UNIVERSITY OF CALIFORNIA,
IRVINE

Sampling Strategies for Efficient Image Synthesis

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Kartic Subr

Dissertation Committee:
Professor James Arvo, Chair
Professor Frédo Durand
Professor Gopi Meenakshisundaram
Professor Sharad Mehrotra

2008

The dissertation of Kartic Subr
is approved and is acceptable in quality and form for
publication on microfilm and in digital formats:

_____

_____

_____

_____

Committee Chair

University of California, Irvine
2008

DEDICATION

To my parents
Geetha Shankar and Sankara Narayanan
who have been my pillars of support

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I wish to thank several people whose contributions have positively impacted this thesis in different ways. First, I express my sincere thanks to my adviser, James Arvo, for his patient guidance. His expertise, experience and timely advice immensely enriched my experience in graduate school and made this thesis possible. I also thank Frédo Durand for his unhesitating support during times of difficulty. Frédo's insight and sincere advice provided exposure and encouragement during the crucial final stage of my Ph.D.

A special thanks goes out to Cyril Soler for valuable discussions and encouragement. I also thank Cyril for sharing the rendering software framework, which we extended, to produce the images in Chapter 4. I am grateful to François Sillion and Nicolas Holzschuch for arranging and supporting my visit to INRIA, Grenoble where much of the work for Chapter 4 was done.

My sincere thanks to Gopi Meenakshisundaram, for all the hours of motivating philosophical and research advice very early in my Ph.D. I also thank Gopi for serving as an internal committee member. I thank Kevin Novins for his understanding advice, encouragement and several discussions, while we shared an office, during the early stages of my Ph.D. I thank my friend and labmate, Pablo Diaz-Gutierrez, for valuable discussions and suggestions.

I am deeply indebted to my parents for their unconditional love, support and encouragement. I am also indebted to my fianceé, Ajitha Rajan, for her immense support during the writing of this dissertation. Thanks to Rajiv Bhat for his helpful comments on final touches to this dissertation.

ABSTRACT OF THE DISSERTATION

Sampling Strategies
for Efficient Image Synthesis

By

Kartic Subr

Doctor of Philosophy in Information and Computer Science

University of California, Irvine, 2008

Professor James Arvo, Chair

The gargantuan computational problem of *light transport* in physically based image synthesis is popularly made tractable by reduction to a series of sampling problems. This reduction is a consequence of using Monte Carlo integration at various stages of the transport process. In this document we describe analytic and computational tools for efficient sampling, and apply them at three stages of the light transport process: Sampling the image, sampling the camera aperture and sampling direct illumination due to distant light sources. We also adapt a standard statistical technique of inductive inference to assess different Monte Carlo sampling strategies that solve the light transport problem.

First, we derive a closed-form parameterization that allows the generation of *stratified samples* according to a linear density function with triangular support. We use this for stratified sampling of importance functions that are piecewise linear.

Next, we describe a new importance sampling strategy with the novel ability to draw samples from a *dynamic steerable importance function*. Contrary to existing techniques, the steerability of the importance function ensures that no wasted samples are generated in regions where the steering function is zero. We demonstrate its

effectiveness in the context of direct illumination from distant light sources, where the incident all-frequency illumination is steered by a dynamically orientable positive cosine lobe that is a function of the local normal.

We extend existing theory for studying the *radiance function* in the frequency domain: We define operators for frequency domain light transport and use them to present a novel analysis of finite aperture cameras in the Fourier domain. Using this analysis, we derive a new sampling algorithm that performs an order of magnitude better than current techniques for simulating depth of field correctly.

Finally, we discuss a novel adaptation of standard statistical hypothesis tests for assessing and comparing Monte Carlo estimators. We demonstrate that this framework can be used to make assertions about the means and/or variances of Monte Carlo estimators in image synthesis, upto a chosen level of significance. Besides comparing estimators, we verify that the framework can be used to detect errors in estimators and sampling algorithms.

Chapter 1

Introduction

Georges-Louis Leclerc, Comte de Buffon is often credited with having used the first known Monte Carlo[1] algorithm in his famous "needle experiment", in 1777, to estimate the value of π. He was one of several mathematicians in the seventeenth and early eighteenth centuries who were motivated by games of chance to form sequences of random events based on observations of successive trials. However, it was not until the nineteenth and early twentieth centuries when mathematicians made the observation that the mean of a function of continous random variables took the form of an integral. It was followed by the realization that, in principle, one could randomly draw numbers and proscribe transformations such that the random numbers could be used to approximately solve integration problems that contained no inherent probabilistic structure.

By the late nineteenth century, Lord Rayleigh [87] showed that a one dimensional random walk could be used to approximately solve a parabolic differential equation. Following this result, Courant et al. [4] demonstrated that a particular finite differ-

[1]The term "Monte Carlo" was coined almost 200 years later. Today, Monte Carlo methods encompass all techniques that use statistical sampling to approximate solutions to quantitative problems.

ence equation could be used to approximate a solution to the Dirichlet boundary-value problem of partial differential equations. Subsequently, they showed that a recursive form of the solution to a two dimensional random walk on a square grid within a closed regions, under certain conditions, produced an identical difference equation. Around the same time, Kolmogorov derived the relationship between Markov stochastic process and certain integro-differential equations. Petrowsky generalized the result of Courant et al. by showing the asymptotic connection between a random walk whose sequence of locations formed a Markov chain and the solution to an elliptic partial differential equation; Petrowsky called this the *generalized Dirichlet problem*.

In the early thirties, Enrico Fermi used the Monte Carlo method to run simulations of particle transport through isotropic media (neutron diffusion) that were central to the research towards building the atomic bomb. Fermi later developed the *Fermiac* which was a Monte Carlo mechanical device used to calculate criticality in nuclear reactors. The associated multidimensional problems proved too formidable for the popular difference equation approach and inspired John von Neumann and Stanislaw Ulam to suggest that sampling experiments using random walk models on the newly developed digital computer could provide useful approximations.

Ulam is credited with inventing the name "Monte Carlo"[2] and, with the help of von Neumann and Nicholas Metropolis, the name soon caught on to refer to methods that employed statistical sampling to approximate solutions to quantitative problems.

[2]Ulam described the incident as follows: "The first thoughts and attempts I made to practice [the Monte Carlo Method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than abstract thinking might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later [in 1946, I] described the idea to John von Neumann, and we began to plan actual calculations."

Figure 1.1: Left: *A portrait of Georges-Louis Leclerc, Comte de Buffon by the French painter François-Hubert Drouais.* Middle: *Stanislaw M. Ulam.* Right: *John von Neumann. Although Count Buffon is commonly credited with the earliest known use of a Monte Carlo algorithm, Ulam was responsible for naming and formalizing the method. Much of the theoretical foundation for the method was laid by John von Neumann.*

Ulam and Metropolis published the first paper [69] describing this method, as it is known today. The use of Monte Carlo methods spread rapidly to several different scientific disciplines.

Developments in the field of computational complexity, in the seventies, began to provide a more precise and persuasive rationale for using the Monte Carlo method. The theory identified a class of problems for which exact solutions often led to algorithms that executed in times that were, at best, exponential with respect to the size of the input. The identification of a certain structure in these problems could be exploited to provide exact solutions in times that were bounded, above, by polynomials in the size of the input. Without this structure, problems that belonged to this class seemed to pose a formidable hurdle to solve.

There was a rising interest in trying to resolve the question of whether Monte Carlo could be used to estimate solutions to problems in this intractable class to within some

Figure 1.2: *The Monte Carlo casino in Monaco. Ulam named the method after this casino where his uncle would borrow money to gamble.*

statistical accuracy in a time bounded, above, by a polynomial in the size of the input. Several attempts were made in the eighties: Karp estimated reliability in a planar multiterminal network with randomly failing edges [53] ; Dyer et al. estimated the volume of a convex body in m-dimensional euclidean space [38]; Broder estimated the permanent of a matrix or, equivalently, the number of perfect matchings in a bipartite graph [17].

Integro-differential equations were applied to problems in radiative transfer [22] which inspired research in neutron transport [100] and hydrologic optics [83]. Recognizing the similarities of these problems to that of light transport for global illumination (see Section 1.2), Kajiya presented a simplified integro-differential equation [52] that he called *the rendering equation*. The rendering equation sufficiently represented the flow of radiant light energy under the many assumptions that were considered practical for use in computer graphics related problems. Further, it provided the means to express

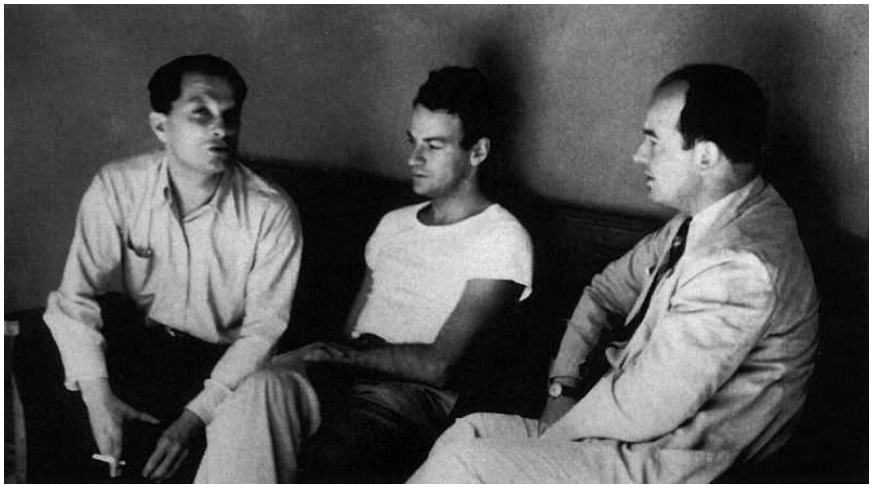


Figure 1.3: *Stanislaw M. Ulam, Richard P. Feynman and John von Neumann. The Monte Carlo method was inspired by the problems encountered conducted during the development of the atomic bomb. (Picture scanned at the American Institute of Physics)*

the transformations of radiant light energy while accounting for several geometric optical effects. The realization that the solution of the rendering equation (and its many variants) would yield global illumination effects like multiple inter-reflection, refractions, scattering within media, penumbrae of shadows, etc. sparked off a flurry of Monte Carlo research within the graphics community.

Despite the mathematical sophistication that the Monte Carlo method is often imbued with, it is the simplicity of the method that has brought about much of its popularity. Ulam, von Neumann and others recognized that the Monte Carlo method could be modified in ways that produced solutions to the original problems with a specified error bound, at considerably reduced cost. Although some of these *variance reduction techniques* were already commonly used by statisticians, others owe their origin to the Monte Carlo method. Collectively, these procedures now represent the central focus of the Monte Carlo method by exploiting available structure that the method fundamentally ignores.

During the early years of the "computer age", the application of variance reduction techniques was essential in practicably estimating solutions to large numerical problems. The design of these techniques was far from trivial and thus took a considerable amount of time to develop. Although many of these techniques were general, the efficiency to be gained by tailoring them to a particular application was so large that analysts typically spent a large amount of time performing the customization.

The dramatic increase in computational power over the last couple of decades triggered two remarkable changes: it became feasible to run Monte Carlo simulations on small, commodity microcomputers; supercomputing power became powerful enough that problems of much larger scale were solvable. The result of this stupendous increase in computational power also spawned the need for assessing whether it was more beneficial to just throw large amounts of computing power at problems rather than recruit analysts to design specialized variance reduction schemes.

Nevertheless, the motivations for sophisticated variance reduction techniques are many: Problems of substantial size still remain; certain applications demand that problems be solved in lesser time than currently possible; certain other problems demand extremely high statistical accuracy in the estimated solutions. Thus, the benefit of using and designing new variance reduction techniques cannot be undermined.

Variance reduction strategies can be classified, based on their philosophy, into at least two different categories: Some strategies modify the way in which random samples are generated and adjust the parameter estimator of interest in a way that variance is reduced. e.g. Importance sampling, stratified sampling, correlated sampling, etc.; Other strategies operate by leaving the sampling mechanism unaffected—instead, they collect ancillary data that are used to estimate already known parameters. The variance reduction due to the latter is achieved by incorporating these data into the estimator of the unknown parameter of interest. e.g. Control variates.

In this dissertation, we focus on the first of the two classes of variance reduction schemes. *We exploit certain structure that is known to exist in some light transport problems in computer graphics to propose sampling strategies that cause a variance reduction in estimated solutions for those problems.* We also present an adaptation of the statistical framework for testing hypotheses, that can be used to assess qualities of estimators, upto specified levels of statistical significance.

1.1 Digital image synthesis

Digital photography produces images where each pixel represents the incoming radiant light energy over a small area on the sensor within a small set of directions in a controlled length of time. Light energy propagates from light sources in the scene and potentially travels through a sequence of infinite bounces on multiple objects before passing through the camera lens and aperture and finally impinging on the camera sensor. The image obtained is a snapshot of the result of several physical processes involving the transport of light energy from luminaires through the scene to the sensor in the camera.

A popular problem in the field of computer graphics is to produce images by mimicking photography starting from geometric and physical descriptions of the scene of interest (see Figure 1.4). This transformation, from geometric and physical information into images is called *image synthesis*. One way of solving this problem to obtain "realistic" images, is to make certain assumptions about the scene and to simulate the process, respecting physical laws to some degree. This is referred to as physically based image synthesis. The ultimate goal of physically based image synthesis is to produce images that are indistinguishable from photographs of the real world, by simulating the physical process involved.

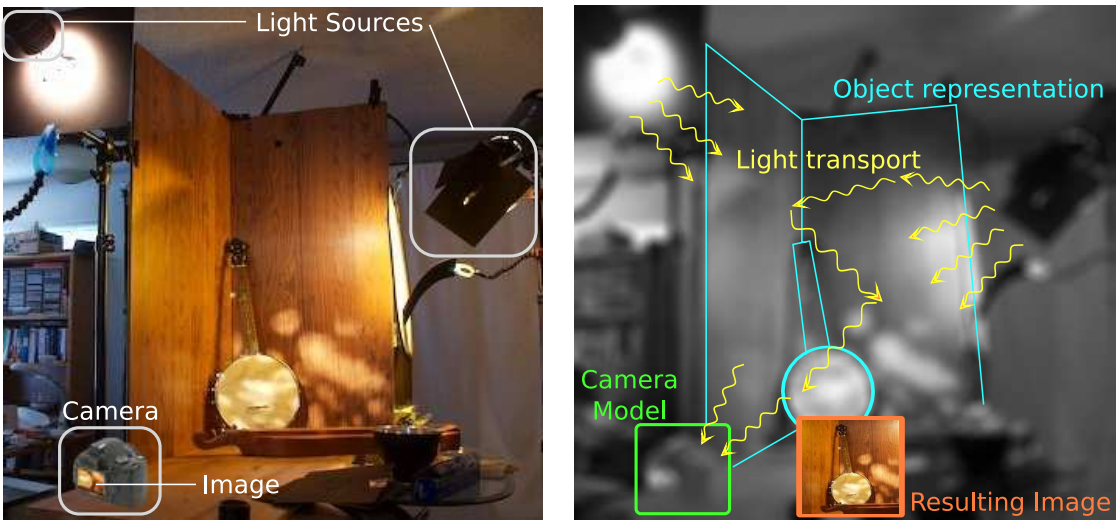


Figure 1.4: *Physically based mage synthesis is the process of producing images by simulation of light transport to mimic the photographic process. Suitable models are chosen for the camera and objects in the scene.*

Physically based image synthesis incorporates results from four large fields of study: (1) mathematical, physical and structural representation of objects, (2) digital signal processing, (3) the interaction of matter and light and (4) the human visual system. Extensive research in these fields has resulted in a large body of literature and, consequently, sophisticated methods for several interesting problems in the field of physically based image synthesis.

1.2 Light transport

A significant fraction of the computational effort in physically based image synthesis is dedicated towards simulation of specific optical phenomena. The simulation of the propagation of light energy is referred to as the *light transport* problem. Several light transport algorithms exist for simulations with varying degrees of accuracy and subject to dramatically different constraints. For example, the focus is on absorption

Figure 1.5: *Measured, simulated and error images of a scene. This famous scene, called the Cornell Box, was setup by researchers in Cornell University's Light Measurement Laboratory. The potential for multiple interreflections between diffuse surfaces and the availability of measured parameters of illumination of reflection made this scene a popular choice for verifying global illumination algorithms. (Source: Cornell University Light Measurement Laboratory)*

and scattering processes in biomedical imaging while, in image synthesis, a lot of effort is directed towards improving reflection models. Applications in hydrologic optics [83], like biomedical imaging applications, consider scattering processes in great detail but demand higher precision of the estimates.

Global illumination algorithms are those that solve the light transport problem for physically based image synthesis. These algorithms approximately simulate the potentially infinite interactions of light with matter, before finally entering the optical system of the virtual camera. The degrees of accuracy to which simulations are run in physically based image synthesis—since the goal is to produce images that are indistinguishable from photographs—is governed by the limits of human perception.

Solutions to several light transport problems are inspired by transport solutions adopted in heat transfer [31] and neutron transport [100]. In a seminal work in image synthesis, Kajiya proposed an integral equation [52] which expressed the radiant light energy leaving a point, along a certain direction, as the sum of the emitted radiant energy in that direction and reflectance-weighted radiant energy incident at the point from all possible directions. The presentation of the light transport prob-

lem in this form, captured the commonality of different global illumination algorithms that existed at the time.

The potentially unpredictable behaviour of the functions in the rendering equation, coupled with the high dimensionality of the domain and the complex interaction of multiple physical processes make general analytical solutions unfathomable. The equation is usually solved either using Monte Carlo or finite element methods.

1.3 The Monte Carlo method

Ulam and Metropolis proposed a strategy [69] that used statistical sampling to numerically solve quantitative problems, which they called the Monte Carlo method. They were inspired by large and complex quantitative problems for which analytical methods were hopeless and typical numerical methods collapsed.

Monte Carlo methods typically consist of two distinct processes: transformation of the problem into an expectation and simulation. The former reduces the problem to one of estimating $E(X)$ where X is a random variable. Although this is usually simple, as in the case of Monte Carlo integration, it can be a tricky problem if the goal is, say, to solve parabolic or elliptical equations.

The second step involves the simulation of random variables under the distribution of X. Mathematically, this means that a sequence of random variables $(X_i,\ 1 \leqslant X \leqslant N)$ is obtained, such that the X_i follow the distribution of X. This is typically achieved by computationally transforming random variables uniformly distributed in the unit interval into the appropriate domain. Finally the required expectation is approxi-

mately estimated as

$$E\left(X\right) \approx \frac{1}{N}\left(X_1 + X_2 + ... + X_N\right) \tag{1.1}$$

One of the most popular uses of Monte Carlo methods has been for estimating the value of integrals. The rest of this section provides a basic introduction to Monte Carlo integration with the help of simple examples. Consider the numerical estimation of the integral

$$\int_0^1 f(x)\,\mathrm{d}x. \tag{1.2}$$

There exist many numerical methods of the form $\sum_0^n w_i f(x_i)$ where the w_i are non-negative weights that sum to unity and $x_i \in [0,1]$. e.g. Trapezoidal integration ($w_i = 1/n, 0 < i < n$, $w_0 = w_n = 1/(2n)$ and $x_i = 1/n$), Gaussian integration, Simpson's rule, etc. The basic Monte Carlo integration algorithm assumes the same form, with $w_i = 1/n, 1 \leqslant i \leqslant n$ and x_i that are randomly drawn from the domain $[0,1]$. The convergence of this Monte Carlo integration scheme is $O(1/\sqrt{n})$. Although the rate of convergence seems poor when compared to other methods for this one dimensional integration, the great advantage of this method is that it is insensitive to the dimensionality of the domain. Typically numerical integration methods will require n^d points when the domain is the d-dimensional unit hypercube $[0,1]^d$ for estimates with constant error.

Consider the multidimensional integral

$$I = \int_{\mathcal{D}} f(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{1.3}$$

where the domain $\mathcal{D} = [0,1]^d$ and the variable of integration $\mathbf{x} = (x_1, x_2, ..., x_d) \in \mathcal{D}$. Following the first step of the Monte Carlo method, we set $X = f(U_1, U_2, ..., U_d)$ where $(U_1, ..., U_d)$ are independent random variables distributed uniformly in $[0,1]$ so that we can write

$$E(X) = E(f(U_1, U_2, ..., U_d)) = \int_{\mathcal{D}} f(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{1.4}$$

Thus, we have completed the first stage of the Monte Carlo method, by writing the quantity that we wish to compute as an expectation.

In the simulation phase, a sequence (U_i) is generated such that each U_i is uniformly distributed in $[0,1]$. Then random variables X_i are constructed so that $X_1 = f(U_1, U_2, ..., U_d)$, $X_2 = f(U_{d+1}, U_{d+2}, ..., U_{2d})$, etc. The required integral is estimated as

$$I \approx \frac{1}{N}(X_1 + X_2 + ... + X_N) \tag{1.5}$$

Often, the integrand is expressible as the product of two functions, $f(\mathbf{x}) = g(\mathbf{x})h(\mathbf{x})$ where h is non-negative and integrates to unity. In such cases, the integral can be written in the form $E(g(Y))$ if Y is a random variable distributed according to $h(\mathbf{x})$. Consequently, the integral can be approximated as

$$I \approx \frac{1}{N}(g(Y_1) + g(Y_2) + ... + g(Y_N)) \tag{1.6}$$

where the Y_i are distributed according to $h(x)$. Thus the problem of integration is reduced to one of generating samples according to a certain distribution. This technique is referred to as *importance sampling* in Monte Carlo literature, and $h(x)$ is called the *importance function*. Two of the most attractive features of importance sampling

are that 1) the distribution used to reduce variance need only be an approximation, and 2) no bias is introduced so long as we can correctly compute the density of the samples generated.

The use of this deceptively simple method for general integration problems often warrants sophisticated mathematical verification to ensure that the correct quantity is being estimated, and with an acceptable amount of error in the estimates.

The strong law of large numbers imposes a theoretical limit on the Monte Carlo method: The method can only be used with integrable random variables. The central limit theorem can be used to derive a random variable, that is asymptotically equal to the error, which suggests that the distribution of $E(X) - \frac{1}{N}(X_1 + X_2 + ... + X_N)$ resembles a centered guassian.

1.4 The problem of sampling

The sampling process assumes different flavours, depending on the application domain. In statistics, a number of interesting sampling strategies were born out of the need for estimating characteristics about populations [27] that were too large for complete surveys to be conducted. In *survey sampling*, a small but carefully chosen sample[3] is used to represent the population. The sample is selected so that it reflects the characteristics of the population that are of interest. In this context, the benefit is that characteristics about the general population may be inferred from the samples, without having to incur the cost of a comprehensive survey.

In signal processing, sampling refers to periodic measurements of a signal[4]. Thus,

[3] In statistics the term sample is used to mean a set of observations. In computer graphics, each of the observations is called a sample.

[4] A physical quantity, usually measurable through time or space.

sampling is central to all digital signal processing problems that deal with analog signals. When digital signals are involved, clever sampling strategies allow for compact representations. If the original signals need to be reconstructed from sampled representations, care is taken that the sampling strategies possess desireable characteristics so that the reconstruction is of high fidelity.

Monte Carlo techniques use samples, drawn from meticulously designed parent distributions, to solve a host of different computational problems. One of the most popular uses has been to solve integration problems. As seen in Section 1.3, Monte Carlo integration reduces the integration problem to one of sampling.

Sampling methods are broadly classified as either probabilistic or non-probabilistic. In probabilistic sampling, each member of the population has a known non-zero probability of being selected. eg. random sampling, systematic sampling and stratified sampling. In non-probabilistic sampling, members are selected from the population in some deterministic manner. eg. convenience sampling, judgment sampling, quota sampling and snowball sampling. The advantage of probabilistic sampling is that *sampling error*[5] can be calculated.

1.5 Sampling problems in image synthesis

In a Monte Carlo *path tracer*, an image is formed by computing the solution to the light transport problem at each pixel, which is obtained by adding contributions from a set of light paths. Each of the paths in a path tracer is constructed using a random sequence of sampling procedures. The paths begin at the eye and are shot through chosen locations on the virtual camera sensor. Subsequent vertices are chosen by randomly choosing a direction and finding the first point along that direction where

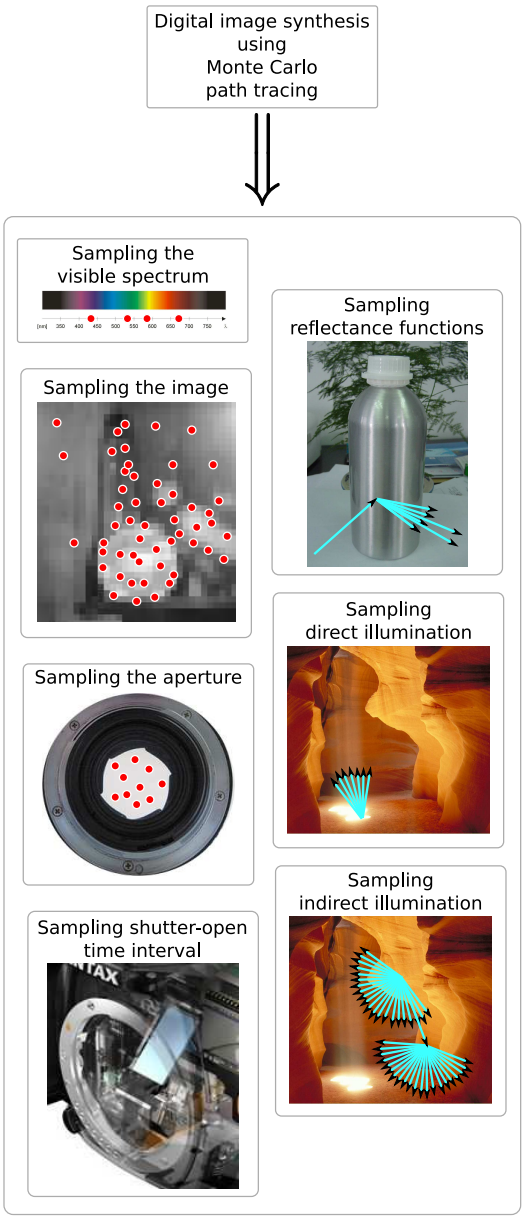
[5]Sampling error is the degree to which a sample might differ from the population.

the next interaction occurs. The random direction chosen at each vertex, is based on a distribution that characterises the interaction of light with matter at that vertex. A plethora of sampling strategies have been proposed in the literature that account for several different types of light-matter interactions along the path.

Using Monte Carlo integration in the physically based image synthesis process reduces the problem of light transport to a series of sampling problems: (1) sampling the pixel area on the sensor or prefiltering; (2) sampling the camera aperture for simulating depth of field; (3) sampling in time to simulate controlled camera shutter speed; (4) sampling the reflectance or transmittance function to simulate glossy reflection or transmission; (5) sampling the solid angle subtended by luminaires for simulating penumbrae; (6) sampling paths for indirect illumination (due to interreflection); (7) sampling in wavelength to account to simulate spectral effects (see Figure 1.6).

1.6 Importance sampling in image synthesis

Variance reduction strategies are crucial elements of Monte Carlo global illumination algorithms. Without them, it is generally regarded as impractical to obtain adequately converged Monte Carlo solutions, particularly for environments that incorporate challenging lighting distributions and/or surface scattering functions. Since the earliest systematic study of Monte Carlo algorithms in image synthesis [52, 29, 94], both importance sampling and stratification have been recognized as being particularly relevant variance reduction strategies, although it has often been a challenge to incorporate them without simultaneously introducing statistical bias [57, 58]. Both importance sampling and stratification are now commonplace in illumination computations, and often appear in several guises within a single algorithm. While improvements to both strategies continue to be an active area of research, importance



Visible spectrum: Monte Carlo sampling of the visible wavelengths of light allows simulation of optical phenomena like dispersion [46, 39, 119, 33].

Image space: Adaptive image subsampling algorithms allow fewer rays to be cast and result in reduced aliasing artifacts [34, 28, 72, 101, 10, 70, 71].

Aperture: Depth of field effects are simulated by integrating light paths sampled over the aperture [81, 28].

Exposure time: Integrating light paths sampled over time produces motion blur effects [30, 82, 55].

Reflectance functions: Glossy reflection and transmission are simulated by integrating paths distributed according to the reflectance distribution [11, 116, 62, 12, 64, 24].

Light sources: Direct illumination computation involves integration of paths over the solid angle subtended by the light source [94, 2, 6, 8, 48, 26, 24].

Indirect illumination: Integrating paths that perform multiple bounces before reaching a light source simulates indirect illumination due to interreflections [79, 105].

Figure 1.6: *Each step of the image synthesis pipeline can be simulated using Monte Carlo integrations [30, 65]. Therefore the problem can be reduced to a series of sampling problems over different domains. A number of solutions have been proposed for each of these sampling problems. The integration domains are described on the right with references to existing solutions for each domain.*

sampling offers the largest potential payoff, with the total elimination of variance being theoretically achievable [88]. The remainder of this section describes the evolution of importance sampling in image synthesis chronologically.

1990-1995

Shirley compared the effectiveness of importance sampling in reducing variance, in his thesis [93], against stratified sampling. He described how the method of inverting the cumulative distribution may be used to generate samples according to a given distribution. The technique was later extended by Arvo and used for stratified sampling of 2-manifolds.

Smits et al. defined importance [98] with respect to a viewpoint by propagating importance from the viewpoint. similar to the transport of light energy from light sources. While the paper showed a significant gain in computational efficiency by performing low resolution radiosity solutions for less important areas, the notion of importance is very different from the use of importance functions in Monte Carlo algorithms.

Dutré et al. presented an importance sampling algorithm [37] for efficiently estimating solutions to the rendering equation. They introduced the concept of adaptive probability distribution functions (pdfs), where the sampling density underwent sequential modifications after each sample was drawn. To begin with, samples are drawn from a constant density. Then, the domain is partitioned and the sample drawn, at each step, is used to estimate the integral; based on the computed estimate density in the corresponding interval of the pdf is modified.

Veach and Guibas presented a new perspective on importance sampling [112], with a conservative strategy that avoided insufficient sampling of regions where the integrand

was large. They decomposed the integrand into functions, identified regions in the domain where any of these functions was large and ensured heavy sampling of these regions. Although regions where the integrand is low could potentially be over-sampled, they demonstrated the effectiveness of their technique using compelling experimental evidence. They called their technique *Multiple Importance Sampling* (MIS).

MIS could be viewed as an extension to stratified sampling where the strata are not strictly partitions of the sampling domain. That is, there could be overlap between strata. In such a situation, samples drawn from different strata may correspond to the same region of the sampling domain and, thus, need to be combined appropriately to avoid bias. To address this problem, the estimate—which is obtained as a weighted average of the results of a host of different estimators—is chosen to be produced by an estimator that outperforms the others.

1996-2000

Shirley et al. further stressed the effectiveness of importance sampling as a variance reduction strategy in a paper [95] where they derived densities for estimating direct illumination. They derived the densities to sample the solid angle subtended by illuminaires of a few common shapes, making the calculation of direct illumination from several sources more efficient. Their importance function did not account for the BRDF or visibility.

La Fortune et al. invented a class of primitive functions [62] with non-linear parameters for representing reflectance functions. They approximated the reflectance distributions by sets of cosine lobes which made them simple, flexible and easy to use in a Monte Carlo algorithm for sample generation. The class was powerful enough to represent a wide variety of materials. This work was a significant contribution in the context of importance sampling because sampling reflectance distributions, which

poses a significant hurdle for realistic materials, was simply reduced to appropriately sampling cosine lobes. Another similar method that unified definitions for a good visual approximation for many materials was presented by Neuman et al. [74]. Their model allowed fast importance sampling of physically plausible reflectance functions.

Since the notions of probability density and variance are not applicable in the context of deterministic quasi Monte Carlo (QMC), the extension of importance sampling is not straightforward to a QMC setting. Szirmay-Kalos et al. presented a QMC algorithm with importance sampling [106], by using variable transformation. The transformation was designed so that its Jacobian matrix was inversely proportional to the integrand, thus resulting in a constant transformed integrand (which corresponds to minimum quadrature error). They derived this transformation was derived by first propagating direct illumintion using a photon tracing procedure.

Pietrek and Peter presented a method to adaptively construct pdfs for sampling indirect illumination [79]. This work was an extension to the work by Dutré et al. and was similar, in concept, to Szirmay-Kalos et al.'s method. Pietrek and Peter built a hierarchical set of density functions that were successively refined as the estimate for indirect illumination was estimated to more precision. By considering diffuse surfaces and tesselating the surfaces into large patches, they reduced the 6 dimensional density down to two dimensions per patch. They demonstrated using experiments with two representations for the density functions— Haar and linear B-spline bases— that there was no advantage of using higher order basis functions over piecewise constant Haar bases. They concluded that, while the use of B-spline bases avoided certain artifacts in specifically constructed examples, Haar wavelet bases performed better for larger scenes without visible artifacts.

Bekaert et al. introduced the notion of *weighted importance sampling* (WIS) to the image synthesis community, and used it to estimate form factors between patches

while accounting for partial occlusion. In WIS, samples are drawn from an "easy-to-sample" source function and are used in a way that suggests that they were drawn from a different, more effective, target importance function. For unbiased estimates, multiplication is required by weights given as the ratio of the target and source functions evaluated at the sample locations. In their paper, Bekaert et al. used uniform area sampling of patches in a radiosity solution as the source importance function and mimicked a target importance function corresponding to cosine distributed directional sampling. Their experimental results indicate a reduced variance, although they reported a bias when only a small number of samples were drawn.

2001-2008

Agarwal et al. defined an importance metric [3] for sampling direct, distant illumination by conservatively accounting for visibility and illumination. While sampling direct illumination, giving importance purely to the magnitude of solid angle subtended undersamples small bright lights. On the other hand, considering illumination without accounting for the solid angle subtended oversamples small bright sources. In an attempt to strike a balance, Agarwal et al. proposed the use of an importance function that considered a carefully chosen combination of both, illumination energy and solid area subtended. The precise blend was based on an empirical analysis of visibility maps.

The environment map was first stratified based on thresholding functions applied to the radiance values associated with each pixel. Then, sample allocation within strata was based on the importance metric, and the pixels of each stratum were clustered according to the allocation. During rendering, a random location was chosen within each cluster and used to compute the estimate for direct illumination. Their paper also describes a few optimizations: (1) approximating the environment map with a

number of directional sources in a preprocess step; (2) eliminating banding artifacts in (1) by using jittered sampling for visibility testing; (3) sorting light sources based on their contribution and only considering the first few in the list so that the error in the estimate is below a certain threshold.

Lawrence et al. presented a BRDF factorization technique [64] that allowed efficient importance sampling of bidirectional reflectance functions (BRDFs) while simultaneously maintaining compact representation. They demonstrated, using analytic and measured BRDFs, that their technique was more efficient than fitting Lafortune or Blinn-Phong lobes and also more compact than tabulating the reflectance functions. They represented the 4D, reparameterized BRDFs as the sum of a number of terms, each of which was the product of a view-dependent 2D function and two 1D functions. Importance sampling was achieved by numerical inversion of the 1D factors.

Ostromoukhov et al. presented a robust and practical algorithm [77] for generating samples according to a 2D density function. While the method is effective in generating samples that satisfy desirable blue noise properties and with aspecified sampling density within a local neighborhood, it is unclear how the weights associated with these samples are to be normalized when used in the context of Monte Carlo integration. The paper demonstrates the use of this sampling algorithm to estimate direct illumination from environment maps.

The use of control variates as a variance reduction strategy has not been explored in depth by the image synthesis community. Szécsi et al. discussed the effectiveness of using control variates (also called correlated sampling) [104] for sampling in Monte Carlo integration in their paper which also presented a scheme to combine the benefits of importance sampling and correlated sampling. Their approach was to introduce a parameter which governed the weightage of estimates resulting from each sampling strategy and then optimize the resulting estimate for minimum variance.

Typically importance sampling had been used for drawing samples distributed according to the local reflectance distribution, or illumination (both distant and nearby illuminaires) independently. Clarberg et al. [23] generalized wavelet products to higher dimensions and applied it to sample from a product of the local reflectance function and distant illumination. The algorithm exploits the property of wavelet products that they can be evaluated top down. The paper then warped a set of uniformly distributed points to match the approximated product distribution. However, the constraint that all BRDFs in the scene be resampled as wavelets, makes it impractical in scenes with a large number of BRDFs. Also, the choice of wavelets as bases inherently restricts rotation into local coordinate frames (in which BRDFs are conveniently represented). To work around this problem, wavelet decomposition of the environment map was stored for different orientations. Cline et al. presented [25] a similar approach, except that they use hierarchical partitioning of the environment map à la McCool and Harwood [67] in conjunction with summed area tables instead of wavelets.

Another work that sampled from both illumination and reflectance function was invented in the same year by Burke et al. [19] who performed the sampling in two stages: samples were drawn from the first distribution and then resampled according to the second distribution. They introduced the terminology *sampling-importance resampling* to represent a process that is quite similar to WIS. Two forms of *bidirectional importance sampling* (BIS) were presented: one using rejection and another using resampling. While rejection leads to increased sampling expense, resampling only allows samples to be drawn from an approximate distribution.

1.7 Original contributions

The original contributions of this thesis are described chapterwise, below.

- **Linear stratified sampling (Chapter 2)**:

  We derive parameterizations whose Jacobian determinants are proportional to a linear density. Then we use this to generate linear stratified samples over triangular and tetrahedral domains, where the linear densities are specified by vertex weights.

- **Steerable importance sampling (Chapter 3)**:

  We define a new technique that uses a steerable function as an importance function.

  Parameterized probability tree: We define a data structure, called the parameterized probability tree, where the traversal is probabilistic with branching probabilities defined as a function of some parameter.

  Efficient direct illumination: We construct a low variance estimator for direct illumination from distant illumination by defining a piecewise linear, steerable importance function which is the product of incident illumination and the local clamped cosine lobe. The reduction in variance is due to a combination of the importance function and stratification that is achieved using the parameterized probability tree and linear stratification algorithm.

- **Adaptive, bandwidth-based sampling(Chapter 4)**:

  We use conservative estimates of local bandwidth for efficient simulation of depth of field effects.

  Fourier depth of field: We present a novel analysis of finite aperture camera models in the Fourier domain.

Adaptive image subsampling: Using the theoretical analysis of depth of field we design a new frequency propagation scheme that allows conservative prediction of bandwidth, locally over the image. We show that these bandwidth estimates are used to obtain sampling densities that are close to optimal for non-objectionable reconstruction of the images.

Adaptive aperture sample allocation: We use the bandwidth prediction algorithm to estimate the variance of the integrand over the aperture, in depth of field simulations. Since the error in Monte Carlo is proportional to the variance of the integrand and inversely dependant on the number of samples, we increase the number of samples where the variance is estimated to be high.

- **Assessing Monte Carlo estimators (Chapter 5)**:

  We use an adaptation of the statistical hypothesis testing framework to compare first and second order statistics of estimators.

  Comparing estimators: We design tests to compare means and variances of estimators. These tests allow the assertion of hypotheses regarding bias and efficiency of estimators, upto a chosen level of significance. We confirm the dependability of the tests by comparing estimators with known qualities.

  Verifying sample distributions: By adapting a goodness-of-fit test, we verify the correctness of analytic sampling algorithms by comparing them against samples generated using rejection.

  Detecting errors in estimators: We introduce errors into common estimators and demonstrate the ability to use the framework for error detection.

Chapter 2

Stratified sampling

In certain sampling scenarios there is benefit in partitioning the sampling domain and drawing samples from each partition rather than directly sampling the domain. There are several reasons for this: (1) To obain samples that are less likely to be "clumped"; (2) A heterogenous distribution over the domain may be split into homogenous partitions, which increases the precision in the estimates of characteristics over the entire domain; (3) Sampling problems may differ markedly within the domain; (4) Higher precision might be required in certain parts of the domain than others; (5) Partitioning the domain may simplify implementation. Of these, (4) and (5) appear rarely in image synthesis.

Definition 2.1. *Stratified sampling is the procedure of drawing independent samples from partitions of the sampling domain. Each partition is referred to as a "stratum", and the process of partitioning the domain is called stratification. In stratified random sampling, a simple random sample is drawn from each stratum.*

Low *discrepancy* is a useful quality measure [92] for sample distributions, especially in the context of Monte Carlo integration. A low discrepancy indicates that samples are

evenly distributed within the domain; thus, for any partitioning of the domain, the different partitions are highly likely to contain the same numbers of samples. Suitably partitioning the sampling domain may dramatically reduce discrepancy. This has been one of the strongest reasons for the popularity of stratified sampling in image synthesis.

Typically, in image synthesis, an estimator's variance is used as a direct indicator of *sampling precision*. Stratification reduces variance of estimates in situations when certain parts of the domain are known to contain information that contribute more to the final estimate. Consider the problem of Monte Carlo integration, over the hemisphere defined by the local tangent plane, for estimating irradiance at a point. Partitioning the hemispherical domain into regions of incident direct and indirect illumination, followed by proportional allocation of samples, is an effective variance reduction strategy. In addition, an appropriate importance sampling strategy may be used for each partition, to fully exploit any known structure of the integrand.

An important consideration while performing stratified sampling is the stratification strategy. Although the strata are typically decided during design, it is not uncommon to choose the strata based on the distribution to be sampled (as seen in the example of irradiance estimation). The theory of stratification describes how to partition the sampling problem and then combine the individual estimates to obtain a higher precision estimate over the whole domain.

In this chapter, we first discuss the theoretical aspects of stratified sampling in its abstract form (Section 2.1). Then we describe image synthesis literature that discusses stratified sampling (Section 2.2), making connections to the theory. Finally we present a new stratification scheme for sampling linear densities for triangular and tetrahedral domains (Section 2.3).

Notation

Symbols with a h as subscript denote quantities associated with the stratum h.

L	number of strata
$\mathcal{D}$	sampling domain
$D = \int_{\mathcal{D}} \mathrm{d}y$	volume of the sampling domain
$\mathcal{D}_h$	subdomain defined by partition h of $\mathcal{D}$
$D_h = \int_{\mathcal{D}_h} \mathrm{d}y$	volume of the subdomain $\mathcal{D}_h$
$\mathbf{n}_h$	number of samples in stratum h
$\mathbf{n} = \sum_{h=1}^{L} \mathbf{n}_h$	total number of samples
f_{hi}	i^{th} sample in stratum h
$W_h = D_h/D$	stratum weight
$w_h = \mathbf{n}_h/D$	sampling fraction in stratum h
$\overline{Y}_h = \frac{1}{D_h} \int_{\mathcal{D}_h} f(y)\, \mathrm{d}y$	true mean for stratum h
σ_h^2	true variance for stratum h
$\overline{y}_h = \frac{1}{\mathbf{n}_h} \sum_{i=1}^{\mathbf{n}_h} f_{hi}$	estimated mean for stratum h

For simplicity we use $\int_{\mathcal{D}} dy$ to represent the volume of the sampling domain, which is strictly $\int_{\mathcal{D}} d\mu(y)$ where $\mu(y)$ is a volume measure defined on the domain $\mathcal{D}$.

2.1 Theoretical aspects of stratified sampling

2.1.1 Proportional stratification

The mean over the entire domain, $\overline{y}_{st}$, is obtained by combining the means within strata, correctly accounting for the strata weights. That is,

$$\overline{y}_{st} \;=\; \frac{1}{D}\sum_{h=1}^{L} D_h\,\overline{y}_h = \sum_{h=1}^{L} W_h\,\overline{y}_h. \tag{2.1}$$

In general this is different from the mean estimated using the samples, which is

$$\overline{y} \;=\; \frac{1}{\mathbf{n}}\sum_{h=1}^{L} \mathbf{n}_h\,\overline{y}_h. \tag{2.2}$$

These two estimates for the mean coincide when

$$\frac{\mathbf{n}_h}{\mathbf{n}} \;=\; \frac{D_h}{D} \qquad \textit{or} \qquad \frac{\mathbf{n}_h}{D_h} \;=\; \frac{\mathbf{n}}{D} \qquad \textit{or} \qquad w_h \;=\; w. \tag{2.3}$$

The stratification procedure which guarantees this condition is called stratification with *proportional allocation*. This type of stratification, with proportional allocation of $\mathbf{n}_h$, yields a self-weighting sample which simplifies the estimation process since the computation of D_h and D can be avoided.

In the case of finite, discrete sampling domains (e.g. statistical surveys within a finite population), D and D_h may be available and the gain due to proportional allocation is most significant when numerous estimates need to be made. When the sampling domain under consideration is infinite (e.g. subsampling a pixel or sampling the surface of a manifold), not having to compute D and D_h almost always simplifies the estimation process.

2.1.2 Bias and Variance

Stratified sampling yields unbiased estimates when the correct strata weights are used.

Theorem 2.2. *If the $\overline{y}_h$ are unbiased estimates of the means within strata, then $\overline{y}_{st}$ is an unbiased estimate of the mean over the entire domain.*

Proof.

$$E\left(\overline{y}_{st}\right) \;=\; E\left(\sum_{h=1}^{L} W_h \overline{y}_h\right) \;=\; \sum_{h=1}^{L} W_h \overline{Y}_h \tag{2.4}$$

since, for unbiased estimates within strata, $E\left(\overline{y}_h\right) = \overline{Y}_h$. The mean over the entire domain, $\overline{Y}$, is

$$\overline{Y} \;=\; \frac{1}{\mathbf{n}} \sum_{h=1}^{L} \sum_{i=i}^{\mathbf{n}_h} f_{hi} \;=\; \frac{1}{\mathbf{n}} \sum_{h=1}^{L} \mathbf{n}_h \overline{Y}_h \;=\; \sum_{h=1}^{L} W_h \overline{Y}_h \tag{2.5}$$

Hence, $E\left(\overline{y}_{st}\right) = \overline{Y}$. □

While stratification does not alter the expected value for the mean, the variance of the estimate for the mean is different from that using simple random sampling. Further, the variance due to stratification can be expressed in terms of the variance of the estimates within strata.

Theorem 2.3. *The variance of the mean estimate over the entire domain using stratified sampling is related to the variance of the mean estimates within strata as*

$$V\left(\overline{y}_{st}\right) \;=\; \sum_{h=1}^{L} W_h^2 \, V\left(\overline{y}_h\right), \tag{2.6}$$

if samples between strata are drawn independent of each other.

Proof. We know that

$$\overline{y}_{st} \;=\; \sum_{h=1}^{L} W_h\, \overline{y}_h, \tag{2.7}$$

where $\overline{y}_{st}$ is a random variable which is a linear function of $\overline{y}_h$ and fixed coefficients W_h. Hence we can write

$$V\left(\overline{y}_{st}\right) \;=\; \sum_{h=1}^{L} W_h^2\, V\left(\overline{y}_h\right) \;+\; 2 \sum_{h=1}^{L} \sum_{k>h}^{L} W_h\, W_k\, Cov\left(\overline{y}_h\, \overline{y}_k\right), \tag{2.8}$$

using the standard statistical result that describes the variance of linear functions of random variables. If samples between strata are drawn independently, then the covariance terms vanish. □

Corollary 2.3.1. *If simple random samples are drawn within the strata,*

$$V\left(\overline{y}_{st}\right) \;=\; \sum_{h=1}^{L} \frac{W_h^2\;\sigma_h^2}{\mathbf{n}_h}. \tag{2.9}$$

This result is obtained by substituting $V\left(\overline{y}_h\right) \;=\; \sigma_h^2/\mathbf{n}_h$ *into the result from Theorem 2.3.* ⌟

Corollary 2.3.2. *If simple random samples are drawn with proportional allocation (Section 2.1.1), we obtain*

$$V\left(\overline{y}_{st}\right) \;=\; \frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h\;\sigma_h^2 \tag{2.10}$$

by substituting $\mathbf{n}_h = \mathbf{n} W_h$ *into Corollary 2.3.1.* ⌟

For any non-trivial stratification scheme (at least two strata), $W_h < 1, \quad \forall\; h$. The typical goals while designing the stratification scheme are: (1) To choose strata such

that the variance within each of the strata, $V(\overline{y}_h)$, is low and (2) That the variance is lower in strata with larger weights.

Theorem 2.3 suggests that $\overline{Y}$ can be estimated without error if $\mathcal{D}$ can be partitioned into $\mathcal{D}_h$ such that the function is constant within each $\mathcal{D}_h$ since $V(\overline{y}_h)$ will be zero for all strata. Note that theorems 2.2 and 2.3 do not make any assumptions about the sampling strategies used within the strata except that they be unbiased.

In the context of image synthesis, stratified sampling is more commonly used to estimate integrals rather than averages.

Theorem 2.4. *Let the estimate of the integral* $\int_{\mathcal{D}} f(y)\,\mathrm{d}y$ *be* $\hat{Y}_{st} = \overline{y}_{st} D$. *Then,*

$$V\left(\hat{Y}_{st}\right) = \sum_{h=1}^{L} \frac{D_h^2\ \sigma_h^2}{\mathbf{n}_h} \tag{2.11}$$

in the case of stratified random sampling.

Proof. The variance of the estimate of the integral is D^2 times the variance of estimate of the mean (Corollary 2.3.1), where D is the volume of the domain. □

Since the relation between statistics of the estimators for mean and integral is straightforward, the results we present in the rest of this chapter for the mean estimator may be simply extended to estimates of integrals.

2.1.3 Benefit from stratification

While stratified sampling can be used as a variance reduction technique in general, there may exist stratification strategies that actually increase the variance of the estimates. Although it is rare that stratification actually increases variance, this is

theoretically possible. To achieve any gain from stratified sampling, the stratification, allocation and sampling strategies need to be chosen carefully for the function or class of functions that are being sampled.

In situations where the total budget is known, there exists a particular allocation of samples that minimizes variance of the mean estimates. This allocation is commonly called Neyman allocation after work by Neyman (in 1934) [75] which popularized it, although the proof for this minimum allocation existed, as early as 1923, in work by Tschuprow [109]. We will refer to this allocation strategy as *optimal allocation*. For a given total of $\mathbf{n}$ samples, optimal allocation minimizes the variance of the mean estimator by using

$$\mathbf{n}_h \;=\; \mathbf{n}\frac{W_h\,\sigma_h}{\sum W_j\,\sigma_j} \;=\; \mathbf{n}\frac{D_h\,\sigma_h}{\sum D_j\,\sigma_j}. \tag{2.12}$$

Intuitively, optimal allocation is achieved when the number of samples in each stratum is proportional to the volume of the stratum and the variation —more specifically standard deviation— of the function within the stratum, or $\mathbf{n}_h \propto D_h\,\sigma_h$. The variance in the estimate when when optimal allocation is used is

$$V_{min}(\overline{y}_{st}) \;=\; \frac{\left(\sum W_h\,\sigma_h\right)^2}{\mathbf{n}}. \tag{2.13}$$

Although it possible that stratified sampling results in increased variance of estimates, fortunately, using optimal or proportional allocation guarantees that the resulting stratified sampling estimators will not be worse (considering variance) than the simple random sampling. Using V_{opt}, V_{prop} and V_{ran} to denote the variance in mean estimates while using optimal allocation, proportional allocation and simple random sampling, we now describe the relation between these three quantities.

Theorem 2.5. *$V_{opt} \leqslant V_{prop} \leqslant V_{ran}$ for finite strata volumes D_h.*

Proof. Recall (from Equation (2.13) and Corollary 2.3.2) that

$$V_{opt} = \frac{1}{\mathbf{n}} \sum_{h=1}^{L} (W_h \, \sigma_h)^2, \quad V_{prop} = \frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \, \sigma_h^2 \quad \textit{and} \quad V_{ran} = \frac{\sigma^2}{\mathbf{n}}.$$

By definition, $V_{opt} \leqslant V_{prop}$. But the relationship between V_{prop} and V_{ran} is to be proven. The variance, σ^2, of the sampling distribution, f, is given by

$$\begin{aligned}
\sigma^2 =& \frac{1}{D} \int_{\mathcal{D}} \left(f(y) - \overline{Y} \right)^2 \mathrm{d}y \\
=& \frac{1}{D} \sum_{h=1}^{L} \int_{\mathcal{D}_h} \left(f(y) - \overline{Y} \right)^2 \mathrm{d}y \\
=& \frac{1}{D} \sum_{h=1}^{L} \int_{\mathcal{D}_h} \left(f(y) - \overline{Y}_h \right)^2 \mathrm{d}y \; + \; \sum_{h=1}^{L} D_h \left(\overline{Y}_h - \overline{Y} \right)^2 \\
=& \frac{1}{D} \sum_{h=1}^{L} D_h \, \sigma_h^2 \; + \; \sum_{h=1}^{L} D_h \left(\overline{Y}_h - \overline{Y} \right)^2 \\
=& \sum_{h=1}^{L} W_h \, \sigma_h^2 \; + \; \sum_{h=1}^{L} W_h \left(\overline{Y}_h - \overline{Y} \right)^2
\end{aligned} \tag{2.14}$$

V_{ran} can then be expressed in terms of the variances in the strata and the weighted variances of the means of the strata from the true mean. That is,

$$\begin{aligned}
V_{ran} =& \frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \, \sigma_h^2 \; + \; \frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \left(\overline{Y}_h - \overline{Y} \right)^2 \\
=& V_{prop} \; + \; \frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \left(\overline{Y}_h - \overline{Y} \right)^2
\end{aligned} \tag{2.15}$$

Clearly, from Equation (2.15), $V_{ran} \geqslant V_{prop}$. The equality occurs when the means within the strata are all identical to the mean over the entire domain. □

To clearly see where the benefit of stratified sampling with optimal allocation comes

from, let us first consider the difference between optimal and proportional allocation.

$$\begin{aligned} V_{prop} - V_{opt} &= \frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \ \sigma_h^2 - \frac{1}{\mathbf{n}} \sum_{h=1}^{L} (W_h \, \sigma_h)^2 \\ &= \frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \, (\sigma_h - \hat{\sigma})^2, \end{aligned} \tag{2.16}$$

where $\hat{\sigma}$ is the weighted mean of the standard deviations within the strata. Substituting this result into Equation (2.15), we obtain the relation between variances due to simple random sampling and stratified random sampling with optimal allocation,

$$V_{opt} = V_{ran} - \underbrace{\frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \, (\sigma_h - \hat{\sigma})^2}_{\textit{Strata Deviations}} - \underbrace{\frac{1}{\mathbf{n}} \sum_{h=1}^{L} W_h \left(\overline{Y}_h - \overline{Y}\right)^2}_{\textit{Strata Means}} \tag{2.17}$$

From Equation (2.17), we see that the reduction in variance with optimal allocation stratified sampling over simple random sampling is from two sources: the first is due to the elimination of differences between the strata means; the second is due to the elimination of differences between the standard deviations between the strata. The latter causes the reduction in variance between proportional allocation and optimal allocation. Note that if larger strata exhibit greater deviations in their mean from the average, the reduction in variance becomes more pronounced.

2.2 Stratified sampling in image synthesis

Stratified sampling theory has existed for almost a century and was mainly used by statisticians for increasing precision in analysing surveys. The application of stratified sampling was originally limited to finite, discrete populations within which samples were drawn to estimate certain charecteristics of the population. Several results on efficient allocation for stratified sampling were proposed, of which the results on

optimal allocation [75, 109, 16] gained popularity within the statistics community in the early forties. About three decades later, Bayes studied the benefit due to stratified sampling and its relation to Markov chain sampling [15]. More recently, Valliant analysed properties of different estimators [110] using stratified sampling and proposed extensions to stratified sampling like stratified two-stage sampling [111, 110].

Recent work [51] in stratified sampling derived an algorithm which varied the allocation at each sampling step based on information derived using samples drawn in the previous step. The algorithm was shown to converge at the rate of convergence of optimal allocation.

Stratified sampling has been applied to solve a wide variety of problems in different fields. e.g. applied mathematics [56], software testing [80], biological sciences [54], optimization [89], Monte Carlo quadrature [44], etc.

1986

Cook argued that visual artifacts in image synthesis were not a result of point sampling [28], rather, that they were results of the sampling being regular. In earlier work, Yellott's observation [120] that retinal cells were distributed with a Poisson disk distribution led him to believe that these sampling patterns could be useful in eliminating aliasing artifacts. It is well known that, with uniform point sampling, frequencies in the function being sampled that are higher than the half the frequency of the distribution of point samples would be aliased. Cook presented a study of uniform sampling and jittered uniform point sampling (a form of stratified sampling) and demonstrated that the latter is a good approximation of the Poisson disk sampling. He studied the problem of aliasing in the frequency domain and concluded that the advantage of using Poisson disk sampling was that the aliasing artifacts were transformed into noise that was less visually disturbing. Cook also describes a variant

to stratified sampling, where the sampling within each stratum is not simple random. Instead, he suggests using a gaussian distribution within each stratum and demonstrates that this reduces aliasing further. Cook applied jittered uniform sampling to distributed ray tracing and demonstrated a reduction in variance.

1987

Mitchell pursued the problem of eliminating aliasing artifacts due to sampling, but particularly in the context of image space sampling [70]. By revisiting Yellott's work on retinal photoreceptor distributions [120] and work in the field of image halftoning (that had been around for a couple of decades already), Mitchell stressed on the importance of *blue noise* properties in sampling distributions. He showed that although Cook's method of jittered uniform sampling reduced aliasing artifacts, it still produced sampling patterns with too much energy in the lower and intermediate frequency bands. To ameliorate this effect, Mitchell recommended strategies at two levels. First, he adapted algorithms from image halftoning literaure to perform true Poisson disk sampling, instead of using jittered sampling as an approximation. Second, he presented an adaptive sampling algorithm that would sample regions with higher frequencies in image space more profusely. He used local image contrast as the metric for adaptive subdivision, rather than local variance— which he claimed to be a poor measure of visual perception of local variation.

1990

Shirley adapted Arvo's technique of *backward ray tracing* [5] to decrease variance in estimating radiance due to illumination reflected of multiple specular as well as diffuse surfaces. Arvo had introduced the term backward ray tracing to refer to the process of tracing light paths from the light sources into the scene, where their contributions

would be recorded for lookup during shading while tracing from the eye. Shirley embellished Arvo's technique by using different algorithms to estimate hard and soft illumination. While Arvo's technique was effective in accounting for light paths that bounced of multiple specular objects before they impinged on diffuse surfaces, Shirley used radiosity to account for soft lighting due to diffuse-diffuse interactions.

Shirley's algorithm employed stratified sampling at two distinctly independent levels: first, stratification of the local hemisphere of directions—that is, strata corresponding to rays arriving from specular surfaces and those from diffuse surfaces; second, further stratification of the those strata corresponding to rays from specular surfaces, by extending Cook's jittered sampling algorithm. In his paper, Shirley only identifies the latter as a form of stratification.

1991

Earlier work on stochastic sampling by [28] and the introduction of the rendering equation by Kajiya in 1986 [52] sparked off a flurry of research in sampling techniques for distributed ray tracing and Monte Carlo integration respectively. The year of 1991 saw three important papers related to sampling.

Observing the importance of evaluating and assessing different sampling techniques, Shirley proposed that discrepancy of point sets could be used as a quality measure for point samples [92]. In the context of sampling for Monte Carlo integration, the variance of estimators may be used as a direct indicator of precision, and hence can be considered a quality measure for the estimator. However, the quality of an estimator could depend on other factors than simply the sampling algorithm. In addition, image space sampling was gaining importance for reducing aliasing artifacts. Shirley demonstrated that the discrepancy of point samples was a quality measure consistent with results of analysing variance or aliasing artifacts. In his paper, Shirley

discussed simple random sampling, jittered sampling, half-jittered sampling, Poisson disk sampling and N-rooks sampling and ranked each of these sampling methods in specific contexts, according to the discrepancy of the resulting samples. In addition, he presented a scheme to generate non-uniform stratified samples, by generating uniform samples and then warping them with a transformation. This idea was extended later by Arvo to sample 2-manifolds. Finally, Shirley proposed a method to estimate discrepancy for stratified samples from non-uniform distributions.

Mitchell analysed the problem of reconstructed images from samples [71] in the frequency domain for extensions of jittered sampling to higher dimensions. In accordance with his earlier thesis, that sampling patterns containing higher frequencies tended to reduce perceptible artifacts, he proposed a scanning sampling algorithm to optimally sample higher dimensions for application in distributed ray tracing.

While there were a number of papers analysing sampling techniques for their efficiency, there was little work on testing whether sampling algorithms were introducing errors in estimates. Kirk and Arvo observed that the adaptive sampling algorithms that were being used were biased [59]. In addition they proposed an adaptive sampling scheme that yielded unbiased estimates.

1995

Although stratified sampling had become popular in image synthesis, it was mainly used in the form of jittered sampling where a regular grid of samples were perturbed to yield a uniform stratification of rectangular domains. Similar to Shirley's warp for stratified sampling of non-uniform distributions, Arvo proposed an area preserving mapping [7] from the unit square to the spherical triangle that allowed stratified sampling of spherical triangles. Using this method, the solid angle subtended by arbitrary polygons could be sampled with stratification.

1996

Stratification theory predicts that the convergence of mean estimates improves from $O(N^{-1})$ to $O(N^{-2})$ when simple random sampling is replaced with stratified random sampling in 2D. Mitchell studied the benefit of stratification [72] in practice, particularly in the context of pixel supersampling. He concluded that, in practice, stratified random sampling of images resulted in convergence rates anywhere from $O(N^{-1})$ to $O(N^{-2})$, with a rate of $O(N^{-3/2})$ for pixels containing edges. His experimental results were accordant with earlier theoretical bounds derived based on discrepancy by Beck and Chen.

An important result that Mitchell deduced was that, in general, the convergence of mean estimates with stratified random sampling in d dimensions is $O(N^{-1-2/d})$. This result triggered the realization within the image synthesis community that stratified random sampling would not be very effective as a variance reduction tool, for higher dimensions. The scope of other forms of stratified sampling, than just stratified random sampling have not been explored much in image synthesis literature.

1997

In contrast to the theoretical and experimental approaches taken by Arvo, Shirley, Cook and Mitchell, McCool and Harwood presented a completely algorithmic sampling strategy using probability trees [67]. The main strategy in this method was to subdivide the sampling domain and arrange the domains into a hierarchy where the subdivisions were leaves. For generating samples according to given distributions, their algorithm could be adapted in two ways: the classical approach of inverting the cumulative distribution for each dimension independently, which was less practical and a hierarchical algorithm where they ensured that the distribution function at

each internal node of the hierarchy represented the integral of the function over its child nodes. Although they provided means to adaptively generate stratified samples, the lack of theoretical justification in their paper makes it unclear what their allocation scheme is, or even whether or not the method introduces bias in mean estimates.

1999

By this time stratified sampling had been shown to be useful in image synthesis for a number of point sampling problems: reducing perceptible aliasing artifacts while sampling images; reducing variance in depth of field, motion blur and other distributed raytracing contexts; and for sampling the hemisphere while integrating to compute incident illumination. Evans and McCool used stratification in conjunction with importance sampling [39], to reduce variance while integrating over wavelengths along paths. They argued that using a single spectral sample associated with each path led to both— a drastic increase in variance of the combined spectral estimates and objectional levels of spectral incoherence between neighboring pixels. To remediate these problems, they used the spectral power distribution of light sources to importance sample wavelengths. As a further variance reduction tool, the samples drawn were stratified. They generated images demonstrating effects like spectral caustics and chromatic aberrations.

2001

Arvo's method to generate stratified samples on spherical triangles was useful in sampling solid angles projected by arbitrary polygons while estimating incident illumination. In 2001, Arvo provided a more general recipe [9] to generate stratified samples on a 2-manifold. Any homeomorphic mapping from the unit square to a

2-manifold could be used to map samples in the unit square to samples on the manifold. However, to ensure that there these samples do not introduce a bias, when used in a Monte Carlo integration, the Jacobian of the mapping would need to be introduced in the integrand. This follows from a simple change of variables due to the parametrization. Arvo observed that, for those parametrizations that the Jacobian is a constant, the original samples could be used directly for computing unbiased Monte Carlo estimates. Based on this observation, he outlined a scheme to derive a constant Jacobian mapping from the unit square to any 2-manifold, and suggested that this be used to map stratified samples on the unit square to stratified samples on the 2-manifold. The samples thus generated, would be uniformly distributed over the surface of the 2-manifold.

2002

Kollig and Keller presented a stratification scheme [61] which they used in combination with a randomized quasi Monte Carlo algorithm for integrating square integrable functions. Randomized quasi Monte Carlo algorithms are quasi Monte Carlo algorithms where the samples are uniformly distributed in the domain.

2004

Wang and Hwang adapted Arvo's recipe for stratified sampling of 2-manifolds to sample ellipses [115]. They used their algorithm to compute form factors and demonstrated that the samples were uniformly distributed. While Arvo's scheme was useful for sampling 2-manifolds, the stratification produced would not be effective as an approximation to the Poisson disk distribution of 3D models represented as polygon meshes. This is fundamentally because satisfying area preserving maps within each polygon of the mesh, becomes progressively less meaningful as the mesh is refined. It

would also be necessary that the triangles be sampled with probabilities proportional to their areas. Nehab and Shilane presented an algorithm for stratified sampling of 3D models [73] and demonstrated that the samples produced were visually appealing. They achieved this by first voxelizing the model based on certain criteria and then drawing samples from the voxels, which also guaranteed a controllable minimum distance between points. Although this method produced good looking samples, it is not clear that the samples could be used in Monte Carlo estimation.

2005

One method of suppressing aliasing artifacts in images is to reduce the bandwidth of the image according to the minimum distance between samples. One common choice for the filter to perform this bandlimiting is a recursively applied box filter. A recursively applied box filter is equivalent to a B-spline filter. An alternative to actual convolution by the filter kernel in image space is to draw samples according to the distribution defined by the filter. Stark and Shirley derived an algorithm to generate stratified samples for cubic B-spline pixel filtering [101]. They derive the algorithm for sampling the cubic B-spline kernel in 1D and use the property that B-spline filters are separable to sample in 2D.

2006

By 2006, environment maps had gained immense popularity in image synthesis due to the richness in illumination that they provided. Several algorithms had been proposed for sampling environment maps, several of which were efficient but biased. Xing et al. proposed an extension [68] to Debevec's biased light probe sampling technique [32]. Debevec proposed the use of a median cut algorithm to partition the environment map into regions of equal integrated energy and represented each partition with a single

point light located at its centroid. In their extension, Xing et al., used stratified importance sampling within each partition.

2.3 Analytical parametrizations for stratification

When the sampling domain is not a rectangle, stratified sampling in the form of jittered sampling cannot be applied directly on the domain, and the stratification algorithm demands more attention. Consider the example of sampling the hemisphere of directions for integrating incident radiance at a point. The hemisphere is typically stratified into regions that are projections of light sources onto the hemisphere (and the complement of the union of those regions). This stratification is an effective variance reduction technique since between strata, the mean light energy could potentially be drastically different, while within each stratum the variation is usually much less dramatic. However, the process of discovering the strata as projections of the light sources is not simple in general since the illuminaires could be of arbitrary shapes. One method would be to project each polygon of the light source onto the hemisphere and further stratify the projection using Arvo's algorithm to sample spherical triangles [7].

Another approach to stratify arbitrary domains was suggested by Shirley [92] and then developed for 2-manifolds by Arvo [9]. Arvo's method involved the construction of a constant Jacobian parametrization from the unit square to the manifold that was to be stratified. As a consequence, the change of variables from the differential area element on the square to that on the 2-manifold, was trivialized. Thus the ratios of strata on the unit square to the regions they map to on the manifold would be constant. While this technique allows stratified random sampling with proportional allocation on arbitrary 2-manifolds, it is (in the form proposed) limited to constant distributions.

However, Arvo suggests ways to extend this for non-uniform distributions, and points out that the true hurdle lies in the step involving inversion. In this section, we present a slight modification of Arvo's algorithm for 2-manifolds, that allows sampling from a non-uniform density (Section 2.3.1). Specifically, we derive the parametrizations for sampling linear densities with triangular (Section 2.3.2) and tetrahedral support (Section **??**).

2.3.1 Non-uniform stratification of 2-manifolds

With a minor modification to Arvo's algorithm [9], we describe an analytic method for non-uniform stratification of 2-manifolds in $\mathbb{R}^n$. We retain the notation introduced by Arvo and invent similar notation for new quantities.

Let $\mathcal{M}$ represent the 2-manifold that we wish to sample and $w(p),\ p \in \mathcal{M}$ be the density according to which we need to sample $\mathcal{M}$. We seek to derive a mapping $\psi : [0,1] \times [0,1] \to \mathcal{M}$ the Jacobian of which is proportional, locally, to the w. In Arvo's algorithm, the goal is to derive ψ with a constant Jacobian. We derive ψ starting from an aribtrary smooth bijection, on all but a set of zero measure, from the unit square to the 2-manifold, $\phi : [0,1] \times [0,1] \to \mathcal{M}$.

We define the weighted surface area 2-form, $\sigma_w : [0,1] \times [0,1] \to \mathbb{R}$ such that the integral of this function over the parameter space yeilds the weighted integral over the manifold using the weighting function. Specifically,

$$\int_{\mathcal{P}} \sigma_w(\mathbf{x})\ \mathrm{d}\mu_{\mathbf{1}}(\mathbf{x}) \ = \int_{\phi(\mathcal{P})} w(\mathbf{y})\ \mathrm{d}\mu_{\mathbf{2}}(\mathbf{y}) \tag{2.21}$$

where $\mathcal{P}$ is a region in the unit square and μ_1, μ_2 are measures of area in the parametric space and manifold surface respectively. The rhs of Equation (2.21) is simply the

Algorithm 2.1 *Algorithm for stratification of 2-manifolds given a target density*

1. Select a smooth bijection $\phi : [0,1] \times [0,1] \rightarrow \mathcal{M}$, from the unit square to the 2-manifold $\mathcal{M} \subset \mathbb{R}^n$. Let $w(\mathbf{y})$ be the density we wish to sample according to, where $\mathbf{y}$ is a point on $\mathcal{M}$.

2. Define $\sigma_w : [0,1] \times [0,1] \rightarrow \mathbb{R}$ as

$$\sigma_w(\mathbf{x}) \equiv w(\phi(\mathbf{x})) \sqrt{\langle \phi_{x1},\, \phi_{x1} \rangle \langle \phi_{x2},\, \phi_{x2} \rangle - \langle \phi_{x1},\, \phi_{x2} \rangle^2} \tag{2.18}$$

   where $\phi_{x1} = \left(\frac{\partial \phi_1}{\partial x1},\, \frac{\partial \phi_2}{\partial x1},\, \frac{\partial \phi_3}{\partial x1},\, ...,\, \frac{\partial \phi_n}{\partial x1} \right)$ and $\mathbf{x} \equiv (x_1, x_2)$.

3. Define cumulative distributions

$$\begin{aligned} F(x_1) &\equiv \frac{\int_0^1 \int_0^{x_1} \sigma_w(u,v)\, du\, dv}{\int_0^1 \int_0^1 \sigma_w(u,v)\, du\, dv} \\ G_{x1}(x_2) &\equiv \frac{\int_0^{x_2} \sigma_w(x_1,\, v)\, dv}{\int_0^1 \sigma_w(x_1,\, v)\, dv} \end{aligned} \tag{2.19}$$

4. Invert the two cumulative distributions

$$\begin{aligned} f(z) &\equiv F^{-1}(z) \\ g(z_1, z_2) &\equiv G^{-1}_{f(z1)}(x_2) \end{aligned} \tag{2.20}$$

5. Define the resulting stratification parametrization, $\psi : [0,1] \times [0,1] \rightarrow \mathcal{M}$, as

$$\psi(\mathbf{z}) \equiv \phi(f(z_1),\, g(z_1,\, z_2))$$

   where $\mathbf{z} \equiv (z_1, z_2)$. $\psi(.)$ has the property that equal areas on the unit square map onto regions on the manifold with equal integrals of the density function $w(.)$.

integral of the sampling density over the region on the manifold that $\mathcal{P}$ maps onto. In the case of a constant sampling density, this would simply be $\mathbf{area}(\phi[\mathcal{P}])$, as used by Arvo.

For 2-manifolds imbedded in 2D and 3D spaces, we find simple expressions for $\sigma_w(\mathbf{x})$. In 2D, this is given by the determinant of the Jacobian of the initial parametrization ϕ, weighted by the local density,

$$\sigma_w(\mathbf{x}) \equiv w(\phi(\mathbf{x}))\, det(J_x(\phi)). \tag{2.22}$$

In 3D, this is expressed in terms of the partial derivatives of the parametrization along the two axes, ϕ_{x1} and ϕ_{x2}, in parametric space and the sampling density:

$$\sigma_w(\mathbf{x}) \equiv w(\phi(\mathbf{x}))\, \|\phi_{x1}(\mathbf{x}) - \phi_{x2}(\mathbf{x})\|. \tag{2.23}$$

This step of the algorithm differs from Arvo's in that the resulting surface area 2-form contains an additional term corresponding to the target sampling distribution.

The rest of the procedure is exactly as described by Arvo: We derive cumulative distributions and invert them to finally derive ψ. However, since σ_w is potentially more complicated than just σ, the process of integrating to define the cumulative distributions and inversion to derive ψ, are slightly more complex. Their actual complexity depends on $\phi(.)$ and $\mathcal{M}$. The complete algorithm, retaining as much similarity to Arvo's algorithm as possible, is shown in Algorithm 2.1.

While the extension of Arvo's algorithm to non-uniform distributions is straightforward, the inclusion of the weighting function increases the complexity of steps 3, 4 and 5 in Algorithm 2.1. In the following sections, we derive the mapping for linear stratification of triangles and tetrahedra respectively.

2.3.2 Linear stratification of triangles

In this section we present a simple and compact algorithm that allows generation of stratified samples [102]according to a linearly-varying density function over a triangle with vertices A, B, C and vertex weights w_a, w_b and w_c. $\mathbf{A}, \mathbf{B}, \mathbf{C}$ denote the position vectors of the vertices. We follow the steps highlighted in Algorithm 2.1, fixing $\mathcal{M}$ to refer to a triangle, and obtain a parametrization for stratification of triangles according to a linear density.

1. We start with a simple barycentric mapping defining position and density:

$$\begin{aligned} \phi(x_1, x_2) &= (1 - x_1)\,\mathbf{A} + x_1(1 - x_2)\,\mathbf{B} + x_1 x_2\,\mathbf{C}; \\ w(x_1, x_2) &= (1 - x_1)\,w_a + x_1(1 - x_2)\,w_b + x_1 x_2\,w_c. \end{aligned} \tag{2.24}$$

2. After simplification of the expression for σ_w, we obtain

$$\sigma_w(x_1, x_2) = 2\,a\,x_1\,w(x_1, x_2) \tag{2.25}$$

   where a is the area of the triangle.

3. For $F(.)$ and $G(.)$, we obtain

$$\begin{aligned} F(x_1) &= \alpha\,x_1^3 + \beta\,x_1^2 \\ G_{x1}(x_2) &= \gamma_{x1}\,x_2^2 + \rho_{x1}\,x_2 \end{aligned} \tag{2.26}$$

where

$$\begin{aligned}
\alpha &= \frac{w_b + w_c - 2w_a}{w_a + w_b + w_c}, \\
\beta &= \frac{3w_a}{w_a + w_b + w_c}, \\
\gamma_s &= \frac{s(w_c - w_b)}{s(w_c - w_b) + 2(1-s)w_a + sw_b}, \\
\rho_s &= \frac{2(1-s)w_a + sw_b}{s(w_c - w_b) + 2(1-s)w_a + sw_b}.
\end{aligned}$$

4. Since $F(.)$ and $G(.)$ are cubic and quadratic, $f(.)$ and $g(.,.)$ are easily obtained by analytically or numerically inverting them.

5. Pseudocode for realizing linear stratification of triangles is shown in Algorithm 2.2

Algorithm 2.2 *Linear stratification of triangles*

function *SampleTriangle* (ξ_1, ξ_2, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, w_a, w_b, w_c)
  $s \leftarrow f(\xi_1, w_a, w_b, w_c)$
  $t \leftarrow g(\xi_2, s, w_a, w_b, w_c)$
  $w \leftarrow (1-s)w_a + s(1-t)w_b + stw_c$
  $\mathbf{p} \leftarrow (1-s)\mathbf{A} + s(1-t)\mathbf{B} + st\mathbf{C}$
  return $(w, \mathbf{p})$

function f (ξ, w_a, w_b, w_c)
  $X \leftarrow (w_b - w_a)/3 + (w_c - w_b)/6$
  $Y \leftarrow w_a/2$
  $\alpha \leftarrow X/(X+Y)$
  $\beta \leftarrow Y/(X+Y)$
  return $RootOf(\alpha x^3 + \beta x^2 - \xi)$

function g (ξ, s, w_a, w_b, w_c)
  $t \leftarrow s(w_c - w_b) + 2(1-s)w_a + sw_b$
  $\gamma \leftarrow s(w_c - w_b)/t$
  $\rho \leftarrow 2((1-s)w_a + sw_b)/t$
  return $2\xi/(\rho + \sqrt{\rho^2 + 4\gamma\,\xi})$

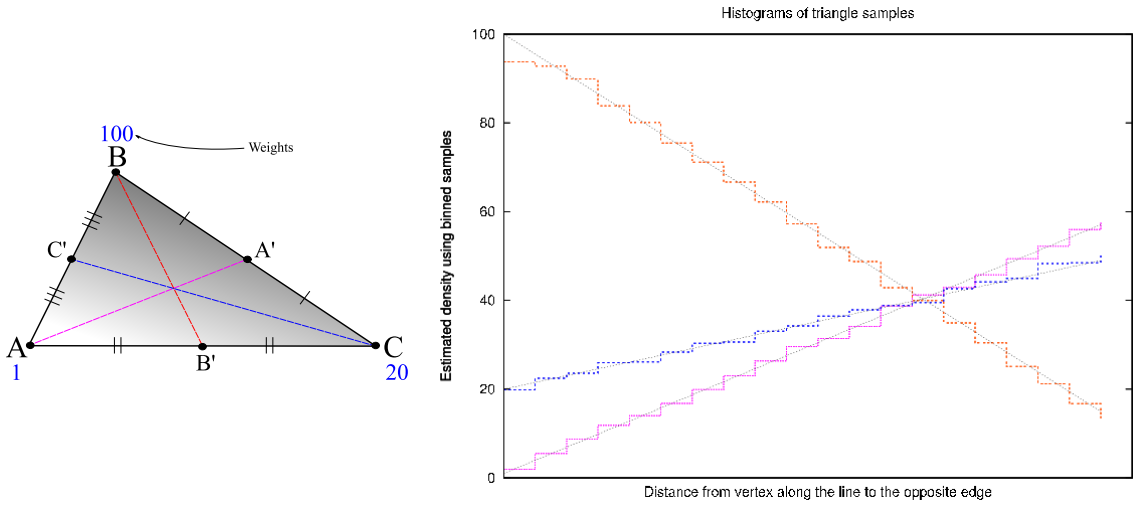


Figure 2.1: *Triangle ABC (left) was sampled using the linear stratified sampling algorithm (see Algorithm 2.2). Samples along AA' (magenta), BB' (red) and CC' (blue) were collected in 20 bins (for each line) and estimates of the density in each bin are plotted. Also shown are the analytically computed expected densities (black lines) along each line.*

Chapter 3

Steerable Importance Sampling

Despite copious amounts of literature exploring importance sampling of static functions in Monte Carlo image synthesis, there has been little work on importance sampling dynamically varying functions. One reason for this is a requisite step of the importance sampling procedure: Computing the correct probability density associated with each sample is crucial. This step involves normalization, or integrating the importance function over the entire domain. For dynamically varying functions, repeated estimation (or computation) of this integral can make the sample-drawing process deterrently expensive. In this chapter, we present a technique [103] that defines a steerable importance function as the product of a static component and a steering function. We present an abstract description of the method before describing an example application. However, we first review the notion of steerable functions before describing the steerable importance sampling method.

3.1 Steerable functions

Functions whose transformed versions can be expressed as a linear combination of a fixed set of bases are called *steerable functions*. The coefficients of the linear combination are dependent on the parameters of the transformation and are called steering functions.

Consider the example of a shifted sinusoid $\sin(x+k)$ which can be represented as the linear combination of a fixed sinusoid and cosinusoid,

$$\sin(x+k) \equiv \cos k \, \sin x \; + \; \sin k \, \cos x. \tag{3.1}$$

The translation of the sinusoid by the parameter, k, can be expressed as a linear combination of the bases $\sin x$ and $\cos x$. Thus a sinusoid is steerable under translation, where the steering functions are the coefficients in the linear combination ($\cos k$ and $\sin k$).

As another example, consider the 2D function $f(x,y)$ and the problem of finding its partial derivative along a direction given by θ degrees from the reference X direction. If the partial derivatives along the reference X and Y directions, $f'_X(x,y)$ and $f'_Y(x,y)$ respectively, have already been computed then explicit computation of the partial derivative, $f'_\theta(x,y)$, may be avoided. Instead, we may use the relation

$$f'_\theta(x,y) \equiv \cos\theta f'_X(x,y) \; + \; \sin\theta f'_Y(x,y) \tag{3.2}$$

In this case the partial derivatives along the two linearly independent, arbitrary reference directions are the fixed bases and the sinusoid and cosinusoid of the rotation angle θ are steering functions. This is another example of a single parameter

transformation; however, the notion may be extended to transformations of multiple parameters.

Definition 3.1. *A function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{C}$ is steerable under a k-parameter transformation $\mathcal{K}_\tau$ if the transformed version of f can be written as a linear combination of a fixed, finite set of basis functions $\{\phi_i(\mathbf{x})\}$:*

$$(\mathcal{K}_\tau f)(\mathbf{x}) \;=\; \sum_{i=1}^{m} \psi_i(\tau)\; \phi_i(\mathbf{x}) \;=\; \langle \Psi(\tau),\, \Phi(\mathbf{x}) \rangle \,. \tag{3.3}$$

Here ψ_i are called the steering functions of f associated with the basis $\{\phi_i\}$ and depend solely on τ, the k-parameter vector that characterizes the transformation.

The set of basis functions required to steer a function is not unique. For example, any linear transformation of the bases can be used so long as the transformation is non-singular.

Theorem 3.2. *If a function f is steerable with a set of bases Φ, then each of the bases, ϕ_i, is itself steerable with the same set of bases.*

Proof. Under a transformation $\mathcal{K}_\tau$, by definition $(\mathcal{K}_\tau f)(\mathbf{x}) = \Psi^T(\tau)\Phi(\mathbf{x})$. The set of basis can be written in terms of a number of linearly independent transformed versions of f, with different parameters τ_1, τ_2, τ_3, ..., τ_n. That is,

$$\Phi = \begin{bmatrix} \Psi^T(\tau_1) \\ \Psi^T(\tau_2) \\ \cdots \\ \Psi^T(\tau_n) \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{K}_{\tau 1} f \\ \mathcal{K}_{\tau 2} f \\ \cdots \\ \mathcal{K}_{\tau n} f \end{bmatrix} \tag{3.4}$$

Since each $\mathcal{K}_{\tau i} f$ is steerable with bases Φ, the theorem holds. □

3.1.1 Brief history

The name "steerable" was introduced by Freeman and Adelson [42] for filter kernels whose rotated versions could be expressed in terms of a set of bases; they provided an analytic method to derive the basis kernels. Their work was extended [96, 97] to include Euclidean transformations (translations and scaling) by Simoncelli et al., who called the property "joint-shiftability". This extension also provided an interesting link between wavelet theory and steerability. Another term for steerable functions that was introduced by Perona [78] is "deformable functions". Perona used singular value decomposition to guarantee optimal bases for two parameter kernel families. In its various flavours, steerability has been used (sometimes without being acknowledged) predominantly in computer vision for image analysis, motion estimation and pattern recognition. In computer graphics, steerability has found applications in texture antialiasing, illumination textures, and texture synthesis.

A common method of feature detection in images involves the use of linear filters. Typically, a template of the feature to be detected is constructed and correlated with the input image. Regions in the output with high values for correlation indicate the presence of the feature. To be able to detect multiply oriented versions of the same feature, one option would be to construct rotated version of the templates and test for correlation against each. However, since correlation is a linear operation, Freeman and Adelson recognized that the result of the correlation against the differently oriented templates could be expressed as the linear combination of the results from a small number of basis templates [42]. Thus, the notion of *steerable filters* was introduced for detecting local image features with different orientations. The set of features was later extended along with the ability to detect features at different scales [96]. Two dimensional steerable filters used for feature detection in images were extended to 3D and used in motion estimation. Since image motion can be viewed as orientation in a

3D spatio-temporal domain [1, 118], the extension of 2D image analysis was natural.

3.1.2 Steerable functions in computer graphics

The earliest, explicit application of steerability in computer graphics was in the purely 2D problem of texture filtering for antialiasing, by Gotsman [47]. In this paper, Gotsman described how the convolution of a filter with an image could be approximated in constant time for space-variant filters of arbitrary sizes. He reduced the problem of filtering in constant time to reasonably approximating a parametric family of filter kernels with a constant number of basis kernels. Although the proposed technique involved precomputation of the convolution by a family of gaussian kernels, the savings were shown to be significant when kernels with large supports were used.

Nimeroff et al. [76] proposed an efficient algorithm for dynamic relighting in naturally lit scenes by capitalizing on the linearity of rendering: the image resulting from two illuminaires is simply the sum of two images rendered considering the illuminaires independently; scaling the strength of an illuminaire causes the image to be scaled by the same factor. Rather than recomputing the entire global illumination solution for each given illumination setting in an animation, they proposed a technique that approximated each illuminaire by a set of *steerable illuminant bases* and used the bases to relight the scene for other illumination settings. By using the theory of steerable functions, they derived the bases for several natural lighting settings (overcast skylight, clear skylight, steerable sunlight and general skylight). During rerendering the coefficients with which the bases were weighted was constructed depending on user input like cloudiness, sun position, etc. The method was later extended to efficiently rerender indoor scenes [35], theatrical lighting design, directional spot lights, area light sources and combinations [107].

Ashikhmin and Shirley revisited the use of steerable illuminant bases [13], and proposed bases for lighting bumpy surfaces (height fields) . Their method involved the precomputation of a set of illumination textures that captured (global) illumination effects on a bumpy surface due to distant, distributed light sources. Given a new light direction, they generated an image using a linear combination of the precomputed set of illumination textures; this generated image captured illumination effects for the bumpy surface, including shadowing and interreflection. The approach taken, in this method, was to first numerically compute the bases and then fit analytical functions to it. That is, they first chose a light source with certain properties and tried to find steerable functions that approximated it well.

3.1.3 Designing steerable bases

As shown earlier (Section 3.1), sinusoids (hence cosinusoids) are steerable. As a consequence, any bandlimited function with a discrete Fourier power spectrum is steerable given enough basis functions. The strategies to design basis functions for steerable functions can be broadly categorized into three categories: analytic, numerical and mixed. The strategy adopted by Freeman and Adelson [42] was to choose a set of basis functions and then find functions with the desired properties, within the linear space spanned by the set of basis functions. A different approach was to compute, numerically, the basis and steering functions required to steer a given function under a family of transformations [78]. Numerical approaches were general although the functions were only available in sampled form rather than as convenient analytic expressions.

Since steerability of a function f implies steerability of its bases ϕ_i, the concept of steerability can be naturally expressed in terms of a function space (the space

spanned by $\{\phi_i\}$). Teo exploited this property, and defined a more sophisticated analytical approach where he defined steerability using the mathematical theory of Lie transformation groups. Although this provided a powerful way to obtain almost optimal set of basis functions, the process was quite complicated and its application was mostly restricted to one-parameter Lie groups and multi-parameter Abelian groups. Neither of these application groups included the group of 3D rotations which is one of the most commonly encountered group of transformations in computer graphics.

The third category of steerable basis design combined the benefits of the analytical and numerical approaches. Teo and Hel-Or chose to use the analytical approach to obtain a large set of basis functions which they then reduced using numerical techniques. Ashikhmin and Shirley extended this idea by first using the numerical approach to arrive at a set of basis functions and then fitting analytic approximations to the sampled functions.

3.2 The parametrized probability tree

Consider a family of parametrized density functions, $\rho : \mathcal{D} \times S \rightarrow \mathbb{R}$, over a spatial domain $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup ... \cup \mathcal{D}_n$ and parameter domain S; assume that $\mathcal{D}_i \cap \mathcal{D}_j = \{\} \; \forall \; i \neq j$. The density within a differential spatial neighborhood about a point $\mathbf{x}_0 \in \mathcal{D}$ is $\rho(\mathbf{x}_0, \mathbf{s})$ where $\mathbf{s} \in S$. Also consider the scenario where ρ is to be used as an importance function.

Two important requirements for ρ to be useful as an importance function are: (1) the ability to generate random variables distributed according to this density and (2) the knowledge of the exact normalized probability density associated with drawing each sample. The normalized density would simply be the value of ρ at each sample

if $\int_{\mathcal{D}} \rho(\mathbf{x}, \mathbf{s})\ d\mu(\mathbf{x}) = 1$ for all values of the parameter $\mathbf{s}$; here $\mu(\mathbf{x})$ is a volume measure associated with the domain $\mathcal{D}$. If the density does not integrate to unity, the normalization would involve a division by value of this integral.

The goal of this section is to describe a data structure for efficient generation of random variables in $\mathcal{D}$ according to ρ. This is possible provided two conditions are met.

The first condition is that there exist ways of generating the random variables in each of the subdomains $\mathcal{D}_i$ according to ρ. In other words, the data structure will be used to randomly select a subdomain $\mathcal{D}_i$ with a probability proportional to the integral of ρ within it. Let $F_i(\mathbf{s})$ be the integral of ρ over subdomain $\mathcal{D}_i$. The probability that a random varaible $\mathbf{y} \in \mathcal{D}$, drawn from the distribution $\rho(., \mathbf{s})$, lies in $\mathcal{D}_i$ is given by

$$\frac{F_i(\mathbf{s})}{\sum_{j=1}^{n} F_j(\mathbf{s})} \equiv \frac{\int_{\mathcal{D}_i} \rho(\mathbf{x}, \mathbf{s})\ \mathrm{d}\mu(\mathbf{x})}{F(s)} \tag{3.5}$$

(3.6)

where $F(s)$ is the integral of the density function over all domains, for a given value of the parameter:

$$F(s) = \sum_{j=1}^{n} \int_{\mathcal{D}_j} \rho(\mathbf{x}, \mathbf{s})\ \mathrm{d}\mu(\mathbf{x}). \tag{3.7}$$

The second condition is that these integrals are relatively easy to compute and can be compactly represented. The first condition reduces the problem of generating the random variable $\mathbf{y}$ according to $\rho(., \mathbf{s})$ to one of randomly selecting a subdomain $\mathcal{D}_i$ (based on the probability in Equation (3.5)), for a given value of the parameter $\mathbf{s}$.

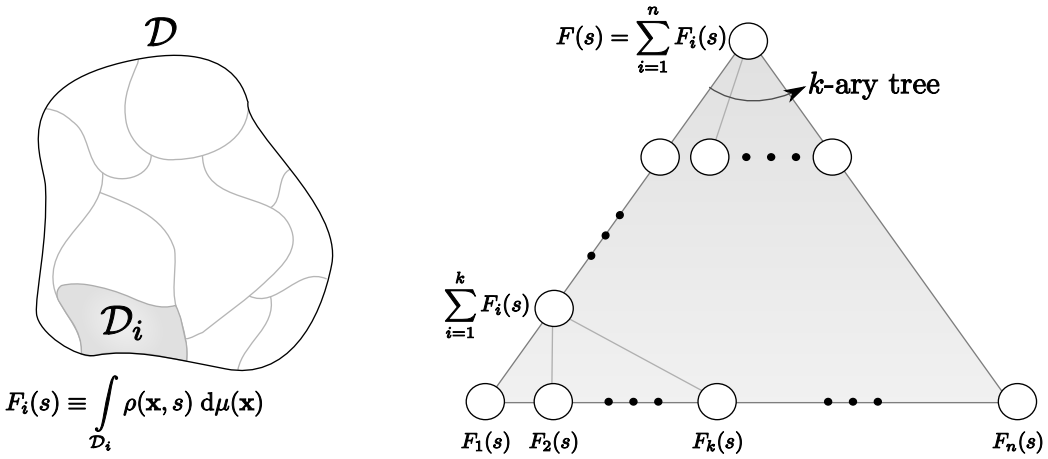


Figure 3.1: *Each node of the tree stores a functional expressing the integral of a density function over that part of the domain which is spanned by the node: Each leaf node k spans a partition $\mathcal{D}_k$; each internal node spans a subdomain that is a union of all the partitions spanned by the leaves in its subtree.*

The data structure presented here is similar in spirit to the hierarchical probability tree introduced by McCool and Harwood [67]. They used k-D trees to store piecewise constant approximations of their probability density function, and generated stratified samples by traversing down the tree based on branching probabilities proportional to the integral of the density function in each branch. However, their application was for a fixed importance function rather than a parametrized importance function family. Another disadvantage of their scheme was their dependence on explicitly integrating the function within each node of their hierarchical structure. The method to perform these integrations in constant time, using the data structure described in this section for a parametrized family of importance functions, is described later in this chapter (see Section 3.4).

3.2.1 The data structure

The data structure consists of a k-ary tree whose nodes store functions (how the functions are represented is an implementation issue). The n leaves of the tree store

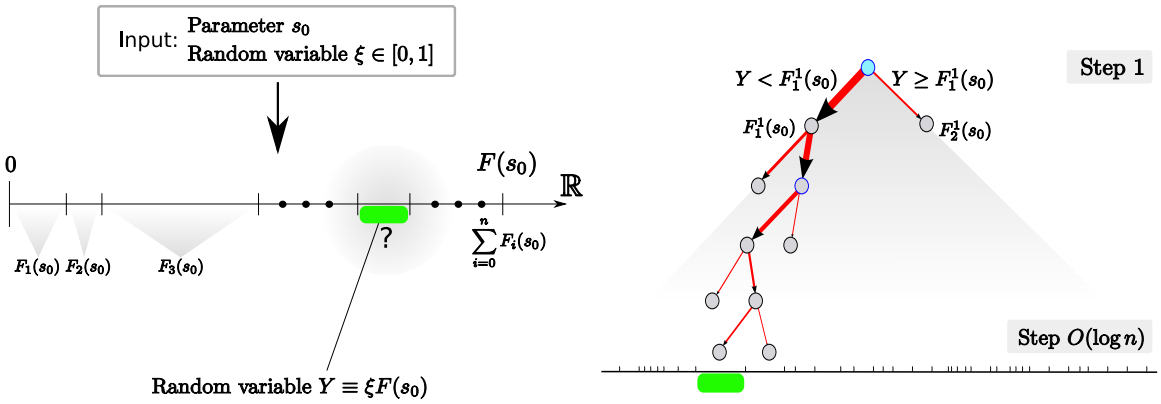


Figure 3.2: *Given the input parameter value, s_0, and a random variable $\xi \in [0,1]$, the tree can be used to select one out of n subdomains with $O(\log n)$ branching steps. The domain corresponding to the leaf with the largest value of the integral of the density function, is most likely to be chosen. The problem of subdomain selection is transformed into a simple 1D range query of $Y \equiv \xi F(s_0)$ within $[0, F(s_0)]$ amongst the intervals.*

the probabilities, $F_i(s)$, of each subdomain $\mathcal{D}_i$. Each internal node stores a function that is the sum of all the functions of its children. If the functions are represented by storing coefficients resulting from their projection onto some standard set of basis functions, internal nodes are computed by simply adding coefficients (see Figure 3.1). Thus the root of the tree stores the integral of the density over the entire domain, and it is this value that will be used for normalization of the density associated with each sample.

3.2.2 Tree traversal

The parameterized probability tree is used for performing range queries on a uniformly distributed random variable, against the various partitions of the domain. The inputs to the traversal routine are: (1) a parameter value $\mathbf{s}_0$ and (2) a uniformly distributed random variable, $\xi \in [0,1]$. While fixing $\mathbf{s}_0$ defines the density function over the spatial domain, ξ defines a tree traversal path from the root to a leaf. The leaf at

which the traversal ends is the selected subdomain. Thus, starting with a uniformly distributed random variable, the data structure can be used to select a subdomain with a probability proportional to the integral of the density function within this subdomain.

The first step in the traversal algorithm is to construct a new random variable $Y = \xi F(s)$. The problem of selecting a subdomain is equivalent to concatenating the different $F_i(s)$ in any arbitrary order, keeping track of the cumulative distribution, and finding that interval into which Y fits (see Figure 3.2). Since each node of the tree stores a function of the parameter, simply plugging $\mathbf{s}_0$ into each internal node yields a value that corresponds to the combined volume of all the leaves it its subtree.

At each point in the traversal, a simple comparison is made of Y against the volume of each of the child nodes. The child node corresponding to the interval into which Y falls is visited next. Thus in $O(\log n)$ asymptotic time, a path is obtained from the root to the subdomain corresponding to the interval into which Y falls. This leaf node has been chosen with a probability proportional to the integral of the density function within it.

Finally, once the subdomain has been selected and a random sample location has been drawn within this subdomain (the assumption is that there is a way to do so), the normalized density associated with choosing that sample location is simply the density evaluated at that location divided by $F(s_0)$.

3.3 Steerable importance sampling

This section describes an extension to importance sampling that allows the use of a dynamically varying importance function which is steerable (or can be reasonably

approximated by a steerable function). The abstract notion of steerable importance sampling is first provided (Section 3.3.1), followed by an example of its applicability in estimating direct illumination from distant light sources (Section 3.4).

3.3.1 Motivation

Consider the Monte Carlo integration of functions that can be expressed as a product of two functions, where the latter is steerable. That is,

$$\begin{aligned} I(\mathbf{s}) &= \int_{\mathcal{D}} h(\mathbf{x}, \mathbf{s})\, \mathrm{d}\mathbf{x} \\ &= \int_{\mathcal{D}} f(\mathbf{x})\, g(\mathbf{x}, \mathbf{s})\, \mathrm{d}\mathbf{x}, \end{aligned} \tag{3.8}$$

where $\mathcal{D}$ is any domain of integration and $g(\mathbf{x}, \mathbf{s})$ is a steerable function, or can be reasonably approximated by a steerable function. Importance sampling is a commonly used strategy for efficiently estimating such integrals. While the choice of the importance function alone cannot introduce a bias in the estimates, the variance of the estimates depends on this choice.

It is well known that the more closely an importance function resembles the integrand, the more effective the importance sampling will be. If $I(\mathbf{s})$ is to be estimated for several values of $\mathbf{s}$, choosing $g(\mathbf{x}, \mathbf{s})$ as an importance function could be a reasonable choice. Although it is easy to construct specific situations where $f(\mathbf{x})$ would be a better importance function, choosing $g(\mathbf{x}, \mathbf{s})$ to be the importance function and factorizing $h(\mathbf{x}, \mathbf{s})$ in a way that $f(\mathbf{x})$ is not a rapidly varying function can be expected to reduce the variance of the estimates. Although the factorization of the integrand is just as important for reducing the variance in the estimates, it is a problem that depends on the actual function, and thus is closely tied to specific applications. Hence

we focus on the problem of using a steerable function as an importance function.

The integral in Equation (3.8) can be expressed as a sum of integrals over a number of subdomains

$$I(\mathbf{s}) = \int_{\mathcal{D}_1} f(\mathbf{x})\, g(\mathbf{x}, \mathbf{s})\, \mathrm{d}\mathbf{x} + \int_{\mathcal{D}_2} f(\mathbf{x})\, g(\mathbf{x}, \mathbf{s})\, \mathrm{d}\mathbf{x} + ... + \int_{\mathcal{D}_n} f(\mathbf{x})\, g(\mathbf{x}, \mathbf{s})\, \mathrm{d}\mathbf{x},$$

where $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup ... \cup \mathcal{D}_n$ such that $\mathcal{D}_i \cap \mathcal{D}_j = \{\}\ \forall\ i \neq j$. This domain partitioning is useful since it is usually easier to sample from the steerable importance function, $g(\mathbf{x}, \mathbf{s})$ (or its approximation), over the subdomains rather than over $\mathcal{D}$. For example, $g(\mathbf{x}, \mathbf{s})$ could be approximated with a piecewise constant function and the subdomains could be simplices of the appropriate dimension. The disadvantage of partitioning the domain and sampling independently from each domain is that it becomes more difficult to make guarantees about the quality of distribution of samples between subdomains.

The overall strategy for drawing each sample, from a steerable function proceeds in two steps: (1) randomly choosing a subdomain, accounting for the integral of the importance function within it; (2) drawing samples from within that subdomain, according to the distribution within it. The data structure described in Section 3.2 can be used for selecting a subdomain. The latter step is closely related to the actual approximation and partitioning of the domain. If a piecewise constant approximation is used in conjunction with simplices as the subdomain, the second step reduces to simply generating random samples within simplices. We illustrate the overall procedure using an example, describing the weighting process in detail (see Section 3.4).

3.3.2 Selecting a subdomain

Given the partitioned domain $\mathcal{D}$ and its partitions $\mathcal{D}_i,\ i = 1, 2, 3, ..., n$ the problem, that we are interested in, is to randomly select a subdomain $\mathcal{D}_k$ according to the integral of the importance function $g(\mathbf{x}, \mathbf{s})$ within the partitions; the importance function is defined for a given parameter value $\mathbf{s} = \mathbf{s}_0$.

One possible algorithm to solve this problem would be to actually integrate the importance function over each partition, and use a hierarchical structure structure similar to probability trees [67]. The problem, however, is that the integral of the importance function within each domain is dependent on the value of the parameter $\mathbf{s}$. Thus, for a new value of $\mathbf{s}$, the integrations would have to be repeated. In this section we describe a way in which the steerability of the importance function could be exploited, to construct a more efficient algorithm.

If $g(\mathbf{x}, \mathbf{s})$ is a steerable function we can write it as an inner product of a steering function and a steerable basis vector. Further, if the integral of the importance function over each subdomain $\mathcal{D}_i$ is steerable we can write

$$F_i(s) = \langle \Psi(\mathbf{s}),\, \Phi(\mathbf{x}) \rangle \,, \tag{3.9}$$

where $\Psi(\mathbf{s})$ is the steering vector and $\Phi(\mathbf{x})$ is the basis vector. This is trivially satisfied if the importance function is approximated by a piecewise constant function over a triangular domain, for instance. Although this is not a harsh constraint (especially since approximation by a steerable function is sufficient), care needs to be taken to ensure that this is actually the case.

Recall that the data structure presented in Section 3.2 allowed the selection of a subdomain problem in logarithmic worst case asymptotic time, of the number of sub-

domain. The assumptions made by the data structure were that: (1) the probabilities $F_i(s)$ could be computed in constant time at each of the leaves ; (2) at each internal node, the sum of probabilties of the child nodes could be computed in constant time. The first of these two requirements can be satisfied by storing precomputed $\Phi(\mathbf{x})$ vectors at each leaf. The vectors associated with internal nodes are simply the sum of the vectors stored in the child nodes.

The tree traversal, for a given value of the parameter $\mathbf{s}$ is performed by first computing $\Psi(\mathbf{s})$ and then performing innerproducts at each stage from the root down to a leaf node. The number of dot products required is $O(k \log n)$ where k is the number of bases used in the steerable representation of $F_i(s)$. For functions which can be represented with few bases, k can be considered a constant and the method is logarithmic in the number of partitions of the domain. The method is practicable if the integral of the importance function within each partition of the domain is narrow-band or at least reasonably approximated by a narrow-band function.

3.3.3 Sample weight computation

The weight associated with each sample must take into account two factors: (1) the probability of choosing the particular subdomain from which the sample was drawn ; (2) the probability density with which the sample was chosen within the subdomain. The exact weight used depends on the partitioning scheme and the approximation within the subdomains of the importance function (see Section 3.4 for an example).

3.4 Application: Environment map sampling

Importance sampling strategies appear in a wide variety of forms, from sampling incident illumination using a simple cosine distribution, to finely adapting the sampling to a particular BRDF, or to features of the environment. In recent years considerable attention has been given to importance sampling of environment maps. There are two justifications for this focus: First, environment maps frequently encode high-dynamic range light sources [32] and therefore represent a significant challenge for efficient sampling. Secondly, light from distant sources, as represented by an environment map, is spatially independent, which greatly simplifies the task of importance sampling by reducing the dependence of such distributions to direction only.

In the context of estimating reflected radiance, a variance reduction strategy must meet several inherent requirements [58], plus an additional property that should be met if at all possible:

1. Estimate the distribution of incident illumination
2. Generate samples distributed according to the estimated illumination
3. Compute the density of each sample
4. Maintain stratification (if possible)

If the incident illumination is defined by an environment map, the first requirement is partially met; the only additional aspects that should be addressed are occlusion and weighting by the cosine of the incident angle, as the incident radiance is always integrated with respect to projected solid angle. The second requirement can be met by approximating the incident illumination using piecewise constant functions, or other simple approximations [63], which admit sampling algorithms. The third

requirement is that of computing the density with which a given sample was drawn which requires that the pdf be normalized. This can always be accomplished through numerical integration of the approximating function. However, such normalization is generally significantly more costly than drawing samples, as it involves the entire importance sampling function. We refer to the latter as *the renormalization problem*, as it is frequently a significant challenge to achieving unbiased importance sampling that is computationally feasible.

This section presents a strategy to sample a product of two functions and demonstrate that it can be used to efficiently sample high-dynamic-range environment maps to estimate reflected radiance. Several methods have been proposed to efficiently sample environment maps and some of them even sample from the product of illumination and surface reflectance functions (see Section 3.4.1). Here, we describe how this could be achieved by defining a steerable importance function to be the product of sharply varying incident illumination and a smooth steerable function; we also detail the scheme to draw correctly-weighted samples from this importance function. By sampling from an importance function that is the product of illumination over the sphere of directions and the positive cosine lobe defined by the surface normal, the variance in the estimate can be reduced since (1) "wasted" samples that lie below the tangent plane are not generated and (2) directions that are close to the horizon are down-sample.

While the illumination is known a priori, the importance function also depends on a dynamically oriented clamped-cosine lobe. When accounting for changes in the distribution of incident illumination above the local tangent plane and/or weighting by the cosine of the incident direction, all but the first requirement become more difficult to meet: generation of samples, computation of the densities, and maintaining stratification. These difficulties stem, in part, from the problem of renormalizing the

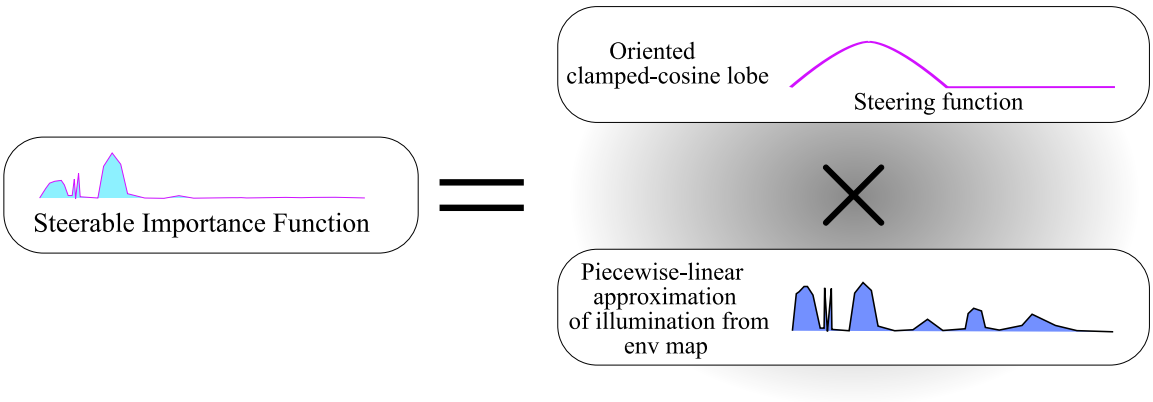


Figure 3.3: *Figure shows the lower dimension analog of our steerable importance function for one orientation of the clamped cosine and a 2D environment map which is a function of* $u \in \mathcal{S}$.

constantly-changing distribution.

We derive a method that solves the renormalization problem decisively by means of a novel hierarchical organization that encodes all possible variation very efficiently in advance using what amounts to *steerable* functions. We consider the surfaces being rendered with respect to the environment maps to steer the importance function using a cosine lobe defined by the surface normal. This lobe is *clamped* to zero at the tangent plane of the surface, which has the effect of ignoring all illumination that arrives from below the tangent plane.

3.4.1 Reflected radiance

Reflected radiance $\mathbf{L}(x, \omega_\mathbf{r})$ due to direct illumination from distant sources, can be expressed as the integral

$$\int_{\mathcal{S}^2} \mathbf{L}(\omega_\mathbf{i}) \rho(x, \omega_\mathbf{i}, \omega_\mathbf{r}) max(\mathbf{n}.\omega_i, 0) V(x, \omega_\mathbf{i}) d\omega_\mathbf{i} \tag{3.10}$$

since radiance incident from distant sources, $\mathbf{L}(\omega_\mathbf{i})$, is only a function of direction. Here $\mathbf{x}$ is a point, with normal $\mathbf{n}$, on a surface with ρ as its bidirectional reflectance distribution function (BRDF) and the integration is over the sphere of incoming directions $\omega_\mathbf{i}$. We will refer to the third and fourth terms in the integrand as the *clamped-cosine* and visibility terms respectively.

A number of sampling strategies have been proposed to efficiently estimate this integral; these methods fundamentally differ in their choice of an importance function (see Section 1.6). The method of structured importance sampling [2] defines an importance function that is a carefully chosen combination of illumination density and solid angle separating the samples. The samples are distributed using a point relaxation algorithm and the incident illumination is approximated with several point light sources. Further, as an acceleration technique, the light sources are sorted in decreasing order of power and sampled deterministically in that order. Another method that approximates the illumination with point light sources [77], extends hierarchical Penrose tiling to quickly sample the 2D environment map; the samples also satisfy certain noise properties. While both these techniques require far fewer samples than naïve sampling of only the clamped-cosine term to produce images of similar visual quality, many (about half) of the samples generated are likely to lie below the tangent plane and thus be rejected. In addition, the cosine term is ignored which means that bright sources near the horizon are sampled as profusely as sources of similar power at the pole.

Some methods include the surface BRDF [18, 26, 24] in the importance function. This allows efficient sampling of a combination of high frequency illumination and glossy surfaces with a large specular component. Lawrence et al.. [63] introduced a fairly general numerical method for approximating and numerically inverting cumulative distribution functions, which lends itself to both stratification and importance

sampling. Ghosh et al.. [45] proposed a method to account for temporal coherence in animation sequences involving environment maps. In this paper, we describe a method that uses the clamped-cosine weighted illumination as an importance function thus automatically accounting for the cosine importance given a normal direction, and also ensuring that all samples are generated above the tangent plane.

Ramamoorthi and Hanrahan first observed that a clamped cosine lobe could be very effectively approximated using a small number of spherical harmonic basis functions; indeed, nine such coefficients attains a fit that is deemed sufficiently accurate for most graphics applications [85]. They also observed that the spherical harmonic representation of a rotated lobe is no more complex than a static one in that no higher-order terms are added as a result of any rotation. It is precisely these observations that we build upon here to obtain an importance sampling function that can dynamically account for any incident surface orientation by pre-computing its response to a steerable lobe; in this case, a clamped cosine lobe.

We shall see that this solves the renormalization problem by making renormalization of an arbitrary piecewise linear importance sampling function equivalent in cost to re-weighting a single point in the environment map. We first approximate the environment map as a piecewise-linear 2D function; Section 3.4.2 explains how the piecewise-linear function can be re-weighted and re-normalized very efficiently by a clamped-cosine lobe, thus making the entire importance-sampling function "steerable". Stratified sampling of our piecewise-linear importance sampling function is performed using the method described in Section 2.3.1. Figure 3.3 shows such an importance sampling function, that is dynamically re-weighted and re-normalized via a steerable clamped cosine lobe along with the stratified samples drawn from it.

3.4.2 The steerable importance function

By partitioning the domain of integration in Equation (3.10), $\mathcal{S}^2$, into spherical triangles $\mathcal{S}_i$, $i = 1, 2, .., M$ we rewrite the r.h.s. of the equation as the sum of integrals over $\mathcal{S}_i$

$$\sum_{i=1}^{M} \int_{\mathcal{S}_i} \mathbf{L}(\omega)\rho(x, \omega, \omega_{\mathbf{r}}) f\,(\mathbf{n}, \omega)\, V(x, \omega)\, \mathrm{d}\omega, \tag{3.11}$$

where $f\,(\mathbf{n}, \omega) = \max(\langle \omega,\, \mathbf{n} \rangle\,,\, 0)$.

To efficiently estimate this integral, we use the method outlined in Section 3.3.1. Here, the sphere of directions, $\mathcal{S}^2$, corresponds to the domain $\mathcal{D}$ and the spherical triangles $\mathcal{S}_i$ correspond to subdomains $\mathcal{D}_i$. The importance function is approximated by a piecewise constant function, defined on the planar triangle defined by the vertices of the spherical triangles of each subdomain. In this section, we show that this leads to a steerable representation for the integral of the importance function, within each spherical triangle.

Consider one of the spherical triangles, $\mathcal{S}_i$, and the planar triangle $\triangle(i)$ defined by the vertices of $\mathcal{S}_i$. Let $\mathbf{p} = (p_0, p_1) \in [0, 1]^2$ be a point on $\triangle(i)$ defined using the parametrization $\psi : [0, 1]^2 \rightarrow \triangle$. Disregarding visibility and the BRDF for the moment, and switching to the above parameterization, we get

$$\int_0^1 \int_0^1 L(\mu_i(p_0, p_1))\; f\,(\mathbf{n}, \mu_i(p_0, p_1))\;\; \varphi_i(p_0, p_1)\;\; |J_i(p_0, p_1)|\; \mathrm{d}p_0\; \mathrm{d}p_1 \tag{3.12}$$

where $\varphi_i(p_0, p_1)$ arises as a result of using a change of variables from the plane onto

the sphere and $\mu_i(p_0, p_1)$ is the unit vector along $\psi_i(p_0, p_1)$. That is,

$$\mu_i(p_0, p_1) \;=\; \frac{\psi_i(p_0, p_1)}{\|\psi_i(p_0, p_1)\|}, \quad \varphi_i(p_0, p_1) \;=\; \frac{\mu_i(p_0, p_1) \cdot \mathbf{n}_{\triangle(i)}}{\|\psi_i(p_0, p_1)\|^2} \tag{3.13}$$

and $\mathbf{n}_{\triangle(i)}$ is the unit normal of $\triangle(i)$.

We normalize the function $|J_i(p_0, p_1)|$, to make it a pdf and obtain the Monte Carlo estimator

$$G_i \sum_{j=1}^{N} L(\mu_i^j) f\left(\mathbf{n}, \mu_i^j\right) \varphi_i^j \tag{3.14}$$

where samples ψ_i^j drawn from the pdf that is proportional to $|J_i(p_0, p_1)|$ are used to obtain μ_i^j and φ_i^j. We derive ψ_i such that the Jacobian is linear in both parameters and equal to the illumination weighted by the clamped cosine at each vertex of $\triangle(i)$ (see Appendix). The normalization factor G_i is simply the integral of the Jacobian and is given by

$$G_i = \int_0^1 \int_0^1 |J_i(p_0, p_1)| \, dp_0 \, dp_1. \tag{3.15}$$

Replacing the BRDF and visibility terms and adding the estimates over all spherical triangles $\mathcal{S}_i$, we arrive at our estimate of the total reflected radiance along $\omega_\mathbf{r}$ as

$$\sum_{i=1}^{M} \quad G_i \sum_{j=1}^{N} \mathbf{L}(\mu_i^j) \rho(x, \mu_i^j, \omega_\mathbf{r}) f\left(\mathbf{n}, \mu_i^j\right) V(x, \mu_i^j) \varphi_i^j \tag{3.16}$$

Note that the piecewise linear importance function is a linear interpolation of the product of illumination along directions given by the vertices in the partition of the sphere of directions and their corresponding clamped cosines for a given normal.

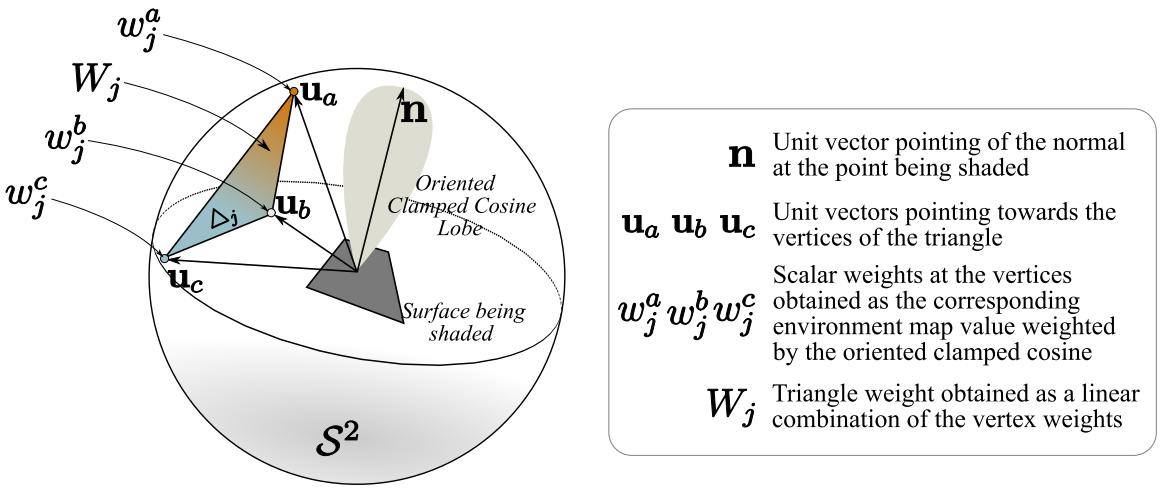


Figure 3.4: *The weight of each triangle vertex is determined by the environment map and its position relative to the surface under consideration. The weight of the entire triangle is a linear combination of its vertex weights.*

3.4.3 Hierarchical steerable bases

To represent the steerable function in a form that can be stored in the parametrized probability data structure, we shall use the spherical harmonic approximation of the clamped cosine lobe first proposed by Ramamoorthi and Hanrahan [85] in the context of fast approximations of irradiance due to distant sources. Our application will differ fundamentally, but will nonetheless enjoy the benefits of concise representation and fast evaluation. First, observe that the function

$$f(\mathbf{u}, \mathbf{n}) \;=\; \max(\langle \mathbf{u},\, \mathbf{n} \rangle, 0), \tag{3.17}$$

hich is what we have been referring to as a clamped cosine lobe, can be approximated by a finite linear combination of spherical harmonics (SH):

$$f(\mathbf{u}, \mathbf{n}) \approx \sum_{i=0}^{k} \mathbf{a}_i(\mathbf{n}) \mathbf{Y}_i(\mathbf{u}), \tag{3.18}$$

where we have treated the spherical harmonics as functions defined on the sphere, $\mathcal{S}^2$, rather than the more traditional function of two angles. We have also "linearized" the indexing of the basis functions, which are traditionally indexed with double subscripts denoted by ℓ and m, with $\ell = 0, 1, 2, \ldots$, and $-\ell \leqslant m \leqslant \ell$. In particular, our ordering coincides with the subscripts $(0,0)$, $(1,-1)$, $(1,0)$, $(1,1)$, $(2,-2)$, $(2,-1)$, $(2,0)$, $(2,1)$, and $(2,2)$, etc. Here $\mathbf{a}(\mathbf{n})$ is the vector of SH coefficients after being rotated using the normal $\mathbf{n}$.

The product of the incident distant illumination along $\mathbf{u}$ and the clamped cosine at $\mathbf{u}$ for a given normal $\mathbf{n}$ can be expressed as

$$\begin{aligned} \mathbf{L}(\mathbf{u}) f(\mathbf{u}, \mathbf{n}) &\approx \mathbf{L}(\mathbf{u}) \sum_{i=0}^{k} \mathbf{a}_i(\mathbf{n}) \mathbf{Y}_i(\mathbf{u}) \\ &= \sum_{i=0}^{k} \mathbf{L}(\mathbf{u}) \mathbf{a}_i(\mathbf{n}) \mathbf{Y}_i(\mathbf{u}) \\ &= \langle \mathbf{a}(\mathbf{n}),\, \mathbf{w}(\mathbf{u}) \rangle \end{aligned} \tag{3.19}$$

where $\mathbf{w}(\mathbf{u}) = \mathbf{L}(\mathbf{u})\mathbf{Y}(\mathbf{u})$ is a vector containing the SH bases associated with a direction $\mathbf{u}$, weighted by the illumination along that direction. Thus the product $\mathbf{L}(\mathbf{u}) f(\mathbf{n}, \mathbf{u})$ is steerable.

We define the function $|J_j(p_0, p_1)|$ in each triangle $\triangle(j)$ with vertices A, B and C as a linear combination of products of illumination and the clamped cosines at vertices. The integral of the Jacobian (see Equation (3.15)) within each triangle is simply the volume of the truncated triangular prism defined by the triangle. Given that the Jacobian varies linearly within each triangle $\triangle(j)$, we can write G_j as $\langle \mathbf{a}(\mathbf{n}),\, \mathbf{W}_j \rangle$ where

$$\mathbf{W}_j = \frac{\text{Area}(\triangle(j))}{3} \left(\mathbf{w}_j^a + \mathbf{w}_j^b + \mathbf{w}_j^b \right). \tag{3.20}$$

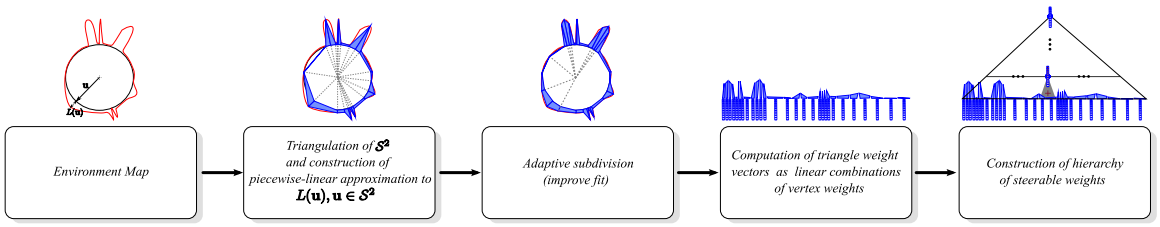


Figure 3.5: *2D illustration of the algorithm used to construct the tree with steerable weights. Here segments are equivalent to triangles in the 3D setting. The vectorsof weights associated with the triangles are at the leaves of the tree, and are propogated up to the root; weights at internal nodes are computed simply as the sumof the weights of their children.*

We precompute and store $\mathbf{w}(\mathbf{u})$ at each vertex in the partition of the sphere of directions and a weight W_j associated with each triangle $\triangle(j)$. Given a normal $\mathbf{n}$ we first compute $\mathbf{a}(\mathbf{n})$ and then G_j in constant time for each triangle $\triangle(j)$ with just one dot product.

Further, we can compute the integral of the piecewise linear importance function over a set of triangles The next observation is crucial: If Q is any set of triangle indices, then

$$\begin{aligned}\sum_{j\in Q} W_j &= \sum_{j\in Q} \langle \mathbf{a}, \mathbf{W}_j \rangle \\ &= \left\langle \mathbf{a}, \sum_{j\in Q} \mathbf{W}_j \right\rangle \\ &= \langle \mathbf{a}, \mathbf{W}_Q \rangle ,\end{aligned}$$

here $\mathbf{W}_Q$ is a new collection of nine coefficients. Thus, the total weight of all the triangles combined is, once again, represented by a collection of the *same* number of spherical harmonic coefficients; *summing the contributions of any number of triangles in any orientation does not introduce higher-order terms.*

To fully exploit this property, we organize the triangles in the partition of the sphere

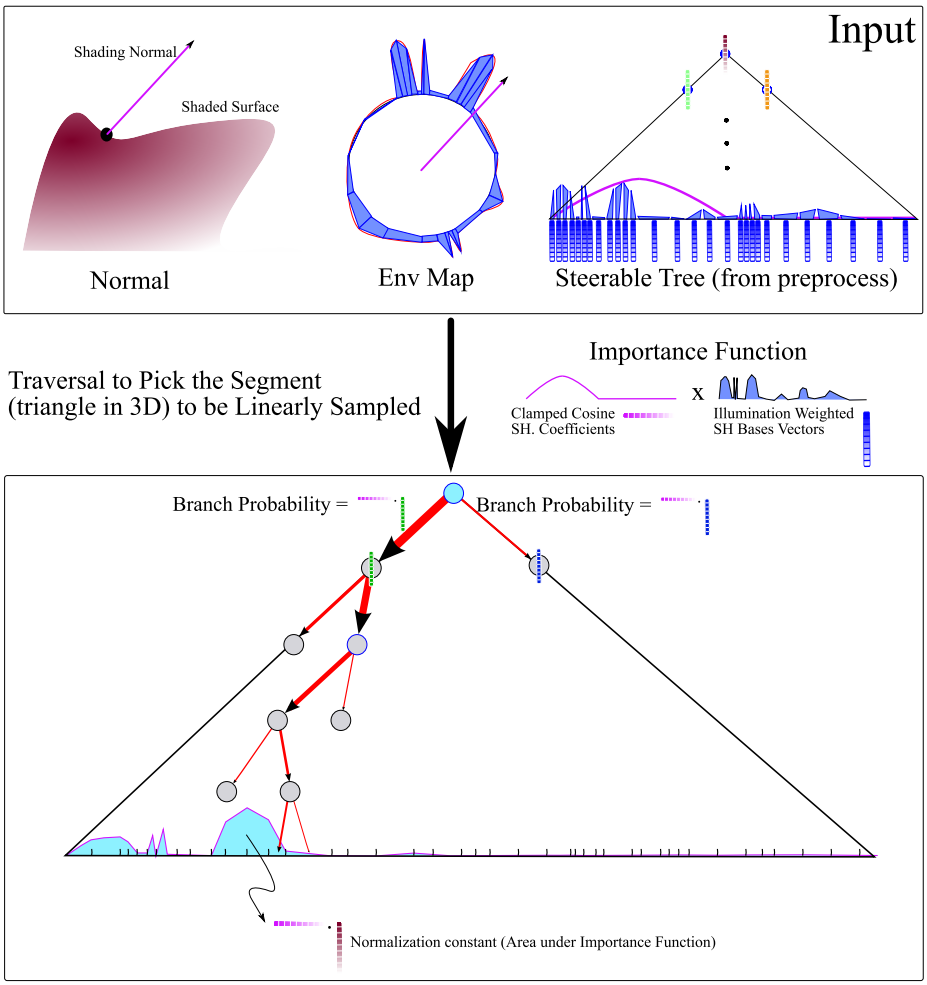


Figure 3.6: *After the spherical harmonic representation of the weight at each vertex has been propagated up to the root, stratified sampling of a function of surface orientation is straightforward: As the cosine lobe is changed, the branching probabilities along each path is altered. To reach a leaf triangle with the correct probability, only 9-element dot products along the path to that triangle are computed. Thus, the cost of generating a sample and computing it correct density is* $O(\log n)$*, where* n *is the number of triangles.*

of directions then organize hierarchically as a binary tree. Each triangle is assigned a vector of nine values, which is then propagated up the tree, adding the weights of the children at each internal node, until the root is reached (see Figure 3.5). The algorithm for generating samples from the resulting piecewise-linear function is illustrated in Figure 3.7, and the algorithms for generating the rotated cosine lobe coefficients and for traversing the three structure are shown in Algorithm 3.1, respectively.

3.4.4 Algorithm

Preprocess: The preprocess step is composed of two main stages– triangulation of the environment map and construction of a reasonably balanced binary tree. While a balanced tree is not required for correctness of the algorithm, balance ensures an $O(S\,log\,N_{\triangle})$ asymptotic bound on the execution time if S stratified samples are required to be drawn for any given normal vector and the triangulation consists of $N_{\triangle}$ triangles. The domain is triangulated by uniform subdivision of an icosahedron followed by a step of adaptive subdivision. During adaptive subdivision, triangles are subdivided if the deviation of the linear approximation within them from the actual illumination is found to be greater than a threshold. After subdivision, vertex and triangle weights are computed and stored.

We build a binary tree that has the triangles randomly distributed as its leaf nodes. Each triangle is associated with a weight, which is the volume of the truncated prism formed by raising its vertices by the appropriate heights. We approximate this volume with one third the area of the triangle times the average height at its vertices. Although this is an approximation and makes the importance function deviate slightly from the actual function on the sphere, it does not introduce a bias so long as the weights computed are in accordance with the densities that samples are drawn from.

This approximation converges to the correct volume as the triangulation is refined.

The internal nodes of the tree represent clusters of triangles and their volumes can each be written as a sum of the volumes of their respective child nodes. Thus we sum up the individual basis vectors of the children to compute the basis vector at each internal node. The actual volume, including the cosine weighting, is computed by a dot product of this weighted basis vector with the coefficient vector of the clamped cosine which is provided during query. Thus we build the tree in a bottom-up fashion, at each node summing up and storing the basis vectors of child nodes (see Figure 3.5).

The volume of the root, which represents the volume under the importance function, is computed for a given normal direction by just one dot product which trivializes the cost of renormalization.

Sample Generation: Given a normal direction and two random variables chosen uniformly in $[0, 1]$ we draw a single sample from our importance function in three steps: Selecting the triangle to sample from, drawing a sample from that triangle according to the weights defined at the vertices and actually computing the density with which the sample was chosen.

Starting with the root we evaluate the volume at each internal node (one dot product each) and use the information to guide the path down to the leaf. At each level the path favors the child with a higher volume (see Figure 3.7). Thus using one of the random variables, and $O(log\, N_{\triangle})$ inner products (each of 9 coefficient vectors), we pick a triangle proportional to the integral of the linearly-varying densities (See Algorithm 3.1). Once we pick a triangle we re-scale the random value used to traverse the tree to $[0, 1]$ and sample from the triangle using the two random variables as shown in Section 2.3.1.

Computing the density with which the given sample was chosen is trivially obtained

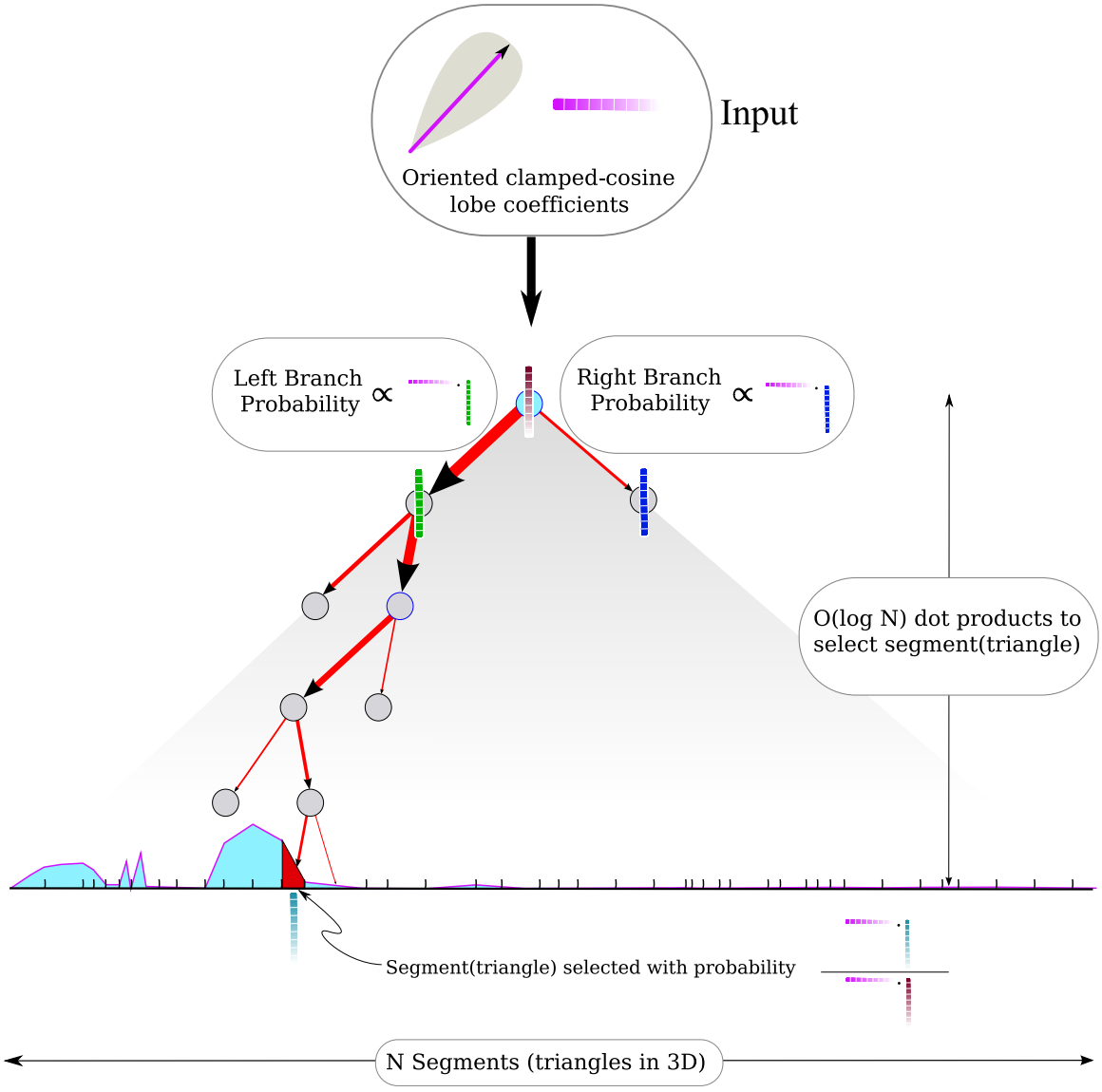


Figure 3.7: *At each node during the traversal the integral of the importance function over the leaf nodes, under its subtree, is approximated with just one dot product. One of the children is visited, depending on the branching probabilities which are proportional to the integrals in each subtree.*

Algorithm 3.1 *The basic algorithm for stratied sampling of the dynamically reweighted piecewise-linear importance function. The variables ξ_1 and ξ_2 are assumed to be stratified random variables in $[0,1] \times [0,1]$. Note that step 18 introduces a bias which can be eliminated by a slight increase in computational cost (Algorithm 3.2). c1 = 0.429043, c2 = 0.511644, c3 = 0.743125, c4 = 0.886227, c5 = 0.247708 according to Ramamoorthi and Hanrahan [85].*

function *Sample* ($\mathbf{n}$, ξ_1, ξ_2)

1: $\mathbf{a} \leftarrow$ *RotateLobeCoeffs*($\mathbf{n}$)
2: $\mathbf{w} \leftarrow$ weight coefficients of tree
3: $v \leftarrow$ root of tree
4: **while** v is not a leaf **do**
5: $\quad w_l \leftarrow \langle \mathbf{a}, \mathit{LeftWeightCoeffs}(v) \rangle$
6: $\quad w_r \leftarrow \langle \mathbf{a}, \mathit{RightWeightCoeffs}(v) \rangle$
7: $\quad w \leftarrow \dfrac{w_l}{w_l + w_r}$
8: $\quad$ **if** $\xi_1 < w$ **then**
9: $\quad\quad \xi_1 \leftarrow \dfrac{\xi_1}{w}$
10: $\quad\quad v \leftarrow \mathit{LeftChild}(v)$
11: $\quad$ **else**
12: $\quad\quad \xi_1 \leftarrow \dfrac{\xi_1 - w}{1 - w}$
13: $\quad\quad v \leftarrow \mathit{RightChild}(v)$
14: $\quad$ **end if**
15: **end while**
16: $(\mathbf{s}_\triangle, \rho_\triangle) \leftarrow \mathit{SampleTriangle}(\ \mathit{Triangle}(v),\ \xi_1,\ \xi_2\ )$
17: **if** $\langle \mathbf{n}, \mathbf{s}_\triangle \rangle < 0$ **then**
18: $\quad \mathbf{s}_\triangle \leftarrow -\mathbf{s}_\triangle$
19: **end if**
20: **return** $\left(\mathbf{s}_\triangle, \dfrac{\rho_\triangle}{\langle \mathbf{a},\ \mathbf{w} \rangle} \right)$

function *RotateLobeCoeffs* ($\mathbf{n}$)

$\mathbf{a}_0 \leftarrow c_4$
$\mathbf{a}_1 \leftarrow 2c_2\mathbf{n}_y$
$\mathbf{a}_2 \leftarrow 2c_2\mathbf{n}_z$
$\mathbf{a}_3 \leftarrow 2c_2\mathbf{n}_x$
$\mathbf{a}_4 \leftarrow 2c_1\mathbf{n}_x\mathbf{n}_y$
$\mathbf{a}_5 \leftarrow 2c_1\mathbf{n}_y\mathbf{n}_z$
$\mathbf{a}_6 \leftarrow c_3\mathbf{n}_z^2 - c_5$
$\mathbf{a}_7 \leftarrow 2c_1\mathbf{n}_x\mathbf{n}_z$
$\mathbf{a}_8 \leftarrow c_1\left(\mathbf{n}_x^2 - \mathbf{n}_y^2\right)$
return a

by the ratio of the actual height at that sample (which is obtained by interpolating between the heights of the vertices) and the total volume associated with all the triangles (which is the volume of the root). Both are evaluated in constant time.

3.4.5 Results and discussion

Figure 3.8 shows the importance function as the product of the environment map values and the oriented clamped cosine lobe for two different environment maps, each with a differently oriented lobe; the resulting samples drawn are also shown.

Figure 3.10 shows images rendered by using steerable importance sampling. Results with different numbers of samples of a scene with diffuse, glossy and specular materials are shown.

The benefit of using the clamped cosine as the steering function for importance sampling is realized when most of the bright luminaires in the environment lie below the tangent plane or close to the horizon. In such situations, traditional environment mapping algorithms like structured importance sampling, that do not account for the local normal, are inefficient because most of the samples are either below the tangent plane or contribute little to the integral. We compare steerable importance sampling against *standard stratified importance sampling* by obtaining irradiance estimates for a set of normal directions by varying the polar angle and comparing variances in the estimates. The standard stratified importance sampling method treats the environment map image as a discrete 2D function from which stratified samples are drawn using numerical inversion [2].

The gain due to the steerability can be seen to achieve significantly lower variance, especially when the number of samples is few or the normal is facing away from

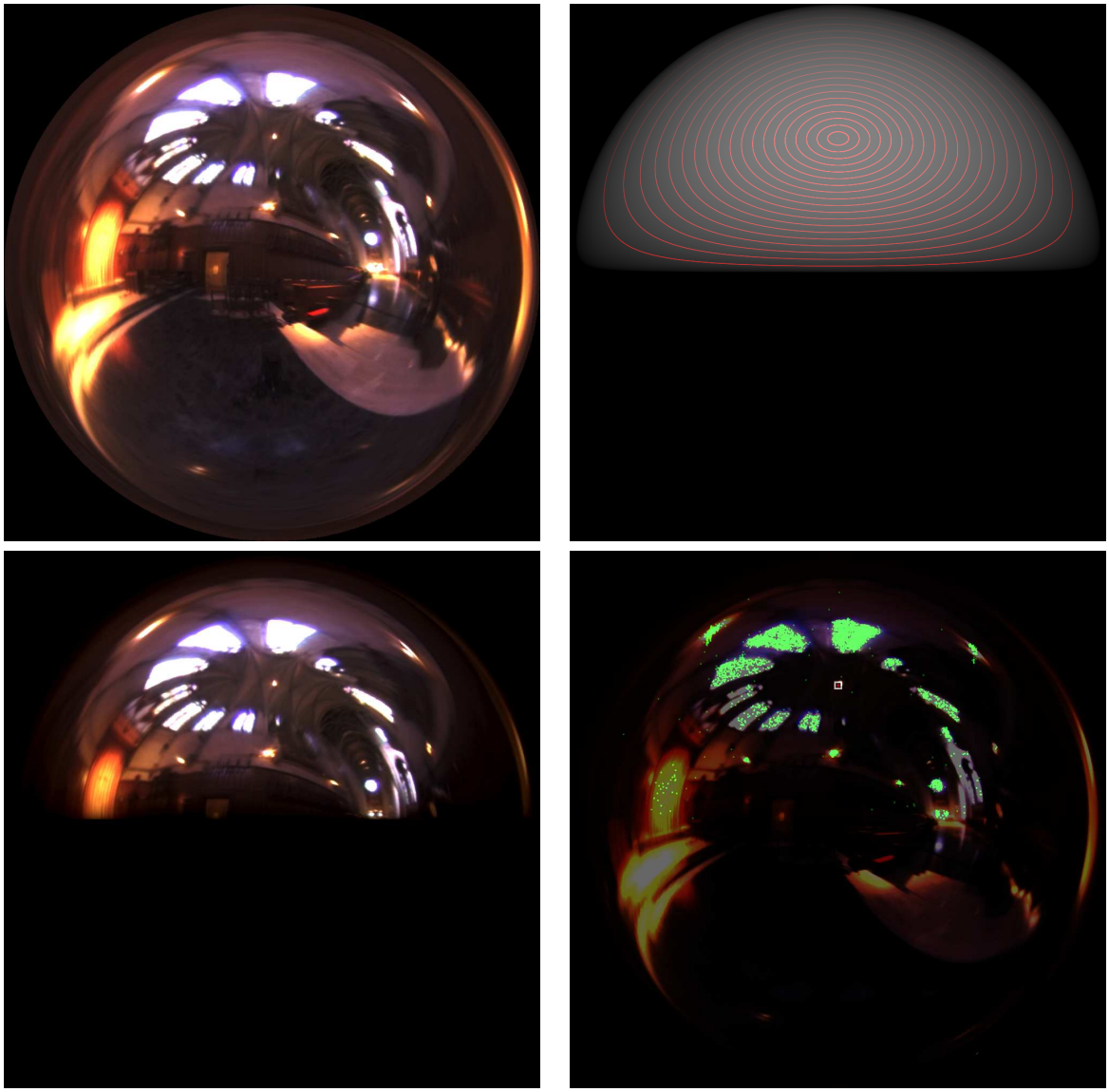


Figure 3.8: *Top left: Input map of Grace Cathedral. Top right: clamped-cosine function (with iso-polar lines in red). Bottom left: the importance function (clamped-cosine weighted input). Bottom right: Samples(green) drawn from the importance function (juxtaposed on dimmed input). Very few of the samples lie in the low-intensity regions of the map and none in the hemisphere below the tangent plane. A large number(100,000) of samples is shown to highlight the effectiveness of the method.*

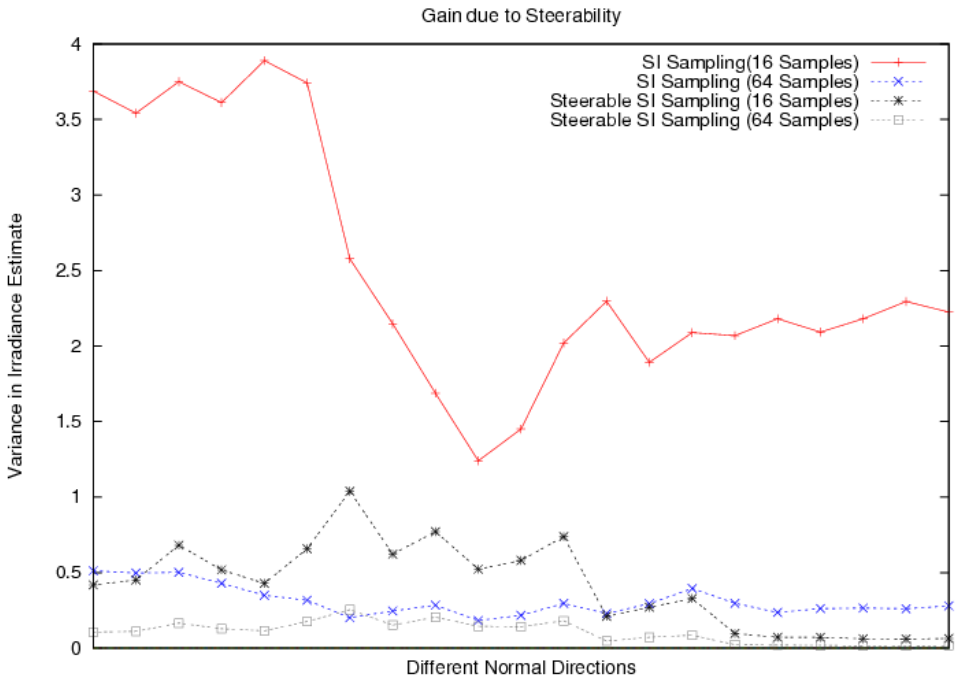


Figure 3.9: *Comparison plots of variance in irradiance estimates using steerable importance sampling against standard stratied importance sampling. The latter uses the 2D density of the illumination in the environment map as an importance function. Tests were run using the St.Peters Basilica environment map, on a set of nor- mal directions by varying the polar angle in the interval* $(0, \pi)$*. There is a significant increase in variance for normals facing away from the illumination using the standard method as a result of not considering local surface orientation in the importance function.*

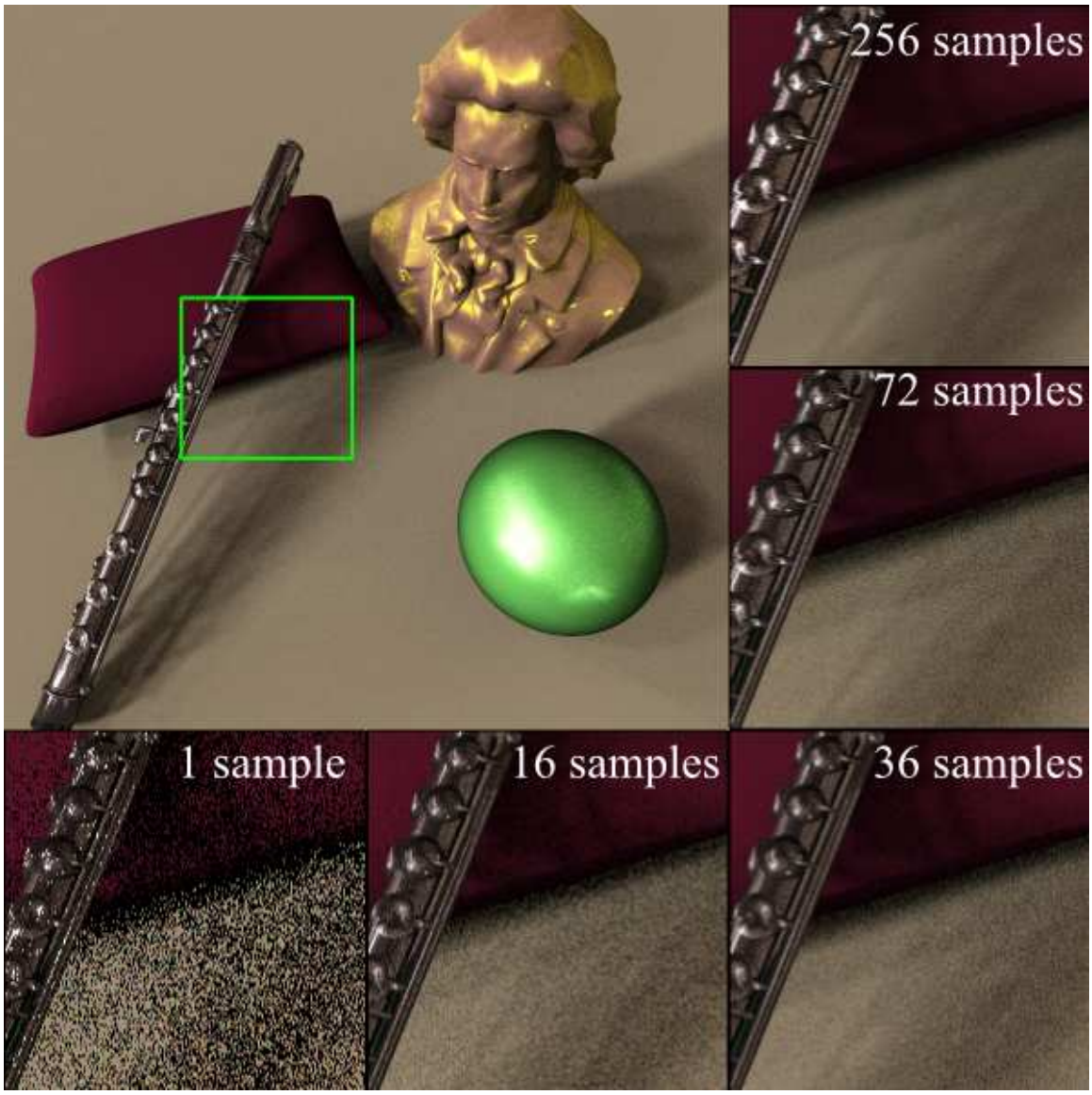


Figure 3.10: *Figure shows images rendered using our sampling algorithm within the "Galileo's Tomb" environment map. Insets show that the variance is tolerable even with few samples and quickly converges as the number of samples is increased.*

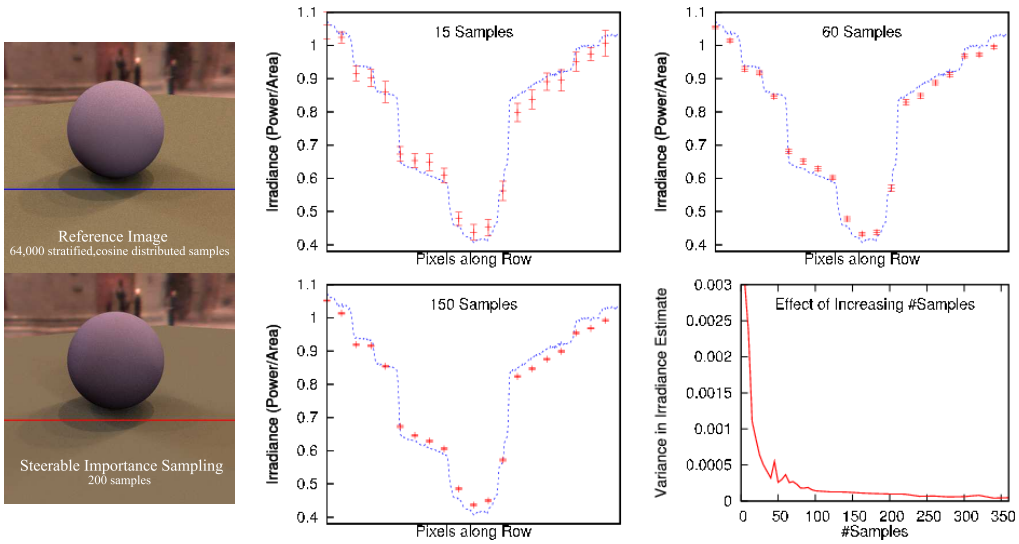


Figure 3.11: *Comparison of the mean of the steerable importance sampling (SIS) estimator against a brute force Monte Carlo estimator (cosine-sampling of hemisphere). Blue dashed lines in the plots are for the trusted estimator while red error bars show the mean and variance for the SIS estimator. Finally, a plot of variance of the SIS estimator, against increasing number of samples is shown (bottom right).*

bright illumination. Figure 3.9 shows the variances in irradiance estimates using stratified importance sampling and steerable stratified importance sampling for 16 and 64 samples. However, there are two possible ways that a bias might be introduced in the estimator.

Ringing in the lobe approximation: Because the clamped cosine lobe is approximated by its projection onto a finite set of basis functions, there is a small amount of ringing near the derivative discontinuity. The ringing causes the approximation to become slightly negative where the lobe is clamped to zero. This is easily fixed by adding an offset (approximately 0.09) to the coeffiient corresponding to the constant basis function. This will increase the values uniformly, thus somewhat reducing the effectiveness of the importance sampling by decreasing the overall variation. This approach completely eliminates the possibility of negatively weighted triangles and negative densities, and introduces no additional bias. However, uniformly raising the

value of the function causes some stray samples, with low probability, to be generated in triangles that should not have been sampled and hence marginally increases the variance of the estimator.

Samples in the wrong hemisphere: A second minor source of bias is due to samples that are occasionally generated below the horizon. This results from an approximation to the clamped cosine lobe that is not exactly zero in the hemisphere below the horizon. Consequently, there is a small probability that it will be sampled. This problem is exacerbated by the global offset that guarantees the function is non-negative. One solution is to simply ignore such samples which amounts to rejection. Another solution is to ignore the bias due to reflecting them into the positive hemisphere; as they occur infrequently, there is little error in any case. However to remove this bias, we increase the density of all samples generated to account for the density of those that arrived there through reflection. Thus, whether a sample falls in the correct hemisphere or not, we add the densities of the two antipodal directions, as shown in Algorithm 3.2. This policy will generate a very low-probability "ghost" of the opposite hemisphere, and is therefore likely to produce a small number of samples that are not very useful, but the resulting estimator will be unbiased.

Algorithm 3.2 *A modified version of the basic algorithm (Algorithm 3.1) for stratified sampling of the dynamically re-weighted piecewise linear importance function. The difference lies in the way samples are weights (lines 17 onwards).*

function *Sample* ($\mathbf{n}$, ξ_1, ξ_2)

1: $\mathbf{a} \leftarrow \mathit{RotateLobeCoeffs}(\mathbf{n})$
2: $\mathbf{w} \leftarrow$ weight coefficients of tree
3: $v \leftarrow$ root of tree
4: **while** v is not a leaf **do**
5: $\quad w_l \leftarrow \langle \mathbf{a}, \mathit{LeftWeightCoeffs}(v) \rangle$
6: $\quad w_r \leftarrow \langle \mathbf{a}, \mathit{RightWeightCoeffs}(v) \rangle$
7: $\quad w \leftarrow \dfrac{w_l}{w_l + w_r}$
8: $\quad$ **if** $\xi_1 < w$ **then**
9: $\quad\quad \xi_1 \leftarrow \dfrac{\xi_1}{w}$
10: $\quad\quad v \leftarrow \mathit{LeftChild}(v)$
11: $\quad$ **else**
12: $\quad\quad \xi_1 \leftarrow \dfrac{\xi_1 - w}{1 - w}$
13: $\quad\quad v \leftarrow \mathit{RightChild}(v)$
14: $\quad$ **end if**
15: **end while**
16: $(\mathbf{s}_\triangle, \rho_\triangle) \leftarrow \mathit{SampleTriangle}(\ \mathit{Triangle}(v),\ \xi_1,\ \xi_2\ )$
17: $\rho^- \leftarrow \mathit{GetDensity}(\mathbf{a}, -\mathbf{s}_\triangle)$
18: **if** $\langle \mathbf{n}, \mathbf{s}_\triangle \rangle < 0$ **then**
19: $\quad \mathbf{s}_\triangle \leftarrow -\mathbf{s}_\triangle$
20: **end if**
21: **return** $\left(\mathbf{s}_\triangle, \dfrac{\rho_\triangle + \rho^-}{\langle \mathbf{a}, \mathbf{w} \rangle} \right)$

{*Modification to eliminate bias*}

Chapter 4

Adaptive, bandwidth-based sampling

Over the last couple of decades, the problem of light transport in image synthesis has grown to become tremendously sophisticated, inspiring theoretical innovations and development of efficient algorithms. Several techniques have been proposed, that address a wide variety of problems in this area: (1) Physically based models and algorithms for simulating complex optical phenomena; (2) efficient algorithms that solve the light transport problem in the context of large, complex scenes that involve billions of optical interactions; (3) algorithms that exploit recent developments in graphics hardware for approximating solutions to simple light transport problems in real-time; (4) strategic preprocessing techniques that relieve the computational burden during the main algorithm's execution; (5) reduced error compression schemes for storing precomputed data. Carefully-chosen, informed sampling strategies play a crucial role in substantially improving the efficiency for each of the above classes of problems.

Smart sampling techniques are indispensible in the context of compact representation and high-fidelity reconstruction of functions. The general nature of this well-studied problem has resulted in a vast body of literature, spanning multiple fields. Several sampling algorithms, with different qualities, have been proposed depending on the characteristics exhibited by the functions under consideration.

Monte Carlo integration is another important context where intelligent sampling has a dramatic effect. The error in Monte Carlo estimates is inversely proportionate to a function[1] of the number of samples used. One strategy for reducing error is to simply increase the number of samples used in estimation. This quickly becomes impractical, since integrals that appear in light transport are of high dimensionality [30, 114]. Another, more practical solution is to control the allotment of number based on the expected error. This amounts to predicting the rate at which the integrand varies, since the error in Monte Carlo integration is proportionate to the variance of the integrand.

In this chapter, after providing a brief introduction to some common analysis tools (see Section 4.1), we present a study of the radiance function in image synthesis from a signal processing perspective. For this we use a framework proposed by Durand et al [36] which is described (with a few embellishments) in Section 4.2. Then, we explore the use of the analysis framework to suggest sampling strategies for both kinds of problems discussed above—adaptive sampling for reconstruction and prediction of the variance of integrands. We apply the sampling strategies for image synthesis with a finite-size-aperture camera model (see Section 4.3) since the two problems form an interestingly antagonistic pair while simulating depth of field.

[1] The exact function depends on the sampling technique used. The convergence of naïve Monte Carlo is $O(1/\sqrt{n})$.

4.1 Frequency analysis: A brief review

The study of *signals*—physical quantities, usually measurable through time or space—is of great interest in engineering; the description of mathematical tools for analysing signals constitutes a substantial body of literature. One such tool for analysis involves projection of a signal onto a number (possibly infinite) set of basis functions and studying the distribution of the results of the projection. The particular characteristics of the signal that this distribution provides insight into depend on the properties of the bases.

Commonly, the set of basis functions is chosen so that the functions represent different "scales" of variation within the domain. Decomposing signals as a linear combination of such functions—that vary at different rates—provides invaluable information about the overall trends of the original signal. This process, called *frequency analysis*, has become a tremendously popular tool with applications in a wide variety of fields.

This section provides a brief review of concepts in digital signal processing that will be used in the following sections of this chapter.

4.1.1 The Fourier series

A real signal, $f : \mathbb{R} \to \mathbb{R}$, is called a *periodic signal* with period T if $f(x) = f(x+T)$. A *finite signal* in time (or space), $g : [a, b] \to \mathbb{R}$ is one that is defined over an interval $[a, b]$. The *duration* of a finite signal in time is defined as the difference between its intervals, $b - a$. Using the finite signal g, a new signal can be constructed as

$$g'(x) = \begin{cases} g(x) & \text{if } x \in [a, b] \\ 0 & \text{otherwise} \end{cases},$$

which can be used to define a periodic signal

$$\sum_{m=-\infty}^{m=\infty} g'(x - mT)$$

with period $T = b - a$. The periodic signal is simply the sum of all versions of the finite signal that have been shifted in time by multiples of the latter's duration. Thus a periodic signal can be defined in terms of a finite signal, which represents one period, and a finite signal can be defined in terms of a periodic signal, by extracting one period.

Any periodic signal, and consequently any finite signal, can be described as the sum of sinusoidal signals. This result, known as the *Fourier series*, was proposed in the nineteenth century by Joseph Fourier. The computation and study of Fourier series is known as *harmonic analysis*, and is extremely useful in decomposing approximations or solutions to large periodic signals into smaller chunks that are relatively simpler to solve. For example, consider the problem of solving a linear, homogeneous ordinary differential equation: If the equation can be solved in the case of a single sinusoid, solutions to larger, more complicated functions can be approximated by representing the functions as sums of sinusoids and appropriately summing the individual solutions.

As initially proposed by Fourier the series expansion of $f(x)$, $x \in [-\pi, \pi]$, is given as

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos nx + \sum_{n=1}^{\infty} b_n \sin nx \tag{4.1}$$

where

$$
\begin{aligned}
a_0 &= \frac{1}{\pi}\int_{\pi}^{\pi} f(x)\ \mathrm{d}x \\
a_n &= \frac{1}{\pi}\int_{\pi}^{\pi} f(x)\cos nx\ \mathrm{d}x \\
b_n &= \frac{1}{\pi}\int_{\pi}^{\pi} f(x)\sin nx\ \mathrm{d}x.
\end{aligned}
\tag{4.2}
$$

The result can be generalized to other intervals than $[-\pi, \pi]$ by a simple change of variables to transform the limits of integration. Further, the notion of Fourier series may be extended to complex coefficients: If f_T is periodic in $[-T/2, T/2]$ the complex Fourier series expansion of f_T can be written as

$$
\begin{aligned}
f_T(x) &= \sum_{n=-\infty}^{\infty} A_n\, e^{\,i(2\pi nx)/T} \\
A_n &= \frac{1}{T}\int_{-T/2}^{T/2} f_T(x)\, e^{\,-i(2\pi nx)/T}\ \mathrm{d}x.
\end{aligned}
\tag{4.3}
$$

The fourier series converges to the function at points of continuity and the average of the two directional limits at points of discontinuity.

4.1.2 The Fourier transform

The complex coefficients in the Fourier series expansion (see Equation (4.3)) are functions of the time period of the original signal. We define new coefficients

$$\begin{aligned} F_n(\nu) &\equiv T A_n(T) \\ &= \int_{-T/2}^{T/2} f_T(x)\, e^{-i(2\pi\nu x)}\, \mathrm{d}x \end{aligned} \tag{4.4}$$

by expressing A_n in terms of $\nu = n/T$. The Fourier series expansion in terms of these new coefficients is given as

$$f_T(x) = \frac{1}{T} \sum_{n=-\infty}^{\infty} F_n(\nu)\, e^{i(2\pi\nu x)}. \tag{4.5}$$

As T increases, ν decreases, causing the resolution within the summation to increase. That is, the larger the time period of the signal, the finer the terms in the Fourier series. In the limit, we obtain

$$f(x) \equiv \lim_{T\to\infty} f_T(x) \;=\; \int_{-\infty}^{\infty} F(\nu)\, e^{i(2\pi\nu x)}\, \mathrm{d}\nu \tag{4.6}$$

where

$$F(\nu) = \int_{-\infty}^{\infty} f(x)\, e^{-i(2\pi\nu x)}\, \mathrm{d}x \tag{4.7}$$

is known as the *Fourier transform* of $f(x)$ and is defined for any signal provided the integral in Equation (4.7) converges. Equation (4.6) represents the *inverse Fourier transform* of $F(\nu)$.

Definition 4.1. *The Fourier transform and its inverse are typically denoted using*

operators $\mathcal{F}$ *and* $\mathcal{F}^{-1}$*:*

$$\begin{aligned}(\mathcal{F} \circ f)\,(x) &\equiv \int_{-\infty}^{\infty} f(x)\; e^{\,-i(2\pi\nu x)}\; \mathrm{d}x \\ (\mathcal{F}^{-1} \circ F)\,(\nu) &\equiv \int_{-\infty}^{\infty} F(\nu)\; e^{\,i(2\pi\nu x)}\; \mathrm{d}\nu\end{aligned} \tag{4.8}$$

The result of applying the Fourier transform on a signal is a function of the "frequency" ν. If $f(x)$ is represented as a sum of sinusoids of different frequencies with different amplitudes and phases, $F(\nu)$ defines the amplitude and phase of each sinusoid at frequency ν so that the combination yields $f(x)$. Specifically, if $F(\nu) = p_\nu + iq_\nu$, the magnitude of the sinusoid at frequency ν is given by $|F(\nu)| = \sqrt{p_\nu^2 + q_\nu^2}$ and the phase is given by $\arg F(\nu) = \arctan q_\nu/p_\nu$.

The function $f(x)$ can be visualized to exist in two different forms, in different domains[2]—one being the original domain of the signal (usually time or space) and the other being the "Fourier frequency domain". Transformations applied to $f(x)$ result in corresponding transformations in $F(\nu)$ that are different, in general, from the transformations in the primal space. One such pair of equivalent transformations that is of tremendous importance is the multiplication-convolution pair.

Theorem 4.2. *The Fourier transform of the convolution of two functions is equivalent to the product of the Fourier transforms of the functions. That is,*

$$(\mathcal{F} \circ (f \;*\; g)) \;= (\mathcal{F} \circ f)\,(\mathcal{F} \circ g)\;. \tag{4.9}$$

[2]the domains of $f(x)$ and $F(\nu)$ are called the primal and dual domains respectively.

Proof. The convolution of the two functions can be written as

$$\begin{aligned}(f \ * \ g) &\equiv \int_{-\infty}^{\infty} g(y) f(x-y)\, \mathrm{d}y \\ &= \int_{-\infty}^{\infty} g(y) \left(\int_{-\infty}^{\infty} F(\nu)\, e^{\,i(2\pi\nu(x-y))}\, \mathrm{d}\nu \right) \mathrm{d}y. \end{aligned} \tag{4.10}$$

Rearranging terms and swapping the order of integration, this becomes

$$\begin{aligned}(f \ * \ g) &= \int_{-\infty}^{\infty} F(\nu) \left(\int_{-\infty}^{\infty} g(y)\, e^{\,-i(2\pi\nu y)}\, \mathrm{d}y \right) e^{\,i(2\pi\nu x)}\, \mathrm{d}\nu \\ &= \int_{-\infty}^{\infty} F(\nu) G(\nu)\, e^{\,i(2\pi\nu x)}\, \mathrm{d}\nu \\ &= \left(\mathcal{F}^{-1} \circ (F(\nu)G(\nu))\right). \end{aligned} \tag{4.11}$$

Applying the Fourier transform on both sides yields the desired result. □

4.1.3 The Fourier transform in higher dimensions

The Fourier transform's definition can be extended to higher dimesional domains by simply increasing the dimensionality of the time or spatial variable and, correspondingly, the frequency variables. The k dimensional fourier transform of a function, f, can be written as

$$\begin{aligned}\left(\mathcal{F} \circ f\right)(\mathbf{x}) &\equiv \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} f(\mathbf{x})\, e^{\,-i(2\pi\langle\Gamma,\mathbf{x}\rangle)}\, \mathrm{d}x \\ \left(\mathcal{F}^{-1} \circ F\right)(\Gamma) &\equiv \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} F(\Gamma)\, e^{\,i(2\pi\langle\Gamma,\mathbf{x}\rangle)}\, \mathrm{d}\Gamma, \end{aligned} \tag{4.12}$$

where $\mathbf{x} \in \mathbb{C}^k$ is the spatial variable and $\Gamma \in \mathbb{C}^k$ is the frequency variable. When $k = 2$, we get

$$
\begin{aligned}
(\mathcal{F} \circ f)\,(x_1, x_2) &\equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2)\; e^{\,-i(2\pi(\nu_1 x_1 + \nu_2 x_2))}\; \mathrm{d}x_1\; \mathrm{d}x_2 \\
(\mathcal{F}^{-1} \circ F)\,(\nu_1, \nu_2) &\equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\nu_1, \nu_2)\; e^{\,i(2\pi(\nu_1 x_1 + \nu_2 x_2))}\; \mathrm{d}\nu_1\; \mathrm{d}\nu_2.
\end{aligned}
\tag{4.13}
$$

The Fourier transform is separable, which means that the 2D Fourier transform can be obtained by first applying a 1D Fourier transform with respect to x_1 and then applying another, on the result, with respect to x_2; similarly for the inverse transforms.

4.1.4 Fourier analysis and sampling

The distribution of $|F(\nu)|$ is known as the *power spectrum*. The power spectrum of a function provides intuition about the distribution of the energy of the function, over the range of frequencies. The interval between the minimum and maximum non-zero values in the power spectrum is called the *bandwidth* of the function. Continuous signals with little variation typically have a low bandwidth (bandlimited signals have a finite bandwidth) while functions with discontinuities typically have infinite bandwidth.

There are two categories of applications in which sampling problems constantly arise: (1) acquisition and reconstruction of continuous signals in discrete time and (2) sampling according to a distribution for Monte Carlo integration. We consider each of these categories and describe how a Fourier analysis of the signal can be used to make predictions about the consequences of sampling rates on the fidelity of the reconstructed signal.

If a fixed, uniform rate is used for sampling a signal, it is likely that variations in the original signal that are too rapid to be observed with the given sampling rate will not be recsontructible. The question naturally arises:What sampling rate guarantees that the reconstructed signal will be identical to the input? Although the *Nyquist-Shannon sampling theorem* provides a partial answer by detailing a condition that is sufficient for this requirement, the necessary conditions are not always straightforward to arrive at. The Nyquist-Shannon theory states that bandlimited signals can be reconstructed perfectly if the sampling rate is greater than a certain frequency called the *Nyquist frequency*. Although frequencies, ν, above the Nyquist frequency, ν_N, are observable in the sampled signal, they are ambiguous since they cannot be distinguished from $\nu_N j - \nu$ and $\nu_N j + \nu$ for non-zero integers j; this ambiguity is called *aliasing*.

Theorem **4.3.** *If a function $f(t)$ contains no frequencies higher than ω cycles per unit, it is completely determined by giving its ordinates at a series of points spaced $1/(2\omega)$ units apart.*

4.1.5 The short-time Fourier transform

Definition **4.4.** *The short-time Fourier transform $\tilde{f}(\nu_1, \nu_2, \tau_{x1}, \tau_{x2})$ of a function $f(x_1, x_2)$ is defined as*

$$\tilde{f}(\nu_1, \nu_2, \tau_{x1}, \tau_{x2}) \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \beta(x_1 - \tau_{x2}, x_2 - \tau_{x2}) f(x_1, x_2)\, e^{-i(2\pi(\nu_1 x_1 + \nu_2 x_2))}\, \mathrm{d}x_1\, \mathrm{d}x_2 \tag{4.14}$$

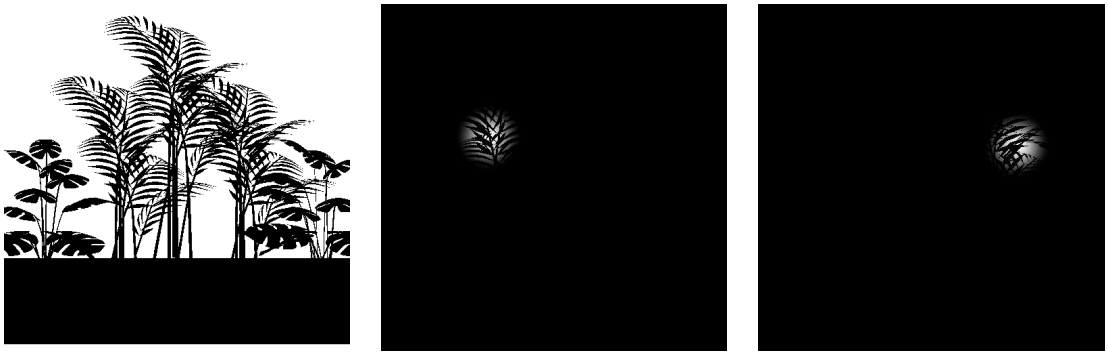


Figure 4.1: *Visibility function along a chosen direction* (left) *and two windowed visibility functions, with windows centered at different locations. The windowing function is a cosine to the fourth power.*

where $\beta(x_1, x_2)$ *is a windowing function centered around zero. Thus,*

$$(\mathcal{F} \circ f)\,(x_1, x_2) = \int\int \tilde{f}(u, v, \tau_x, \tau_y)\,\mathrm{d}\tau_{x1}\,\mathrm{d}\tau_{x2} \tag{4.15}$$

While the power spectrum of a function is useful in analysing the overall presence of high or low frequencies in the input signal, the Fourier transform itself provides no information about which parts of the signal contains the sharp variations. To do this, typically, the input signal is partitioned into intervals and the fourier transform of each interval is examined as a separate function. This "chopping up" of the input signal is equivalent to multiplying the original signal with several small pulses with widths and centres corresponding to the intervals. These pulses are examples of *windowing functions* and the resulting transform is referred to as the *windowed Fourier transform.*

According to the convolution theorem, the resulting spectrum for each partition will be a convolution of the spectrum of the original function with the spectrum of the windowing function. Since pulses have infinite bandwidth, typically, windowing functions

are chosen to be smooth. Regardless, the presence of the frequencies of the windowing function introduce "leakages" into the results of windowed Fourier transforms. With appropriately chosen windowing functions, these spurious frequencies can be negligible for reasonably wide widows. As the width of the windows are reduced, higher frequencies are introduced causing the resolution in frequency space to drop. Thus, in the limit, the resolution in Fourier frequency space is zero for infinitesimally small intervals in the original signal[3].

4.2 Frequency analysis of light transport

Radiance—a 5D function (three spatial and two directional)—undergoes several potentially complex physical interactions with matter before reaching the camera sensor. Intelligent sampling techniques play a vital role in efficient light transport for two reasons: without adaptive sampling techniques, discrete intermediate representations for the inherently analog quantities, like radiance, either tend to become leviathan structures (if densely sampled) or prone to high reconstruction errors (if sampled too sparsely); errors in Monte Carlo integrations could prove utterly frustrating without the use of carefully chosen importance functions.

Intelligent sampling, however, needs to be dependent on the characteristics of the signal. For this reason, frequency analysis of the radiance function has been considered an important problem in image synthesis. However, the high dimensionality, presence of arbitrary discontinuities, non-stationary nature of the phenomena involved (reflection, refraction, occlusion, etc.) and the substantially complex filtering operations that the radiance function undergoes in its interaction with matter make it a considerably challenging target for frequency analysis. Several works in the image synthesis

[3]This is in accordance with the uncertainty principle.

literature analyse the light transport problem from a signal processing perspective; however, they make restrictive assumptions so that the problem remains tractable.

While analysing radiance functions in the frequency domain, two distinctly different types of frequencies, that each explains a particular set of optical phenomena, can be observed: spatial frequencies that represent the variation in the radiance with position and angular frequencies that represent the variation of radiance at a point, as a function of angle. Intuitively, high spatial frequencies can be imagined to be "injected" into the transport process by sharply varying textures, intricate light occluders, small light sources, etc. High angular frequencies, on the other hand, correspond to effects like highly glossy or specular reflection (or refraction) and effects due to reflection off points with high local curvature. Almost all the literature on frequency analyses of radiance functions either studies strictly spatial or strictly angular frequencies. In this section we describe a theoretical framework [36] that performs a frequency analysis of light transport considering both spatial and angular frequencies. This work determines the effect of certain global light transport transformations on the local light spectra. Later, we build on this theory to propose means of propagating bandwidth information. Further, we extend the theory to account for depth of field effects due to finite sized apertures and suggest an efficient simulation algorithm using the predicted bandwidth.

4.2.1 Local lightfield parameterization

Shinya and Takahashi introduced paraxial approximation theory to the image synthesis community and suggested ways of using the theory for raytracing light pencils rather than rays. In their paper [91], they considered a local neighborhood of light paths where rays were parametrized with respect to the axial ray. Thus each ray was

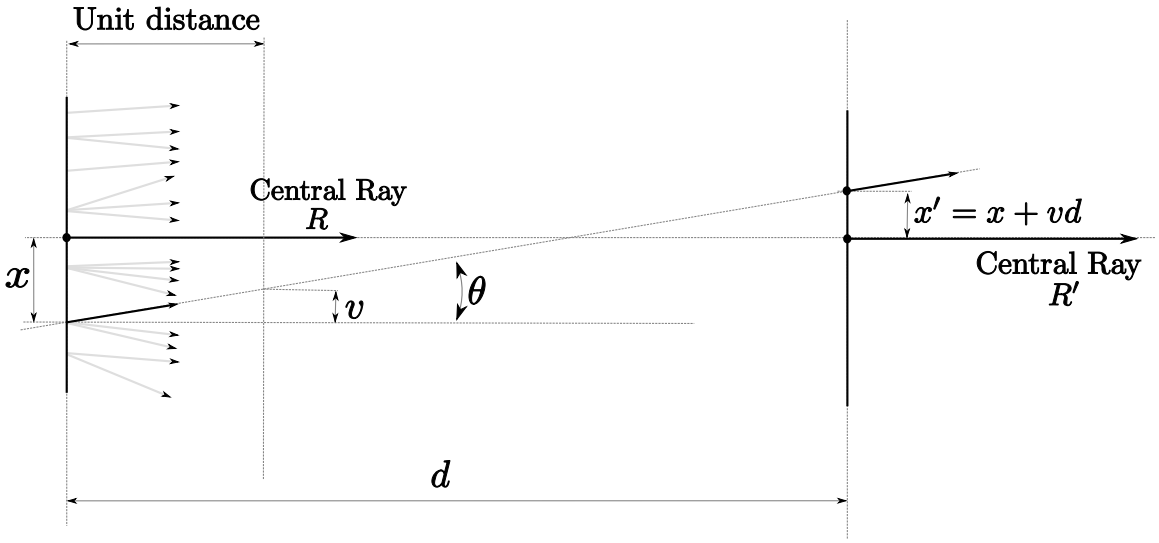


Figure 4.2: *The two parameterizations,* (x, v) *and* (x, θ)*, for rays in the local neighborhood of a central ray are equivalent under the paraxial approximation [91], when* $v = \tan\theta \approx \theta$*.*

represented using a 4D vector: two dimensions to define the vector from the axial ray to the ray under consideration, on the transverse plane and the remaining two dimensions to define the angular deviation from the axial ray. Using simple linear paraxial approximations, Shinya and Takashi constructed *system matrices*, using Snell's laws, that represented the ray vector changes due to basic transport processes.

Durand et al studied the radiance in the neighborhood of a ray as it was affected by transport processes. They studied a 4D slice of radiance at a virtual plane orthogonal to a central ray and called this slice the *local lightfield*. Depending on the transport phenomenon being studied, one of two parametrizations were used: the two plane parametrization [21] using the intersection at a parallel plane at unit distance and the plane-sphere parametrization [20]. The two parametriations are equivalent under the paraxial approximation (see Figure 4.2).

Following the example of Durand et al, we illustrate the radiance field and its transformations in flatland. We retain their notation for the radiance field in the neigh-

borhood of a central ray R, $\ell_R(x, v)$, and its Fourier transform, $\widehat{\ell}_R(\Omega_x,\ \Omega_v)$. Thus we have

$$\widehat{\ell}_R(\Omega_x,\ \Omega_v) = (\mathcal{F}\ \circ\ \ell_R)\ \ (x,\ v), \tag{4.16}$$

and the power spectrum of the lightfield is $\left|\widehat{\ell}_R(\Omega_x,\ \Omega_v)\right|$.

4.2.2 Transformations due to transport processes

In this section we describe the transformations undergone by the local lightfield undergoes transformation and the corresponding transformations to the spectra, through the different transport phenomena.

Emission

Consider the neighborhood of rays around a point light source emitting light equally in all directions. Along the spatial axis, the local lightfield is zero everywhere except at the location of the central ray. At the location of the central ray, since emission is constant with respect to direction, every point along the angular axis contains a constant non-zero value. Thus the lightfield is an impulse (Dirac delta) along the spatial dimension and constant along the angular dimension. Since the Fourier transform of a Dirac delta is a constant, the spectrum of the lightfield is a constant along the spatial dimension and a Dirac delta along the angular dimension. This suggests that the local lightfield contains no angular frequencies and infinite spatial frequencies. If the light source is not a point, then the geometry of the light source defines the lightfield along the spatial and angular dimensions. Distant illumination is constant in space which means that the spectrum is a Dirac delta in space. Thus,

unoccluded distant illumination contains purely angular frequencies.

Travel through free space

Radiance along a ray remains unchanged during travel through free space. However, the central ray undergoes a spatial reparameterization. Radiance at a point x after transport through a distance d is obtained as

$$\ell_R(x, v) = \ell_{R'}(x - vd, v), \tag{4.17}$$

suggesting that as light travels through free space, the local lightfield is sheared along the spatial direction. Intuitively, this means that directional variation at a source, for example, is transformed into angular variation at a receiver located at some distance from the source. Performing the appropriate change of variables to account for the reparameterization within the integral for the Fourier transform, the spectrum after the travel can be expressed in terms of the original spectrum as

$$\widehat{\ell}_{R'}(\Omega_x,\ \Omega_v) = \widehat{\ell}_R(\Omega_x,\ \Omega_v + d\Omega_x). \tag{4.18}$$

That is, travel through free space effects a shear in the lightfield spectra along the angular or directional dimension. When the distance traveled increases, spatial frequencies are pushed farther along the directional axis, thus resulting in high angular frequencies. We define a linear operator $\mathcal{S}_d$ that effects the transformation on the local lightfield spectrum, due to transport through free space:

$$\widehat{\ell}_{R'}(\Omega_x,\ \Omega_v) = (\mathcal{S}_d\ \circ\ \widehat{\ell}_R)\ \ (\Omega_x,\ \Omega_v)\ \equiv\ \widehat{\ell}_R(\Omega_x,\ \Omega_v + d\Omega_x). \tag{4.19}$$

Material surface interaction

Surface interaction basically involves two important steps apart from the various reparameterizations: accounting for curvature and the BRDF. Reflected radiance is obtained by convolving the differential irradiance with the BRDF and hence the operation on the spectra is a multiplication. That is, the relfected lightfield spectra are bandlimited versions of the incident spectra, where the bandwidth is determined by the bandwidth of the BRDF. Diffuse surfaces have zero bandwidth while specular surfaces have infinite bandwidth in the angular dimension.

Durand et al provide a detailed analysis of surface interaction phenomena. They derive the relationship between reflected radiance and incident radiance accounting for different cases. For flat, diffuse surfaces, the reflected radiance is the incident differential irradiance integrated over all directions and multiplied by the surface albedo. So the spectrum of the reflected radiance is simply the angular slice of the result of the incident spectrum convolved with the spectrum of the Jacobian of lightfield parameterization[4].

For curved surfaces with isotropic BRDFs, the shading process is decomposed into seven steps: (1) reparameterization of the lightfield in the local tangent frame; (2) accounting for curvature (results in a shear of the spectrum along the spatial direction); (3) computation of the differential irradiance; (4) reparameterization along specular direction since isotropic BRDFs mostly depend on the difference between pure specular reflection and the outgoing direction; (5) Convolution by the BRDF in the primal, which implies that the incident spectrum is bandlimited by the spectrum of the BRDF; (6) inverse of step 2; and (7) inverse of step 1.

In the case of anisotropic BRDFs, Durand et al factored the BRDF [36] into a Fresnel

[4]Since $\int f(x)\mathrm{d}x = (\mathcal{F} \circ f)(0)$

term (only dependent on incident angle) and a term that is dependent on the difference between the mirror and exiting angle. This affects only step 5 of the process for isotropic BRDFs. Thus, for anisotropic BRDFs, before being bandlimited by the spectrum of the BRDF, the incident spectrum is first convolved by a slice of the spectrum of the Fresnel term.

Modulation of reflected radiance by a texture function is a multiplication in the primal and hence a convolution in the Fourier domain. The texture modulated lightfield spectrum is obtained by convolving the reflected spectrum with the spectrum of the texture function. Since textures only contain spatial frequencies, the convolution is purely spatial.

Occlusion

When a radiance field $\hat{\ell}_R(\Omega_x,\ \Omega_v)$ encounters an obstacle with a binary visibility function $V(x, v)$, the resulting lightfield $\hat{\ell}_{R'}(\Omega_x,\ \Omega_v)$ is given as the product $\hat{\ell}_R(\Omega_x,\ \Omega_v)V(x, v)$. Consequently, the spectrum after occlusion is obtained as a convolution

$$\hat{\ell}_{R'}(\Omega_x,\ \Omega_v) = \hat{\ell}_R(\Omega_x,\ \Omega_v)\ *\ \hat{V}(\Omega_x,\ \Omega_v) \tag{4.20}$$

We define a linear operator $\mathcal{C}_V$ to represent the transformation on the local lightfield spectrum, due to occlusion:

$$\hat{\ell}_{R'}(\Omega_x,\ \Omega_v) = (\mathcal{C}_V\ \circ\ \hat{\ell}_R)\ (\Omega_x,\ \Omega_v)\ \equiv\ \hat{\ell}_R(\Omega_x,\ \Omega_v)\ *\ \hat{V}(\Omega_x,\ \Omega_v) \tag{4.21}$$

4.2.3 Case study: Analysing soft shadows

Consider the simple scenario where a planar occluder blocks light from a planar lambertian light source, causing a shadow on a planar lambertian receiver. Assume for simplicity that all three planes are parallel. Let R be a ray emanating at a point on the light source and R' be the ray before it impinges on the receiver after passing by the occluder. The spectrum of the lightfield in the neighborhood of R' can be expressed in terms of the spectrum at R by using the frequency domain transport operators for transport through free space and occlusion (see Equation (4.19) and Equation (4.21)):

$$\hat{\ell}_{R'}(\Omega_x,\ \Omega_v) = (\mathcal{S}_{d2}\ \mathcal{C}_V\ \mathcal{S}_{d1}\ \circ\ \hat{\ell}_R)\ \ (\Omega_x,\ \Omega_v), \tag{4.22}$$

where $V(x, v)$ is the binary visibility function at the plane of the occluder and d_1 and d_2 are the distances between the planes of the source and occluder and occluder and receiver respectively. The size of the penumbra region in the shadow depends on a number of factors: size of the source; distance between the source and occluder; size of the occluder; distance between occluder and receiver. We now analyse the effect of each of these factors, using the frequency analysis framework of Durand et al, and the frequency light transport operators in Equation (4.22).

Size of the source: The spectrum at R, $\hat{\ell}_R(\Omega_x,\ \Omega_v)$, has both angular and spatial content. The actual bandwidth of $\hat{\ell}_R(\Omega_x,\ \Omega_v)$ depends on the geometry of the light source. Intuitively, the smaller the light source, the higher will be the spatial bandwidth.

Distance between source and occluder: Recall that transport through free space ($\mathcal{S}_{d1}$ operator) shears the spectrum along the angular dimension, converting spatial frequencies into angular frequencies. The extent of the shear depends on the distance

travelled. If d_1 is large, then the transformation of spatial to angular frequencies is more dramatic. The angular bandwidth after transport to the plane of occlusion is proportional to the spatial bandwidth of $\widehat{\ell}_R(\Omega_x,\ \Omega_v)$ and the distance d_1. The spatial bandwidth, however, only depends on the geometry of the light source.

Visibility spectrum of occluder: Occlusion ($\mathcal{C}_V$ operator) amounts to a convolution of the transported lightfield at the plane of the occluder and the visibility spectrum (purely spatial frequencies) of the occluder. If the occluder contains small, intricate features, then its visibility spectrum has a high spatial bandwidth. A convolution with this visibility spectrum spreads the incident spectrum along the spatial direction, thus increasing spatial bandwidth. Note that the information about how close to the occluder the ray passes, is encoded in the phase of the convolved spectrum and not in its power spectrum. The angular bandwidth is unaffected by occlusion but the spatial bandwidth is now proportional to the size of the source and the visibility spectrum of the occluder.

Distance between occluder and receiver: This transport ($\mathcal{S}_{d1}$ operator) results in another shear in the angular dimension, this time proportional to the distance d_2. Convolution by the visibility spectrum of the occluder spread the spectrum in the spatial direction and, thus, the directional shear increases the angular bandwidth.

Finally, at R', the angular bandwidth is inversely proportional to the size of the light source and directly proportional to d_1 and a combination of the visibility spectrum of the occluder and d_2. The spatial bandwidth is inversely proportional to the size of the light source and directly proportional to the bandwidth of the visibility spectrum.

The importance of distinguishing between angular and spatial bandwidths can be realized when a series of transport phenomena need to be performed in succession where each phenomenon affects the spatial and angular bandwidths differently. For

example, reflection off the diffuse receiver kills angular frequencies in $\widehat{\ell}_{R'}(\Omega_x,\ \Omega_v)$, occlusion increases only spatial bandwidth, transport through free space increases angular bandwidth alone, etc.

4.3 Application: Depth of field

The pinhole camera model produces images that are completely sharp because every image element corresponds to a single ray in the scene. Real-life optical systems such as photographic lenses, however, must collect enough light to accommodate the sensitivity of the imaging system, and therefore combine light rays coming through a finite-sized aperture. Focusing mechanisms are needed to choose the distance of an "in-focus plane", which will be sharply reproduced on the sensor, while objects appear increasingly blurry as their distance to this plane increases. The visual effect of focusing can be dramatic and is used extensively in photography and film, for instance to separate a subject from the background.

Potmesil and Chakravarty proposed an algorithm [81] that generated a sharp image using the pinhole model and then, in a postprocess, applied an adaptive blur depending on the depth, at each pixel. The problem with this technique occurs when defocused objects are present in the foreground, since the visibility at each pixel is tested with only the ray through the optical center.

Cook et al. identified this problem, and proposed a solution [30] which integrates contributions over the aperture. They traced a ray through the center, to first find a point on the plane in focus. Then they averaged the contributions of several rays from points distributed over the aperture to the point on the plane in focus. Although the algorithm accounted for visibility correctly, it is extremely expensive since several

Figure 4.3: *Approximating the depth of field effect by applying a depth dependent blur on a sharp image does not account for occlusion correctly. Two different focus settings of the kitchen scene are shown (left and right columns).* Top row: *Results of applying a depth dependent blur where the blurring kernel is varied according to the circle of confusion at each pixel.* Bottom row: *Results of using a bilateral blurring filter where the depths at each pixel are taken into account, in addition to the distance from the central pixel of the kernel. (Thanks to Cyril Soler for providing the images.)*

rays need to be traced for each pixel.

Although the use of different camera models, in computer graphics, has been studied for more than two decades [14, 60], the simulation of depth of field effects is rarely used in practice because of its high cost: The lens aperture must be densely sampled to produce a high-quality image. This is particularly frustrating because the defocus produced by the lens is not increasing the visual complexity, but rather removing detail!

In the remainder of this section, we study the problem from a signal processing perspective, in the Fourier domain [99]. Then, we describe an algorithm that estimates local image bandwidth. This allows us to reduce computation costs in two ways, by adapting the sampling rates over the image and lens aperture domains. Finally, we analyse the algorithm and compare it with the naïve algorithm for simulating the depth of field effect.

4.3.1 Fourier depth of field

We present a theoretical analysis of the frequency content of the lightfield at the sensor plane of a camera with a finite sized aperture. For effective exposition, we present a flatland analysis where the lightfield is two dimensional: one spatial and one angular dimension; in 3D space the corresponding quantities and transforms are four dimensional.

Consider a point P in the scene (see Figure 4.5). We assume that we know the local lightfield at P[5], $\ell_P(x, v)$, and its spectrum, $\widehat{\ell}_P(\Omega_x,\ \Omega_v)$. We describe the transport of $\ell_P(x, v)$ to $\ell_Q(x, v)$ where Q is in the plane with the camera sensor and derive

[5]For brevity, we use "local lightfield at P" to mean "local lightfield in the neighbourhood of the ray at P in a certain direction"

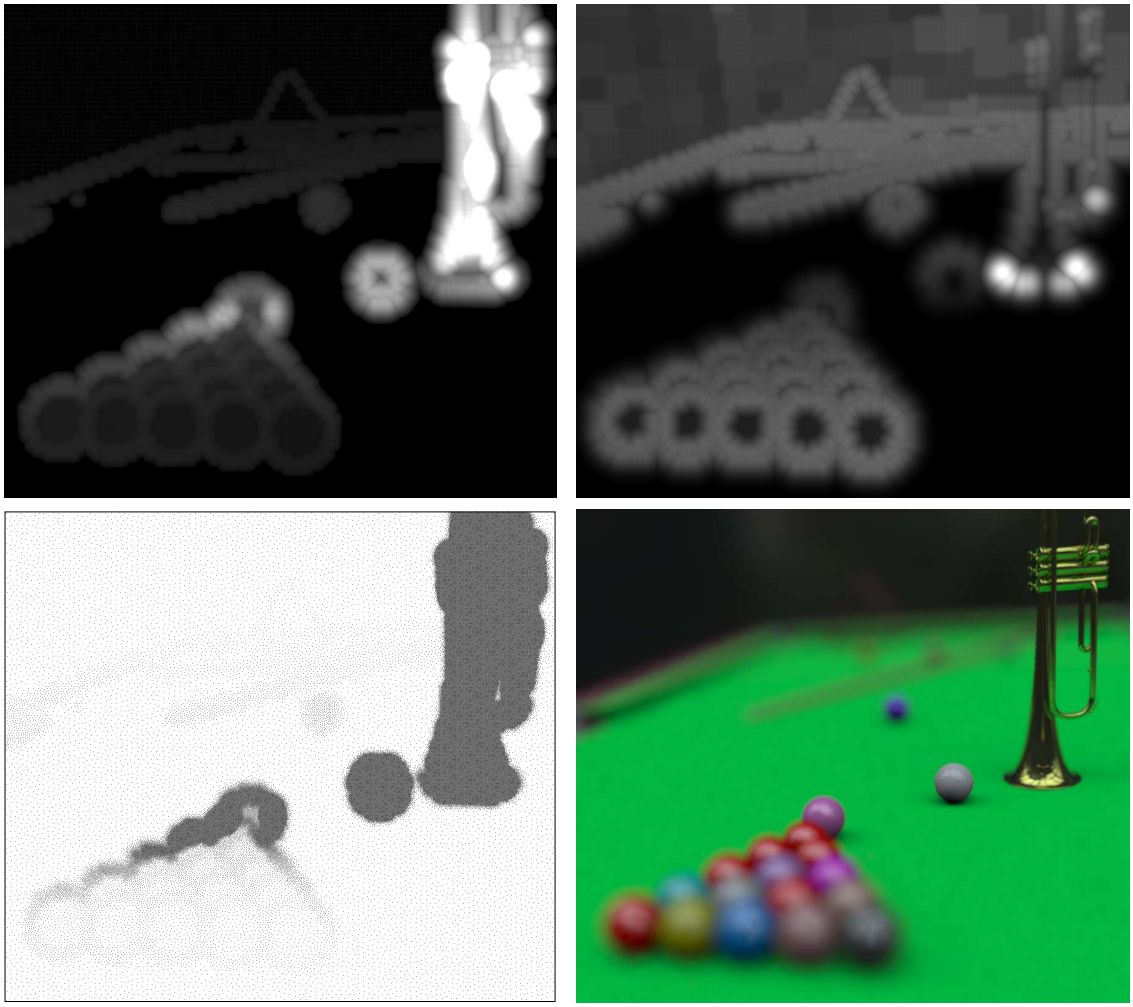


Figure 4.4: Top left: *The image sampling density predicts that the specular regions of the trumpet, with high curvature and in focus need to be sampled most profusely in the image.* Top right *The aperture density predicts that defocused regions need to be sampled densely while the ball in focus requires very few samples over the aperture.* Bottom left*: Adaptive bandwidth-based image samples.* Bottom right*: The final image, reconstructed from scattered radiance estimates at image sample locations, using Monte Carlo path tracing.*

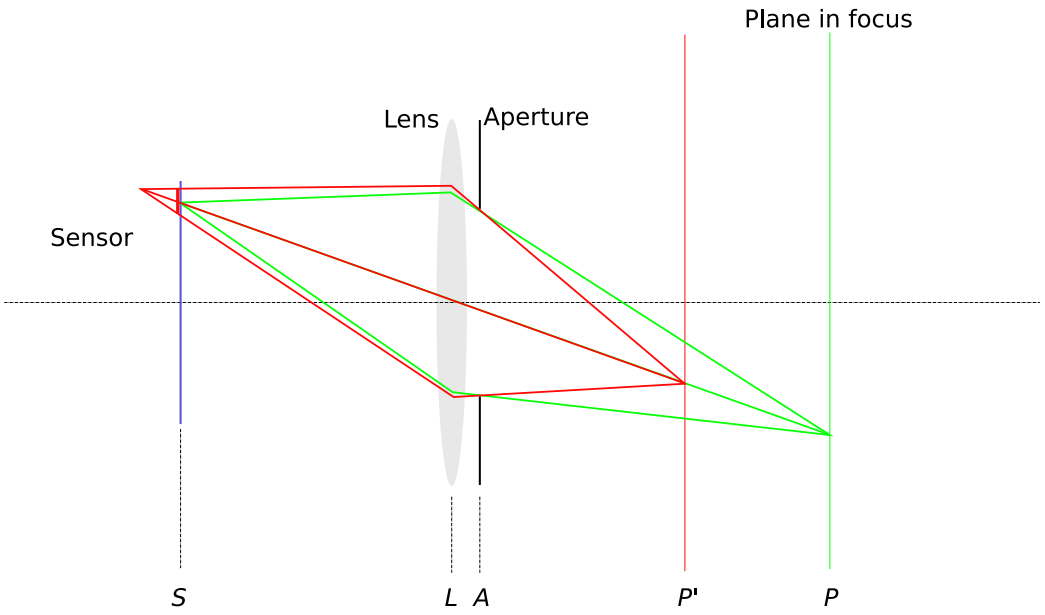


Figure 4.5: *Finite aperture (thin lens) camera model: Rays from points that lie in front of (resp. behind) the plane in focus converge behind (resp. in front of) the sensor plane, after passing through the lens, resulting in finite blurry regions on the sensor called circles of confusion.*

the transformations undergone by $\hat{\ell}_P(\Omega_x,\ \Omega_v)$ corresponding to this transport. The complete process is illustrated in Figure 4.6.

Transport from P to the lens:

To begin with, the light from P travels in free space towards S. From earlier work [36], we know that free-space traveling corresponds to a re-parameterization of the light-field, *i.e.* a shear in the angular domain of its Fourier spectrum. Recall that this transformation is expressed using the $\mathcal{S}$ operator (see Equation (4.19)):

$$\hat{\ell}_{P'}(\Omega_x,\ \Omega_v) = (\mathcal{S}\ \circ\ \hat{\ell}_P)\ \ (\Omega_x,\ \Omega_v). \tag{4.23}$$

If the light from P passes by an occluder en route to L, this occluder also affects

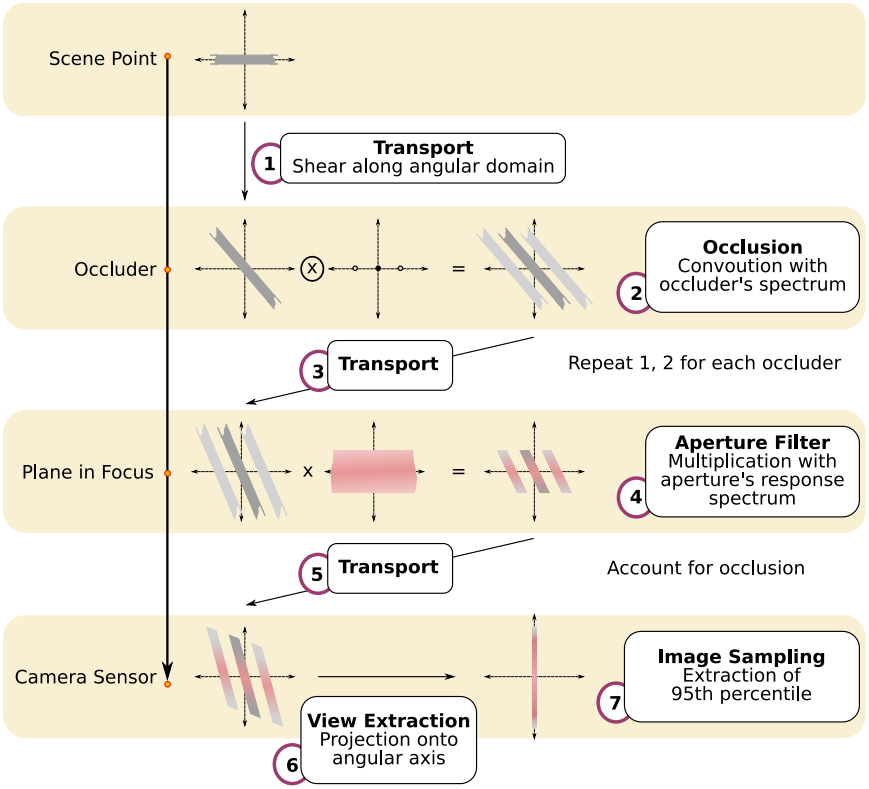


Figure 4.6: *Flatland illustration of the transformations at different locations undergone by power spectra of local lightfields after last bounce in the scene as they travel to the camera sensor.*

the lightfield. We express this by the operator $\mathcal{C}$(see Equation (4.21)), which is a convolution of the spectrum of the local lightfield with that of the occluder. If the occluder were planar, the effect of $\mathcal{C}$ would be to inject spatial frequencies at the plane of occlusion. For non planar occluders, this is a continuous process through the width of the occluder.

The spectrum of the local lightfield at the lens after passing by a single occluder is obtained by passing the spectrum at P through a simple composition of the above linear operators:

$$\widehat{\ell}_L(\Omega_x,\ \Omega_v) = (\mathcal{S}\,\mathcal{C}\,\mathcal{S}\ \circ\ \widehat{\ell}_P)\ \ (\Omega_x,\ \Omega_v) \tag{4.24}$$

In the general case, light travelling from P to L will encounter m different occluders, and $m+1$ shears (with different values for the shear parameter d). In this case we can write $\widehat{\ell}_L(\Omega_x,\ \Omega_v)$ as

$$\widehat{\ell}_L(\Omega_x,\ \Omega_v) = (\mathcal{S}\ \ (\mathcal{C}\,\mathcal{S})^m\ \circ\ \widehat{\ell}_P)\ \ (\Omega_x,\ \Omega_v) \tag{4.25}$$

Lens integration

The result of a finite-sized aperture is that, at each location Q on the sensor, there is an integration of the cone of incident rays from the lens to the scene, defined by the aperture a. We choose to model this integration as an operation over the lightfield at the lens. This integration corresponds to a convolution in ray-space at L, and thus the lightfield just after L is actually

$$\ell_{L_+}(x,v)\ =\ \ell_{L_-}(x,v)\quad *\ a\,(x,v). \tag{4.26}$$

In this equation L_+ (resp. L_-) represent the lightfield after (resp. before) the lens. The equivalent transform in Fourier space is a product and can be written as

$$\hat{\ell}_{L_+}(\Omega_x,\ \Omega_v)\ =\ \hat{\ell}_{L_-}(\Omega_x,\ \Omega_v)\ \ \hat{a}\,(\Omega_x,\ \Omega_v). \tag{4.27}$$

To understand what $\hat{a}\,(\Omega_x,\ \Omega_v)$ looks like, observe that the set of rays over which the lightfield is integrated, converge at a point P_f in the plane in focus (see Figure 4.5). Therefore, at this point, the integration filter is a box in angles and a Dirac in space. Its Fourier transform is thus a sinc in angle and a constant in space. At L, $a\,(x, v)$ is the same function sheared from the distance between P and L. In 3D, the box is circular, and its Fourier transform is consequently a Bessel function in angles.

As a consequence, the lightfield at L_+ (*i.e.* just after the lens) is *bandlimited* by the spectrum of the aperture response function. Constricting the aperture of a camera spreads the width of $\hat{a}\,(\Omega_x,\ \Omega_v)$ resulting in increased angular bandwidth at L_+.

Finally, because we have already accounted for the integration at the lens, and because the free-space traveling from the lens to the sensor is usually very small, we will neglect this very last phase of the transport to Q.

Consequences on lens integration and image-space frequencies

When numerically performing the lightfield integration at the lens, it is preferable to adapt the integration accuracy to the frequency content of the lightfield at L_- so as to ensure a desirable precision while keeping the computation cost as low as possible. This information is available in $\hat{\ell}_{L_-}(\Omega_x,\ \Omega_v)$ and will be used in the algorithm to drive the lens sampling.

When computing an image, it is preferable to adapt the image sampling to the fre-

quency content of the image and interpolate between samples, rather than explicitly compute all pixels. At the sensor, the result of the integrated lightfield is the radiance at point Q, corresponding to a pixel into the image. Seen from the lens, image frequencies correspond to angular frequencies of the lightfield at L_+ at the center of the lens (see Figure 4.5). In Fourier space, this means that we need to rate angular frequencies in $\hat{\ell}_{L_+}(\Omega_x,\ \Omega_v)$ integrated over the spatial domain.

Since view extraction is a projection onto the angular axis, a wider $\hat{a}\,(\Omega_x,\ \Omega_v)$ results in higher frequencies in the image. Intuitively, reducing the aperture size causes more regions of the image to be "in focus". In the limit we obtain a pinhole camera which retains all frequencies.

4.3.2 Adaptive depth of field rendering

We increase the efficiency with which depth of field effects can be simulated by adaptively varying the image space samples and the number of samples over the aperture at each image sample. The former are obtained according to conservatively predicted bandwidths over the camera sensor and, at each of these samples, the latter are obtained by estimating the variance of the integrand over the aperture. The computation of both, the bandwidth and the estimate of the variance, are enabled by the propagation of local light field spectra after last bounce off surfaces in the scene towards the camera sensor.

To adaptively distribute effort between sampling the image and aperture, we consider the different transport phenomena between a visible object and the camera sensor. We propagate the spectral information of local lightfields after last bounce off visible objects. To do this, we sample the power spectrum of the lightfield and adjust these samples during the different stages of transport to reflect the power spectrum density

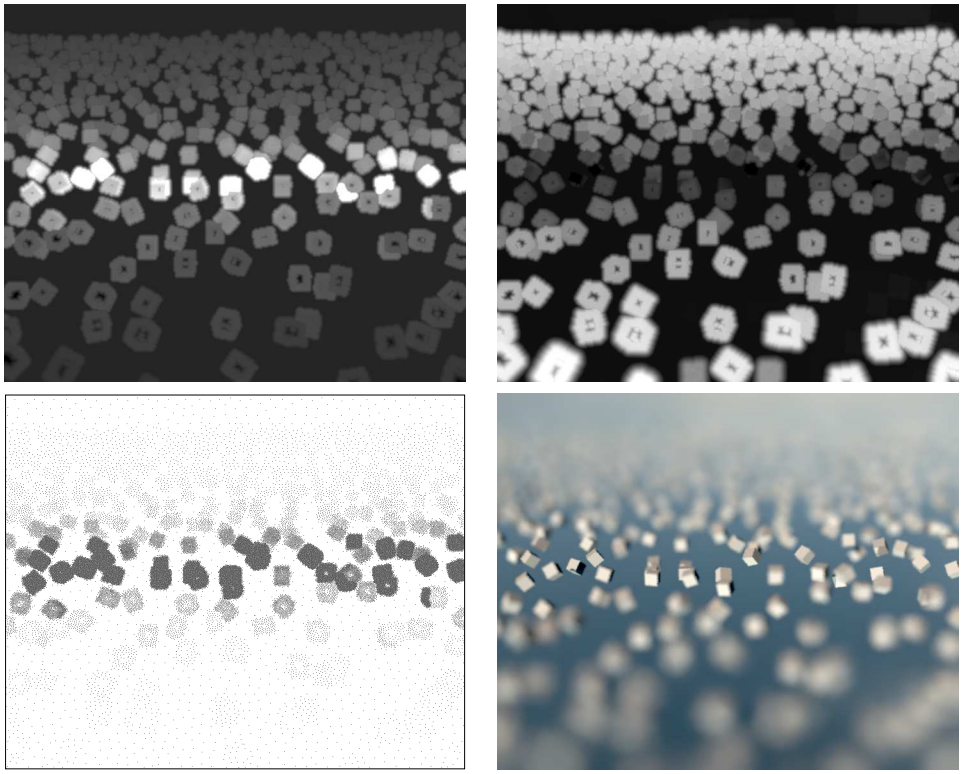


Figure 4.7: Top Left*: Image density depicting local bandwidth at each pixel.* Top Right*: Lens density indicating expected variance in the aperture integral.* Bottom Left*: image samples at which incoming radiance is estimated;* Bottom Right*: reconstructed image, using adaptive gaussian splatting. Blurry regions of the image are sampled sparsely, but require profuse sampling of the lens.*

locally. Using a depth map to detect occlusion along the transport, we are able to efficiently estimate frequency propagation towards the camera sensor.

Using the frequency information of the lightfields at the sensor, we extract a slice to obtain an image space density that predicts bandwidth locally over the camera sensor. This operation is performed for a subset of image pixels on a regular grid, namely one every ten to one hundred pixels, and the frequency information is splatted using a max across the image. This makes the whole process very fast. Slices of the spectra at the plane of focus are used to estimate the variance of the integrand over the aperture. We use the density yielded by this slice to derive the number of lens samples for each pixel.

The next stage of our algorithm samples the image density and estimates the number of lens samples required at each of those sample locations. Given this information, we estimate incident radiance at those locations on the camera sensor using a Monte Carlo path tracer. The final image is reconstructed from the scattered radiance estimates. In contrast to images created by pinhole cameras, blurry images produced by cameras with finite sized apertures pose a greater challenge for reconstruction from sparse samples. This is because conventional "hacks" that splat upto material boundaries are not usable when the boundaries are blurry. Occlusion effects caused by drastic visibility changes over the aperture of the lens pose another frustrating hurdle in simulating depth of field. Many approximations that blur an input image using depth maps to decide the kernel size fail to handle the effect of occlusion correctly. Contrarily, our algorithm takes into account the effect of occlusion correctly since we add the effect of the aperture in a separate step.

To arrive at a simple algorithm, we conservatively assume that local lightfields at surfaces after last bounces to the camera have infinite bandwidths in space although bandlimited in angle by the spectra of the corresponding reflectance functions. We

propagate these local lightfield spectra to the camera sensor by:

1. transporting to the plane in focus accounting for nearby occluders en route.
2. performing the lens convolution at the plane in focus. In the Fourier domain this amounts to a multiplication of the local lightfield spectra with the spectrum of the aperture response function.
3. transporting to the camera sensor accounting for nearby occluders en route.

In practice, we cast primary rays by through uniform samples on the the camera sensor and perform the above steps from the points of first intersections back to the sample locations.

Sampling local lightfield spectra

Let Q be a point on the sensor from where a primary ray $\mathbf{r}$ is cast and let P be the point of intersection of this primary ray with the scene. We represent the power spectrum of the local lightfield at P, $\left|\hat{\ell}_p(\Omega_x,\ \Omega_v)\right|$, by a set of random variables

$$\{(\omega^s{}_i,\ \omega^a{}_i)\} \sim \mathcal{P}\left(\ \left|\hat{\ell}_p(\Omega_x,\ \Omega_v)\right|\ \right),\ 0 < i < n_s. \tag{4.28}$$

$|\omega_i^s| < \infty$ and $|\omega_i^a| < \Omega_p$ are independent random variables representing the spatial and angular components of a 2D frequency sample. Ω_p is half the angular bandwidth of the reflectance function at $\mathbf{P}$. $\mathcal{P}$ is a projection of the four dimensional power spectrum down to two dimensions, one in each, namely space and angle. The projection down to two dimensions implies that we assume isotropy independently in space and in angle which makes the computation, representation and propagation of the spectra

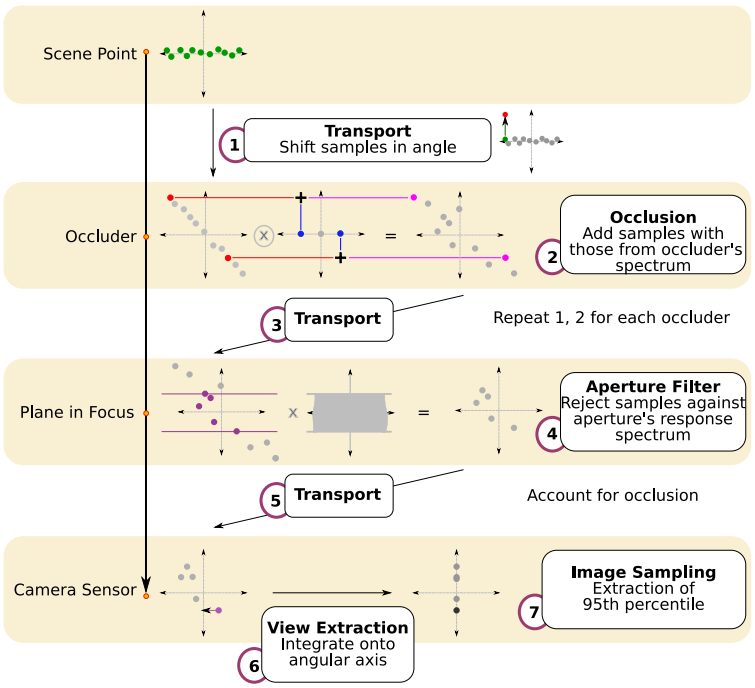


Figure 4.8: *Sampled power spectra are propagated from the scene to the camera sensor. Transformations to the spectra are performed by independently modifying each sample.*

practicable. In practice this assumption is reasonable since we are only interested in maximum frequencies and not in accurate estimates of the spectra themselves.

Local lightfields in the scene can of course be arbitrarily complex, as can their corresponding 4D spectra. The existance of discontinuities in the lightfield implies that the range of frequencies is infinite. Although, after reflection they are restricted in the angular domain by the bandwidth of the reflectance function, they could contain arbitrarily high spatial frequencies. This results in a very conservative prediction of bandwidth at Q and thus we generate more samples than the optimal number.

Since we are only interested in predicting bandwidth, hence maximum frequency, we project the 4D lightfield down to 2D (space and angle). Note that this would not be a reasonable assumption if we were estimating spectra. Since we are not interested in details such as spatial or angular anisotropy in the lightfields, we focus on 2D spectra

with one dimension for angle and one for space. In the scene, the local spectra are bounded in angle, by the bandwidth ρ of the BRDF. In space, the local lightfield spectrum is unbounded *a priori*, because we have no information about shadow or other boundaries and what spatial frequency conetent they inject.

Associated with each primary ray is a set of samples– ray $\mathbf{r}$ is initialized with $\{({\omega^s}_i,\ {\omega^a}_i)\}$ from the power spectrum at P, as above. The range of useful frequencies in the image plane is always bounded by the maximum number of samples N_s per square pixel in image space, and by the maximum number of lens samples N_l, in angle, which are user defined parameters. Also, in practice, anticipating the shear from the point to the sensor, we can restrict the spatial bounds to be such that the resulting frequencies stay below the maximum angular frequency at the sensor.

Propagation of the frequency content along the ray until Q requires that the samples be appropriately updated at each step in the transport from P to Q. These updates are simple and inexpensive to compute (see Figure 4.8).

Propagating local lightfield spectra

Transport through free space shears the power spectrum along the angular direction proportional to the distance transported. Starting from the original samples, obtaining samples that are distributed according to the sheared distribution involves simply shifting each of the samples in the angular dimension. That is, each sample $({\omega^s}_i,\ {\omega^a}_i)$ is updated to be $(\omega_i^s,\ \omega_i^a + d\omega_i^s)$ as a result of the free space transport by a distance d.

Occlusion involves a convolution of the spectrum with the local lightfield by the spectrum of the occluder. Random variables representing the spectra of the lightfield and the occluder when added are representatives of the convolution of the two distribu-

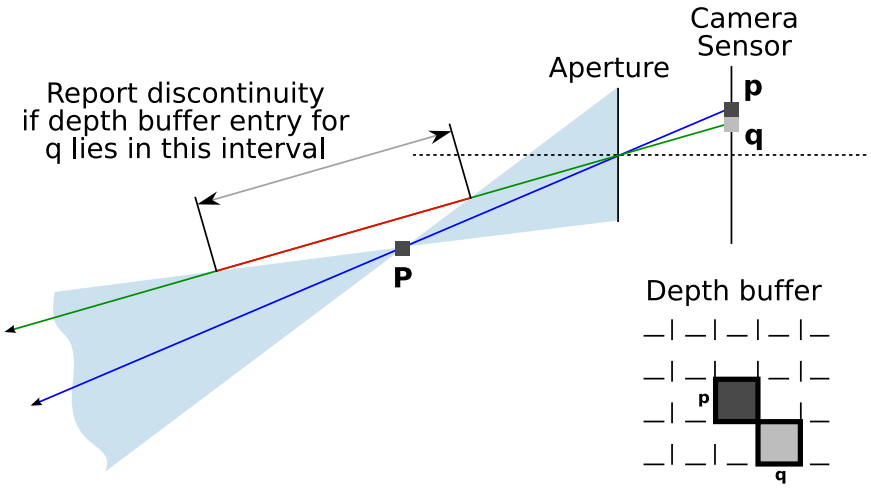


Figure 4.9: *A depth map of the scene is used to build the lists of occluders, along with their distances, for each primary ray. P is the point of intersection of the primary ray through pixel p and the scene. This defines the double cone where a ray from the lens can hit the point P. The above figure illustrates the interval of depth values for a neighboring pixel q within which a discontinuity is reported.*

tions. Thus if we are able to draw samples $\{({\nu^s}_i,\ {\nu^a}_i)\},\ 0 < i < n_s$ from an occluder's spectrum then we can simply update our samples $({\omega^s}_i,\ {\omega^a}_i)$ to be $(\omega_i^s + \nu_i^s,\ \omega_i^a + \nu_i^a)$.

Occlusion is a transport process that injects high spatial frequencies at multiple points along the transport towards the sensor. These spatial frequencies are spread along the angular dimension due to the transport through free space. Given these points and some approximate occluder spectra at each of those points, we can use the algorithm shown in Figure 4.8 to update the frequency samples appropriately

For each ray $\mathbf{r}$ we use the depth map to build a list of occluders and the points along the ray the occlusions occur. To achieve this, we search the depth map for discontinuities and splat these discontinuities in an occlusion buffer. Each discontinuity is splatted to influence a region as large as its circle of confusion. Given a pixel p and a pixel q in its neighborhood, the test to determine if q corresponds to a discontinuity where occlusion needs to be accounted for is illustrated in Figure 4.9.

At each occlusion point, the power spectrum of the occluder is assumed to be a Dirac in angle and proportional to $1/\omega_x$ in space. This conservative choice is due to the fact that visibility functions contain zero-order discontinuities and thus produce a spectrum with first-order fall-off. The effect of this is seen in the regions surrounding the foreground cubes in Figure 4.11 where the predicted effect of occlusion is more conservative than its measured counterpart. While the method works well for conservative estimates, better representations might be required if specifics of the occluder are of interest (see Section 4.4).

The effect of a finite aperture is to cut off high angular frequencies at the plane in focus. Updating samples to represent the result of applying this operator involves rejecting angular frequencies with a probability defined by the shape of the aperture power spectrum. Although this will increase the variance of the estimate, it is reasonable since we are interested in information about maximum frequencies and not complete spectra.

Bandwidth, variance and reconstruction

Sampling the image: To obtain image space samples, the first step is to conservatively estimate bandwidth over the camera sensor using the incoming local lightfield spectral information. That is, we project the samples onto the angular axis (view extraction) and compute the highest angular frequency in the local neighborhood of each pixel. In practice, to decrease sensitivity to outliers, we use the 98^{th} percentile of energy ξ_s as a representative of the maximum value at each point $s \in [0, W) \times [0, H)$. Here W and H are the width and height of the image respectively. The distribution of ξ_s over the image serves as an indicator of regions that need to be sampled more densely. Further, since ξ_s represents the maximum local frequency, we can estimate

the optimal number of samples required locally (samples per square pixel) at s as

$$\rho(s) \;=\; \frac{4\,{\xi_s}^2\,f_h\,f_v}{W\,H}, \tag{4.29}$$

where f_h and f_v are the horizontal and vertical fields of view. However, since we predict bandwidth conservatively for increased reconstruction quality, the number of samples over the image may be suboptimal. After computing the density, image samples are generated according to $\rho(s)$ using a technique that produces samples with desireable noise properties [77]. The total number of samples is dependant on the integral of $\rho(s)$ over the image rather than a user defined parameter.

Sampling the aperture: Using Monte Carlo integration over a finite aperture, the variance of the estimates depend on the variance of the integrand[6]. The goal is to sample the aperture more profusely at image locations where the variance of the lens integrand is high. We use the lightfield spectra at the plane of focus to estimate the angular variance of the lightfield, since according to Parseval's theorem, the variance of a function is the integral of its power spectrum minus the DC term:

$$\sigma^2 = \int y_p(\Omega_v)^2 - y_p(0)^2$$

In this equation, y_p is the predicted spectrum at the plane in focus, projected onto the angular axis. The central limit theorem predicts that the Monte Carlo estimates of each of these integrals using uniform sampling over the aperture has itself a variance of $O(n_s^{-1})$. While, in theory, stratification can improve the variance up to $O(n_s^{-2})$, Mitchell showed [72] that in practice it is about $O(n_s^{-1.5})$ for pixels with edge boundaries. Using this conservative estimate for stratified sampling of the aperture, we

[6]The variance of a function describes the rate at which the value of the function changes while the variance of an estimator describes the error in the estimates.

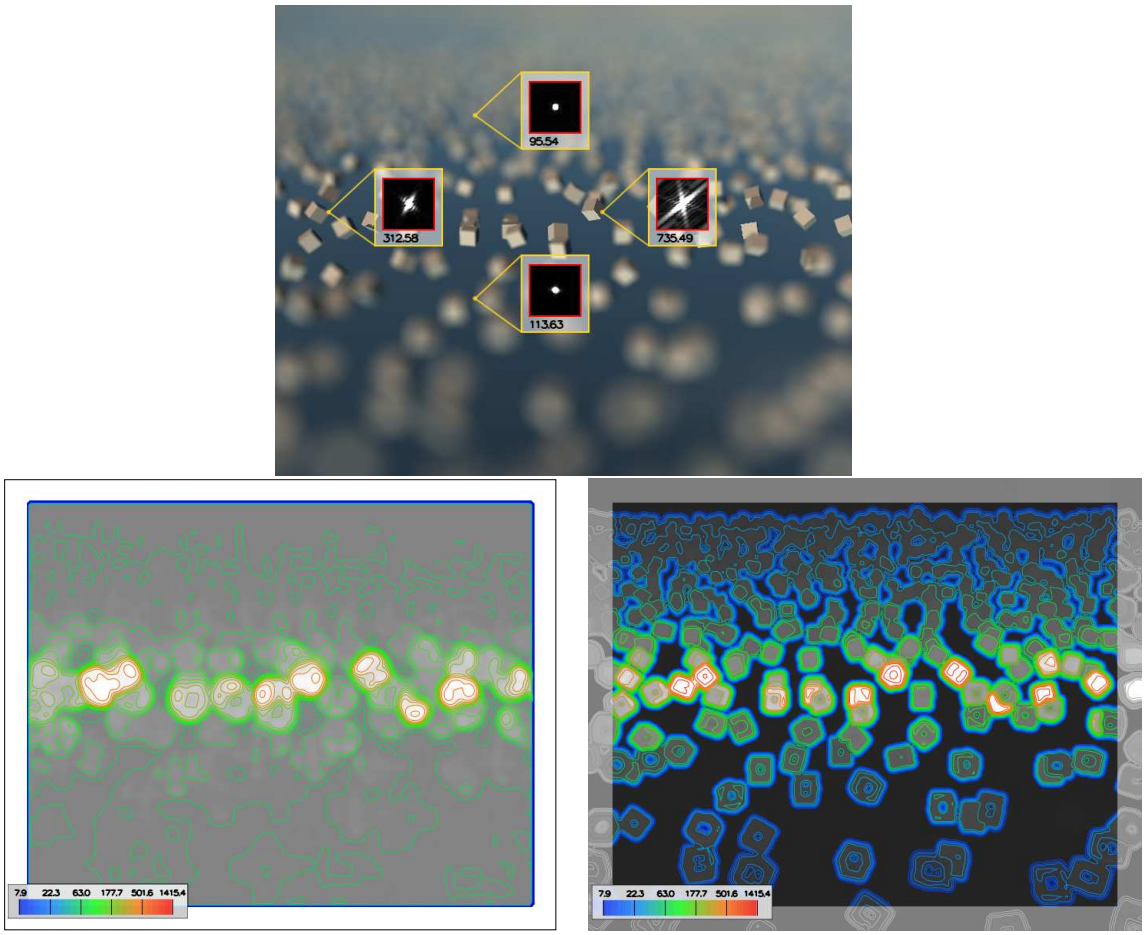


Figure 4.10: *Comparison between measured and predicted image-space frequencies.* top: *image space frequencies are measured in the reference image by extracting the maximum* 98 *percentile (radially) in a 2D spatial spectrum computed using a* 64×64 *windowed Fourier transform around the point. Inlays show the spectra and image-space frequencies in pixel*$^{-1}$ *at four points.* bottom left: *measured values across the image.* bottom right: *using sampled spectra for conservative estimates. Conservative estimates using sampling not only gives qualitatively the same profile of frequencies but also produces a conservative estimate of the actual values. Note that in the domain of low frequencies, the measured frequencies become higher than our estimate since the measurement method can not produce very low frequencies because of the* 64×64 *window resolution. In addition, the windowed fourier transform has an averaging effect whereas we estimate a purely local frequency, hence the difference in blurriness of two approaches.*

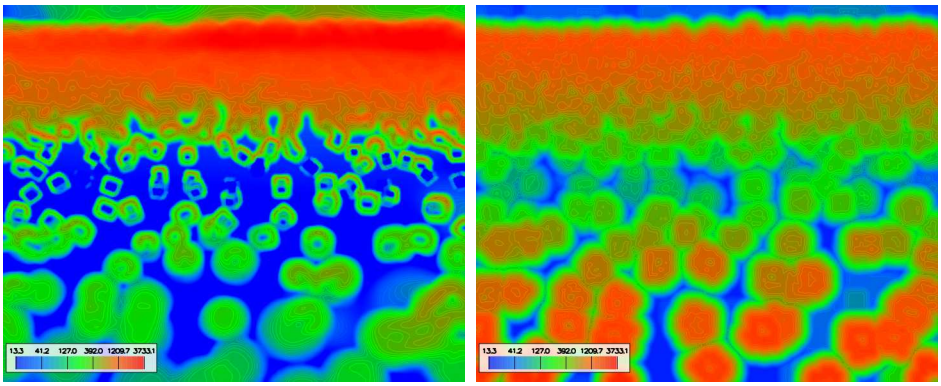


Figure 4.11: *Comparison of variance measured over the rays converging to each pixel of the* cubes *scene (*left*), with the variance predicted by our method (*right*). Both images are displayed using the same scale. Our prediction is comparable to the actual measured values both in its distribution over the image, but also qualitatively, except in the foreground where it is a more conservative estimate. This makes it usable for adaptive lens sampling.*

determine the number of samples as

$$n_s \;=\; k\,\left(\sigma^2\right)^{\frac{1}{1.5}} \tag{4.30}$$

The constant of proportionality, k can be used to control the expected error consistently over the entire image.

Image reconstruction: We obtain image samples that respect this The image space density obtained after frequency propagation directly provides an estimate of the number of samples per square pixels required over the image. We reconstruct the image using the radiance estimates at each of the image sample locations. The color at each pixel is computed as a weighted average of a constant number of neighboring samples. Since the samples are distributed according to a density, choosing a constant number of neighboring samples involves adaptively varying the radius of contribution of each pixel so that a constant number of samples (independent of the local density) contribute to the color at each pixel. In practice, we use a gaussian weighting term with a variance that is proportional to the square root of the local density.

For each pixel

$$p(x) = \frac{1}{\sum_i g_i(x)} \sum_i g_i(x) p(x_i) \quad \text{with} \quad g_i(x) = e^{-\frac{|x-x_i|^2}{\sigma_i^2}} \tag{4.31}$$

In practice, we first splat gaussians and gaussian weights for all image samples, and divide the result by the total weight at each pixel. The splatting radii r_i as well as the constants σ_i for each sample are computed such that the number of samples contributing to neighboring pixels is constant n throughout the image. This means choosing:

$$r_i = \frac{W\sqrt{n}}{\pi f_h \xi_s} \quad \text{and} \quad \sigma_i = \frac{r_i}{\epsilon} \tag{4.32}$$

In this expression, f_h is the horizontal image field of view, W is the width of the image, and ϵ is the minimum weight splatted for each sample. In practice we take $n = 6$ and $\epsilon = 0.1$.

We emphasize that sparsely sampled images resulting from simulation of depth of field cannot be splatted upto material or depth discontinuities (as is done for pinhole camera simulation), due to the integral over the aperture. Blurred discontinuities in the image need to be sampled adequately, which requires a systematic treatment of occlusion and aperture effects.

4.3.3 Validation and results

We compare our conservative predictions of the local image bandwidth and lens variance against experimental measurements. To verify our predictions of the image density, at each pixel s_i (in the reference image) we compute a windowed Fast Fourier Transform (FFT) with the window centered at s_i and record the 98^{th} percentile. Figure 4.10 shows a comparison of such a measured 98^{th} percentile image against our image space sampling density. The measurement is not entirely local due to a fundamental property of the windowed FFT. Depending on the choice of window size the measured frequencies are either heavily blurred (large window) or restricted heavily in the range of measured frequencies (small window). To avoid border effects, the measurements are limited to the interior part of the reference image. From the figure, it is evident that our prediction both appears to qualitatively match the distribution of measured frequency and is of the same order of magnitude. In fact, we obtain a much more local prediction than observed with the windowed FFT.

To verify our estimates of the variation of the integrand over the aperture, we use stratified samples to estimate and record the variance in the lens integrals at each pixel. In Figure 4.11 we compare the predicted variance at each pixel using Equation (4.30) to the actual variance measured during Monte Carlo integration over the aperture for the reference image. From the comparison we observe that, although our predicted distribution resembles the measured variance, we predict higher frequencies around the blurry cubes in the foreground since our prediction is conservative.

Computation times

The table in Figure 4.12 summaries computation cost for the various scenes and focus settings with our algorithm. Kitchen 1 and 2 correspond to the kitchen scene with

the plane in focus set on the foreground and background respectively. Clearly, the accumulated cost of propogating, computing and splatting frequency information, along with image reconstruction (using splatting) is quite negligible compared to the cost of naïve stratified Monte Carlo integration over the aperture at all pixels (see table in Fig. 4.13. This suggests that our adaptive algorithm significantly (at least by an order of magnitude) increases the efficiency of synthesizing images with depth of field effects. The shallower the depth of field, the blurrier the image; this is when the adaptive algorithm provides maximum gain.

Scene	**Size**	**Frequency computn.** (seconds)	**Path tracing** (seconds)	**Reconstr.** (seconds)	**Image space samples**	**Primary rays**
Cubes	721 × 589	45	3150	3	76 000	13 M
Snooker	904 × 806	90	4 500	10	119 335	25M
Kitchen 1	897 × 679	60	7401	8	867 000	113 M
Kitchen 2	897 × 679	60	6849	3	2 000 000	144 M

Figure 4.12: *Execution times for the different steps in our algorithm and number of primary rays cast are shown for different scenes.*

The number of image samples is indicative of the number of pixels where radiance needs to be estimated. For images with larger regions in focus (large depth of field), this number would be very close to the number of pixels in the image. In those regions, the gain from using our algorithm is due to the extremely sparse lens sampling, again implying that fewer radiance estimates are required. Note that focused images are reconstructed faster, which is consistent since samples require smaller splatting radii.

We use the total number of primary rays cast to compare our technique with the non-adaptive stratified sampling technique. By distributing the total number of primary rays cast in our method amongst all pixels for the stratified sampling method, we generate images of similar computational cost. The table shown (see Fig. 4.13) shows the number of rays cast for similar image quality as those images used for measurements in Fig. 4.12. We also tabulate the theoretical speedup by dividing the

number of primary rays in the reference technique by the number of primary rays cast by our algorithm.

Scene	Number of lens rays/pixel	Number of . primary rays	Speedup due to our method
Cubes	450	191M	14.7
Snooker	600	437M	17.3
Kitchen 1	1100	2 719M	24.0
Kitchen 2	1100	2 719M	18.9

Figure 4.13: *Number of rays cast using stratified sampling Monte Carlo integration for similar appearance quality as for the images tabulated in Fig.4.12. The last column shows the speedup gained by using our method, obtained by dividing the middle column by the last column in Fig.4.12.*

Examples

We present, in Figure 4.14, example renderings with direct illumination of a scene lit by area and point light sources. The frequency maps conservatively capture the various effects which can produce high image-space and lens frequencies such as focused regions for the former, and highly curved specular regions for the later. The image samples as well as the lens samples are automatically adapted so as to produce an image of constant quality. The image resolution is 897×679, and we used maximum values of $N_s = 4$ image samples per square pixel and $N_l = 2500$ lens samples per pixel. The total number of primary rays is 44, 000, 000 and 77, 000, 000 in the two settings respectively.

We compare our results to what we can obtain using a stratified lens sampling (with image space stratification for antialiasing) for the same computation cost. We do this by setting the number of lens samples so that the total number of primary rays is the same as with our method samples (70 and 129 for the foreground and background focus settings respectively). In both cases our algorithm results in images that are less noisy. Our algorithm performs particularly well in regions of high angular variance

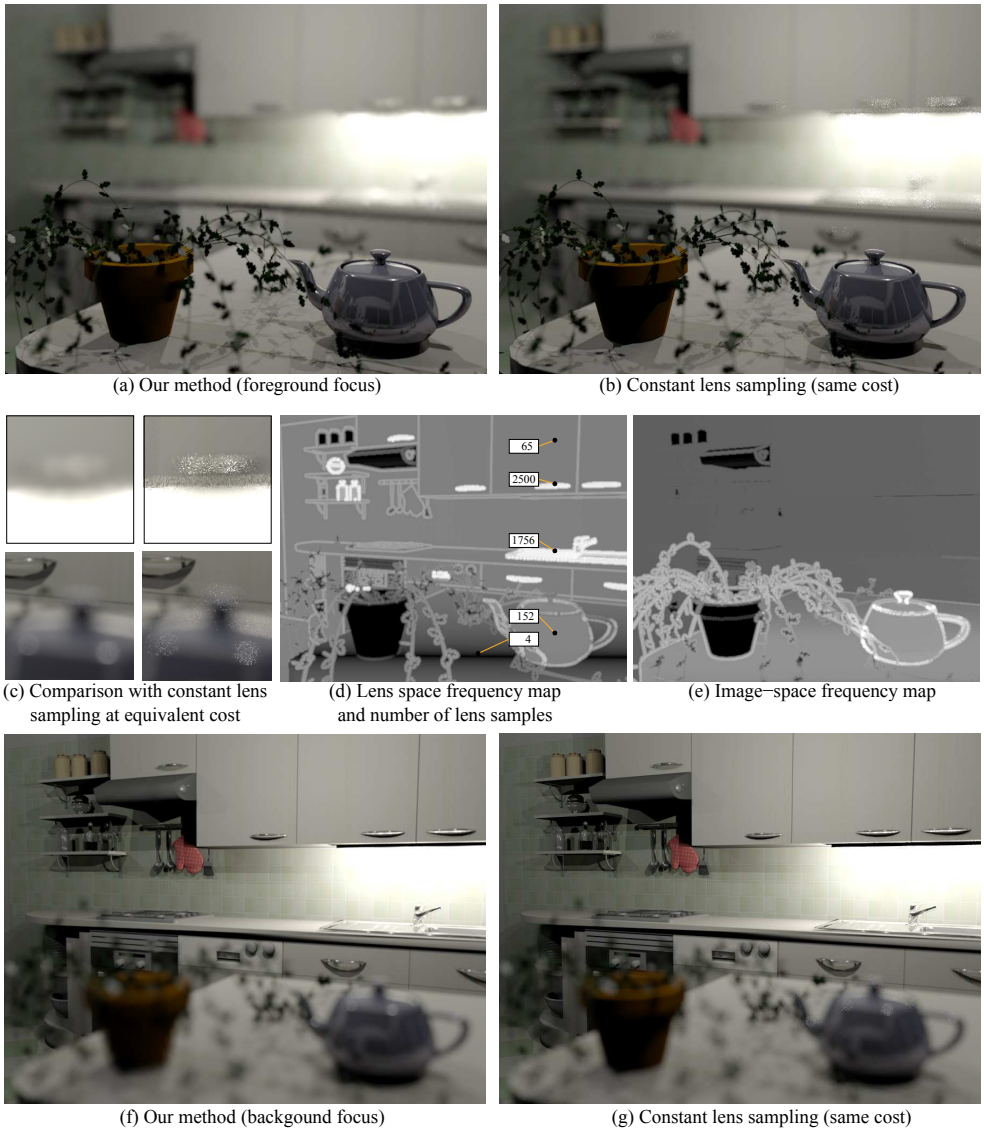
such as the handles of the cabinet. Despite the total cost being the same, the reason that the naïve method does not produce regions with less noise, is that many regions of the image are wastefully oversampled because of its non-adaptive nature.

Discussion of the various approximations

Ignoring phase information of the local lightfield spectra, as we do in our model, implies approximations in the computation of convolutions between spectra. In practice, this means that we neglect the relative positions of multiple obstacles close to the same ray. The convolution is then over-estimated, and tends to produce higher frequencies when multiple obstacles lie between the eye and the scene. This approximation is therefore conservative with respect to image-space frequency and lens variance.

By reducing dimensionality from 4D spectra to 2D spectra, we implicitly make assumptions about the isotropy in the spatial and angular domains independently. In practice, since we only use the spectra to conservatively predict bandwidth, we do not observe artifacts that could be due to this projection.

Our choice of using conservative spectra such as maximum spatial frequencies when a textured surface is detected and angular frequencies equal to the bandwidth of the BRDF on all surfaces results in suboptimal sampling. Thus we are not able to take special advantage of knowing the local bandwidth of a region with texture. In addition we do not take illumination into account while sampling.



(a) Our method (foreground focus)

(b) Constant lens sampling (same cost)

(c) Comparison with constant lens sampling at equivalent cost

(d) Lens space frequency map and number of lens samples

(e) Image−space frequency map

(f) Our method (backgound focus)

(g) Constant lens sampling (same cost)

Figure 4.14: *Example of renderings using our method, with two settings of the focus plane* (a) *and* (f)*. In both cases, we compare our result to sampling the lens constantly throughout the image and by shooting the same number of total rays than in our method. The images obtained are much more blurry in regions of high variance, such as door handles which are highly curved very specular materials. In (c) we zoom on specific image locations and compare our method (at left) to the uniform constant sampling (at right). In* (d) *and* (e) *we show the lens and image-space frequency maps (logarithmic tone mapping) that we used to sample the lens and image, as well as the number of lens samples used at some locations.*

4.4 Visibility spectra

The visibility function $V(x, v)$ is defined with respect to a virtual plane, which makes it a directional function. In other words, an occluder's visibility function is different for central rays with different directions. Consider a planar occluder in the interval $[a, b]$ on the virtual plane that defines $V(x, v)$. Since $V(x, v) = 0,\ \forall v\ \forall x \in [a, b]$, the visibility spectrum is a Dirac delta in the angular domain and a sinc along the spatial dimension, corresponding to the Fourier transform of a pulse of width $b - a$. For non planar occluders, the visibility function, and hence the spectra, have information in the spatial and angular dimensions. We limit ourselves to studying opaque occluders. That is, occluders with binary visibility functions.

The goal of this section is to describe representations for visibility spectra that we experimented with. and to study the reason for their inapplicability in a practical setting. We present an analytic and a numerical approximation for precomputation and storage of spectra.

4.4.1 Approximate analytical representation

Along a given ray direction, let us consider a non-planar occluder as being composed of infinitesemal transverse slices, each of which can be considered a planar occluder. Each planar occluder is a square pulse which has a 0 value representing points inside or on the occluder[7]. Thus the occluder can be approximated by a sequence of N square pulses. We are interested in defining the visibility function along the given direction as $N \to \infty$. We present a study of convex occluders without holes, in flatland.

Let $V(x, \theta)$ be the visibility function defined at the virtual plane P that is at the

[7]This corresponds to negative logic, where a low signal indicates the pulse—in this case a 0 indicates occlusion.

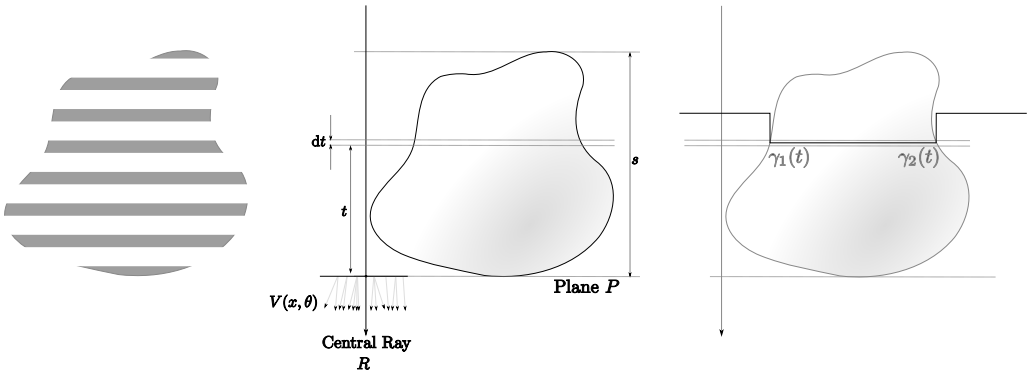


Figure 4.15: Left: *A 2D occluder approximated by a number of slices transverse to the ray under consideration.* Middle: *In the limit, each slice* dt *at distance* t *from* P *contributes to the visibility* $V(x, \theta)$. Right: *The slice* dt *can be considered as a planar occluder, causing the visibility at the plane to be a pulse.*

farthest plane transverse to the ray direction that the occluder lies in (see Figure 4.15). Let the depth interval spanned by the occluder be $[0, s]$ where a depth of s corresponds to the plane P. Consider the single slice of the occluder at a distance t from P. Let the interval spanned by this slice of the occluder be $[\gamma_2(t), \gamma_1(t)]$. We represent the occlusion due to this slice by a square pulse $\prod_{\gamma_2(t)}^{\gamma_1(t)}(x)$. If this were the only slice approximating the occluder, the lightfield at P can be written as

$$V(x, \theta) = \prod_{\gamma_2(t)}^{\gamma_1(t)} (x - t\theta), \tag{4.33}$$

where the reparameterization to $x - t\theta$ accounts for the transport from the slice to P. Approximating the occluder with N planar occluders, we can express $V(x, \theta)$, approximately, as the product of the pulses at each of N slices.

$$V(x, \theta) = \prod_{i=0}^{N} \left[\prod_{\gamma_2(t_i)}^{\gamma_1(t_i)} (x - t_i\theta) \right]. \tag{4.34}$$

$\gamma_2(t)$and $\gamma_1(t)$ characterise the left and right boundaries of the occluder respectively by defining the variation (with t) of the start and end of the pulse at each slice.

Using a common engineering trick that uses logarithms to replace product with sums, and taking the limit $N \to \infty$, we get

$$V(x,\theta) = \exp\left[\int_0^s \ln\left(\prod_{\gamma_2(t)}^{\gamma_1(t)} (x - t\theta) \right) \mathrm{d}t \right]. \tag{4.35}$$

Since the term inside the logarithm could go to zero, we approximate the occlusion due to a slice by a combination of exponential terms.

Approximating the Square Pulse

The square pulse $\prod_{\gamma_2(t)}^{\gamma_1(t)}(x)$ can be written as

$$\prod_{\gamma_2(t)}^{\gamma_1(t)}(x) = 1 - \mathrm{H}(x - \gamma_1(t)) + \mathrm{H}(x - \gamma_2(t)), \tag{4.36}$$

where $\mathrm{H}(x)$ is the Heaviside step[8] function. The Heaviside step function can be approximated by

$$\mathrm{H}(x-a) \approx \frac{\exp[\,(x-a)\,\alpha\,]}{1 + \exp[\,(x-a)\,\alpha\,]}, \tag{4.37}$$

where α is a parameter that can be used to control the approximation. The accuracy of the approximation increases as α is increased. Under this approximation, and the approximation

$$\frac{1 + \exp[\,A+B\,] + \exp[\,B\,]}{(1+\exp[\,A\,])(1+\exp[\,B\,])} \approx \frac{1 + \exp[\,A+B\,]}{(1+\exp[\,A\,])(1+\exp[\,B\,])}, \tag{4.38}$$

[8]The heaviside step function is the antiderivative of the Dirac delta

Equation (4.36) becomes

$$\prod_{\gamma_2(t)}^{\gamma_1(t)}(x) \approx \frac{1+\exp\left[\ ((x-\gamma_2(t))+(x-\gamma_1(t)))\alpha\ \right]}{(1+\exp\left[\ (x-\gamma_2(t))\alpha\ \right])\ (1+\exp\left[\ (x-\gamma_1(t))\alpha\ \right])}.$$

The windowed visibility spectrum

Substituting the expression for the approximation of the square pulse into Equation (4.35), we obtain

$$\begin{aligned} V(x,\theta) = \exp&\left[\ \int_0^s ln(1+\exp\left[\ ((2x-2t\theta-\gamma_2(t)-\gamma_1(t))\alpha)\ \right])\ \mathrm{d}t\ \right] \\ &\times \exp\left[\ -\int_0^s ln(1+\exp\left[\ ((x-t\theta-\gamma_1(t))\alpha)\ \right])\ \mathrm{d}t\ \right] \\ &\times \exp\left[\ -\int_0^s ln(1+\exp\left[\ ((x-t\theta-\gamma_2(t))\alpha)\ \right])\ \mathrm{d}t\ \right]. \end{aligned} \tag{4.39}$$

Observe that the functionals defined by the exponential terms in Equation (4.39) take a common form:

$$\psi_{\varsigma}^{a,b,c}(x,s,\theta) \equiv \exp\left[\ a\int_0^s ln(1+\exp\left[\ ((bx-\varsigma(t,\theta))c)\ \right])\ \mathrm{d}t.\ \right] \tag{4.40}$$

We rewrite the visibility as a product of three functionals Ψ_1, Ψ_2 and Ψ_3:

$$\begin{aligned} V(x,\theta) &= \Psi_1\ \Psi_2\ \Psi_3 \\ &\equiv \psi^{1,2,\alpha}_{\gamma_1(t)+\gamma_2(t)+2t\theta}(x,s,\theta) \quad \psi^{-1,1,\alpha}_{\gamma_1(t)+t\theta}(x,s,\theta) \quad \psi^{-1,1,\alpha}_{\gamma_2(t)+t\theta}(x,s,\theta). \end{aligned} \tag{4.41}$$

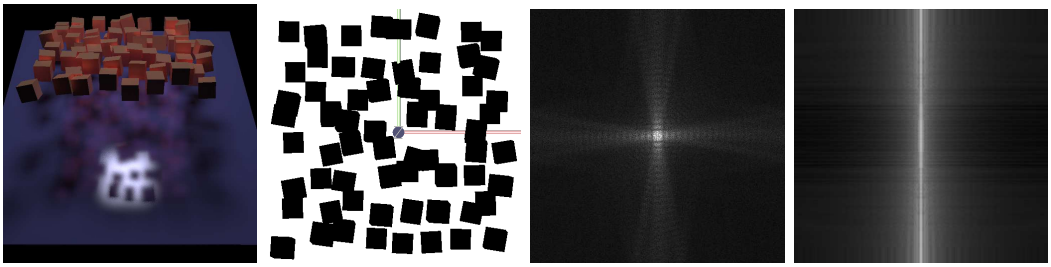


Figure 4.16: From left to right*: Scene containing a set of cubes as occluders; the binary visibility function (vertical viewing direction); the visibility spectrum (vertical viewing direction); 2D spectra reduced to 1D spectra (rows) along multiple directions, by integrating along circles about the central* $(0, 0)$ *frequency.*

The windowed Fourier transform of the visibility field is therefore

$$\begin{aligned} \hat{V}(\Omega_x, \Omega_\theta) &\equiv (\mathcal{F} \circ \beta V)\ (x, \theta) \\ &= \hat{\beta}(\Omega_x, \Omega_\theta)\ *\ \hat{\Psi}_1\ *\ \hat{\Psi}_2\ *\ \hat{\Psi}_3. \end{aligned} \tag{4.42}$$

Discussion

One simple algorithm to account for non-planar occluders would be to consider them as a set of slices and treating each as an occluder. However, simply performing a series of shears and convolutions to approximate the visibility spectrum results in an algorithm that performs as many convolutions as there are slices.

The expression obtained for the visibility function (see Equation (4.39)) consists of a product of three terms, irrespective of the number of slices used in the approximation. Choosing a set of slices to represent the occluder amounts to sampling the integrals in the exponents of the equation. While the accuracy of the visibility function increases when many slices are chosen, the expression for the visibility spectrum consists of four convolutions.

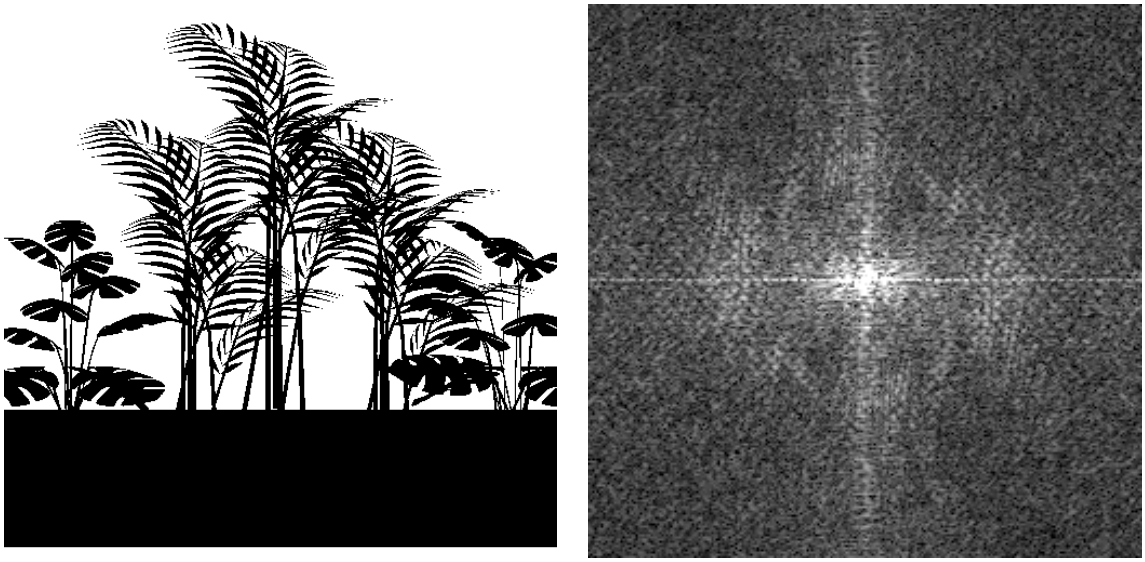


Figure 4.17: *Binary visibility function of a set of occluders* (left) *in a given direction, along with its Fourier power spectrum* (right).

The difficulties of using this representation in a practical algorithm are: (1) the occluder must not have holes; (2) obtaining $\gamma_1(t)$ and $\gamma_2(t)$ is not practical for occluders that are commonly modeled as meshes; (3) $\gamma_1(t)$ and $\gamma_2(t)$ are directional functions.

4.4.2 Numerical representation for visibility spectra

Given unlimited memory, we could imagine storing the Fourier transforms of binary snapshots of the occluder from every view. Each snapshot is a white image with the occluder rendered in black. In such a setting, the visibility spectrum along the direction of a given central ray would simply be a lookup into the stored spectra.

Obviously, for the method to be practical the number of views must be limited and the spectra for arbitrary viewing directions obtained using interpolation. In addition, rather than storing the results of the Fourier transforms of the snapshot, only the resulting power spectra can be stored. Associated with each occluder, we can imagine

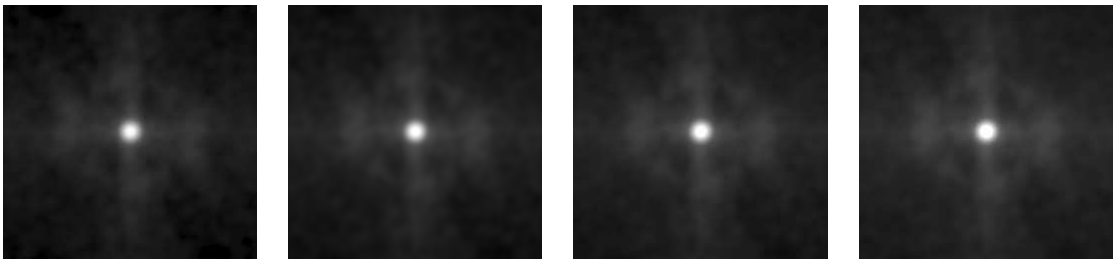


Figure 4.18: *Approximate power spectra for the visibility function shown in Figure 4.17 computed by averaging randomly centred, equally wide, windowed Fourier transforms of the visibility function. The spectra shown were computed using 64, 144, 256 and 900 windowed Fourier transforms from left to right, respectively. The windowing function chosen was a cosine to the fourth power.*

storing a sequence of power spectra corresponding to different central ray directions. To further compact the representation, for each view, we can integrate along circles about the origin in the Fourier domain. This yields a 1D vector corresponding to each snapshot. Thus, associated with each occluder is an image whose rows correspond to 1D visibility spectra obtained along different directions (see Figure 4.16).

Experiments with this representation reveal two fundamental limitations that can both be attributed to the lack of phase information: (1) this representation is unable to distinguish between the visibility spectra along two parallel rays at different distances from the occluder. (2) It is not obvious how correlation between the spectra of multiple obstacles must be accounted for.

The problem of determining "closeness"

Clearly any ray that passes at a finite distance past an occluder picks up some frequencies depending on the visibility function along its direction. However, in a practical setting, rays that are not "close" to an occluder can be reasonably assumed to remain unaffected. The use of numerically computed spectra without phase information, does not distinguish between rays at different distances from the occluder if they are

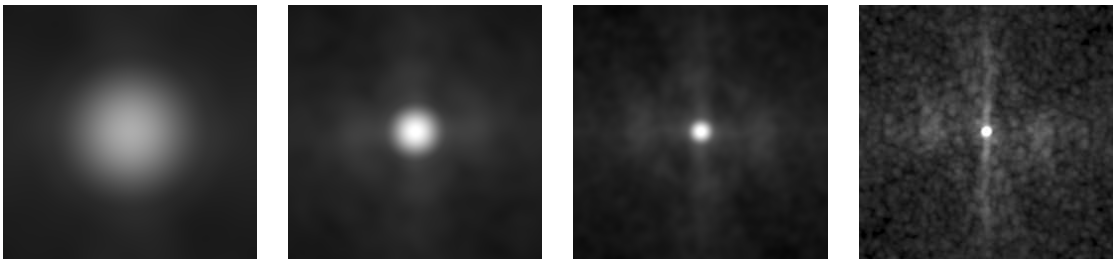


Figure 4.19: *The effect of increasing window size on visibility spectra computed by averaging* 100 *randomly centred, windowed spectra. The window sizes, from left to right, were 20, 50, 100 and 200 pixels respectively for a* 514×514 *image. Resolution in the frequency domain increases as the window size is increased. Certainty in the frequency predictions is necessarily forgone while trying to gain resolution in space.*

parallel. This inherent deficiency in the method demands an auxiliary mechanism that artificially accounts for the distance to the occluder. This mechanism must, in addition, determine when a ray is close enough to the occluder that the convolution with the visibility spectrum must be performed.

One possible solution is to compute and store the average power spectrum, for each direction, from several windowed Fourier transforms where the windows centres are uniformly randomly distributed (see Figure 4.18). The resulting average 2D visibility spectrum can be collapsed to 1D as before, for efficient storage. The advantage of this method is that the stored spectrum corresponds to the average result of a ray that passes by occluders at a distance corresponding to the size of the window used in the precomputation. Since averaging power spectra disregards phase, this is only an approximation.

A hierarchy of average spectra can be computed where each level in the hierarchy stores spectra computed with a wider window than the previous. A spectrum can be chosen from this hierarchy depending on the distance of the central ray for which the visibility spectrum query is made. The effect of window size on the averaged spectrum is shown in Figure 4.19.

The problem of scale and correlation

Consider a ray that passes by two obstacles, each with their own precomputed directional visibility spectra. Accounting for the two obstacles individually and performing two independant convolutions of the visibility power spectra with the lightfield spectra completely disregards correlation between the two obstacles. One possible remedy would be to hierarchically compute spectra, by clustering objects together. This raises more questions like how the level in the hierarchy of spectra must be chosen for a particular ray, how the clustering must be done, etc.

Chapter 5

Statistical assessment of estimators

In a valid *deductive* argument the premises logically entail the conclusion, where such entailment means that the truth of the premises provides a guarantee of the truth of the conclusion. An *inductive* logic is a system of reasoning that extends deductive logic to less-than-certain inferences. In a good inductive argument or assertion, the premises should provide some degree of support for the conclusion, where such support means that the truth of the premises indicates with some degree of strength that the conclusion is true.

In the context of Monte Carlo image synthesis one is often faced with the task of supporting an assertion that a given algorithm is superior in that it can produce images with the same first-order statistics (generally the expected value at each pixel), while exhibiting different second-order statistics (generally a reduction in variance). For example, algorithms for importance sampling or stratified sampling, when properly implemented, will exhibit precisely these characteristics; that is, reducing variance while leaving the mean intact. On the other hand, biased estimators are sometimes specifically constructed, primarily to reduce variance in the estimate or to simplify the

algorithm. Such results are commonly demonstrated with comparison images showing a reduction in the "graininess" of the image and/or a reduction in running time by virtue of the proposed algorithm. Plots of the first- and second- order statistics of the estimators are used to help in the assessment.

Novel rendering algorithms are often proposed in order to compute a given image faster or to allow effective trade-offs between speed and accuracy. In either case, the question naturally arises as to how one can demonstrate that a proposed algorithm meets the stated criteria. Presently it is widespread practice within the rendering community to employ a combination of objective and subjective criteria; running time is an objective criterion that is easy to measure and compare, while image quality, which presents a much greater challenge, generally rests upon subjective criteria such as visual inspection of two images or variance-plots.

There are numerous disadvantages to relying on subjective assessments such as visual comparison of images or plots: 1) they are only weakly quantitative, since comparisons are usually binary 2) the absolute variance is not a useful indicator of the quality of the estimator unless some assertions can be made about the mean 3) subtle errors can go undetected, and 4) the comparison cannot be automated.

In this chapter, we explore the use of a well known set of tools in statistics to objectively assess Monte Carlo estimators in image synthesis using simple criteria based on first and second order statistics. In addition we demonstrate the use of these tools for verifying sampling algorithms and detecting errors.

5.1 Statistical tests of hypotheses

Statistical inference is in the nature of inductive logic, aiming to make generalizations from particular observations. For the logic of inductive arguments to be of any value, the measure of support it articulates must satisfy the Criteron of Adequacy (CoA) : As evidence accumulates, the degree to which the collection of true evidence statements comes to support a hypothesis, as measured by the logic, should tend to indicate that false hypotheses are probably false and that true hypotheses are probably true. Methods of statistical inference are prone to controversies since they inherently involve generalization from observed data. While several methods exist for inference, their efficacies may only be meaningfully ranked in the context of a specific application.

Once a hypothesis has been formed, one often is interested in testing it against the observed data. While testing hypotheses may be more useful in cases where the hypotheses have been framed based on heuristics, the tests might also be used to compare different techniques of inference for a given application. There are numerous types of statistical tests, associated with different forms of application problems, such as significance tests that determine whether a hypothesis ought to be rejected, parametric tests to verify hypotheses concerning parameter values, goodness of fit tests to determine whether an observed distribution is compatible with a theoretical one, etc. Statistically *significant* results are those that are unlikely to have occurred by chance. *Significance Tests* are procedures for establishing the probability of an outcome, on a null hypothesis of no effect or relationship.

5.1.1 Brief history

In contrast to the Bayesian approach to inductive inference which is based on the inverse probability $Pr(H|x)$ of a hypothesis H given the data x, Fisher urged the adoption of direct probability $Pr(x|H)$ in an attempt to argue "from observations to hypotheses" [40]. If the data deviated from what was expected by more more than a specified criterion, the level of significance, the data was used to reject the null hypothesis. However, Fisher's significance tests are difficult to frame in general since often there exist no natural or well-defined complements to null hypotheses eg. H_0 : *The sample was drawn from the unit normal distribution.*

The terminology *Hypothesis Testing* was made popular by Neyman and Pearson [49, 50] who formulated two competing hypotheses called the null hypothesis (H_0) and the *alternative hypothesis* (H_1). Given a sample [1] from an arbitrary population, the goal of hypothesis testing is to test H_0 against H_1 according to the given data. Although the Neyman-Pearson theory was criticised [41] for only being suited to situations in which repeated random sampling has meaning, it fits well in the context of assessing MC estimators used in image synthesis. While Fisher's view of inductive inference focused on the rejection of the null hypothesis, the Neyman-Pearson theory sought to establish rules for making decisions between two hypotheses. This fundamental difference is exploited in all the tests that we present later in this chapter. The tests described in this chapter are results in statistical inference that can be found in standard textbooks [43, 86] that have been adapted for assessing Monte Carlo estimators in image synthesis.

[1] We remind the reader that, in this chapter, we shall use the term *sample* as it is used in statistics; that is, to refer to a set of observations of a population, not a single observation, as it is commonly used in the graphics literature.

5.1.2 Theory

Definition **5.1.** *Let S denote the sample space of outcomes of an experiment and x be an arbitrary element of S. Let H_0 be a hypothesis which specifies, partly or completely, the probability measure on the Borel field $\mathcal{B}$ of sets in S. The problem of hypothesis testing is to decide, based on an observed x, whether H_0 is true or not.*

In practice, the points of S will be regarded as the realization of a random variable (r.v.) X such that $Pr(A|H) = Pr(X \in A|H)$ for $A \subset S$. We may then write a function T defined over S as a function $T(X)$ of X with the value $T(x)$ when $X = x$. Then $T(X) > \lambda \equiv x : T(x) > \lambda$ and $Pr(\{x : T(x) > \lambda\}) = Pr(T(X) > \lambda)$.

Let X be a r.v. dependent on some observed data. If the probability distribution of X is completely specified by the null hypothesis, the corresponding null hypothesis is referred to as a *simple* hypothesis. In cases where the null hypothesis does not completely specify the probability distribution but, instead, specifies the distribution as particular functions of a set of parameters, the hypothesis is said to be of type *composite*.

Regardless of the type of the hypothesis and the procedure used to test the null hypothesis, the testing process involves two types of errors: (1) that of rejecting H_0 when it is, in fact, true and (2) that of not rejecting H_0 when an alternative hypothesis is true. These are called errors of the first and second kinds or Type I and Type II errors respectively.

We only consider nonrandomized test procedures; that is, one where the sample space is divided into two regions W and $S - W$ and H_0 is rejected if $x \in W$. Here x is an observed outcome and W is called the *critical region*.

For simple hypotheses the probability, α, of errors of the first kind is the probability measure of the set W under the hypothesis H_0,

$$Pr\,(W|H_0) = \alpha. \tag{5.1}$$

α is also called the *level of significance* with which the null hypothesis was accepted. The probability $\beta(h)$, or error of the second kind for a particular alternative $h \in H$ of the set of alternative hypotheses H, is

$$Pr\,(S - W|h) = \beta(h), \tag{5.2}$$

and $\gamma(h) = 1 - \beta(h)$ defined over H is called the *power function*.

For a composite null hypothesis H_0, α is defined as

$$Pr\,(W|H_0) = \sup_{h \in H_0} Pr\,(W|h)\,. \tag{5.3}$$

Neyman and Pearson [49, 50] presented one of the earliest formal theories leading to the clear understanding of problems related to hypothesis testing. They posed the problem as follows: Given a level of significance, we would like to reduce errors of the second kind to as low as possible, or maintain as large a power function as possible.

Lemma 5.2. *Let f_0, f_1, ,,, be integrable functions over space S with respect to a measure v and let W be any region such that*

$$\int_W f_i \, \mathrm{d}v = c_i \; (given), \quad i = 1, 2, ... \tag{5.4}$$

Further, let there exist constants k_1, k_2, ... such that for the region W_0, within which

$f_0 \geqslant k_1 f_1 + k_2 f_2 + \ldots$ and outside which $f_0 \leqslant k_1 f_1 + k_2 f_2 + \ldots$, the conditions in Equation (5.4) are satisfied. Then

$$\int_{W_0} f_0 \, \mathrm{d}v \geqslant \int_{W} f_0 \, \mathrm{d}v \tag{5.5}$$

Consider the case where a simple null hypothesis H_0 is to be tested, against a simple alternate hypothesis H_1. Let $Pr\,(x|H_0)$ and $Pr\,(x|H_1)$ be the probability densities at x under H_0 and H_1 respectively with respect to a σ-finite measure v. The problem is that of determining a critical region such that

$$\int_{W} Pr\,(x|H_0) \, \mathrm{d}v = \alpha \ \textit{(assigned value)} \tag{5.6}$$

$$\int_{W} Pr\,(x|H_1) \, \mathrm{d}v \ \ \textit{is maximum} \tag{5.7}$$

For solving this problem, we use Lemma 5.2. Choosing $f_0 = Pr\,(x|H_1)$ and $f_1 = Pr\,(x|H_0)$, the optimum region W is defined by

$$\{x : Pr\,(x|H_1) \ \geqslant \ k\, Pr\,(x|H_0)\} \tag{5.8}$$

provided there exists a k such that Equation (5.6) is satisfied. Observe that using the r.v. X, the test in Equation (5.8) can be written as

$$T \ = \ \frac{Pr\,(X|H_1)}{Pr\,(X|H_0)} \ \geqslant \ k \tag{5.9}$$

If the distribution of T with respect to H_0 is continuous, then there exists a k such that $Pr\,(T \geqslant k|H_0)$ for any assigned α. In this case, T is dependent on the simple

alternate hypothesis H_1. T is called the *test statistic* and needs to be derived for the hypotheses being tested against. In Section 5.2 we present some standard test statistics and how they could be used in our context.

One-tailed Tests : Tests in which the critical region lies at either the left or right of the distribution $p(x)$ followed by the test statistic. Given the max probability of false rejection α, the two critical values are obtained as $P^{-1}(\alpha)$ and $P^{-1}(1-\alpha)$ which are the the inverse cumulative distribution evaluated at α and $1-\alpha$ respectively. The null hypothesis is rejected if the test statistic that is computed from the data lies below or above the critical values respectively. The appropriate alternate hypothesis may be accepted.

Two-tailed Tests : Tests in which the critical region is equally distributed at both ends of the distribution $p(x)$ followed by the test statistic. Given the max probability of false rejection α, two critical values are obtained as $P^{-1}(\alpha/2)$ and $P^{-1}(1-\alpha/2)$. The null hypothesis is rejected if the test statistic that is computed from the data does not lie between these two critical values.

5.1.3 Procedure summary

The general algorithm for testing hypotheses proceeds in a number of steps. The first step involves formalization of the null hypothesis. After stating the hypothesis in a way that allows the probabilities of samples to be calculated assuming that the hypothesis is true, the next step is to set up a statistical test that will aid in likely reject the null hypothesis in favour of the alternative hypothesis. The *test statistic* is a prescription according to which a number is computed from a given sample— that is, a real-valued function of the sample. Sometimes the test statistic could be a function of two samples, and in such cases the test is called a *two sample test*. Given a sample,

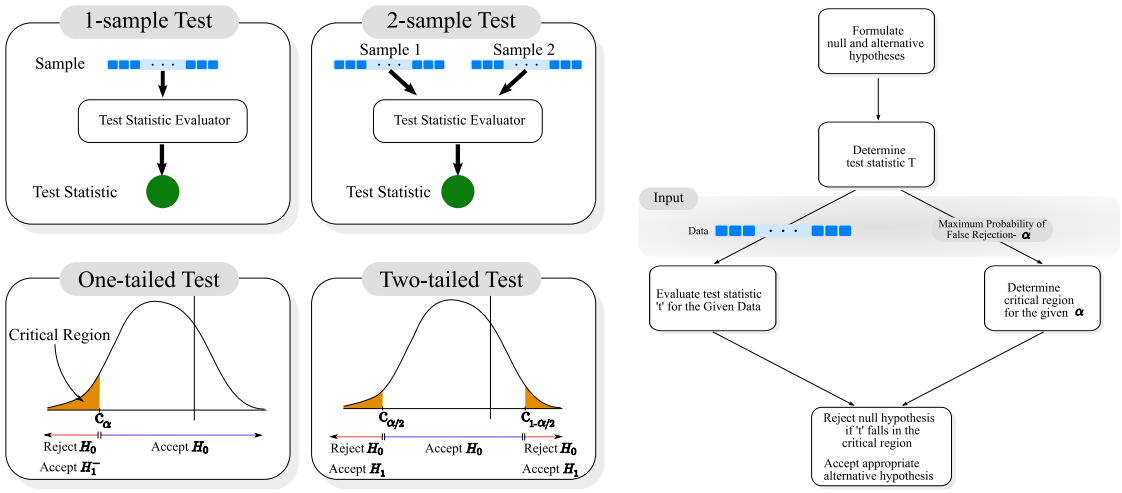


Figure 5.1: *Overview of the general procedure for hypothesis tests presented in this chapter.*

its associated value of the test statistic is used to decide between accepting the null and the alternative hypotheses. Thus there exist probabilities associated with false rejection (Type I) and false acceptance (Type II) errors which are typically denoted by α and β respectively. An acceptable α along with the test statistic defines a region of the parent distribution where H_0 is rejected in favor of H_1; this region is called the *critical region*. α defines the maximum probability of the test statistic falling in the critical region despite the null hypothesis being true and corresponds to the fraction of the time that the null hypothesis is erroneously rejected. If the critical region is chosen to lie either completely at the left tail of the parent distribution or completely at the right tail, the test is called a *one-tailed test* or assymetrical or one-sided test. If the critical region is chosen to equally cover the left and right tails, the test is called a *two-tailed test* or symetrical or two-sided test. α is an input parameter and is typically chosen to be low (see Figure 5.1).

With the hypothesis and test statistic set up and having identified the critical region, the data is examined for evidence to reject the null hypothesis. The test statistic is calculated for the given sample data and tested to check if it lies in the critical region.

If this is the case, then the conclusion is that either the null hypothesis is incorrect or an erroneous result of probability less than α has occurred and in either case we accept the alternate hypothesis. Parametric hypothesis tests that hypothesize about parameters of the parent distribution, such as mean and variance, are intimately tied to the distribution of the population under study and most of the existing techniques only apply to distributions of a restricted type. In fact, the vast majority of the existing theory has been developed for populations with normal distributions.

5.2 Hypothesis Tests for mean and variance

5.2.1 One Sample Mean Test

The goal of this test is to assert with some confidence that the mean of the distribution from which a sample y of size n is drawn, is a specific value μ_0. The test assumes that the distribution from which the sample is drawn is normal but does not make any assumption about its true variance. The null and alternative hypotheses for this test are

$$H_0 : \overline{y} = \mu_0,$$
$$H_1 : \overline{y} \neq \mu_0,$$
$$H_1^+ : \overline{y} > \mu_0,$$
$$H_1^- : \overline{y} < \mu_0.$$

The test statistic is

$$t_\nu = \frac{\overline{y} - \mu_0}{s/\sqrt{n}} \tag{5.10}$$

which follows the Student's t-distribution with $\nu = n - 1$ degrees of freedom. The null hypothesis is tested against the first alternative hypothesis with a two-tailed test and against the other two alternative hypotheses with the appropriate one-tailed tests. If the data do not provide enough evidence, at the given α probability of false rejection, to reject the null hypothesis in favour of any of the alternate hypotheses then we accept that the mean of the sample is not significantly different from μ_0.

5.2.2 One Sample Variance Test

This test allows the variance of the distribution from which a sample y of size n is drawn, to be compared with some confidence against a specific value σ_0^2. The test assumes that the distribution from which the sample is drawn is normal but does not make any assumption about its true mean. The null and alternative hypotheses for this test are

$$H_0 : \sigma^2 = \sigma_0^2,$$
$$H_1^+ : \sigma^2 > \sigma_0^2,$$
$$H_1^- : \sigma^2 < \sigma_0^2.$$

The distribution of observed variances s^2 for samples drawn from some numerical population follows the chi-square distribution, which we use as the test statistic in this case. The test statistic is

$$\chi_\nu^2 = \frac{\nu s^2}{\sigma_0^2} \tag{5.11}$$

where again the degrees of freedom $\nu = n - 1$. An interesting property of this distribution is that the s^2 values average σ^2, the actual (usually unknown) variance of the distribution. Two one-tailed tests are performed to test if the data provides

enough evidence to reject the null hypothesis in favour of either of the alternative hypotheses.

5.2.3 Comparing Means of Two Samples

This test compares the means of two distributions, each of which is represented by one sample, to check for equality without making any assumptions about the variances of the distributions. If the two samples are y_1 and y_2 of sizes n_1 and n_2 respectively, the null and alternative hypotheses are

$$H_0 : \overline{y}_1 = \overline{y}_2,$$
$$H_1 : \overline{y}_1 \neq \overline{y}_2.$$

The test statistic is

$$T_\nu = \frac{\overline{y}_1 - \overline{y}_1}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} \tag{5.12}$$

which follows the Student's t-distribution with

$$\nu = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

degrees of freedom. A two-tailed test is used to determine whether the samples provide enough evidence to reject the null hypothesis in favour of the alternative hypothesis.

5.2.4 Comparing Variances of Two Samples

To compare the variances of two distributions, each of which is represented by one sample, we use the standard F-test. If the two samples are y_1 and y_2 of sizes n_1 and

n_2 respectively, the null and alternative hypotheses are

$$H_0 : s_1^2 = s_2^2,$$
$$H_1^+ : s_1^2 > s_2^2,$$
$$H_1^- : s_1^2 < s_2^2.$$

The test statistic is

$$F_{\nu_1,\nu_2} = \frac{s_1^2}{s_2^2} \tag{5.13}$$

which follows the F-distribution with ($\nu_1 = n_1 - 1$, $\nu_2 = n_2 - 1$) degrees of freedom. The null hypothesis is tested against the alternative hypotheses using two one-tailed tests.

5.3 Assessing Monte Carlo estimators

While completely automatic ranking of estimators is an enormous challenge, *hypothesis tests* may be used provide objective answers to several very basic queries about r.v.'s. If X and Y are r.v.'s, we answer queries such as "Is the mean value of X equal to μ_0?" or "Is the mean value of X equal to the mean value of Y?" or "Is the variance of X less than that of Y?". The structure of such queries is to first pose a *null hypothesis*, such as $E(X) = E(Y)$ and competing *alternative hypotheses* such as $E(X) \neq E(Y)$, $E(X) < E(Y)$ and $E(X) < E(Y)$. Then, solely based on samples drawn from the parent distributions of X and Y, the null hypothesis is either *accepted* or *rejected* with a given level of confidence. The null hypothesis is only accepted if the data do not provide enough evidence to reject it. If the null hypothesis is rejected, further tests are made to decide which alternative hypothesis may be accepted.

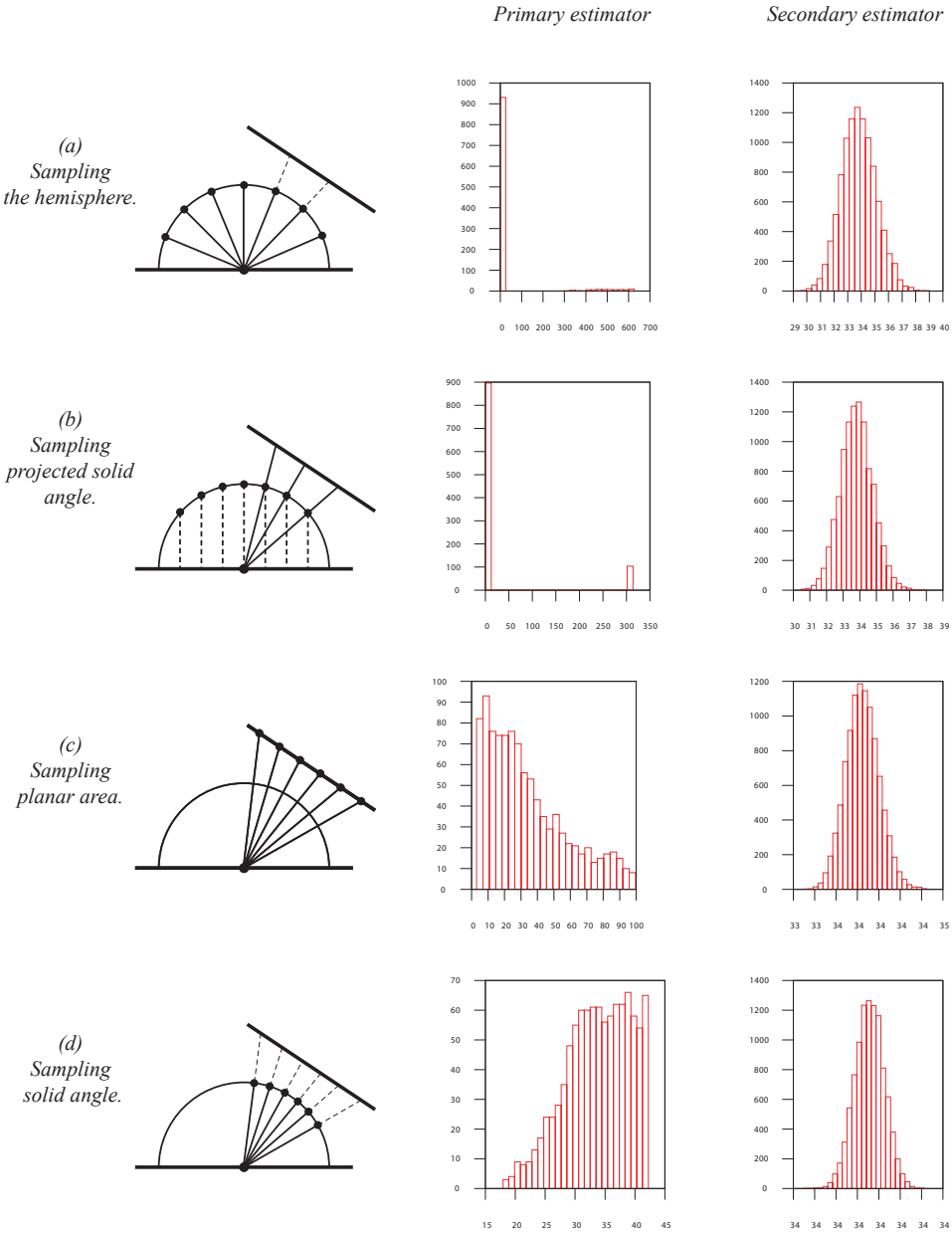


Figure 5.2: *Comparing four different Monte Carlo estimators for computing irradiance. The histograms show frequency vs irradiance for a large number of estimates. The distribution characteristic of the secondary estimators is observed to be close to normal with the same mean but different variances depending on the sampling scheme.*

Previous work in computer graphics has drawn upon similar tools, such as the Chi-Square and Student-t distributions, although they have focused on applications involving the problem of estimating true variance using sample variance for the purpose of stochastic anti-aliasing [34, 66, 84]. Here, however, we present a variety of significance tests for assessing both the mean and variance of the r.v.'s resulting from Monte Carlo estimations for the purpose of verifying that they are indeed estimating what they are intended to estimate; that is, we are not interested in directly assessing the accuracy of an approximation, rather the correctness and efficiency of the estimator.

Two important hurdles in trying to apply statistical tests to populations defined as the outputs of MC estimators are : (1) dealing with estimators whose estimates are not distributed normally and (2) formulating the null hypothesis and setting up the statistical tests

By the central limit theorem, the distribution of the estimated means of samples of MC estimator E rapidly approaches a normal distribution as the size of each sample is increased. To overcome the first of the two hurdles, rather than assess the primary estimator, we simply use distributions obtained from secondary estimators E_s (see Figure 5.2) in our assessment.

To overcome the latter hurdle, we first need to define the goal of the test. In the context of MC estimators two parameters are of interest– mean and variance. Our goal is to hypothesize about each of these parameters in two distinct settings: comparing an estimator with analytically obtained results and comparing two estimators (one- and two- sample tests). We address each of the four different combinations of problems describing the null hypotheses and describe the corresponding well-known statistical tests.

5.4 Applications in image synthesis

In this section, we present a few applications in Monte Carlo image synthesis where hypothesis testing can be used to assess estimators. The goal is to verify that the results of hypothesis testing are consistent with theoretical predictions. Thus, we consider scenarios that are well understood and present hypotheses that are already known to be true (or false) to the hypothesis testing framework and record whether they are accepted (or rejected).

5.4.1 Irradiance

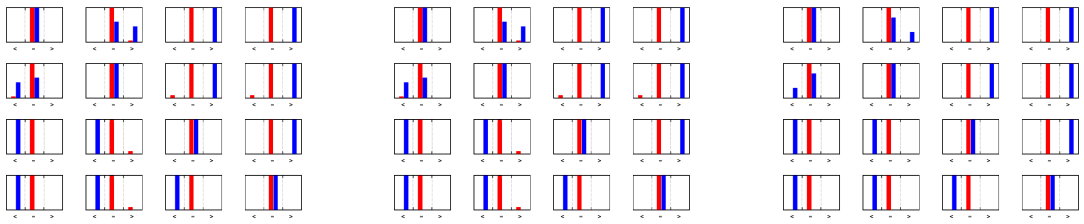
Consider the irradiance at a point $\mathbf{x}$ with normal $\mathbf{n}$ due to a triangular uniform, lamberitian emitter in the absence of occluders. The existence of an analytical solution, commonly known as *Lambert's formula* [6], combined with the availability of several MC solutions for comparison make this problem a good candidate for a case study. The irradiance at point $\mathbf{x}$ is given by

$$E(\mathbf{x}) = \int_{\mathcal{H}^2} L(\mathbf{x}, \omega)(\mathbf{n} \cdot \omega) \, \mathrm{d}\omega, \tag{5.14}$$

where $L(\mathbf{x}, \omega)$ is the incident radiance at $\mathbf{a}$ along ω and $\mathcal{H}^2$ is the hemisphere of directions defined by $\mathbf{n}$. $E(\mathbf{x})$ is estimated using the following methods:

1. Estimator U: uniformly sampling the hemisphere of directions and averaging the cosine weighted incident radiance along those directions.

2. Estimator C: sampling the projected hemisphere and averaging the incident radiance along those directions.



$\alpha = 0.1$ $\alpha = 0.05$ $\alpha = 0.01$

Figure 5.3: *Results from testing four estimators (see Section 5.4.1) to compare their means and variances. Rows and columns in each matrix of plots correspond to estimators U, C, A and S respectively. Frequencies of the results "less than", "equal to" and "greater than" for 2-sample mean (red) and variance (blue) tests are shown in each cell of the matrix from a sequence of 100 runs of each. The results clearly confirm that the means of all the estimators are equal and that $\sigma_U > \sigma_C > \sigma_A > \sigma_S$. The diagonals correspond to testing an estimator against itself and, as expected, the mean and variance tests report equality. For lower values of α, there are fewer false rejections. Observe that there is no clear winner in the test for variance between U and C but on average $\sigma_U > \sigma_C$.*

3. Estimator A: sampling the area of the triangle uniformly and averaging the estimates of irradiance due to each individual area element.

4. Estimator S: uniformly sampling the solid angle subtended by the triangle and averaging the estimates of irradiance along each direction.

We compare means and variances of the above estimators against each other and also compare against the analytical mean obtained using Lambert's formula. The tests are valid in this setting because the secondary estimators for the above yield roughly normal distributions (see Figure 5.2). Thus, each of the tests is repeated a number of times and the average result is reported. All the above estimators are known to be unbiased and the mean tests confirm this on average. We observe that sometimes, depending on the data, the mean test fails. By reducing the value of α, we can verify that the failures approximately correspond to false rejections allowed by the factor α. The result of the variance tests confirm that on average, $\sigma_U > \sigma_C > \sigma_A > \sigma_S$ (see

Figure 5.3).

5.4.2 Verifying sampling distributions

One of the many desirable properties of a BRDF is its suitability to be used in a MC rendering setup. This usually involves being able to sample from the reflectance function or an approximation of this function. In the latter case, so long as the exact density associated with each direction in the sample is known there is no risk of introducing a bias while estimating reflected radiance using the sample, regardless of how weak the approximation. However, the closer the approximation, the lower the variance in the estimated reflected radiance.

The goal of this case study is to use two popular BRDF models proposed by Ashikhmin and Shirley [11] and Ward [117, 113] and test whether the distributions sampled by the two techniques significantly differ from their corresponding reflectance functions. We select an input direction arbitrarily and obtain a sample containing many output directions according to the BRDF. We bin these samples and visualize the 2D histogram as a greyscale image where image intensity is proportional to bin frequency. For comparison, we visualize the histograms obtained by sampling each BRDF using rejection sampling. The test is set up so that the size of the sample obtained using rejection is equal to the size of the sample obtained by sampling the BRDF.

Visual inspection of the histograms is sufficient (see Figure 5.4) to assert that the sampling of the Ward's BRDF does not match the actual reflectance distribution. In the case of the Ashikhmin-Shirley BRDF however, it is not obvious. To assess the Ashikhmin-Shirley BRDF sampling algorithm we use the 2-sample GoF test. Since the test is applicable only to univariate distributions and we have a 2D distribution for a fixed outgoing direction, we linearize this 2D space by using a space filling curve

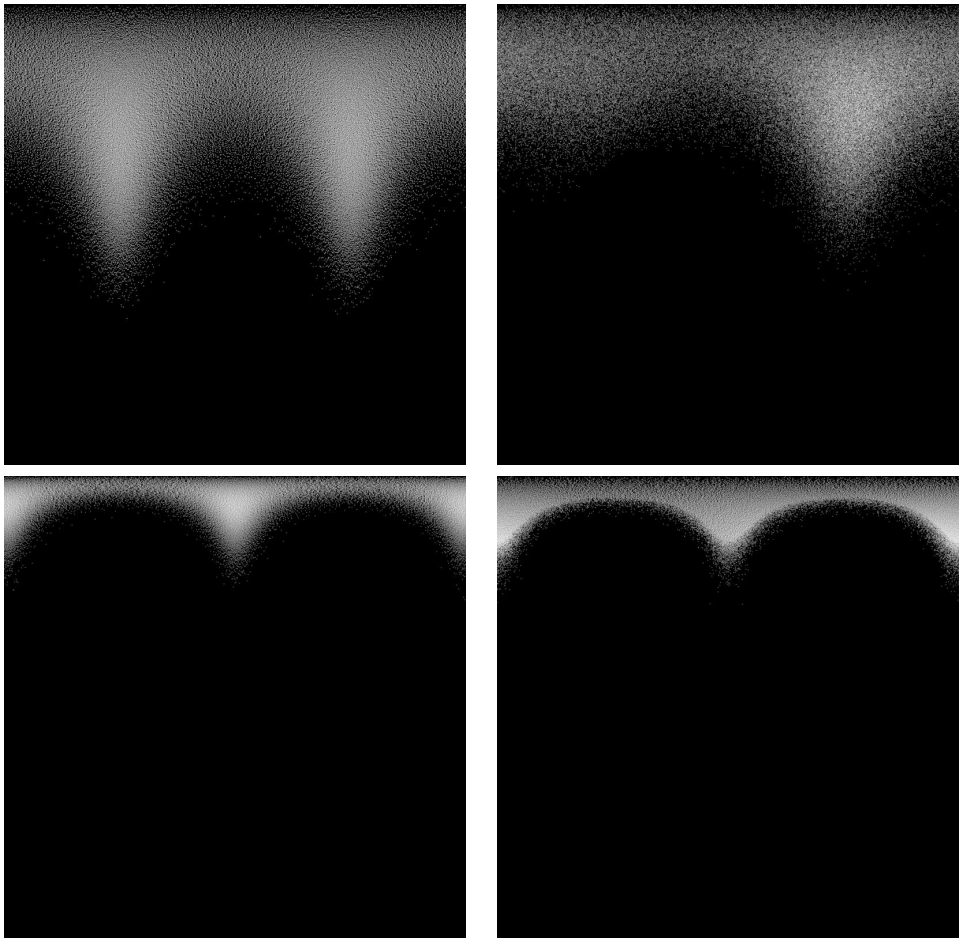


Figure 5.4: *Histograms of sample directions for two anisotropic BRDF's (Ward and Ashikhmin-Shirley) are shown, for a given outgoing direction. Multiple peaks are observed due to the anisotropy. Rows and columns in the image correspond to polar and azimuthal angles respectively. Sampling from the reflectance distribution (left) vs sampling using rejection (right) is shown. While it is evident that the distributions do not match for Ward's BRDF (top row), it is not obvious from visual inspection if the two samples for the Ashikhmin-Shirley BRDF (bottom row) represent the same distribution.*

such as Morton-order [108].

Since the GoF test is not a parametric test, we do not make any assumptions about the distribution other than that it is continuous [11]. Also since we can afford to repeat the experiment multiple times, two of the three major limitations of the K-S test are no longer major concerns in our application. The third limitation of the K-S test suggests that it will be less likely to detect sampling anomalies near the pole or near the horizon. We show that this is not a major concern in practice. If need be this decreased sensitivity to the tails may be made insignificant by adopting a parameterization scheme for the BRDF such as the half-angle parameterization [90] in conjunction with a linearization scheme, thus keeping the interesting changes of the BRDF in the middle of the distribution. The fact that the K-S test does not make assumptions about the distribution from which the samples are drawn is key.

The results of the 2-sample K-S test for a sample directly drawn from Ward's BRDF against one drawn using rejection failed for all levels of significance and any numbers of samples drawn. On the other hand, a similar test for the Ashikhmin-Shirley BRDF passed with $\alpha = 0.005$ for a sample size of less than 100. For larger samples the Ashikhmin-Shirley BRDF failed the test indicating that the distribution being drawn from does not match the reflectance distribution exactly. This is consistent with the sampling technique [11] which derives the scheme for a distribution that is very close to the reflectance function but not identical.

5.4.3 Detecting Errors

One of the applications of the hypothesis testing approaches we have described is catching unintended sources of bias, and determining whether an experimental variance reduction technique is in fact effective.

It is difficult to construct low-variance estimators that remain unbiased, either because of the intrinsic difficulty of correctly normalizing the probability density functions, or simply because they are prone to error. For example a factor of π, a missing cosine factor or an incorrect change of variables (e.g. cosine over distance squared) will lead to erroneous results that nevertheless look plausible and may therefore go unnoticed. Indeed, many sources of bias would be nearly impossible to detect without an objective comparison against either an analytic solution, or a trusted Monte Carlo estimator. For example, if stratified sampling over a 2-manifold is used with a mapping that is not uniform (i.e. a mapping that does not map equal areas in the parameter domain to equal areas on the manifold), there will be a systematic bias unless the strata are weighted according to their respective areas. Similarly, if samples are used both to estimate the mean and to guide adaptive sampling, the result is systematically biased downward [57]. In both cases, the bias may be arbitrarily large, yet offers no obvious visual clue of its existence. Such errors are relatively easy to catch with hypothesis testing.

We intentionally introduced three common unintended sources of bias in the estimator A (see Section 5.4.1) and verified that they could be detected by using the tests described in Section 5.2. In constructing A, Equation (5.14) is rewritten, using a change of variables, as

$$E(\mathbf{x}) = \int_{Area(\triangle)} L(\mathbf{x}, \mathbf{z}) \frac{\mathbf{n} \cdot \mathbf{z}}{\|\mathbf{z}\|} \frac{\mathbf{n}_{\triangle} \cdot \mathbf{z}}{\|\mathbf{z}\|^3} \, \mathrm{d}\mathbf{y} \tag{5.15}$$

where the integral is now over the area of the triangle as opposed to the sphere of directions, with $\mathbf{y}$ as the variable of integration. $\mathbf{n}_{\triangle}$ is the triangle's normal and $\mathbf{z} = \mathbf{x} - \mathbf{y}$ is a vector along ω. The term $(\mathbf{n}_{\triangle} \cdot \mathbf{z}/\|\mathbf{z}\|^3)$ is a factor that appears in the integral due to the change of variables. Specifically, we made the following three

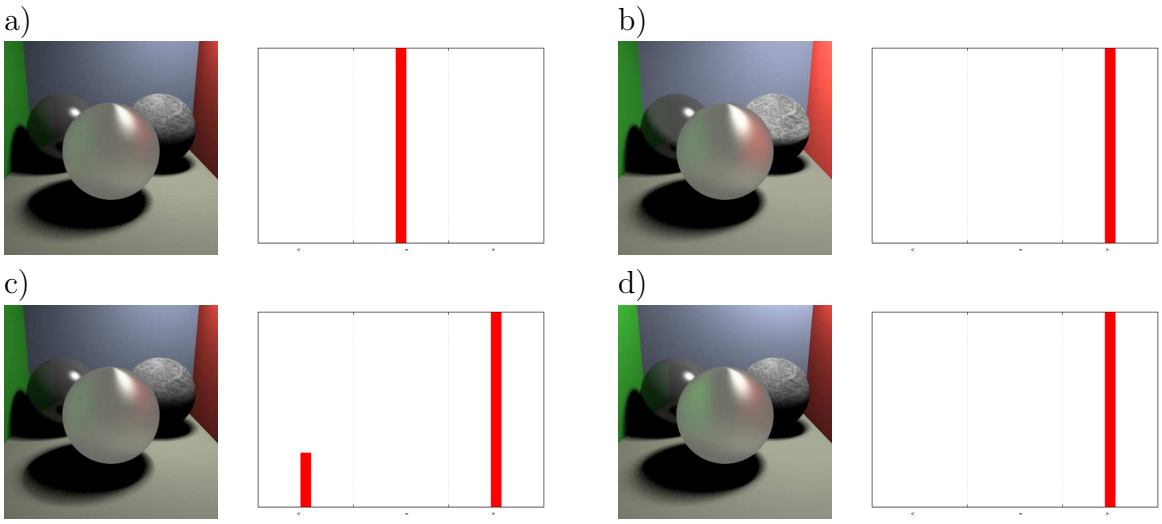


Figure 5.5: *Results of the 2-sample tests comparing the mean of an estimator against a trusted estimator before and after three errors were introduced in the former. The tests were performed with* $\alpha = 0.01$ *to verify that the difference in means after introduction of the errors was detectable. Images generated using the erroneous estimators are shown for a scene with shiny, textured and glossy (Ward's BRDF) spheres. a) Before introducing errors b) Missing cosine term; c) Non-uniform sampling of the illuminaire; d) Incorrect change of variables in Equation (5.15). The errors are not always obvious from just visual inspection.*

alterations

1. *Omitting the cosine term* $(\mathbf{n} \cdot \mathbf{z}/\|\mathbf{z}\|)$ in Equation (5.15)

2. *Non-uniform sampling* of the area of the triangle by using uniform random variables in $[0, 1]$ as barycentric coordinates.

3. *Incorrect change of variables* by omitting the $(\mathbf{n}_{\triangle} \cdot \mathbf{z}/\|\mathbf{z}\|^3)$ in Equation (5.15).

All three errors were promptly detected by running the 2-sample test for means when tested against the unmodified trusted estimator S (see Figure 5.5).

Chapter 6

Conclusion

6.1 Summary

In Chapter 2, we derived a closed-form parameterization that allowed the generation of stratified samples according to a linear density function with triangular and tetrahedral support. The parameterization was constructed so that its Jacobian determinant is proportional to the density. Thus the stratification problem was reduced to one of inverting quadratic and cubic equations.

In Chapter 3, we described a new importance sampling strategy with the novel ability to draw samples from a dynamic steerable importance function. The steerability of the importance function restricted the generated samples to regions where the steering function is non-zero. We demonstrated its effectiveness in the context of direct illumination from distant light sources, where the incident all-frequency illumination is steered by a dynamically orientable positive cosine lobe that is a function of the local normal. The results clearly indicated the benefit of the technique, espcially when shading regions with normals pointing away from the bright portions of the incident

illumination.

In Chapter 4, we performed a Fourier analysis of finite aperture cameras and the depth of field effect in terms of operators that described light transport in the frequency domain light. The algorithm that we derived from this analysis showed a significant improvement over current techniques that correctly account for visibility.

Finally, in Chapter 5, we discussed a novel adaptation of standard statistical hypothesis tests for assessing and comparing Monte Carlo estimators. We showed that this framework could be used to make assertions about the means and/or variances of Monte Carlo estimators in image synthesis, upto a chosen level of significance. We verified that the inferences made using the framework, by comparison against standard, known results.

6.2 Future work

The stratification theory presented in Chapter 2 suggests that there is much to be explored. Stratified sampling research in image synthesis has been restricted to jittered sampling and developing suitable stratification schemes. The true benefit of stratification is realized as a combination of the stratification and allocation schemes. Optimal and cost-evaluated allocation schemes could significantly improveme the overall sampling efficiency.

The use of steerable importance sampling needs to be explored for other problems than direct illumination. The general notion of a dynamically varying importance function whose integral can be computed in constant time seems powerful enough for general use. One important consideration would be the representation for the functionals in the parameterized probability tree. A natural extension of the application presented

in Chapter 3 would be to include the reflectance function in the steering function.

Bandwidth prediction provides several key insights into functions that allows for efficient sampled representation and for deciding allocation schemes. The frequency propagation scheme—using sampled spectra—discussed in Chapter 4 needs to be explored for other applications than depth of field. The main challenge to be overcome seems to be that of efficiently and accurately representing occluder spectra although accuracy may not be an issue for conservative bandwidth prediction. The possible use of conservatively predicted bandwidth for driving allocation schemes for sampling indirect illumination seems exciting.

Although we have presented statistical hypothesis tests as a tool for assessing Monte Carlo estimators (see Chapter 5), it would be interesting to explore the potential utility of other sysems of inductive inference.

Bibliography

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284, 1985.

[2] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. In *Computer Graphics* Proceedings, ACM SIGGRAPH, pages 605–612, 2003.

[3] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. In J. Hodgins and J. C. Hart, editors, *Proceedings of ACM SIGGRAPH 2003*, volume 22(3) of *ACM Transactions on Graphics*, pages 605–612. ACM Press, 2003.

[4] and R.K. Friedrichs and H.Lewy. *Engl. transl. (1956) by P.Fox, ABC Computing Facility, Institute of Mathematical Sciences, New York University*, 48:246–251, 1928.

[5] J. Arvo. Backward ray tracing. In *SIGGRAPH '86 Developments in Ray Tracing course notes*. 1986. also appeared in SIGGRAPH '89 Radiosity course notes.

[6] J. Arvo. The irradiance Jacobian for partially occluded polyhedral sources. In *Computer Graphics* Proceedings, ACM SIGGRAPH, pages 343–350, July 1994.

[7] J. Arvo. Stratified sampling of spherical triangles, Nov. 13 1995.

[8] J. Arvo. Stratified sampling of 2-manifolds. In *State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis, SIGGRAPH 2001 Course Notes*, volume 29, Aug. 2001.

[9] J. Arvo. Stratified sampling of 2-manifolds, July 16 2001.

[10] J. Arvo and D. Kirk. Unbiased variance reduction for global illumination. In *Proceedings of the Second Eurographics Workshop on Rendering*, Barcelona, Spain, May 1991.

[11] M. Ashikhmin and P. Shirley. An anisotropic Phong BRDF model. *Journal of Graphics Tools*, 5(2):25–32, 2000.

[12] M. Ashikhmin and P. Shirley. Steerable illumination textures. *Transactions on Graphics*, 21(1):1–19, Jan. 2002.

[13] M. Ashikhmin and P. Shirley. Steerable illumination textures. *ACM Transactions on Graphics*, 21(1):1–19, 2002.

[14] B. A. Barsky, D. R. Horn, S. A. Klein, J. A. Pang, and M. Yu. Camera models and optical systems used in computer graphics: Part II, image based techniques. In *International Conference on Computational Science and its Applications*, 2003.

[15] A. J. Bayes. A minimum variance sampling technique for simulation models. *Journal of the ACM*, 19(4):734–741, Oct. 1972.

[16] J. Bethel. Minimum variance estimation in stratified sampling. *Journal of the American Statistical Association*, 84(405):260–265, 1989.

[17] Broder. How hard is it to marry at random? (on the approximation of the permanent). In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1986.

[18] D. Burke, A. Ghosh, and W. Heidrich. Bidirectional importance sampling for illumination from environment maps. In *ACM SIGGRAPH 2004 Sketches*. ACM Press, 2004.

[19] D. Burke, A. Ghosh, and W. Heidrich. Bidirectional importance sampling for direct illumination. In K. Bala and P. Dutré, editors, *Eurographics Symposium on Rendering*, pages 147–156, Konstanz, Germany, 2005. Eurographics Association.

[20] E. Camahort, A. Lerios, and D. Fussell. Uniformly sampled light fields. In *Rendering Techniques '98 (Proc. of EG Workshop on Rendering '98)*, pages 117–130. Eurographics, 1998.

[21] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong. Plenoptic sampling. In *Computer Graphics* Proceedings, Annual Conference Series, pages 307–318. ACM SIGGRAPH, 2000.

[22] S. Chandrasekar. *Radiative Transfer*. Oxford University Press, 1950.

[23] P. Clarberg, W. Jarosz, T. Akenine-Möller, and H. W. Jensen. Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Transactions on Graphics*, 24(3):1166–1175, July 2005.

[24] P. Clarberg, W. Jarosz, T.-A. Moller, and H. W. Jensen. Wavelet importance sampling: Efficiently evaluating products of complex functions. In *ACM TOG*, pages 1166–1175, Aug. 2005.

[25] D. Cline, P. K. Egbert, J. F. Talbot, and D. L. Cardon. Two stage importance sampling for direct lighting. In T. Akenine-Möller and W. Heidrich, editors, *Eurographics Workshop/ Symposium on Rendering*, pages 103–113, Nicosia, Cyprus, 2006. Eurographics Association.

[26] D. Cline, P. K. Egbert1, J. F. Talbot, and D. L. Cardon. Two stage importance sampling for direct lighting. *Proc. Eurographics Symposium on Rendering (EGSR'06)*, 2006.

[27] W. G. Cochran. *Sampling Techniques*. John Wiley & Sons, Inc., New York, 1977.

[28] R. L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, Jan. 1986.

[29] R. L. Cook. Stochastic sampling in computer graphics. *ACM TOG*, 5(1):51–72, 1986.

[30] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (Proc. SIGGRAPH 84)*, 18(3):137–145, July 1984.

[31] R. C. Corlett. Direct Monte Carlo Calculation of Radiative Heat Transfer in Vacuum. *Journal of Heat Transfer*, pages 376–382, 1966.

[32] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Computer Graphics* Proceedings, ACM SIGGRAPH, pages 369–378, Aug. 1997.

[33] K. Devlin, A. Chalmers, A. Wilkie, and W. Purgathofer. Tone reproduction and physically based spectral rendering, 2002.

[34] M. A. Z. Dippe and E. H. Wold. Antialiasing through stochastic sampling. *CG*, 19(3):69–78, July 1985.

[35] Y. Dobashi, K. Kaneda, H. Nakatani, and H. Yamashita. A quick rendering method using basis functions for interactive lighting design. *Computer Graphics Forum*, 14(3):229–240, 1995.

[36] F. Durand, N. Holzschuch, C. Soler, E. Chan, and F. X. Sillion. A frequency analysis of light transport. *ACM Transactions on Graphics*, 24(3):1115–1126, Aug. 2005.

[37] P. Dutre and Y. D. Willems. Importance-driven monte carlo light tracing. In *Photorealistic Rendering Techniques (Proceedings of the Fifth Eurographics Workshop on Rendering)*, pages 188–200, New York, 1994. Springer-Verlag.

[38] Dyer, Frieze, and Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *JACM: Journal of the ACM*, 38, 1991.

[39] G. F. Evans and M. D. McCool. Stratified wavelength clusters for efficient spectral monte carlo rendering. In *Graphics Interface '99*, San Francisco, CA, June 1999. Morgan Kaufmann.

[40] R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, 1925.

[41] R. A. Fisher. *Statistical Methods and Scientific Inference*. Oliver and Boyd, second edition, 1959.

[42] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[43] J. E. Freund and R. E. Walpole. *Mathematical Statistics*. Prentice-Hall, Englewood Cliffs, New Jersey, fourth edition, 1987.

[44] L. J. Gallaher. A multidimensional monte carlo quadrature with adaptive stratified sampling. *Commun. ACM*, 16(1):49–50, 1973.

[45] A. Ghosh, A. Doucet, and W. Heidrich. Sequential sampling for dynamic environment map illumination. *Proc. Eurographics Symposium on Rendering (EGSR'06)*, 2006.

[46] J. S. Gondek, G. W. Meyer, and J. G. Newman. Wavelength dependent reflectance functions. *Computer Graphics*, 28(Annual Conference Series):213–220, July 1994.

[47] C. Gotsman. Constant-Time filtering by singular value decomposition. In M. F. Cohen, C. Puech, and F. Sillion, editors, *Fourth Eurographics Workshop on Rendering*, pages 145–156, 1993.

[48] D. Hart, P. Dutre, and D. P. Greenberg. Direct illumination with lazy visibility evaluation. In *Computer Graphics* Proceedings, Annual Conference Series, pages 147–154. ACM SIGGRAPH, Aug. 1999.

[49] N. J. and P. E.S. *On the Use and Interpretation of Certain Test Criteria for Purposes of Statistical Inference, Parts I, II*. Biometrika, 1928.

[50] N. J. and P. E.S. *On the Problem of the Most Efficient Tests of Statistical Hypotheses*. Philosophical Transactions of the Royal Society of London, 1933.

[51] B. Jourdain and P. Etore. Adaptive optimal allocation in stratified sampling methods. 2007.

[52] J. T. Kajiya. The rendering equation. *CG*, 20(4):143–150, Aug. 1986.

[53] Karp and Luby. Monte-carlo algorithms for the planar multiterminal network reliability problem. *COMPLEXITY: Journal of Complexity*, 1985.

[54] A. Kaur, G. P. Patil, S. J. Shirk, and C. Taillie. Environmental sampling with a concomitant variable: A comparison between ranked set sampling and stratified simple random sampling. *Journal of Applied Statistics*, 23(2/3):231–255, June 1996.

[55] D. Kim and H.-S. Ko. Eulerian motion blur. In D. Ebert and S. Merillou, editors, *Natural Phenomena*, pages 39–46, Prague, Czech Republic, 2007. Eurographics Association.

[56] J. Kim. Estimation of optimality gap using stratified sampling. *Applied Mathematics and Computation*, 171(2):710–720, 2005.

[57] D. Kirk and J. Arvo. Unbiased sampling techniques for image synthesis. *CG*, 25(4):153–156, July 1991.

[58] D. Kirk and J. Arvo. Unbiased variance reduction for global illumination. In *Proceedings of the Second Eurographics Workshop on Rendering*, Barcelona, May 1991.

[59] D. B. Kirk and J. Arvo. Unbiased sampling techniques for image synthesis. In T. W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 153–156, July 1991.

[60] C. Kolb, P. M. Hanrahan, and D. Mitchell. A realistic camera model for computer graphics. In *Computer Graphics* Proceedings, Annual Conference Series, pages 317–324. ACM SIGGRAPH, Aug. 1995.

[61] T. Kollig and A. Keller. Efficient multidimensional sampling. *Computer Graphics Forum*, 21(3):557–557, 2002.

[62] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[63] J. Lawrence, S. Rusinkiewica, and R. Ramamoorthi. Adaprive numerical cumulative distribution functions for efficient importance sampling. In *EGSR 2005*, pages 11–20, 2005.

[64] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi. Efficient brdf importance sampling using a factored representation. *ACM Trans. Graph.*, 23(3):496–505, 2004.

[65] M. E. Lee, R. A. Redner, and S. P. Uselton. Statistically optimized sampling for distributed ray tracing. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 61–67, July 1985.

[66] M. E. Lee, R. A. Redner, and S. P. Uselton. Statistically optimized sampling for distributed ray tracing. *CG*, 19(3):61–68, July 1985.

[67] M. D. McCool and P. K. Harwood. Probability trees. In W. A. Davis, M. Mantei, and R. V. Klassen, editors, *Graphics Interface '97*, pages 37–46. Canadian Information Processing Society, Canadian Human-Computer Communications Society, May 1997. ISBN 0-9695338-6-1 ISSN 0713-5424.

[68] X. Mei, M. Jaeger, and B.-G. Hu. An effective stratified sampling scheme for environment maps with median cut method. In *CGIV*, pages 384–389. IEEE Computer Society, 2006.

[69] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44:335–341, 1949.

[70] D. P. Mitchell. Generating antialiased images at low sampling densities. In M. C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 65–72, July 1987.

[71] D. P. Mitchell. Spectrally Optimal Sampling for Distribution Ray Tracing. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, volume 25, pages 157–164, July 1991.

[72] D. P. Mitchell. Consequences of stratified sampling in graphics. In *SIGGRAPH*, pages 277–280, 1996.

[73] D. Nehab and P. Shilane. Stratified point sampling of 3D models. In M. Gross, H. Pfister, M. Alexa, and S. Rusinkiewicz, editors, *Symposium on Point-Based Graphics*, pages 49–56, Zürich, Switzerland, 2004. Eurographics Association.

[74] L. Neumann, A. Neumann, and L. Szirmay-Kalos. Reflectance models with fast importance sampling. In D. Duke, S. Coquillart, and T. Howard, editors, *Computer Graphics Forum*, volume 18(4), pages 249–265. Eurographics Association, 1999.

[75] J. Neyman. On two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society, Series B*, 97:558–606, 1934.

[76] J. S. Nimeroff, E. Simoncelli, and J. Dorsey. Efficient Re-rendering of Naturally Illuminated Environments. In *Fifth Eurographics Workshop on Rendering*, pages 359–373, Darmstadt, Germany, 1994. Springer-Verlag.

[77] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23(3):488–495, Aug. 2004.

[78] P. Perona. Deformable kernels for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):488–499, 1995.

[79] G. Pietrek and I. Peter. Adaptive wavelet densities for monte carlo ray tracing. In V. Skala, editor, *WSCG'99 Conference Proceedings*, 1999.

[80] A. Podgurski and C. Yang. Partition testing, stratified sampling, and cluster analysis. In *SIGSOFT FSE*, pages 169–181, 1993.

[81] M. Potmesil and I. Chakravarty. A lens and aperture CAM model for synthetic image generation. volume 15, pages 297–305, Aug. 1981.

[82] M. Potmesil and I. Chakravarty. Modelling motion blur in computer-generated images. volume 17, pages 389–399, July 1983.

[83] R. W. Preisendorfer. *Hydrologic Optics, Volumes I-VI.* National Oceanic and Atmospheric Administration, Honolulu, HI, 1976.

[84] W. Purgathofer. A statistical method for adaptive stochastic sampling. In A. Requicha, editor, *Proceedings of Eurographics 86*, pages 145–152. Elsiver, North-Holland, 1986.

[85] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Computer Graphics* Proceedings, ACM SIGGRAPH, pages 497–500, 2001.

[86] C. R. Rao. *Linear statistical inference and its applications.* John Wiley & Sons, NY, 1965.

[87] L. J. Rayleigh. On James Bernoulli's theorem in probabilities. *Philosophical Magazine*, 48:246–251, 1899.

[88] R. Y. Rubinstein. *Simulation and the Monte Carlo Method.* John Wiley & Sons, NY, 1981.

[89] R. Rush, J. M. Mulvey, J. E. Mitchell, and T. R. Willemain. Stratified filtered sampling in stochastic optimization. *Journal of Applied Mathematics and Decision Sciences*, 4(1):17–38, 2000.

[90] S. Rusinkiewicz. A new change of variables for efficient BRDF representation. In G. Drettakis and N. Max, editors, *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)*, pages 11–22, New York, NY, 1998. Springer Wien.

[91] M. Shinya, T. Takahashi, and S. Naito. Principles and applications of pencil tracing. *Computer Graphics (Proc. SIGGRAPH '87)*, 21(4), 1987.

[92] P. Shirley. Discrepancy as a quality measure for sample distributions. In *Eurographics '91*, pages 183–94. Elsevier Science Publishers, Amsterdam, North-Holland, 1991.

[93] P. Shirley. Physically Based Lighting Calculations for Computer Graphics: A Modern Perspective. In K. Bouatouch and C. Bouville, editors, *Photorealism in Computer Graphics (Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics, 1990)*, pages 67–81, 1992.

[94] P. Shirley, C. Wang, and K. Zimmerman. Monte Carlo methods for direct lighting calculations. *ACM TOG*, 15(1):1–36, Jan. 1996.

[95] P. Shirley, C. Wang, and K. Zimmerman. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics*, 15(1):1–36, 1996.

[96] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *International Conference on Image Processing*, volume 3, pages 444–447, 23-26 Oct. 1995, Washington, DC, USA, 1995.

[97] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE transactions on informations theory*, 38(2), 1992.

[98] B. E. Smits, J. R. Arvo, and D. H. Salesin. An importance-driven radiosity algorithm. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 273–282, New York, NY, USA, 1992. ACM.

[99] C. Soler, K. Subr, F. Durand, N. Holzschuch, and F. Sillion. Fourier depth of field. In *ACM Transactions on Graphics (under review)*, 2008.

[100] J. Spanier and M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, Reading, MA, 1969.

[101] M. Stark, P. Shirley, and M. Ashikhmin. Generation of stratified samples for B-spline pixel filtering. *Journal of Graphics Tools: JGT*, 10(1):39–48, 2005.

[102] K. Subr and J. Arvo. Statistical hypothesis testing for assessing monte carlo estimators: Applications to image synthesis. In *PG '07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 106–115, Washington, DC, USA, 2007. IEEE Computer Society.

[103] K. Subr and J. Arvo. Steerable importance sampling. In *IEEE Symposium on Interactive Raytracing*, pages 133–140, 2007.

[104] L. Szecsi, M. Sbert, and L. Szirmay-Kalos. Combined correlated and importance sampling in direct light source computation and environment mapping. *Computer Graphics Forum (Eurographics 2004 Proceedings)*, 23(3), sep 2004. To appear.

[105] L. Szécsi, L. Szirmay-Kalos, and C. Kelemen. Variance reduction for russian-roulette. In V. Skala, editor, *Journal of WSCG*, volume 11, feb 2003.

[106] L. Szirmay-Kalos, B. Csebfalvi, and W. Purgathofer. Importance driven quasi-random walk solution of the rendering equation. In V. Skala, editor, *WSCG'98 Conference Proceedings*, 1998.

[107] P. C. Teo, E. P. Simoncelli, and D. J. Heeger. Efficient linear re-rendering for interactive lighting design. Technical Report STAN-CS-TN-97-60, Stanford, CA, 1997.

[108] H. Tropf and H. Herzog. Multidimensional range search in dynamically balanced trees. In *Angewandte Informatik*, pages 71–77, 1981.

[109] A. Tschuprow. 0n the mathematical expectation of the moments of frequency distributions in the case of correlated observations. *Metron*, 2:461–493, 646–683, 1923.

[110] R. Valliant. Conditional properties of some estimators in stratified sampling. *Journal of the American Statistical Association*, 82(398):509–519, 1987.

[111] R. Valliant. Generalized variance functions in stratified two-stage sampling. *Journal of the American Statistical Association*, 82(398):499–508, 1987.

[112] E. Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, New York, NY, USA, 1995. ACM Press.

[113] B. Walter. Notes on the Ward BRDF. In *Technical Report, PCG-05-06CG*, New York, NY, USA, 2005.

[114] B. Walter, A. Arbree, K. Bala, and D. P. Greenberg. Multidimensional lightcuts. *ACM Transactions on Graphics*, 26(3):1081–1088, 2006.

[115] C.-M. Wang and N.-C. Hwang. A stratified sampling technique for an ellipse. *Journal of Graphics Tools: JGT*, 9(1):13–22, 2004.

[116] G. J. Ward. Measuring and modeling anisotropic reflection. *CG*, 26(2):265–272, July 1992.

[117] G. J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 265–272, New York, NY, USA, 1992. ACM Press.

[118] A. B. Watson and J. Albert J. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Am. A*, 2(2):322, 1985.

[119] A. Wilkie, R. Tobler, and W. Purgathofer. Raytracing of dispersion effects in transparent materials, 2000.

[120] J. Yellot. Spectral consequences of photoreceptor sampling in the rhesus retina. *Science*, 221:382–385, 1983.