bidirectional importance sampling*, a method that samples visibility according to an importance derived from the product of BRDF and environment map illumination. The challenge of this approach is to develop an efficient means of drawing samples from this product distribution. The task is complicated by the fact that the 2D BRDF slice varies from point to point on the surface. Furthermore, the environment map is

usually represented relative to a global coordinate frame, while the BRDF is expressed in a local frame that changes with surface orientation. For these reasons, precomputation approaches, such as those that employ a table of the product distribution, are infeasible.

We present two solutions to this problem. The first is a combined rejection sampling and importance sampling scheme that initially generates samples according to the product distribution with rejection sampling, and then uses the resultant samples to estimate local illumination with standard importance sampling. The second approach uses a technique called sampling-importance resampling. In this method, samples are generated from either the lights or the BRDF and then resampled to distribute them according to the product distribution.

While both methods increase the cost of sample generation, we demonstrate significant quality improvements for the same compute time under the assumption of BRDF representations that support efficient evaluation and sampling. Our method creates samples on the fly and does not require expensive precomputation.

The remainder of the thesis is structured in the following manner. In Chapter 2 we review some of the relevant work in environment map rendering. In Chapter 3 we give an introduction to sampling and Monte Carlo rendering, before describing our approach in Chapter 4. Chapter 5 discusses the results of our methods, and we conclude in Chapter 6.

Part of this work has appeared in a different form. Our rejection sampling technique appeared as a Siggraph technical sketch, published with Abhijeet Ghosh and Wolfgang Heidrich [4].

Chapter 2

Related Work

All rendering systems, both global and local, must at some point compute the direct illumination in the scene; that is, the amount of light arriving at surfaces from primary light sources. Unfortunately, this task remains expensive; as such, much research effort has focused on the development of more efficient techniques for computing the direct lighting. Little work deals directly with sampling from environment maps for global illumination applications. Debevec [6] describes the use of high dynamic range environment maps in global illumination in the context of Ward's RADIANCE package [47]; however, no specialized sampling strategies are employed to reduce variance for this specific light representation.

This thesis focuses on direct lighting from environment maps. In this chapter, we present an overview of previous work on computing direct illumination from environment maps. After an introduction to environment mapping in Section 2.1, we address the approach of environment map pre-filtering in Section 2.2. Section 2.3 summarizes work in alternate bases and precomputed light transport, and Section 2.4 discusses point set relaxation techniques.

2.1 Environment Mapping

Environment mapping was first proposed by Blinn and Newell [3]. Environment mapping operates under the assumption of a distant environment: when the distance between an object and its environment is large compared to the object's size, incoming illumination from the environment can be considered directional. That is, the light sources in the environment can be thought of as being located at infinity – for example, think of the daytime sky lighting the surface of the earth. Under this assumption, the incoming illumination at the object can be efficiently stored in a 2D image called an *environment map* (EM). During rendering, the illumination can be indexed by, for example, the spherical coordinates of the reflected ray, as illustrated in Figure 2.1.

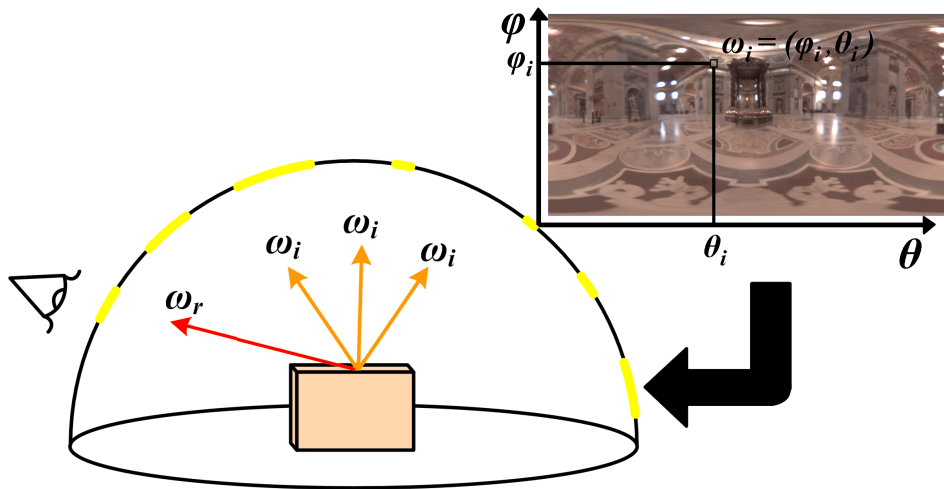


Figure 2.1: Environment mapping. Incoming light is assumed directional, and can thus be sampled and stored in an image. Rendering involves computing the amount of light reflected towards the viewer (direction ω_r); this requires considering light incoming from all directions (ω_i 's).

2.2 Prefiltered Environment Mapping

When computing shading for a non-mirrorlike object, it is generally necessary to consider multiple EM light directions. To mitigate this, the notion of *prefiltering* the environment maps was suggested, first by Greene [12]. The idea is for each EM entry to represent not just the incoming radiance from a single direction, but also the radiance from a larger region that has been integrated against the BRDF lobe of the surface. With this approach, one need only perform a single EM look-up to compute approximate shading for the surface patch. In the case of Greene, the EM was prefiltered with the surface’s diffuse reflectance. Heidrich and Seidel [15] extended the approach to glossy BRDFs by prefiltering specular environment maps with radially symmetric Phong lobes. Kautz and McCool [20] addressed other isotropic BRDFs by developing a representation for more complex BRDFs that can still be used to prefilter environment maps.

A drawback of these approaches is the precomputation required. This drawback was partially reduced by the hierarchical prefiltering algorithm of Kautz et al. [21] which significantly accelerates the preprocessing time. However, the situation remains difficult when the scene contains multiple BRDFs or a surface with a spatially-varying BRDF. In these cases, the environment must be prefiltered separately against each BRDF model, resulting in large computational and storage costs. Also, efficient representations do not exist for many types of BRDFs, particularly anisotropic models or measured BRDFs.

Regardless of whether or not the environment map has been prefiltered, when rendering with a single EM lookup, one does not take visibility or occlusion into account, either during preprocessing or rendering. Thus, one

cannot produce images that contain such effects as shadows and multiple scattering.

2.3 Alternate Bases and Precomputed Transport

In some recent work, the illumination and/or BRDF are projected into finite bases, such as spherical harmonics (SH) and wavelets. These finite bases support efficient compression, as only a small number of terms are necessary for representing the source functions to a reasonable fidelity.

For example, Ramamoorthi and Hanrahan [33] express the lighting in a spherical harmonics basis and then use only the first nine coefficients to render diffuse objects. However, because ringing artifacts occur when high-frequency lighting is expressed in the SH basis, they must heavily down-sample and blur their environment maps before applying their technique. This also implies their technique is unusable for high dynamic range environments. They extend their work in [34] from Lambertian BRDFs to more complex isotropic BRDFs.

Sloan et al. [39] introduces precomputed radiance transfer, where visibility and self-transfer effects are taken into account during an offline process. The precomputed radiance is then expressed in spherical harmonics so that it can be compressed and used for interactive rendering. Like Ramamoorthi, their technique is limited to very low frequency lighting. Ng et al. [29] and Liu et al. [27] factorize with wavelet representations which better preserve frequency detail in the environment, albeit at resolutions that are considerably lower than our technique can handle. Ng et al. [30] presents a formulation for factorizing the rendering equation into separate terms, each of which is expressed in a wavelet basis. Their rendering times are non-interactive.

Conceptually, what these techniques do is essentially render the scene from all viewpoints and then store a compressed version of these images for access at render time. As such, all flavours of factorization and precomputed transport require hours of preprocessing and huge amounts of storage, often in the gigabytes. What is more, the preprocessing must be repeated every time new geometry or material properties are introduced. Such massive precomputation is prohibitive in most every setting, especially in animation when geometry is constantly changing. Our approach can handle high resolution all-frequency lighting and requires virtually no precomputation. Geometry, environment maps, and surface materials can be substituted on the fly at essentially no cost. While our rendering times are longer, at perhaps twenty seconds instead of two seconds, our algorithm comes with the flexibility of requiring no precomputation.

2.4 Importance Sampling and Point Relaxation

A common approach for rendering from environment maps is a technique called *importance sampling*, where the full illumination integral is approximated by considering only a small set of sample directions. These directions are chosen according to their contribution to the reflected radiance. One way to achieve this is to approximate the EM with a small number of point lights representing bright regions in the environment. For example, Kollig and Keller [23] propose a scheme that is based on relaxing a point set until its distribution fits the energy distribution of the environment map. Relaxation schemes attempt to serve as a compromise between light intensity and reasonable spatial coverage. Kollig's algorithm is similar to earlier work by Deussen et al. [8] on distributing stipples for digital halftoning. There is

also LightGen [5], which seeks to reduce the diffuse illumination from high dynamic range environments into a set of directional light sources using a similar relaxation technique.

The problem with point relaxation methods is that they are not proven to converge in 2D, and they are often time consuming for a sufficiently large number of points. As a consequence, techniques that use relaxation precompute only a single relaxed point set, and then use this set to shade all points in the scene. Ostromoukhov et al. [31] presented a technique for distributing 2D point samples which is much faster than relaxation-based approaches, and also appears to produce a good spatial distribution for the points.

Agarwal et al. [1] introduced a sampling method for environment maps where the sampled importance takes into account both the energy in the environment map and the solid angle separating the samples. This way, close clustering of environment map samples is avoided, which reduces redundant shadow tests. Like Agarwal et al., we have included the solid angle in the importance term for the lights; see Section 3.4.

To distribute their samples, Agarwal et al. use a point relaxation algorithm which is different from that of Kollig and Keller; however, a downside of their approach is that the environment map has to be quantized to generate the distribution.

As an extension to their work, Agarwal et al. sort the samples for each shading operation by the magnitude of their contribution to the final illumination. They sample all point lights deterministically, in order of contribution, until the contrast that the remaining lights can add falls below some threshold. This use of the product of BRDF and environment map value is a step towards our approach of drawing samples according to an

importance that is the product of BRDF and light distribution. However, because they precompute a point set via relaxation, they are limited to choosing directions only from a set of precomputed samples, thereby introducing quantization artifacts (aliasing and banding). Also, the sorting introduces bias. In contrast, our methods are asymptotically unbiased, and can create samples efficiently on the fly, so that different sampling patterns can be used throughout the scene.

Secord et al. [36] describes a fast algorithm for distributing stipples according to image intensities based on pre-integrating and inverting the probability density function (PDF) derived from image intensities. This same approach applies to drawing samples efficiently from environment maps, which are themselves images. We use a variant of this method in our current work; see Section 3.3 for details.

2.5 Combined Importance Sampling

Also similar in spirit to our approach is the work by Veach and Guibas [44] in which they combine sampling from the light sources and sampling from the BRDF to reduce the variance of the results. Prior to rendering, a decision is made as to how many samples to draw from each distribution. The general strategy is to draw more samples from the distribution with higher variance. Unfortunately, employing combined sampling results only in a blend between the variances of the individual distributions (see Section 4.1). Our method goes further than their work to reduce variance in that we sample *directly* from the product distribution, rather than just mixing samples taken from the individual distributions.

The work of Szecsi et al. [42] selects sample weights with correlated

sampling, a technique that seeks to separate Monte-Carlo estimators into low and high variance components and combine samples from the components to reduce the overall variance. Unfortunately, their results are not convincing. Furthermore, as with Veach and Guibas, decisions must still be made a priori as to how samples are combined. Our approach is mathematically straightforward and allows for direct sampling of the product distribution without any arbitrary guesswork.

Chapter 3

Monte Carlo Rendering

The general goal of global illumination is the computation of all light interactions in a scene to generate realistic images. Many global illumination algorithms, such as radiosity [11], irradiance gradients [47], and photon mapping [16] consist of two phases: first distributing radiance from the light sources onto surfaces, and then gathering the local radiances to construct an image for a specific viewpoint. A thorough distribution step will permit for a fast gather step, one which can produce quality images in a handful seconds. To date, computing the direct illumination remains the expensive aspect of global illumination; as such, much research effort has focused on coming up with efficient techniques for direct lighting.

The work described in this thesis is an approach for tackling the direct illumination problem in the case where both the light sources and the materials are complex – that is, of arbitrarily high frequency. This chapter builds a foundation for the presentation of our methods. Section 3.1 introduces the mathematics of direct illumination rendering from an environment map. In Section 3.4 extends the method for light sampling to include area considerations. Section 3.2 presents some basic notions of sampling and Monte Carlo integration. From these, we construct methods in Sections 3.3 and 3.5 for drawing samples according to the environment map and BRDF.

3.1 Environment Mapping

While following the discussion in this section, refer to Figure 3.1 for a diagram of the geometry involved. The convention is to denote directions with ω . The subscripts i and r denote quantities that are incoming and reflected (outgoing), respectively; for example, ω_i and ω_r .

The integral for direct illumination is given by the following expression:

$$L_r(x, \omega_r) = \int_{\Omega} f_r(x, \omega_i \rightarrow \omega_r) L_i(\omega_i) (n(x) \cdot \omega_i) V(\omega_i) d\omega_i. \quad (3.1)$$

Equation 3.1 describes the amount of radiance (light) being reflected at a surface point x towards some outgoing direction ω_r . This reflected radiance is computed by integrating the *incoming radiance* $L_i(\omega_i)$ over the hemisphere centered at x of incoming directions ω_i . The reflectance properties of the surface are described with the *bidirectional reflectance distribution function* (BRDF), denoted $f_r(x, \omega_i \rightarrow \omega_r)$. The BRDF is a function indicating how much of the light incident to x along the direction ω_i is reflected away towards ω_r . The $n(x) \cdot \omega_i$ term is known as the *cosine term*; it is an area foreshortening term that scales the incoming radiance $L_i(\omega_i)$ based on orientation of the surface normal $n(x)$ with respect to the incoming light direction ω_i . $V(\omega_i)$ is the *visibility term*, which in our case is binary; that is, $V(\omega_i)$ is 1 if the surface point x can see the light along ω_i and 0 otherwise.

In this work, we express the incoming illumination L_i with an *environment map* (EM), a discrete representation of the continuous hemisphere of lights surrounding x . The environment map in our case is a latitude-longitude map – that is, a 2D image that is projected onto the hemisphere. Recall that environment mapping operates under the assumption of distant illumination; that is, each pixel in the EM is treated as a point light source

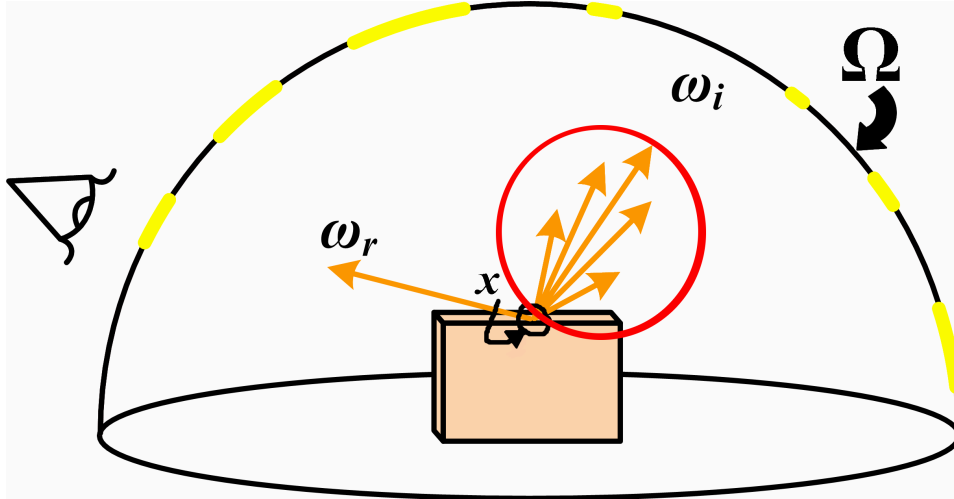


Figure 3.1: Shading for surface point x is computed by considering the light from all incoming directions ω_i of the hemisphere Ω . Each light $L(\omega_i)$ is weighted by the BRDF (circled in red), describing the probability that the surface at x reflects that light towards the viewing direction ω_r .

corresponding to the incoming light summed over a differential patch on the hemisphere. An EM pixel can be uniquely indexed with a direction ω_i expressed in spherical coordinates.

Under illumination from an environment map, we can rewrite the direct illumination integral as an explicit summation over the EM entries:

$$L_r(x, \omega_r) = \sum_{j=1}^N f_r(x, \omega_i(j) \rightarrow \omega_r) L_i(\omega_i(j)) (n(x) \cdot \omega_i(j)) V(\omega_i(j)) S(\omega_i(j)), \quad (3.2)$$

where the incoming light direction ω_i is now a function of the EM pixel index j . The additional weighting term $S(j)$ represents the solid angle subtended by the spherical patch corresponding to element j .

Computing the direct illumination integral involves the evaluation of the

visibility term $V(\omega_i)$. Determining whether or not the surface point x can see the light at ω_i typically involves a *ray cast*. A ray r with origin x and direction ω_i is intersected with the scene geometry. If no intersection occurs, x can see the light $L_i(\omega_i)$, and so $V(\omega_i)$ is set to 1. Otherwise, some part of the scene is occluding x 's line of sight to the distant environment along ω_i . $V(\omega_i)$ is thus set to 0, negating the light $L_i(\omega_i)$'s contribution to the reflected radiance.

Performing the ray-scene intersection test makes up the bulk of the work during rendering. Even with the aid of hierarchical acceleration structures, visibility queries dominate the rendering time. In our implementation, for example, typical timings for ray intersection queries with scenes of modest complexity – half a million triangles – are on the order of 4-5 microseconds on our test machine. If one was to precisely evaluate the reflected radiance as indicated in Equation 3.2 by summing over the entire EM, then for sufficiently interesting environment map resolutions (1024x512 pixels), one would have to wait for a couple of seconds to light a single pixel in the output image. A small output image of size 128x128 would take 12 hours to render; this is obviously far too slow.

Intuitively, one realizes that it is not necessary to sum over the entire environment map when computing direct illumination, because there are certain regions of the environment that contribute far less light than others. As an example, think of a night scene lit by a full moon in a clear sky. A vast majority of the light received by the scene will be coming only from the small set of directions corresponding to the location of the moon in the sky. In fact, for most real-world lighting conditions, the contrast ratio between the lightest and darkest points in the environment is very high, typically five to eight orders of magnitude [7].

This leads us to the notion of *Monte Carlo ray tracing*. When rendering from environment maps, one need only consider a small set of incoming light directions. If these directions are carefully chosen in such a way that their contributions make up a significant amount of the reflected radiance, then one can expect to have computed a decent approximation to the illumination integral.

The challenge is to decide how to distribute this small set of sample rays effectively. Possible approaches are to distribute rays according to the intensity of the lights, as suggested above, or according to the reflectance of the surface. In practice, we do this randomly – hence the name Monte Carlo. These topics are discussed in detail in Sections 3.3 and 3.5. First, we require valid mathematics for approximating integrals; this is the subject of the next section.

3.2 Monte Carlo Estimators and Sampling

We present here an overview of Monte Carlo (MC) integration and basic sampling theory. Many introductory treatises of probability theory and Monte Carlo methods exist in the literature. The reader is pointed to Kalos and Whitlock [19] and Hammersley and Handscomb [14] for basic texts on probability theory and Monte Carlo methods, including Monte Carlo integration, rejection sampling and importance sampling. For introductions to Monte Carlo methods in ray tracing and rendering, see [9, 17, 18, 37].

3.2.1 Monte Carlo Integration

The goal in rendering is to compute the reflected radiance for every surface point visible to the viewer. In our case, this amounts to evaluating the direct

illumination integral for every pixel in the resulting image. As described in the previous section, this is too computationally intense to solve directly. Instead, we employ *Monte Carlo integration*, a stochastic technique that allows us to compute approximations for integrals such as the illumination integral for which an exact evaluation is intractable.

Suppose we wish to evaluate the integral

$$I(f) = \int_{\chi} f(x)p(x)dx, \quad (3.3)$$

where the space χ is the domain of definition of the function. Of course, $I(f)$ need not necessarily be a continuous integral. As is the case molecular dynamics, for example, $I(f)$ could be a very large sum and χ the perhaps uncountable set of possible configurations of the system. In any case, when χ is high-dimensional, $I(f)$ becomes infeasible to compute with deterministic integration methods.

One can approximate such integrals or large sums $I(f)$ with tractable sums $I_N(f)$. Given a set of identically independently distributed samples $X = \{x_1, x_2, \dots, x_N\}$ drawn from some density $p(x)$ defined over the space χ , the tractable sum

$$I_N(f) = \frac{1}{N} \sum_{i=1}^N f(x_i) \quad (3.4)$$

converges asymptotically as $N \rightarrow \infty$ to

$$I(f) = \int_{\chi} f(x)p(x)dx.$$

Let us evaluate the *expectation value* of our estimator $I_N(f)$. The expectation of a function $f(x)$ is defined by

$$\langle f(x) \rangle \equiv \int f(x)p(x)dx. \quad (3.5)$$

for a continuous variable and

$$\langle f(x) \rangle \equiv \sum_x f(x)p(x)$$

for a discrete variable. Among the properties satisfied by expectation values are the following:

$$\langle aX \rangle = a\langle X \rangle; \quad (3.6)$$

$$\langle \sum_i X_i \rangle = \sum_i \langle X_i \rangle, \quad (3.7)$$

where X is the random quantity and a is any constant.

We wish to evaluate the expectation of our estimator $I_N(f)$ from Equation 3.4:

$$\langle I_N(f) \rangle = \left\langle \frac{1}{N} \sum_{i=1}^N f(x_i) \right\rangle. \quad (3.8)$$

Applying Equations 3.6 and 3.7 to the right side of Equation 3.8 yields

$$\langle I_N(f) \rangle = \frac{1}{N} \sum_{i=1}^N \langle f(x_i) \rangle.$$

By the definition of expectation given in Equation 3.5, this can be rewritten as

$$\begin{aligned} \langle I_N(f) \rangle &= \frac{1}{N} \sum_{i=1}^N \int f(x)p(x)dx; \\ &= \frac{1}{N} \sum_{i=1}^N I(f); \\ &= I(f). \end{aligned}$$

The expectation of $I_N(f)$ is therefore $I(f)$. $I_N(f)$ is an unbiased estimate of $I(f)$, and by the strong law of large numbers, will almost surely converge to the target integral $I(f)$ [2].

The strength of Monte Carlo integration over deterministic integration schemes is essentially that MC integration seeks to evaluate $f(x)$ only in those areas where probability is high – that is, where the contributions of $f(x)$ to the integral $I(f)$ are most significant.

Let us now examine the *variance* of $I_N(f)$. The variance of a random quantity X , written $\text{var}(X)$, is the expected squared difference between a realization of the variable and its expectation. That is,

$$\begin{aligned}\text{var}(X) &\equiv \langle (X - \langle X \rangle)^2 \rangle; \\ &= \langle X^2 \rangle - \langle X \rangle^2.\end{aligned}$$

For independent random variables X_i , variance observes the following properties:

$$\begin{aligned}\text{var}\left(\sum_i X_i\right) &= \sum_i \text{var}(X_i); \\ \text{var}(aX) &= a^2 \text{var}(X).\end{aligned}$$

With these relations in mind, we see that

$$\begin{aligned}\text{var}(I_N(f)) &= \text{var}\left(\frac{1}{N} \sum_{i=1}^N f(x_i)\right); \\ &= \frac{1}{N^2} \sum_{i=1}^N \text{var}(f(x_i)); \\ &= \frac{1}{N} \text{var}(f(x)).\end{aligned}\tag{3.9}$$

In rendering, variance manifests itself as per-pixel noise in the output image. Equation 3.9 shows that the variance in a Monte Carlo estimator of an integral $I(f)$ is inversely proportional to the sample size N . This result tells us that we can reduce noise in the output image by increasing the number of sample directions we use to approximate the illumination integral.

However, the error in our estimate actually behaves similarly to the *standard deviation* of the samples; that is, the square root of the variance. Thus is demonstrated the fundamental problem with Monte Carlo integration: the notion of diminishing returns. Because image quality depends on N^2 , one must quadruple N to halve the error.

3.2.2 Sampling a General Distribution

So far, we have not mentioned how to choose the density function $p(x)$ for distributing samples during Monte Carlo integration. Let us defer this discussion for a time, assuming for the moment that we know a good distribution $p(x)$ to sample from. This section presents two methods for redistributing a uniformly distributed variate according to some arbitrary probability density $p(x)$.

Cumulative Distribution Functions and the Transformation

Method

A common technique for sampling from general PDFs is known as the transformation method, which requires the notion of the cumulative distribution function.

The *cumulative distribution function* (CDF) is an alternate way of completely describing the distribution of a real-valued random variable $X \sim p(x)$. For every real number x , the CDF is given by

$$C(x) = p(X \leq x).$$

In other words, $C(x)$ is the probability that the random variable X takes on a value less than or equal to x .

The CDF can be computed with

$$C(x) = \int_{x_0}^x p(x)dx,$$

where x_0 is the smallest value that the variable X may take on, generally $-\infty$. If x_0 is finite, the random variable can always be translated so that $x_0 = 0$.

We can now use the *transformation method* to sample from $p(x)$. Specifically, we choose uniform deviates $u_i \sim \mathcal{U}_{(0,1)}$ and transform them according to $x_i = C^{-1}(u_i)$. Refer to Figure 3.2 for an illustration. Note that since $p(x) \geq 0$, $C(x)$ is monotone increasing, and so $C^{-1}(x)$ exists everywhere.

Discrete Distributions

We now examine the case where X is a discrete random variable. Here, the probability distribution $p(x)$ describing X is represented by a table of values $p[i] = Pr[X = i]$, where $i = 1, \dots, N$ runs through the set of all possible values of X . Such a PDF representation can be created from any discretely sampled function $f[i]$ in $O(n)$ time via the expression:

$$p[i] = \frac{f[i]}{\sum_{i=1}^N f[i]}.$$

We can evaluate a discrete CDF $C[i]$ according to

$$C[i] = \sum_{i=1}^N p[i].$$

Now, to sample from $p(x)$, we use the transformation method as before. That is, choose $u \sim \mathcal{U}_{(0,1)}$ and find the table index i that satisfies $C[i] \leq u < C[i + 1]$. This requires a binary search through $C[i]$, and so sampling this way takes $O(\log N)$ time per generated sample. Practically speaking, one

can often do fairly well to use $i = \lfloor uN \rfloor$ as a starting point for the search. A $O(1)$ sampling can be achieved by pre-inverting C , which takes $O(N)$ time. Depending on the scenario, this may not be worth the effort. For example, if C is constantly changing, or if only a small number of samples will be drawn, sampling directly from C in $O(\log N)$ time may well be faster than pre-inverting C and sampling in $O(1)$ time.

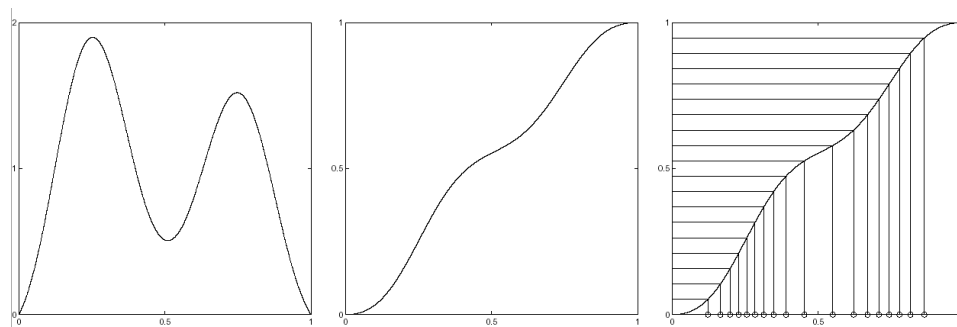


Figure 3.2: Sampling via CDF inversion. Left: a 1D PDF, $P(x)$. Middle: the corresponding CDF, $C(x)$. Right: uniform samples transformed by $C^{-1}(x)$. Note along the x-axis how samples have been redistributed according to $P(x)$. Image courtesy of [36].

Faster Sampling: the Alias Method

Another method for sampling from discrete distributions is the *alias method*, which was proposed by Walker [46]. This ingeniously simple technique computes samples in $O(1)$ time, but with an initialization that is less costly than the CDF inversion and pre-normalization of the transformation method. Strangely, the alias method has been largely overlooked by the graphics community, where the transformation method seems to be the standard technique for sampling from discrete distributions. As such, this section

presents the idea behind the method of aliases, and the variant of Vose [45] that we use in this work.

Let x be a discrete random variable distributed over the set $S = \{0, 1, 2, 3\}$ with corresponding probabilities $P = \{0.1, 0.2, 0.4, 0.3\}$ (see Figure 3.3, left). We want to generate sample values for x according to the distribution P . Marsaglia [28] introduced the following simple approach. Assuming we have a uniform random number generator, one way to sample from P is to generate a uniform random integer i in $[1, 10]$ and use it to index into the set $L = \{0, 1, 1, 2, 2, 2, 2, 3, 3, 3\}$ (see Figure 3.3, right). L has been chosen such that the number of times each element of S appears in L is proportional to the probability of that element occurring, as described by P . Note that the size of L depends on the probabilities in P . If, for example, x was distributed over $\{0, 1\}$ with probabilities $\{1/2953, 1 - 1/2953\}$, this method would require L to have 2954 elements. This simple example distribution can obviously be sampled more efficiently; however, for arbitrarily complex discrete distributions, the required size of L is bounded only by the precision of the random number generator – generally several orders of magnitude.

Marsaglia’s approach was how arbitrary discrete distributions were sampled prior to Walker’s method of aliases, which introduced storage costs that are only linear in the input size while maintaining constant-time sampling.

We now describe the sampling approach behind the alias method. Let $S = \{0, \dots, N - 1\}$ be the domain of distribution of x , as before, with corresponding probabilities $P = \{p_0, \dots, p_{N-1}\}$. The method operates on two lists, *prob* and *alias*, to generate an integer x from S as follows: given a real-valued uniform variate $u \sim [0, N)$, let $j = \lfloor u \rfloor$. If $(u - j) \leq \text{prob}_j$, then choose $x = j$; otherwise, choose $x = \text{alias}_j$.

The particular technique we use to create the lists *prob* and *alias* is

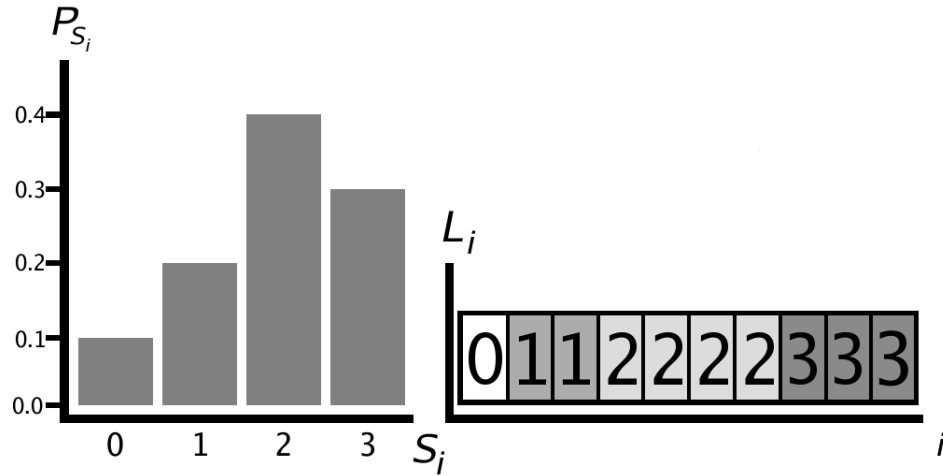


Figure 3.3: Marsaglia’s sampling method [28]. Left: the target distribution P . Right: Sampling is achieved by redistributing a uniform random integer i in $[1, 10]$ according to L .

that of Vose [45]. Consider again the distribution $S = \{0, 1, 2, 3\} \sim P = \{0.1, 0.2, 0.4, 0.3\}$, as illustrated on the left side of Figure 3.4. Vose’s algorithm can be visualized by drawing a horizontal line in the plot of P at probability $\frac{1}{N}$. Elements of P that are over this line are “chopped off” and their contributions redistributed to elements which have a smaller probability, creating tables *prob* and *alias*. On the right side of Figure 3.4, the x-axis corresponds to table indices. The heights of the shaded regions correspond to the real-valued entries of *prob*, while the numbers in the lighter regions correspond to the indices stored in *alias*. For those elements P_j that are $\leq \frac{1}{N}$, $prob_j = NP_j$. For those elements that are $> \frac{1}{N}$, $prob_j = 1$ and j appears in *alias*. Sampling amounts to picking a random index j and determining if it falls within $prob_j$. If so, return j , otherwise return the reference $alias_j$ to a value whose probability overflowed into bin j .

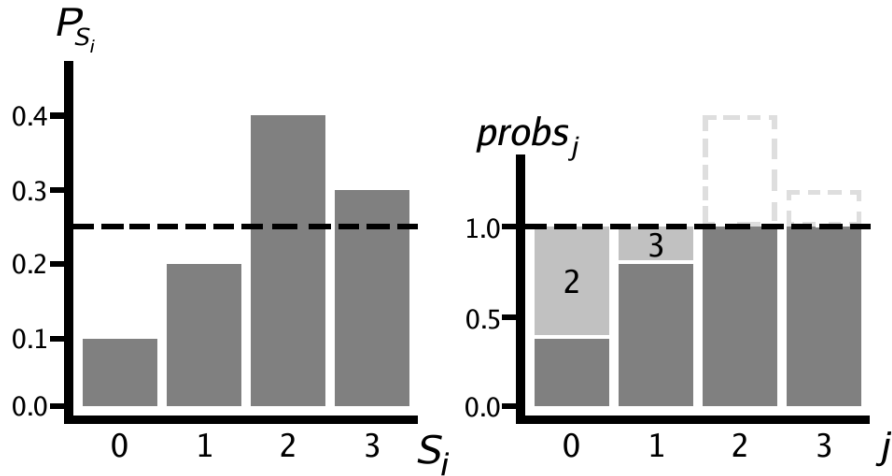


Figure 3.4: Walker’s alias method [46]. Left: the target distribution P and the cutoff line at $\frac{1}{N} = 0.25$. Right: Values with large probabilities have references distributed to bins where probabilities are small.

Refer to Appendix A for a discussion and implementation of the alias method, including Vose’s linear-time algorithm for initializing the lists *alias* and *probs*.

3.3 Sampling the Environment Map

In situations where we need to draw samples according to the importance of the environment map image, we employ an approach similar to Secord et al. [36] for sampling from a 2D PDF. For an RGB image, our notion of importance is the intensity of each pixel – that is, the sum of the intensities of each color channel.

We begin with an overview of the method, followed by a detailed description.

Overview of 2D Sampling

Preprocessing

Given an $N \times M$ RGB image, we perform the following precomputation:

1. For each pixel, compute an intensity $I(x, y) = red(x, y) + green(x, y) + blue(x, y)$
2. For each of the $y = [1, \dots, M]$ scanlines, create a PDF based on the intensities of the pixels in that scanline. At the same time, compute the average intensity of the pixels in that scanline: $I_{avg}(y) = \frac{1}{N} \sum_{i=1}^N I(i, y)$
3. Finally, create a PDF from the distribution of M scanline averages I_{avg}

Sampling

One can now generate samples according to the image intensities:

1. Generate $u_i \sim \mathcal{U}_{(0,1)^2}$.
2. Redistribute $u_{i,y}$ according to the PDF of scanline averages to select a scanline $x_{i,y}$.
3. Redistribute $u_{i,x}$ according to the PDF of the pixel intensities in scanline $x_{i,y}$, to select pixel location $x_{i,x}$.

2D Sampling in Detail

Our goal is to redistribute a set of uniform 2D samples $U = \{u_1, \dots, u_n\}$ where $u_i \sim \mathcal{U}_{(0,1)^2}$ according to a PDF $p(x)$, generating a set $X = \{x_1, \dots, x_n\}$

with $x_i \sim p(x)$. Let $x_{i,x}$ and $x_{i,y}$ denote the x and y coordinate of sample x_i . To determine $x_{i,y}$, we compute the cumulative density function

$$C(y) = \int_0^y m(t)dt, \quad (3.10)$$

where $m(y)$ is the marginal density function of $p(x)$. In our case, we have a 2D image, and so $m(y)$ can be thought of as the average intensity of scanline y in the environment map. Note that $M(y)$ is non-zero for those scanlines containing non-zero intensity, and so using the transformation method we obtain $x_{i,y} = C^{-1}(u_{i,y})$.

Given $x_{i,y}$, the uniform sample $u_{i,x}$ is now redistributed according to the PDF of the respective scanline, $p_{x_{i,y}}(x)$. This can be accomplished via the conditional probability distribution

$$c(x|x_{i,y}) = \frac{p_{x_{i,y}}(x)}{m(x_{i,y})} \quad (3.11)$$

and the corresponding cumulative distribution

$$C(x|x_{i,y}) = \int_0^x c(s|x_{i,y})ds \quad (3.12)$$

As before, the x coordinate of the new point can be found through $x_{i,x} = C^{-1}(u_{i,x}|x_{i,y})$

Our 2D environment map is a discrete PDF, and so we use either the transformation method or the alias method for sampling in $O(1)$ time. The precomputation of scanline PDFs and data structures have linear requirements in both space and time. The precomputation need only be done once per environment map. Also, the preprocessing is fast, at only milliseconds even for large images; it can thus be performed on-the-fly as the image is loaded.

Importance sampling from the EM is now just a simple table lookup, where the table indices are uniformly random points. Practically speaking,

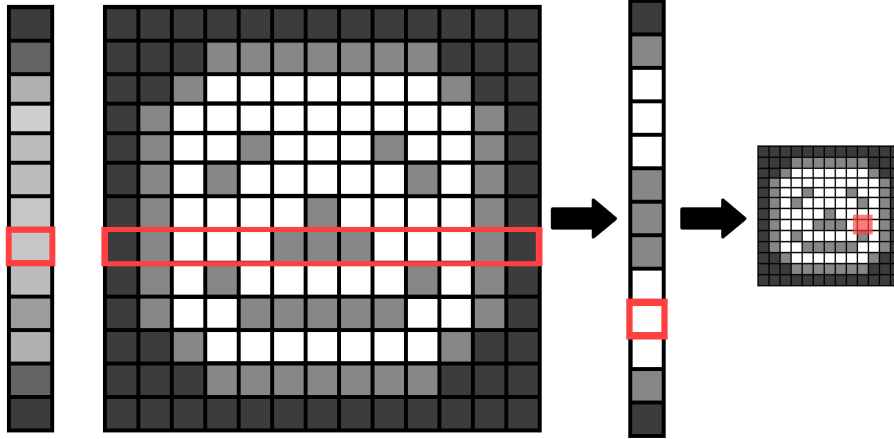


Figure 3.5: Sampling a 2D intensity image. Two 1D passes are performed: a scanline is selected according to the distribution of scanline averages, and a pixel is then sampled from within that scanline.

rather than transforming a uniform distribution, one can use points taken from a low-discrepancy sequence such as the Halton sequence [36] or a blue-noise distribution [31]. These choices will result in a sample set which is more evenly distributed in space, with fewer clusters of close points. For further information, the reader is referred to Keller's thesis [22] for an excellent presentation of quasi Monte Carlo methods and their use in rendering.

3.4 Improving Shadows: Area-weighted Lighting

In principle, there is always a practical lower-bound on the number of visibility tests one can perform per pixel and still achieve a low-variance image. For example, consider a high-frequency environment map that is dark everywhere except for N small, bright area light sources. For a surface patch

which can see all N lights, the best thing to do would be to draw a single sample from each of the light sources. If ever we render with less than N samples, there will always be some amount of variance in the resulting image, because visibility will not have been tested for each light. This noise is particularly noticeable in penumbra regions (soft shadow boundaries), as each pixel is seeing a different random sampling of the lights. In practice, this is often acceptable; as will be shown in Chapter 5, our approaches for sampling from the product distribution still give us nice images even for an extremely small number of samples. Sampling purely from the EM or BRDF takes many more samples (hundreds), and thus implicitly generates shadow boundaries which are generally good.

Up to this point, sampling from the environment map has meant selecting directions (pixels) based solely on pixel intensity – that is, based on light brightness. In practice, however, it is desirable to distribute light samples with a good spatial distribution as well as by intensity.

The motivation for including an area factor is to address the oversampling of small, bright light sources. Once a small light source has been sampled from, there is little additional gain in drawing another sample from this light, because it is behaving essentially as a point source. By drawing a single sample from it, we already have an idea of the intensity and color of the light coming from this source. Furthermore, our shadow quality will not improve or smoothen, because the spatial extent of the light is small, and the visibility function is not likely to vary across the light.

As an example, consider an environment map containing several small light sources with equal area, with one light being significantly brighter than the others. If directions are chosen based solely on EM pixel intensity, the majority of the samples will come from the bright light. However, because

this light source is small, only one or two samples from this light would be enough. We're better off casting a few samples to some of the dimmer lights in order to smooth out shadow boundaries.

To this end, we take the approach of Agarwal et al. [1] and modulate the light intensity by an area term based on the solid angle of the light. The basic idea is to first identify the light sources in the environment map. Then, the importance of all pixels making up a light source are scaled based on the area of that light. In effect, we want to penalize small light sources more than large ones, so that small sources are not oversampled.



Figure 3.6: The Grace Cathedral environment after quantization based on the logarithm of pixel intensity. Observe how area light sources in the environment, such as the altar and windows, have been localized.

Our light classification scheme proceeds in the following manner. We first compute the histogram of the image and quantize it into N intensity levels. The binning is performed on the logarithm of pixel intensity. Because we're interested in high-frequency HDR environment maps, where the contrast

ratio is 5-7 orders of magnitude, the majority of EM pixels are low intensity. So, this classification is performed logarithmically rather than linearly.

Next, connected components are found in the quantized image by running a breadth-first search. The area (solid angle) of each connected component is found by summing the areas (solid angles) of each pixel making up the light source. Importances L_i of the pixels, originally taken just from intensity, are now scaled by this solid angle area. The new importance is given by $L \propto L\Delta\omega$. Specifically, we employ the conclusions of Agarwal et al. [1] and select $L = L(\min(0.01, \Delta\omega))^b$, where b is in the range $[0.1, 0.2]$, depending on the average size of the light sources in the environment map.

The time complexity of the area-weighting algorithm is linear in the size of the EM, and is sufficiently fast so that it can be performed in negligible time during loading.

In practice, the bright areas in measured environments are often reasonably well distributed across the sky, and so this factor does not significantly affect the quality of the sample distribution. Nevertheless, the results presented in Chapter 5 were generated using this augmented light importance.

3.5 Sampling from the BRDF

Another method for choosing sample directions for visibility testing is to distribute rays according to surface reflectances (BRDFs). This is particularly helpful in scenes that have, for example, rather uniform light distributions (think outdoor scenes). Sampling based on lighting intensity in such a scene does not significantly localize important directions. However, sampling from glossy or shiny materials in the scene will do well because of their non-uniform reflection properties. Examples of such surfaces include

metals, plastics, anything with a finish or polish, or surfaces such as asphalt or water when viewed at grazing angles.

Recall that the BRDF $f_r(x, \omega_i \rightarrow \omega_r)$ states what fraction of the power arriving at a surface point x from an incoming direction ω_i will be reflected along an outgoing direction ω_r (see Figure 3.7). We assume that the BRDF is energy preserving; that is,

$$\int_{\Omega} P(\omega_r | \omega_i, x) d\omega_r = \int_{\Omega} f_r(x, \omega_i \rightarrow \omega_r) d\omega_r = 1. \quad (3.13)$$

In other words, we assume that the BRDF reflects all energy incident upon it, and that no energy is generated or absorbed. While not strictly necessary for computing single-bounce direct illumination, this restriction allows for an alternative interpretation of the BRDF: the BRDF f_r can be thought of as a probability density that describes the probability that an incoming ray of light will be randomly scattered towards a particular outgoing direction. Under this view, each incoming ray maps to a single outgoing ray with equal intensity, but directed along directions sampled according to the magnitude of the BRDF. Furthermore, the BRDF is reciprocal, and so incoming and outgoing directions can be exchanged. Hence, another interpretation is to view the BRDF as a function which describes which incoming light directions ω_i will contribute to the light reflecting along ω_r .

There has been much previous work in efficient sampling from BRDFs (for example, [24, 25, 40]). There are several analytical representations for the BRDF that support sampling directly. In particular, we use the energy-preserving (normalized) form [26] of the Phong model [32]:

$$f_r(x, \omega_i \rightarrow \omega_r) = \frac{k_s}{2\pi} (n + 1) \cos^n \theta + \frac{k_d}{\pi}. \quad (3.14)$$

θ is the angle between the view direction ω_r and the reflected light ray R , where R is the incoming light direction ω_i reflected across the surface

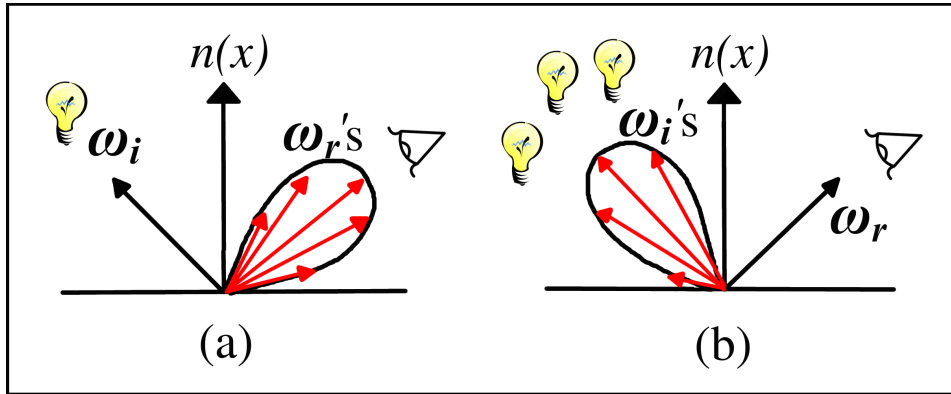


Figure 3.7: Two interpretations for the BRDF. The BRDF describes: (a) the distribution of scattering directions ω_r for light which is incoming along ω_i ; or, (b) the distribution of incoming light directions ω_i that send light towards ω_r .

normal $n(x)$. See Figure 3.8 for an illustration. $k_s(n+1)\cos^n\theta$ is called the *specular term*; it describes the width of shiny highlights observed in the material. k_s and k_d are referred to as the *specular* and *diffuse* coefficients, respectively. When $k_s = 0; k_d = 1$, the BRDF reduces to a constant, and the surface is said to be diffuse, meaning that light is reflected uniformly in all directions. When $k_s = 1; k_d = 0$, incoming light is reflected according to the cosine lobe. As such, some reflected directions are preferred over others, and so the reflected radiance is non-uniform over the surface, resulting in a glint or highlight. As the specular exponent n is increased, the highlights become taller (brighter) and more concentrated. In order that the BRDF preserves energy, its terms are normalized by $\frac{1}{2\pi}$ and $\frac{1}{\pi}$, and the coefficients constrained to $k_s + k_d = 1$.

One can sample the Phong PDF by sampling from the PDF

$$p(\theta, \phi) = \frac{n+1}{2\pi} \cos^n \theta. \quad (3.15)$$

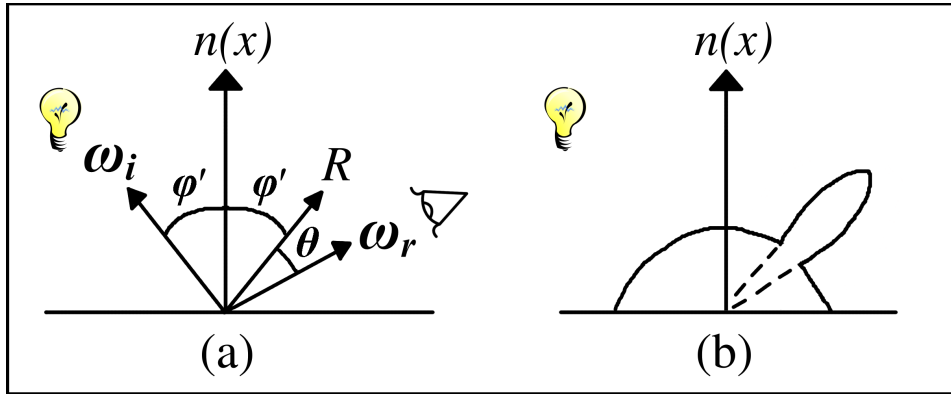


Figure 3.8: The Phong BRDF. (a) The amount of light a viewer sees depends on the angle θ with respect to the reflected light direction R . (b) The diffuse and specular lobes. The specular lobe shape depends on the exponent n .

The pair (θ, ϕ) is a direction expressed in spherical coordinates; θ is the polar angle constrained to the upper hemisphere ($\theta \in [0, \pi/2]$), and ϕ is the azimuthal angle ($\phi \in [0, 2\pi]$). We can sample from Equation 3.15 by transforming a pair of uniform variates (u_1, u_2) according to

$$(\theta, \phi) = (\arccos((1 - u_1)^{\frac{1}{n+1}}), 2\pi u_2). \quad (3.16)$$

Note that our sample directions (θ, ϕ) are distributed about the polar axis. However, the specular term in the BRDF is parameterized about the reflection direction. So, the final step in sampling is to transform our directions from the global polar axis to the local frame of the BRDF.

It is common practice to incorporate the cosine term from the illumination integral (Equation 3.1) into the BRDF, yielding

$$f'_r(x, \omega_i \rightarrow \omega_r) = \left[\frac{k_s}{2\pi} (n+1) \cos^n \theta + \frac{k_d}{\pi} \right] n(x) \cdot \omega_i. \quad (3.17)$$

Recall that the cosine term $n(x) \cdot \omega_i$ serves to scale incoming light based on the orientation of the surface with respect to the incoming light direction.

Observe that the Phong lobe is parameterized about the reflection direction, whereas the cosine term is parameterized about the surface normal. This difference in parameterization makes sampling from the product of the BRDF and cosine term more challenging. In fact, to date, there is no analytical method for directly sampling from Equation 3.17 [38]. Instead, we sample according to f'_r by combing samples drawn exclusively from either the specular term or the diffuse term. The ratio of samples drawn from each term depends on the values k_s and k_d ; in other words, N BRDF samples will be drawn as $k_s N$ specular samples and $k_d N$ diffuse samples. Deciding which term to sample from is made randomly in order to remain unbiased.

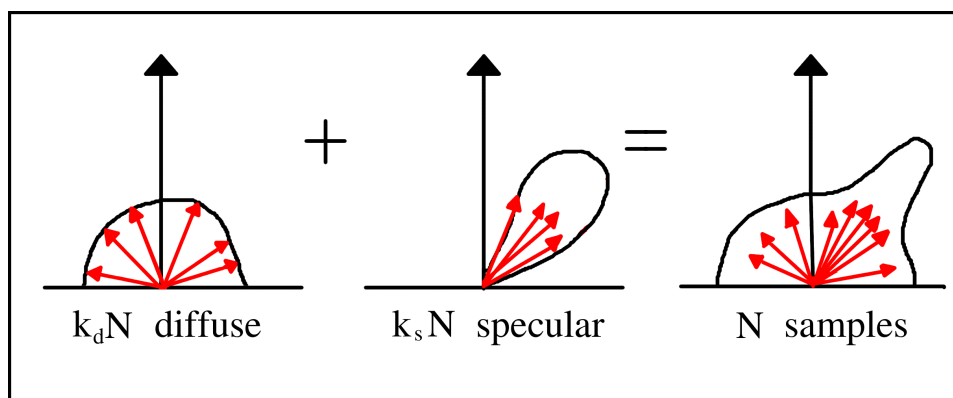


Figure 3.9: Sampling from the product of the Phong BRDF and the area term involves combing samples drawn individually from the diffuse and specular terms. In this case, $k_s = k_d = 0.5$, resulting in (approximately) an equal number of samples being drawn from each term.

Chapter 4

Sampling the Product Distribution

The main contribution of this thesis addresses the problem of direct illumination from environment maps. We propose a bidirectional sampling approach in which both the energy distribution in the environment map and the reflectance of the BRDF are taken into account.

4.1 Bidirectional Importance Sampling

We operate on the assumption that creating samples from only the environment map or only the BRDF model is inexpensive, and that the visibility test dominates the cost. This assumption holds for scenes with complex geometry and for BRDF models optimized for sampling. In the case of procedural shaders, for example, importance sampling is always difficult, but we can still operate if the shader provides information to support efficient importance sampling as proposed by Slusallek et al. [40]. In any case, visibility is typically evaluated by casting a ray through the scene and intersecting it with geometry. A ray cast operation is orders of magnitude more expensive than drawing samples in virtually every case.

Under these assumptions, one can benefit from extra time spent in at-

taining a good sample distribution that takes both the BRDF and environment map into account. Such a distribution selects only directions for visibility testing that have significant contribution to the reflected radiance of the surface under evaluation.

Consider again the direct illumination integral

$$L_r(\omega_r) = \int_{\Omega} f_r(\omega_i \rightarrow \omega_r) \cos \theta_i L_i(\omega_i) V(\omega_i) d\omega_i, \quad (4.1)$$

where L_i is the illumination from the environment map, f_r is the BRDF, and V is the binary visibility term. Conventional approaches perform importance sampling either from the intensity in the environment map according to the probability density

$$q_L(\omega_i) := \frac{L_i(\omega_i)}{\int_{\Omega} L_i(\omega_i) d\omega_i}, \quad (4.2)$$

or from the BRDF according to

$$q_f(\omega_i) := \frac{f_r(\omega_i \rightarrow \omega_r) \cos \theta_i}{\int_{\Omega} f_r(\omega_i \rightarrow \omega_r) \cos \theta_i d\omega_i}. \quad (4.3)$$

Notice that in Equation 4.3 the cosine term has been included in the PDF, as discussed in Section 3.5.

Let us derive estimators for the reflected radiance L_r . When choosing sample directions $\omega_{i,j} \sim q_L(\omega_i)$, Equation 4.1 can be estimated with $L_{N,L}$:

$$\begin{aligned} L_{N,L}(\omega_r) &= \frac{1}{N} \sum_{j=1}^N \frac{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} L_i(\omega_{i,j}) V(\omega_{i,j})}{q_L(\omega_{i,j})}; \\ &= \frac{1}{N} \sum_{j=1}^N \frac{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} L_i(\omega_{i,j}) V(\omega_{i,j})}{L_i(\omega_i)} \int_{\Omega} L_i(\omega_i) d\omega_i; \\ &= \frac{\int_{\Omega} L_i(\omega_i) d\omega_i}{N} \sum_{j=1}^N f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} V(\omega_{i,j}). \end{aligned}$$

When sampling according to the BRDF with $\omega_{i,j} \sim q_f(\omega_i)$, the corresponding estimator $L_{N,f}$ is

$$\begin{aligned} L_{N,f}(\omega_r) &= \frac{1}{N} \sum_{j=1}^N \frac{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} L_i(\omega_{i,j}) V(\omega_{i,j})}{q_f(\omega_{i,j})}; \\ &= \frac{1}{N} \sum_{j=1}^N \frac{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} L_i(\omega_{i,j}) V(\omega_{i,j})}{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j}} \int_{\Omega} f_r(\omega_i \rightarrow \omega_r) \cos \theta_i d\omega_i; \\ &= \frac{\int_{\Omega} f_r(\omega_i \rightarrow \omega_r) \cos \theta_{i,j} d\omega_i}{N} \sum_{j=1}^N L_i(\omega_{i,j}) V(\omega_{i,j}). \end{aligned}$$

Evaluating the variance of these estimators yields

$$\begin{aligned} \omega_{i,j} \sim q_L(\omega_i) &\rightarrow \text{var}(L_{N,L}) = \frac{\int L_i^2}{N} \text{var}(f_r(\omega_i \rightarrow \omega_r) \cos \theta_i V(\omega_i)); \\ \omega_{i,j} \sim q_f(\omega_i) &\rightarrow \text{var}(L_{N,f}) = \frac{\int f_r(\omega_i \rightarrow \omega_r) \cos \theta_i^2}{N} \text{var}(L_i(\omega_i) V(\omega_i)). \end{aligned}$$

When proposing samples from the environment, variance in the result is proportional to the variance in the BRDF. Similarly, when proposing from the BRDF, variance is proportional to the lights. It follows that the greatest reduction in image noise occurs when samples are drawn from the function with greater variance. This is verified by intuition. If the BRDFs are diffuse but the lighting spotty, then directions should be chosen according to the importance of the lights. On the other hand, if light sources in the environment map are relatively broad but the surfaces glossy or shiny, then proposing from the BRDF will be the better thing to do.

Either approach will produce significant noise if both the BRDF and the illumination contain any high frequency information. The solution of Veach and Guibas [44] was to linearly combine samples drawn exclusively from the lights or BRDF. However, a mix of samples still suffers from dependence on

the variances of the individual techniques, because the variance of a sum of terms equates to the sum of the variances of each term.

Our approach is to reduce variance by sampling *directly* from the product distribution,

$$p(\omega_i) := \frac{f_r(\omega_i \rightarrow \omega_r) \cos \theta_i L_i(\omega_i)}{\int_{\Omega} f_r(\omega_i \rightarrow \omega_r) \cos \theta_i L_i(\omega_i) d\omega_i}. \quad (4.4)$$

Observe that the normalization term in Equation 4.4 is the direct illumination integral with the visibility term $V(\omega_i)$ omitted. In other words, this term is the exitant radiance in the absence of shadows. We refer to it as L_{ns} ("radiance, no shadows"):

$$L_{ns} := \int_{\Omega} f_r(\omega_i \rightarrow \omega_r) \cos \theta_i L_i(\omega_i) d\omega_i. \quad (4.5)$$

If we draw sample directions $\omega_{i,j} \sim p(\omega_i)$ according to the product distribution in Equation 4.4, we can estimate Equation 4.1 with $L_{N,p}$, where

$$\begin{aligned} L_{N,p}(\omega_r) &= \frac{1}{N} \sum_{j=1}^N \frac{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} L_i(\omega_{i,j}) V(\omega_{i,j})}{p(\omega_{i,j})}; \\ &= \frac{1}{N} \sum_{j=1}^N L_{ns} \frac{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} L_i(\omega_{i,j}) V(\omega_{i,j})}{f_r(\omega_{i,j} \rightarrow \omega_r) \cos \theta_{i,j} L_i(\omega_{i,j})}; \\ L_{N,p}(\omega_r) &= \frac{L_{ns}}{N} \sum_{j=1}^N V(\omega_{i,j}). \end{aligned} \quad (4.6)$$

We refer to $L_{N,p}$ as the *bidirectional estimator* for the direct illumination integral. The evaluation of Equation 4.6 can be interpreted as taking the unoccluded reflected radiance L_{ns} and scaling it by the average result of N visibility tests performed along directions that contribute most significantly to the radiance.

Evaluating the variance,

$$\omega_{i,j} \sim p(\omega_i) \rightarrow \text{var}(L_{N,p}) = \frac{L_{ns}^2}{N} \text{var}(V(\omega_i))$$

Observe that the variance of the bidirectional estimator for the reflected radiance depends only on the variance in the visibility function. Variance in the lighting and BRDF has been completely factored away by sampling directly from the product distribution. The remaining question is how these samples can be drawn.

Figure 4.1 contains angular plots of the probability densities corresponding to the various proposal distributions. The top image of Figure 4.1 is an image of the importance derived from environment map intensity. The center image is a plot of the BRDF for a surface patch whose normal is pointing at the horizon. The BRDF plotted here is a specular Phong BRDF with exponent 50. The bottom image is the product of the EM and BRDF. It is this product distribution that we'd like to sample from, because it directly corresponds to the reflected radiance of the surface patch.

Figure 4.2 illustrates the benefit of sampling from the product distribution, as opposed to sampling from the EM or the BRDF individually. Sampling from the BRDF alone misses the bright lights in the environment (top image), while samples drawn from the environment do not lie in the BRDF (middle image). The bottom image, however, shows samples drawn according to the product distribution using our SIR technique, as described in Section 4.4.



Figure 4.1: From top to bottom: angular plots of the importance function of the Grace Cathedral EM, a specular Phong BRDF of exp 50, and their product.

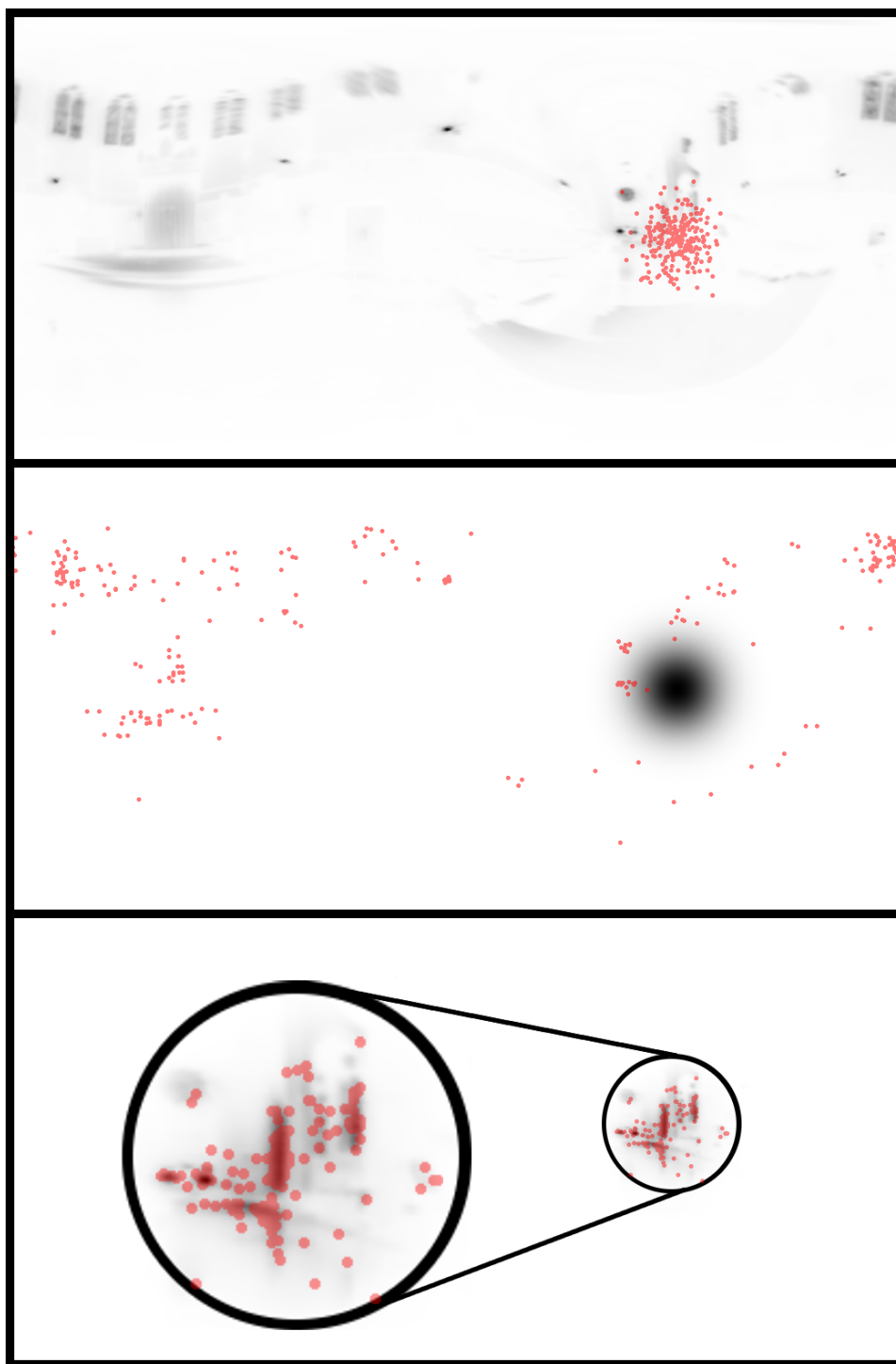


Figure 4.2: Samples drawn solely from the BRDF (top) or the environment (middle) vastly undersample the product distribution (bottom). The sample set in the bottom image was generated with our SIR technique; see Section 4.4.

4.2 The Challenge of Bidirectional Sampling

The challenge in realizing the idea of bidirectional importance sampling is that the product of the BRDF and environment map is too high dimensional to precompute. The BRDF is a 4D function that maps from incoming directions to outgoing directions. The relevant 2D slice of the BRDF, corresponding to a specific outgoing light direction ω_r , varies from point to point in the scene due to changes in the local surface orientation. The environment map is two dimensional, and thus the BRDF-EM product has six dimensions. Even with a coarse discretization of the BRDF, which might cause high frequency features in the BRDF to be lost, precomputing the product distribution and storing it in a table for sampling is obviously prohibitively expensive.

Complicating matters further is the fact that the BRDF and environment map are usually specified relative to different coordinate frames. Typical parameterizations for the BRDF are local, either about the surface normal or the reflected light direction. The EM, on the other hand, is expressed in a global frame. Thus, precomputing the product of the BRDF and the environment for a single BRDF orientation is insufficient. One must either compute a sample rotation or add another two dimensions to the precomputed table, corresponding to a discrete sampling of possible surface orientations.

The high-dimensional nature of the problem makes it difficult to use pre-computation for sample generation. We simply cannot afford to premultiply the two functions together.

We suggest the following process for sampling from the product of the lights and the BRDF. First, create samples according to either the environment map or the BRDF. Then, adjust the sample distribution such that the

directions chosen for visibility testing will be proportional to the product distribution.

We have developed two solutions that realize this approach, one based on rejection sampling and the other on the sampling-importance resampling (SIR) algorithm. Note that the complete algorithm is a two-stage approach. That is, the local illumination integral is always estimated with importance sampling, but individual subproblems are solved with either rejection sampling or SIR.

Our two sampling methods are detailed in the following two sections.

4.3 Sample Generation Through Rejection

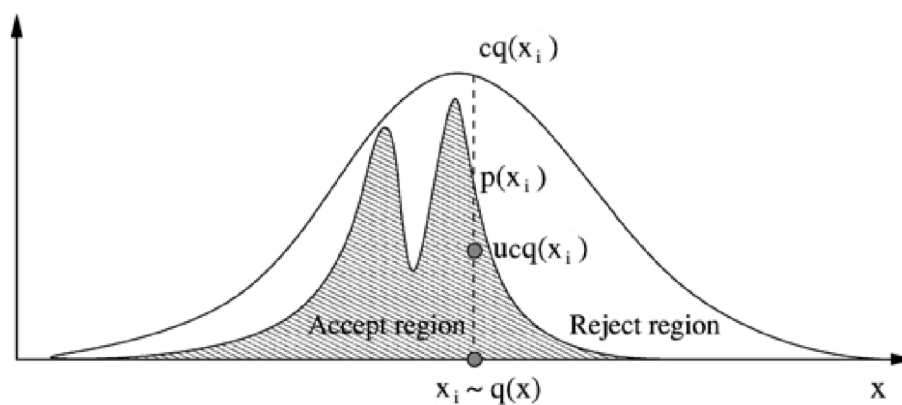


Figure 4.3: Rejection sampling. A sample $x_i \sim q(x)$ is accepted as being a valid sample of the target distribution $p(x)$ if a uniform variate in $[0, cq(x)]$ falls under $p(x)$.

Our first approach for sampling from the product distribution is through *rejection sampling*. Rejection sampling is a technique which allows us to

sample from a probability distribution known only up to a proportionality constant. This allows us to sample according to distributions, where the normalization constant is too expensive to compute explicitly (as is the case with the product distribution from Equation 4.4). Also, rejection sampling provides a simple procedure for sampling from arbitrary distributions that do not have simple analytical forms for sampling.

Rejection sampling is illustrated in Figure 4.3. In order to create samples $\omega_{i,j} \sim p(\omega_i)$, we can approximate $p(\omega_i)$ with a PDF $q(\omega_i)$, such that $p(\omega_i) < c \cdot q(\omega_i)$ for some constant c . We then generate random samples $\omega_{i,j} \sim q(\omega_i)$, and accept them with a probability of $p(\omega_{i,j}) / (c \cdot q(\omega_{i,j}))$.

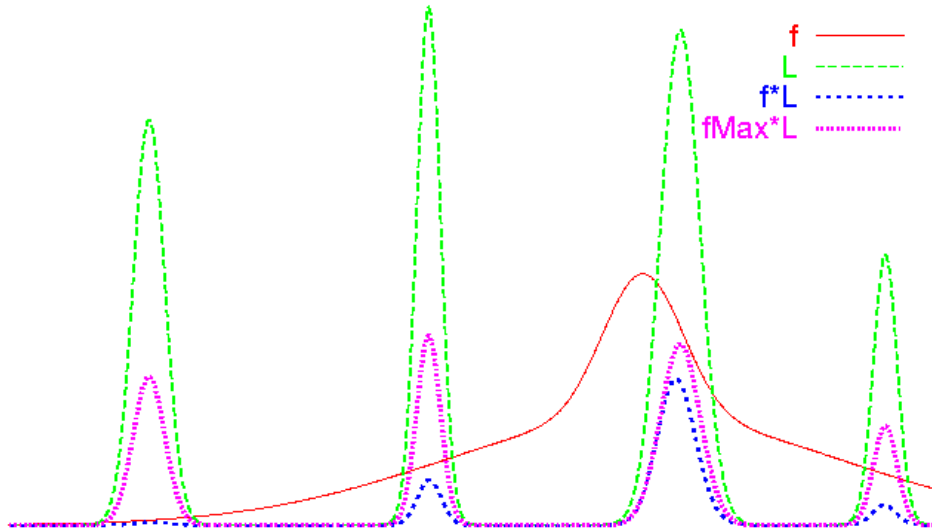


Figure 4.4: Rejection sampling the product distribution. Both $f_{max}L$ and $L_{max}f$ bound the product distribution fL . However, $f_{max}L$ is a tighter fit for this particular BRDF and EM. Thus, the acceptance rate is higher by proposing samples from the lights and rejecting against the max of the BRDF.

In our case, one simple way of bounding $p(\omega_i)$ from Equation 4.4 is to

use the global maximum of the BRDF as a conservative estimate for its contribution as illustrated in Figure 4.4. If L_{\max} and f_{\max} refer to the maximum values of both distributions, i.e.

$$\begin{aligned} L_{\max} &:= \max_{\omega_i} q_L(\omega_i), & \text{and} \\ f_{\max} &:= \max_{\omega_i} q_f(\omega_i), \end{aligned}$$

then we can bound the target PDF $p(\omega_i)$ as $p(\omega_i) < f_{\max} \cdot q_L(\omega_i)$. Note that q_L is just the usual importance from the environment map alone, and so we can sample from it in constant time using CDF pre-integration and inversion or the alias method, as discussed in Section 3.2.2. In order to accept N visibility samples, we will thus have to create $M \approx f_{\max} \cdot N$ environment map samples $\omega_{i,j}$ through importance sampling, and then accept each sample individually with probability

$$\frac{f_r(\omega_{i,j}) \cos \theta_{i,j} \cdot \int_{\Omega} L_i(\omega_i) d\omega_i}{f_{\max} \cdot L_{ns}}.$$

Both this formula and the final radiance estimate from Equation 4.6 require the normalization term L_{ns} from Equation 4.5. We can estimate this term using information that has already been computed during the rejection sampling. Since we already evaluate both BRDF and environment map for M directions $\omega_{i,j} \sim q_L(\omega_i)$, we can estimate L_{ns} as

$$L_{ns} \approx \frac{\int_{\Omega} L_i(\omega_i) d\omega_i}{M} \sum_{j=1}^M f_r(\omega_{i,j} \rightarrow \omega_o) \cos \theta_{i,j}. \quad (4.7)$$

Another interpretation of this method is that we estimate the unoccluded illumination L_{ns} with M samples, using importance sampling from the environment map. However, we only evaluate the visibility for N of

those samples whose light contribution is significant enough to make visibility tests worthwhile. The directions for the visibility tests are chosen in an unbiased fashion.

So far, we have bounded the actual target PDF as a constant times the environment map PDF. This is appropriate if the BRDF contains mostly low frequencies, i.e. if f_{\max} is a close bound of the real BRDF. If this is not the case, then most samples will be rejected and the rejection sampling will become inefficient. Instead, we can perform the same rejection sampling algorithm by approximating the environment map with a conservative bound and then selecting samples according to the real BRDF. Under this scheme, we now have $p(\omega_i) < L_{\max} \cdot q_f(\omega_i)$, which amounts to generating samples from the BRDF alone, and then rejecting according to the product distribution as before.

Given these two ways of rejection sampling, we usually want to draw the initial samples in such a way that the bounding constant is minimized. That is, if $f_{\max} < L_{\max}$ we importance sample from the environment map; otherwise, we importance sample from the BRDF. In practice, we randomly choose which of the two methods to use, where the method with the smaller bounding constant is chosen with a higher probability. This is similar in spirit to the combination of sampling strategies proposed by Veach and Guibas [44].

Our rejection sampling approach has worked well in the experiments that we have performed (see Chapter 5). However, the inherent downside of using rejection sampling is that one cannot guarantee bounds on the execution time for creating a new sample. That is, if the area between $c \cdot q(\omega_{i,j})$ and $p(\omega_{i,j})$ is large, the probability of sample acceptance will be low. One way of dealing with this is to choose a maximum number of sample attempts in

the rejection sampling. So long as at least one sample is accepted, this will still yield an asymptotically unbiased estimate of the reflected radiance.

If no samples are accepted, a possible strategy could be to test visibility for a random subset. Another possibility which is less expensive is to use the unoccluded illumination, where visibility has not been tested at all. While this introduces bias, it will only happen in very dark areas, where the product of illumination and BRDF is very small. In these areas, the visibility term will not have significant impact anyway. In practice, even in the presence of high specular BRDFs and environments, the rejection sampling acceptance probability has been sufficiently high so as to not require resorting to these biased methods. See Chapter 5 for a complete discussion.

The next section presents our second method for sampling from the product distribution, which does not suffer from the unbounded execution time of the rejection sampling.

4.4 Sample Generation Through Sampling-Importance Resampling

In this section, we describe our second technique for sampling from the product distribution by motivating with a description of the importance sampling algorithm.

Recall that the idea behind Monte Carlo integration is to approximate the integral $I(f) = \int f(x)dx$ by sampling from a target density $p(x)$ and computing an estimate to $I(f)$,

$$\langle I_N(f) \rangle = \frac{1}{N} \sum_{j=1}^N \frac{f(x_j)}{p(x_j)}. \quad (4.8)$$

In our case, $p(x)$ is the BRDF-EM product distribution of Equation 4.4.

The idea behind the *importance sampling* algorithm [35] is to sample from $p(x)$ by introducing an arbitrary proposal distribution $q(x)$ that is easier to sample from than $p(x)$ and whose support includes $p(x)$. Equation 4.8 can be rewritten as

$$\langle I_N(f) \rangle = \frac{1}{N} \sum_{j=1}^N \frac{f(x_j)}{w(x_j)q(x_j)},$$

where $w(x_j) \propto \frac{p(x_j)}{q(x_j)}$ is a measure of the importance of sample x_j . Consequently, if one can draw samples according to $q(x)$ and evaluate $w(x_j)$, a possible Monte Carlo estimate of $I(f)$ is

$$\langle I_N(f) \rangle = \frac{1}{N} \sum_{j=1}^N f(x_j)w(x_j). \quad (4.9)$$

This leads to the *sampling-importance resampling* (SIR) algorithm [10, 41, 43]. The goal of SIR is to introduce a resampling step to eliminate samples with low importance ratios $w(x)$. SIR proceeds in two stages. In the first stage, a set of independent random samples $X = \{x_1, \dots, x_M\}$ is drawn from a proposal distribution $q(x)$. In the second stage, a smaller set of samples $Y = \{y_1, \dots, y_N\}$ is drawn from X with sample probabilities $w(x_i)$ proportional to their importance ratio $\frac{p(x_i)}{q(x_i)}$. As the number of first-round samples M approaches infinity, the sample set Y can be shown to have been drawn directly from p . A requirement for the algorithm to be efficient is that $q(x)$ be a good approximation to $p(x)$ in the region of visibility.

We can apply SIR as a solution to the problem of drawing samples from the lighting-BRDF product distribution. It is desirable for $q(x)$, the proposal density for the first round of sampling, to be a reasonable approximation of

$p(x)$. Because $p(x)$ in our case is the product distribution, natural choices for $q(x)$ are probability densities based on either the BRDF or the environment map. In the second SIR stage, we resample the first-round draws according to $w(x_i) \propto \frac{p(x_i)}{q(x_i)}$. The basic intuition here is that directions are proposed from a single distribution, then resampled (filtered) based on their value to the second distribution.

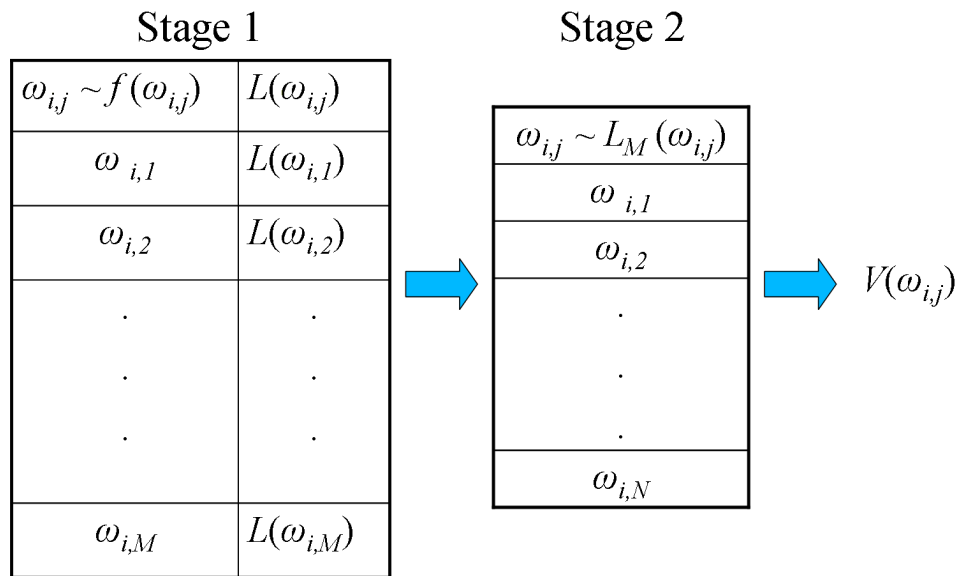


Figure 4.5: Our resampling method. First, M samples are proposed from q_f , the PDF of the BRDF. The candidate directions are then resampled based on the incoming light along those directions, producing N samples for visibility testing. N is generally much less than M .

This leads to our second method for approximating the illumination integral, which proceeds in two stages. In the first stage, M sample directions $\omega_{i,j}$ are drawn according to $q_L(\omega_i)$ or $q_f(\omega_i)$. We store the sample and associated weight $w(\omega_{i,j})$, chosen to be $w(\omega_{i,j}) = f_r(\omega_{i,j})$ when $\omega_{i,j} \sim q_L(\omega_i)$

and $w(\omega_{i,j}) = L_i(\omega_{i,j})$ when $\omega_{i,j} \sim q_f(\omega_i)$. The second stage then resamples the stored sample directions according to the distribution of their weights $w(\omega_i)$. Visibility is tested along each of the N resampled directions.

The total number of samples generated for each pixel is exactly $M + N$. This is an improvement over rejection sampling for two reasons. Firstly, execution time is tightly bounded. We no longer have to wait an indeterminate amount of time (possibly even indefinitely) for the acceptance criteria to pass and accept a sample as being valid. Using the SIR algorithm, samples can be drawn directly from the product distribution in constant time.

The second improvement over rejection sampling is that the sample sizes M and N can be freely chosen, yielding fine control over the tradeoff between quality and time. For example, the BRDF sample size can be adjusted based on the expense of sampling from the BRDF model. Also, it is possible to directly select N – the target number of visibility rays traced per pixel – based on, for example, scene complexity. Because the cost of ray tracing typically dominates rendering time, our general approach for generating results has been to fix N and adjust M so as to increase or decrease the variance in resampled directions. As discussed in the next chapter, typical values of M we use are one to two orders of magnitude larger than N .

It is interesting to note that prior strategies for sampling, which draw directly from the EM or BRDF alone, are just special cases of our SIR technique where $M = N = 1$. For example, proposing samples only from the lights can be achieved by SIR sampling with a single first-round sample drawn from the EM, and then trivially resampled in the second round with probability 1.0 because it is the only sample.

Chapter 5

Results

This chapter presents the results of our techniques in comparison with previous sampling strategies for rendering from environment maps. While both the rejection sampling and SIR methods increase the cost of sample generation, we demonstrate significant quality improvements for the same compute time under the assumption of BRDF representations that support efficient evaluation.

In all of our tests, the illumination comes from high dynamic range (HDR) environment maps, where the contrast ratio between bright and dark areas of the environment is high (several orders of magnitude). In principle, our techniques would work equally well under low dynamic range (LDR) lighting, where the ratio of bright to dark is at most 255 : 1. However, LDR EMs are not of sufficiently high-frequency to be interesting. Furthermore, the trend in rendering is to use HDR environments, which serve as a more accurate representation of real lighting conditions.

The images in this chapter were generated with our ray tracer which uses a voxel grid as the acceleration data structure for intersection queries. Our comparisons examine the output quality of the various rendering algorithms for a fixed amount of computation time. We performed these tests on a single-processor 3.0GHz P4 running Linux.

Figures 5.1, 5.2 and 5.3 contain images of Michaelangelo’s David in the Grace Cathedral environment. We use here the 700k-triangle version of the David model acquired from the Digital Michaelangelo Project [13]. In our implementation, intersecting a ray with the David model takes on average $6.1 \mu\text{s}$ on our test machine. The Grace Cathedral environment is a 1024x512 HDR map with a contrast ratio of $10^7 : 1$. In these tests, each algorithm was given 12.0 seconds to render a 176x248 image. This image resolution was chosen in order to be able to distinguish differences between the images when presented in print form.

In these three figures, the images on the top row show previous sampling strategies, while the images on the bottom row present our techniques. In each case, the upper-left image was rendered with 100 samples drawn from the environment map, while the upper-right image was rendered with 75 samples from the BRDF. The images on the bottom row were each rendered with 15 visibility samples, chosen from 800 candidates; on the left is the SIR approach, and on the right, rejection sampling.

Figure 5.1 shows the David rendered with a Phong BRDF model with coefficients $k_s = 1.0, k_d = 0.0$ and a specular exponent of 10. The BRDF in this case is a broad-lobed gloss. Observe that sampling from the EM produces a nicer image than sampling from the BRDF. This is because Grace is a highly specular environment map, and the BRDF here is glossy but still broad; drawing samples according to the BRDF is less informative than sampling according to the high-frequency lighting. The images generated by our techniques, however, are much better than either simple sampling strategy. Observe also that our methods are able to achieve this quality with several times fewer ray casts – in this case only 15 as opposed to 75 or 100.



Figure 5.1: David in Grace Cathedral. Phong exp 10, $k_s = 1.0$, $k_d = 0.0$. 176x248 image rendered in 12.0 seconds. Clockwise from top left: 100 EM samples; 75 BRDF samples; 15/800 rejection samples; 15/800 SIR samples.



Figure 5.2: David in Grace Cathedral. Phong exp 50, $k_s = 1.0$, $k_d = 0.0$. 176x248 image rendered in 12.0 seconds. Clockwise from top left: 100 EM samples; 75 BRDF samples; 15/800 rejection samples; 15/800 SIR samples.

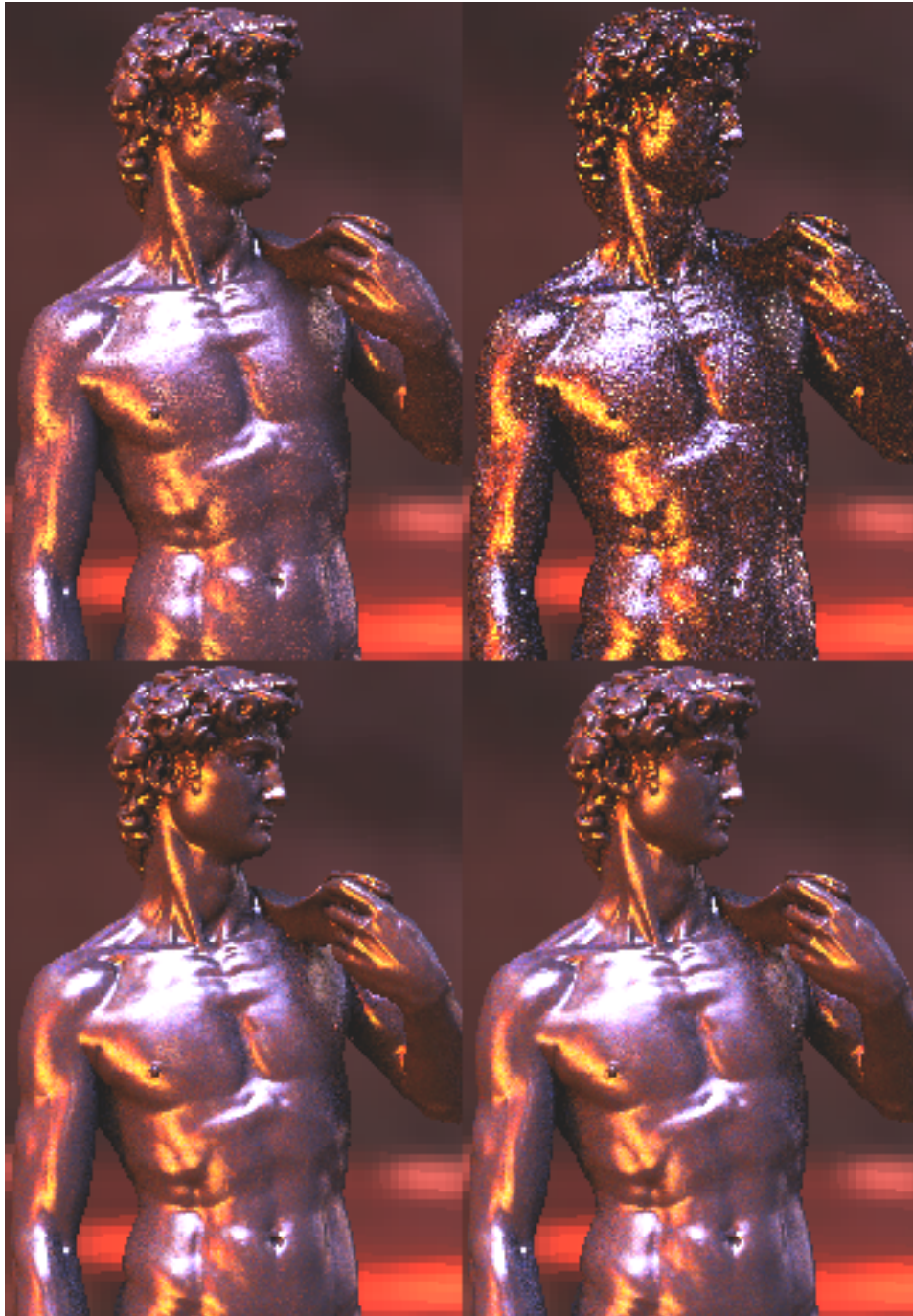


Figure 5.3: David in Grace Cathedral. Phong exp 50, $k_s = 0.5$, $k_d = 0.5$. 176x248 image rendered in 12.0 seconds. Clockwise from top left: 100 EM samples; 75 BRDF samples; 15/800 rejection samples; 15/800 SIR samples.

In Figure 5.2, David’s BRDF is still a purely specular Phong ($k_s = 1.0, k_d = 0.0$), but here the Phong exponent has been increased to 50. Here, the model’s surface is very shiny; accordingly, sampling purely from the BRDF produces a better image than sampling from the environment map. Again, however, our techniques go further. Both the lighting and the BRDF in this scene are high-frequency, and yet our methods produce images that are of much higher quality.

In Figure 5.3, David is again given a shiny Phong BRDF with exponent 50. This time, however, we add a diffuse term ($k_s = 0.5, k_d = 0.5$). Observe how noisy the image produced by sampling purely from the BRDF has become. BRDF sampling makes little sense in this case, because the presence of a diffuse term sends samples in all directions, making the variance high. Our methods still outperform the straightforward sampling strategies, because by sampling directly from the product distribution we disregard those directions which the BRDF accepts but the EM does not.

Figure 5.4 shows David appearing with the same Phong BRDF as in Figure 5.3, but under a different lighting scheme, the St. Peter’s Basilica environment map. Like the Grace Cathedral environment, St. Peter’s is a 1024x512 EM with a contrast ratio of $10^7 : 1$. However, the content of St. Peter’s is somewhat lower frequency than Grace, and as such sampling solely from the EM does not perform as well for this scene as it did in Grace. Consequent, our technique stands out even more.



Figure 5.4: David in St. Peters. Phong exp 50, $k_s = 0.5$, $k_d = 0.5$. 176x248 image rendered in 12.0 seconds. Clockwise from top left: 100 EM samples; 75 BRDF samples; 15/800 rejection samples; 15/800 SIR samples.

How do our techniques do in situations where the BRDF is very low frequency? The images in Figure 5.5 shows a purely diffuse David (Phong coefficients $k_s = 1.0, k_d = 0.1$) in Grace Cathedral. On the left is the image resulting from 100 samples drawn purely EM samples. On the right is the SIR technique with 25 samples ($M = 800, N = 25$). In this case, because the BRDF is purely diffuse and thus smoothly varying, one might think that sampling from the EM alone would be all that is necessary to render a decent image. Nevertheless, our SIR technique produces a comparable image with a quarter of the rays, indicating that sampling from the product distribution is always desirable.

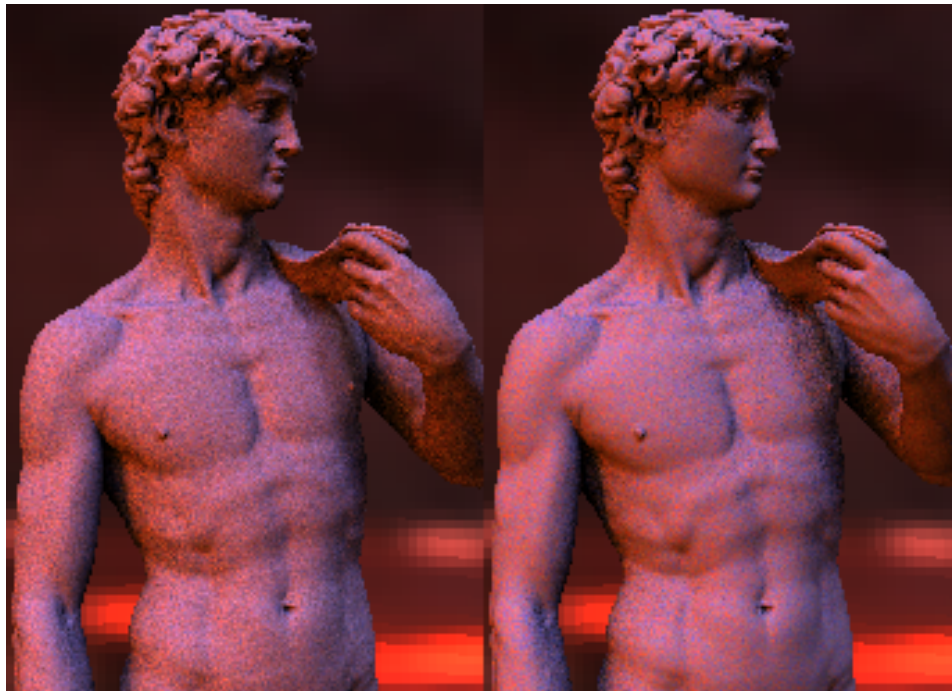


Figure 5.5: David in Grace Cathedral. Diffuse Phong ($k_s = 0.0, k_d = 1.0$). 176x248, 12.0 seconds. Left: 100 EM samples; Right: 25/800 SIR samples.

Figure 5.6 contains unzoomed images of David in Grace Cathedral.

Figures 5.7 and 5.8 contain a comparison of our algorithm to that of Veach and Guibas [44], in which samples drawn from the individual EM and BRDF distributions are combined via a weighted sum to produce an estimate of the reflected radiance. Our technique is able to achieve superior image quality with far fewer rays. The results obtained from the combined sampling approach are an improvement over straightforward sampling approaches; however, our method goes further to reduce variance than simply blending between the variances in the individual distributions.

In summary, the results presented here clearly demonstrate that the approach of sampling directly from the product distribution outperforms previous sampling strategies. What is more, we are able to achieve comparable quality with far fewer rays, meaning that our techniques are particularly beneficial to renderings of complex scenes where ray-scene intersection queries are expensive.

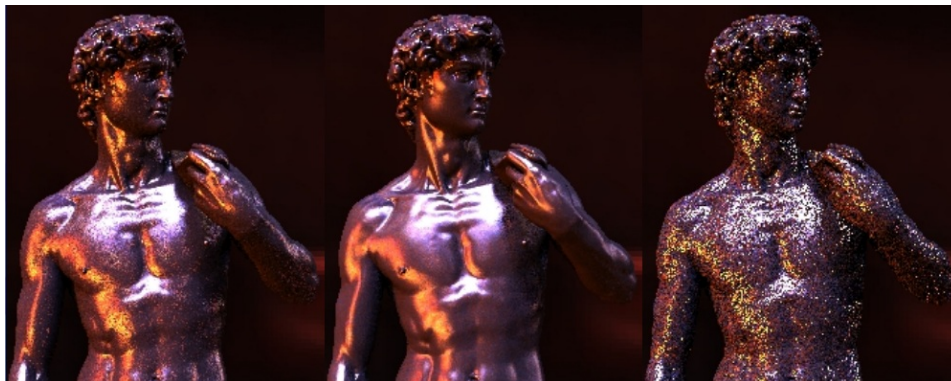


Figure 5.6: David in Grace Cathedral, viewed unzoomed. Phong exp 50, $k_s = 0.5$, $k_d = 0.5$. 228x272 image rendered in 18.0 seconds. From left to right: 110 EM samples; 13/900 SIR samples; 80 BRDF samples



Figure 5.7: Comparison between Veach and Guibas [44] (left column, 100 samples) and our SIR technique (right column, 15/800 samples). 176x248 image rendered in 12.0 seconds. Specular Phong ($k_s = 0.5$, $k_d = 0.5$), exponents 10 (top row) and 50 (bottom row).



Figure 5.8: Comparison between Veach and Guibas [44] (left column, 100 samples) and our SIR technique (right column, 15/800 samples). 176x248 image rendered in 12.0 seconds. Phong BRDF, exp 50, $k_s = 1.0$, $k_d = 0.0$



Figure 5.9: Our SIR algorithm compared to the full solution. 268x220 image of the Dragon model (871k triangles) in Grace Cathedral. Top: converged image (hours); Bottom: SIR algorithm using 150/3000 samples (90 sec)

Chapter 6

Conclusions

We have presented two Monte Carlo strategies for sampling the incident illumination from environment maps that take into account the product of the light and the surface reflectance. By providing a means of sampling from a more complex target distribution, our methods are able to achieve lower variance in the renderings of scenes with high frequency lighting and specular BRDFs, as compared to traditional importance sampling strategies.

Note that although our proposed bidirectional methods take longer to generate samples than simpler approaches, the number of samples required to achieve good quality is considerably less than when sampling according to a simple function. For large datasets with complex structures, the time required to trace shadow rays will dominate the rendering time. In such cases, our methods will provide greater benefit over importance sampling from the EM or BRDF alone.

On-going work for us is the examination of other sampling strategies that exist in the literature, such as iterative SIR, Metropolis-Hastings, and particle filtering [2]. The general idea behind these strategies is to use samples that have already been drawn as a basis for proposing further, fitter samples. It would be interesting to explore how these methods of sampling from more complicated distributions could be applied to other problems in computer graphics.

Bibliography

- [1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *ACM Transactions on Graphics (Proc. Siggraph)*, 22(3):605–612, July 2003.
- [2] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning, 2003.
- [3] J. Blinn and M. Mewell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, 1976.
- [4] D. Burke, A. Ghosh, and W. Heidrich. Bidirectional importance sampling for illumination from environment maps. *ACM Siggraph 2004 Sketches and Applications*, August 2004.
- [5] J. Cohen and P. Debevec. Light-gen. HDRshop plugin, 2001. <http://www.ict.usc.edu/~jcohen/lightgen/lightgen.html>.
- [6] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. of ACM Siggraph '98*, pages 189–198, July 1998.
- [7] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. of ACM Siggraph '97*, pages 369–378, 1997.

-
- [8] O. Deussen, S. Hiller, C. van Overveld, and T. Strothotte. Floating points: A method for computing stipple drawings. In *Proc. of Eurographics*, pages 41–50, August 2000.
- [9] P. Dutre and H. W. Jensen. *Siggraph Course Notes: State of the Art in Monte Carlo Global Illumination*. 2004.
- [10] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis*. Chapman and Hall, London, 1995.
- [11] C. Goral, K. Torrance, D. Greenberg, and B. Battaile. Modeling the interaction of light between diffuse surfaces. In *Proc. of ACM Siggraph '84*, pages 213–222, 1984.
- [12] N. Greene. Environment mapping and other applications of world projections. *IEEE CG&A*, 6(11):21–29, 1986.
- [13] Stanford Graphics Group. Digital Michaelangelo project, 2001. <http://graphics.stanford.edu/projects/mich/>.
- [14] J. Hammersley and D. Handscomb. *Monte Carlo Methods*. Methuen, London, 1964.
- [15] W. Heidrich and H.-P. Seidel. Realistic, hardware-accelerated shading and lighting. In *Proc. of ACM Siggraph '99*, pages 171–178, August 1999.
- [16] H. W. Jensen. Global illumination using photon maps. In *Eurographics Workshop on Rendering*, pages 21–30, 1996.
- [17] H. W. Jensen. *Siggraph Course Notes: State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis*. 2001.

-
- [18] H. W. Jensen. *Siggraph Course Notes: Monte Carlo Ray Tracing*. 2003.
- [19] M. Kalos and P. Whitlock. *Monte Carlo Methods*. John Wiley & Sons, New York, 1986.
- [20] J. Kautz and M. McCool. Approximation of glossy reflection with pre-filtered environment maps. In *Proc. of Graphics Interface*, pages 119–126, May 2000.
- [21] J. Kautz, P.-P. Vázquez, W. Heidrich, and H.-P. Seidel. Unified approach to prefiltered environment maps. In *Eurographics Workshop on Rendering*, pages 185–196, 2000.
- [22] A. Keller. *Quasi-Monte Carlo Methods for Photorealistic Image Synthesis*. PhD thesis, Aachen, 1998.
- [23] T. Kollig and A. Keller. Efficient illumination by high dynamic range images. In *Eurographics Symposium on Rendering*, pages 45–51, June 2003.
- [24] E. Lafortune, S.-C. Foo, K. Torrance, and D. Greenberg. Non-linear approximation of reflectance functions. In *Proc. of ACM Siggraph '97*, pages 117–126, August 1997.
- [25] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (Proc. Siggraph)*, 23(3):496–505, August 2004.
- [26] R. Lewis. Making shaders more physically plausible. *Computer Graphics Forum*, 13(2):109–120, 1994.

-
- [27] X. Liu, P. Sloan, H.-Y. Shum, and J. Snyder. All-frequency precomputed radiance transfer for glossy objects. In *Eurographics Symposium on Rendering*, June 2004.
- [28] G. Marsaglia. Generating discrete random variables in a computer. *Communications of the ACM*, 6(1):37–38, January 1963.
- [29] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics (Proc. Siggraph)*, 22(3):376–381, July 2003.
- [30] R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integral for all-frequency relighting. *ACM Transactions on Graphics (Proc. Siggraph)*, 23(3):477–487, August 2004.
- [31] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Transactions on Graphics (Proc. Siggraph)*, 23(3):488–495, August 2004.
- [32] B. Phong. Illumination for computer generated images. *Communications of the ACM*, 18(6):311–317, June 1975.
- [33] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proc. of ACM Siggraph '01*, pages 497–500, 2001.
- [34] R. Ramamoorthi and P. Hanrahan. Frequency space environment map rendering. In *Proc. of ACM Siggraph '02*, pages 517–526, 2002.
- [35] R. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley and Sons, New York, 1981.

-
- [36] A. Secord, W. Heidrich, and L. Streit. Fast primitive distribution for illustration. In *Eurographics Workshop on Rendering*, pages 215–226, June 2002.
- [37] P. Shirley. *Realistic Ray Tracing*. A K Peters, Natick, 2000.
- [38] P. Shirley. Personal communication. May 2004.
- [39] P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency environments. In *Proc. of ACM Siggraph '02*, pages 527–536, 2002.
- [40] P. Slusallek, T. Pflaum, and H.-P. Seidel. Using procedural renderman shaders for global illumination. In *Proc. of Eurographics*, volume 14, pages 311–324, 1995.
- [41] A. Smith and A. Gelfand. Bayesian statistics without tears: a sampling-resampling perspective. volume 46, pages 84–88, 1992.
- [42] L. Szecsi, M. Sbert, and L. Szirmay-Kalos. Combined correlated and importance sampling in direct light source computation and environment mapping. In *Proc. of Eurographics*, volume 23, pages 585–593, 2004.
- [43] M. Tanner. *Tools for statistical inference*. Springer Verlag, New York, 3rd edition, 1996.
- [44] E. Veach and L. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proc. of ACM Siggraph '95*, pages 419–428, August 1995.

-
- [45] M. Vose. A linear algorithm for generating random numbers with a given distribution. *IEEE Transactions on Software Engineering*, 17(9):972–975, September 1991.
- [46] A. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3(3):253–256, September 1977.
- [47] G. Ward. The RADIANCE lighting simulation and rendering system. In *Proc. of ACM Siggraph '94*, pages 459–472, August 1994.

Appendix A

Details on the Alias Method for Sampling

As mentioned in Section 3.2.2, we use the alias method of Vose [45] for sampling from discrete random distributions. Surprisingly, Vose's work appears to have been overlooked by the graphics community, where the transformation method is the de facto standard for importance sampling. The alias method offers a number of advantages over CDF inversion, the most notable being a fast $O(N)$ -time initialization algorithm.

In his paper, Vose's description of the approach contains ambiguous pseudocode; we therefore present a class called `DiscreteDistribution` which is similar to the class used in our implementation. `DiscreteDistribution` contains two methods: a constructor for initialization, and a method for drawing samples called `sample()`. The constructor operates on an input array of non-negative values which are used as distribution weights:

```
DiscreteDistribution::DiscreteDistribution(double *probs,
                                           unsigned num):
    m_num( num ), m_probs( probs )
{
    // Classify the normalized weights as being
    // either large ( > 1/N ) or small ( <= 1/N ).
    const double stdWeight = 1.0 / m_num;
    Vector small, large;

    for ( unsigned i = 0 ; i < m_num ; ++i )
        if ( m_probs[i] > stdWeight )
```

```

        large.addObject( i );
    else
        small.addObject( i );

// Assign alias indices.
m_alias = new unsigned[m_num];

while ( !large.isEmpty() && !small.isEmpty() ) {
    const unsigned sm = small.getAndRemoveLastObject();
    const unsigned lg = large.getAndRemoveLastObject();

    m_probs[lg] -= stdWeight - m_probs[sm];
    m_probs[sm] *= static_cast<double>( m_num );
    m_alias[sm] = lg;

    if ( m_probs[lg] > stdWeight )
        large.addObject( lg );
    else
        small.addObject( lg );
}

for ( unsigned i = 0 ; i < small.size() ; ++i )
    m_probs[ small[i] ] = 1.0;
for ( unsigned i = 0 ; i < large.size() ; ++i )
    m_probs[ large[i] ] = 1.0;
}

```

One can now draw samples from the distribution by calling the `sample()` method, which returns an index in $[0, N - 1]$.

```

unsigned CDiscreteDistribution::sample() const
{
    // Generate a random number in [0, m_num).
    const double u = uniform() * m_num;
    const unsigned i = static_cast<unsigned>( u );

    return ( u - i > m_probs[i] ) ? m_alias[i] : i;
}

```