Introduction to Sampling Strategies for Efficient Monte Carlo Image Synthesis

Kartic Subr

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

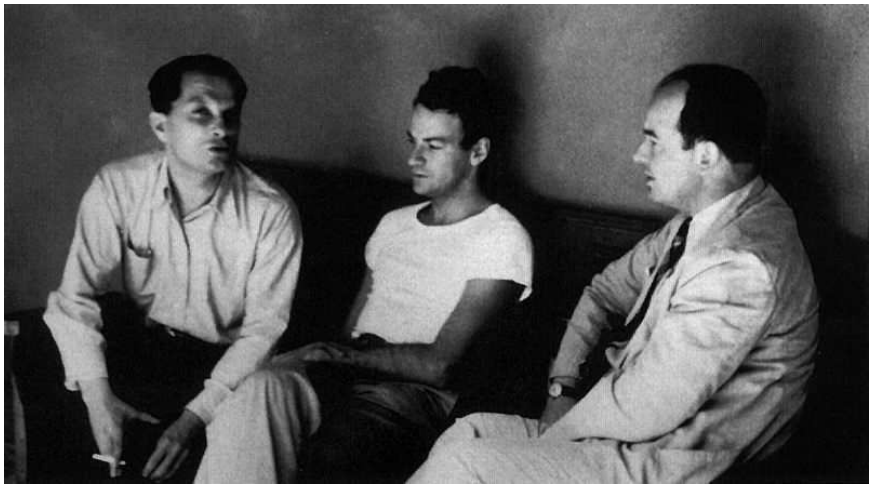


Figure 1.3: *Stanislaw M. Ulam, Richard P. Feynman and John von Neumann. The Monte Carlo method was inspired by the problems encountered conducted during the development of the atomic bomb. (Picture scanned at the American Institute of Physics)*

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.

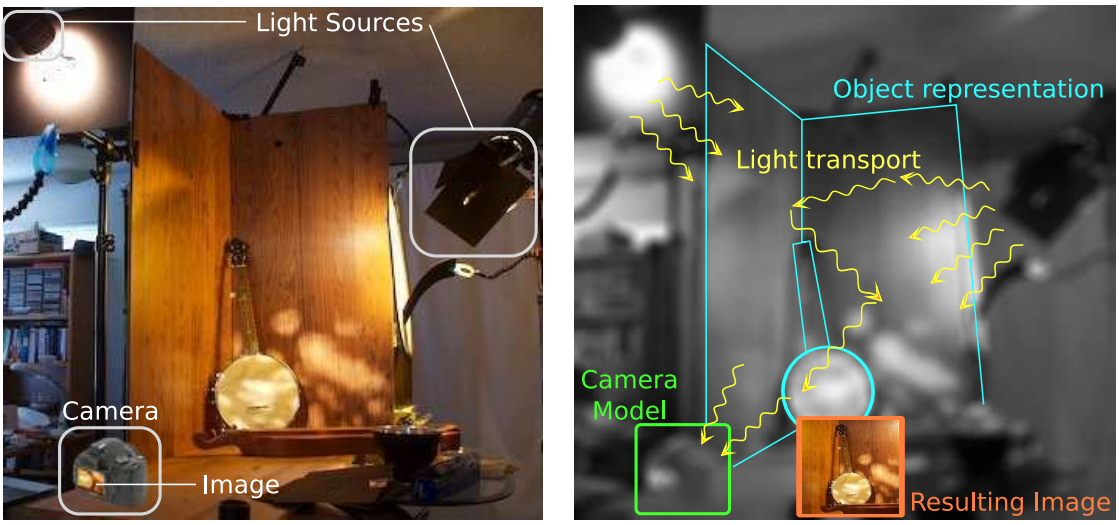


Figure 1.4: *Physically based mage synthesis is the process of producing images by simulation of light transport to mimic the photographic process. Suitable models are chosen for the camera and objects in the scene.*

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

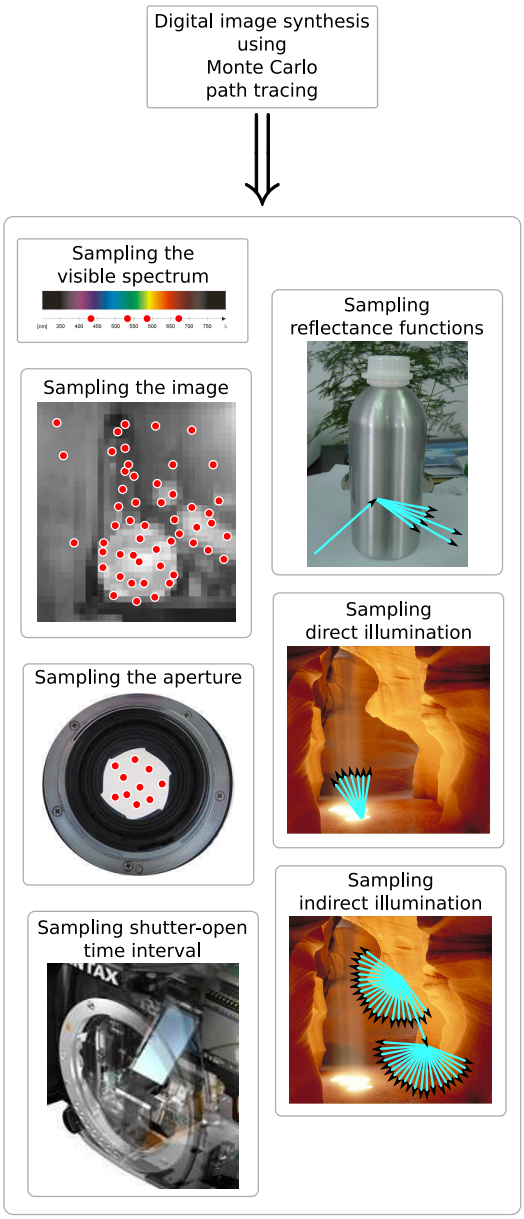
[5]Sampling error is the degree to which a sample might differ from the population.

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.