THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Analysis of Sample Correlations for Monte Carlo Rendering

Citation for published version:
Singh, G, Oztireli, C, Ahmed, A, Coeurjolly, D, Subr, K, Deussen, O, Ostromoukhov, V, Ramamoorthi, R & Jarosz, W 2019, 'Analysis of Sample Correlations for Monte Carlo Rendering', *Computer Graphics Forum*, vol. 38, no. 2, pp. 473-491. https://doi.org/10.1111/cgf.13653

Digital Object Identifier (DOI):
10.1111/cgf.13653

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Computer Graphics Forum

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 01. May. 2024

EUROGRAPHICS 2019
A. Giachetti and H. Rushmeier
(Guest Editors)

Volume 38 (2019), Number 2
STAR – State of The Art Report

Analysis of Sample Correlations for Monte Carlo Rendering

Gurprit Singh[1] Cengiz Öztireli[2] Abdalla G.M. Ahmed[3] David Coeurjolly[4]
Kartic Subr[5] Oliver Deussen[6] Victor Ostromoukhov[4] Ravi Ramamoorthi[7] Wojciech Jarosz[8]

[1]Max-Planck Institute for Informatics, Saarbrücken, [2]Disney Research, Zurich, [3]KAUST, Saudi Arabia, [4]Université de Lyon / CNRS, France, [5]University of Edinburgh, UK, [6]University of Konstanz, Germany, [7]University of California, San Diego, USA, [8]Dartmouth College, USA

Abstract

Modern physically based rendering techniques critically depend on approximating integrals of high dimensional functions representing radiant light energy. Monte Carlo based integrators are the choice for complex scenes and effects. These integrators work by sampling the integrand at sample point locations. The distribution of these sample points determines convergence rates and noise in the final renderings. The characteristics of such distributions can be uniquely represented in terms of correlations *of sampling point locations. Hence, it is essential to study these correlations to understand and adapt sample distributions for low error in integral approximation. In this work, we aim at providing a comprehensive and accessible overview of the techniques developed over the last decades to analyze such correlations, relate them to error in integrators, and understand when and how to use existing sampling algorithms for effective rendering workflows.*

CCS Concepts
• ***Mathematics of computing*** → ***Computation of transforms;*** *Stochastic processes; Number-theoretic computations;* • ***Computing methodologies*** → ***Ray tracing;***

1. Introduction

Over the past decade, Monte Carlo (MC) integration has revolutionized the rendering pipeline within the film and animation industries. This has led to stunningly realistic visual effects by simulating the physics of light. MC-based numerical integration offers an elegant approximation of high-dimensional integrals that are encountered during simulation of the physics of radiant light energy propagating within virtual environments. Many production renderers such as Arnold [GIF*18] or Renderman [CFS*18] generate estimates of the integrals by evaluating integrands at a number of *sample* locations. When N independent random samples are used, the error converges at the rate of $\mathcal{O}(N^{-0.5})$, and this error manifests itself as an undesirable grainy appearance (noise) in the rendered images. Since the integration technique is inherently parallelizable, a common strategy to mitigate approximation error is to invest in large computational resources (CPUs/GPUs). For realistically modeled virtual environments, random sampling is still typically considered prohibitively expensive. This motivates the need for *effective sampling strategies* that lead to a reduction of approximation error.

A popular approach to mitigate error operates by incorporating dependencies, or correlations, between the sampled locations. Such correlations often lead to observable patterns, when visualized, as shown in Figure 1. Two key hurdles in designing such strategies are: understanding and identifying correlations that result in a reduction of error; and designing efficient algorithms to generate samples with the required correlations. Classical statistical measures such as probability density are not very useful in capturing these correlations: all strategies shown in Figure 1 represent samples drawn from the same (constant) distribution. i.e. the expected number of samples within any fixed subdomain will be the same for each strategy.

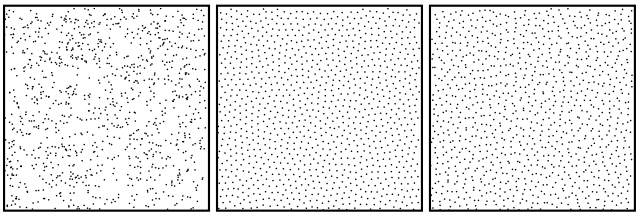


Figure 1: *Each stochastic point set exhibits different correlation structure despite having derived from a constant probability density function.*

As a result, traditionally, practitioners of computer graphics utilize a different set of statistical measures for qualitative and quantitative analysis of error in rendering, such as spectral diagrams or spatial statistics. As observed in the literature, such measures can be interpreted as analyzing stochastic structures represented by point samples. This observation naturally leads to utilizing ideas and techniques from the field of *stochastic point processes* [MW04].

Our focus is to review proposed point pattern quality measures

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

in the framework of point processes to understand how complex correlations affect error in integral estimation for rendering. The reader is going to gain an understanding of:

- existing statistical measures for point patterns with correlations, how they relate to each other, and when and how to use a particular measure for quantitative or qualitative analysis.
- how error in rendering is affected by point patterns in relation to these measures, with exact expressions for bias and variance of the integral estimators.
- what practical sampling algorithms exist, what the trade-offs are, and how to choose a sampling technique in light of the theory.

2. Preliminaries

We start with a brief overview of the setting considered, and the theory of stochastic point processes from a rendering perspective. We cover the theoretical underpinnings on an accessible level, without going into extensive technicalities[†]. This will form the basis of the analysis techniques in the next sections. All the notations used in the following sections can be found in Table 1.

2.1. Estimating Integrals in Rendering

For a *real* function $f : \mathbb{R}^d \to \mathbb{R}^+$ representing a certain part of light transport in d dimensions, rendering requires us to estimate its integral to compute the color of each pixel. We hence would like to study the error in estimating the integral $I := \frac{1}{|V|}\int_V f(\vec{x})\mathrm{d}\vec{x}$, where V is the domain of the function f where $\forall \vec{x} \notin V, f(\vec{x}) = 0$, and $|V|$ is its volume. The estimator for this integral can be written as

$$\hat{I} := \sum_{j=1}^{N} w(\vec{x}_j) f(\vec{x}_j), \tag{1}$$

for some positive weights $w(\vec{x}_j)$ and sample points $\vec{x}_j$. For a fixed sample count N, $w(\vec{x}_j) = 1/(Np(\vec{x}_j))$, where $p(\vec{x}_j)$ is the probability density function (PDF) used to distribute the samples. The goal is to study the error in this estimator in relation to the distribution of the sample points $\vec{x}_j$. In order to quantify the error, we can utilize the theory of point processes. Note that, since we are interested in analyzing the impact of correlated samples on integral estimation, we refer to such estimators as sampling-based estimators. We reserve the term "MC estimator" for pseudo-random MC samples (with no correlations).

2.2. Stochastic Point Processes

A point process is a generating process, e.g. an algorithm or a natural process, for a set of point distributions with common characteristics. Each distribution generated by a point process can be regarded as a realization or instance of that point process. In order to characterize a point process, we need to compute statistics that capture correlations of sample points in all realizations. To quantify such correlations, the study of point processes starts by assigning a random variable $X(\mathcal{B})$ to each Borel set $\mathcal{B} \in V$ for a given domain V. This implies that we actually need *infinitely many* random variables to characterize a given point process. Each of these random variables is assigned to one of the infinitely many sets in the given domain.

In order to make this analysis more tractable, we can first assume that we select a number of sets $\mathcal{B}_1$ to $\mathcal{B}_N$. For these random variables, there exist the joint probability $\mathbb{P}(X(\mathcal{B}_1) \leq b_1, \cdots, X(\mathcal{B}_N) \leq b_N)$ over b_i sub-regions. A point process can be formally described by characterizing this joint probability for *all* different N and group of sets $\mathcal{B}_i$. The most commonly used and familiar random variable is the number of points $N(\mathcal{B})$ that fall into the set $\mathcal{B}$ (see Figure 2). This is indeed a random variable: for each realization of the point process, i.e. a point distribution, this number changes for a fixed set $\mathcal{B}$. Hence, by generating different distributions from the same point process,

Table 1: *Here we summarize all the notations used in the paper.*

Symbol	Description
$\mathcal{P}$	a point process
$\mathbb{P}(\cdot)$	joint probability of $\mathcal{P}$
$\lambda(x)$	first order product density or intensity of $\mathcal{P}$
$\rho(x_1, x_2)$	second order product density of $\mathcal{P}$
$\mathbb{E}[\cdot]$	expectation operator
$var[\cdot]$	variance operator
$f(\cdot)$	light transport function
$s(\cdot)$	sampling function
$w(\cdot)$	weighting function
V	sampling domain
I	integral of a function
$\hat{I}$	Monte Carlo estimator of I
$g(\cdot)$	pair correlation function
$\hat{g}$	Monte Carlo estimator of g
$\mathcal{B}$	Borel set
B	a set of high-dimensional rectangles
$\mathrm{card}(\cdot)$	cardinality of a set
D_N^*	star discrepancy of $\mathcal{P}$
$\mathcal{V}(\cdot)$	total variation of a function
$\mathcal{F}(\nu)$	Fourier transform coefficient at *real* frequency ν
$\mathbf{g}_m$	Fourier series coefficient of g at m-th integer frequency
$\mathbf{f}_m$	Fourier series coefficient of f at m-th integer frequency
$\mathbf{s}_m$	Fourier series coefficient of s at m-th integer frequency

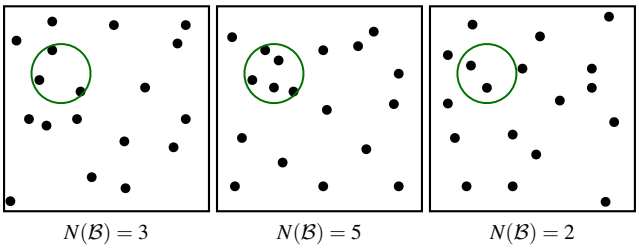


Figure 2: *For a given set $\mathcal{B}$, $N(\mathcal{B})$ gives the number of points that fall into $\mathcal{B}$ for different point distributions generated by an underlying point process. By generating more distributions from the same point process, we can estimate the PDF of $N(\mathcal{B})$.*

† For a complete theoretical exposition on point processes, we refer the reader to excellent books on point processes [MW04, IPSS08].

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

we can compute the PDF of $N(\mathcal{B})$. Similarly, we can consider the joint probability and corresponding PDF of $N(\mathcal{B}_1), \cdots, N(\mathcal{B}_N)$.

2.3. Point Process Statistics

2.3.1. Product Densities

Instead of considering finite sets and the associated probabilities, we can define infinitesimal sets and hence work with probability densities. Let $\vec{x}_j$ denote an arbitrary point in $\mathbb{R}^d$, and $\mathcal{B}_j$ denotes an infinitesimal sphere centered at this point with volume $\mathrm{d}\vec{x}_j = |\mathcal{B}_j|$. We can then define $\mathbb{P}(\vec{x}_1, \cdots, \vec{x}_N)$ as the joint *probability of having a point* of a point process $\mathcal{P}$ in each of the infinitesimal spheres $\mathcal{B}_j$. This probability is typically expressed in terms of the kth order *product density* $\rho^{(k)}$ with $\mathbb{P}(\vec{x}_1, \cdots, \vec{x}_N) := \rho^{(k)}(\vec{x}_1, \cdots, \vec{x}_N)\mathrm{d}\vec{x}_1 \cdots \mathrm{d}\vec{x}_n$. The product densities are thus simply the PDF functions for joint probabilities of these random variables, which can be shown to uniquely define the point process $\mathcal{P}$.

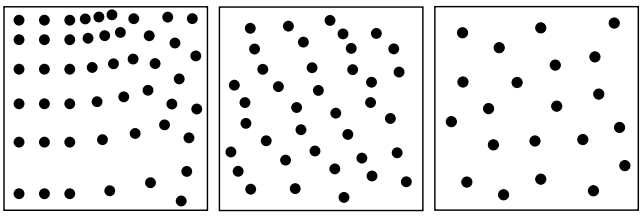
We will see in the next sections that for our purpose of using point processes for analyzing error in integral estimation, it is sufficient to consider product densities of first and second order $\rho^{(1)}$, $\rho^{(2)}$. The first order product density is simply given by $\rho^{(1)}(\vec{x})\mathrm{d}\vec{x} = \mathbb{P}(\vec{x})$. The expected number of points in a set $\mathcal{B}$ is given by $\mathbb{E}_{\mathcal{P}}[N(\mathcal{B})] = \int_{\mathcal{B}} \rho^{(1)}(\vec{x})\mathrm{d}\vec{x}$: with the expectation computed over different realizations (distributions) of a point process $\mathcal{P}$. Hence, $\rho^{(1)}(\vec{x})$ measures the local expected density of points generated by this point process. It is thus usually called the *intensity* of $\mathcal{P}$, and denoted by the symbol $\lambda(\vec{x}) := \rho^{(1)}(\vec{x})$.

We can then similarly define the second order product density $\rho^{(2)}(\vec{x}, \vec{y})\mathrm{d}\vec{x}\mathrm{d}\vec{y} = \mathbb{P}(\vec{x}, \vec{y})$. As we do not need higher order product densities, we denote this product density simply with $\rho(\vec{x}, \vec{y})$. The second order product density gives us the joint probability of finding a pair of points at certain locations in space. It can be estimated by generating infinitely many distributions from the point process, and counting the number of point pairs with each of the two points in the pair falling into one of the volumes around $\vec{x}$ or $\vec{y}$.

2.3.2. Stationary and Isotropic Processes

Although the literature on point processes is very extensive, it gracefully matured in two very important subclasses: stationary and isotropic processes. *Stationary point processes* are processes that are translation invariant, i.e. the point distributions generated by such point processes will have the same statistics regardless of where we look at in space. We show an example point distribution generated by a stationary point process in Figure 3b. For stationary point processes, the intensity becomes a constant $\lambda(\vec{x}) = \lambda$, and the second order product density is a function of the *difference* between locations of points in space $\rho(\vec{x}, \vec{y}) = \rho(\vec{x} - \vec{y})$. This assumption is what makes most of the statistics encountered in the rendering literature meaningful, as we will elaborate on in the next sections. The second order product density is typically given by the normalized *pair correlation function* (PCF) g where $\rho(\vec{x} - \vec{y}) = \lambda^2 g(\vec{x} - \vec{y})$.

We can further assume that a point process is translation *and* rotation invariant (*isotropic*), meaning that the characteristics of distributions generated by that point process do not depend on where we look, and how we are oriented in space. In other words, you may freely rotate the point distributions and will always get the same characteristics (Figure 3c). For the isotropic case, $\lambda(\vec{x}) = \lambda$, and $\rho(\vec{x}, \vec{y}) = \rho(\|\vec{x} - \vec{y}\|)$, and thus $g(\vec{x} - \vec{y}) = g(\|\vec{x} - \vec{y}\|)$. Hence, instead of considering the high dimensional $\rho(\vec{x}, \vec{y})$ statistic for general point processes, we can now simply work with the 1D statistic $g(\|\vec{x} - \vec{y}\|)$ to describe the second order correlations of such processes. This is a significant simplification, especially when we deal with point distributions in high dimensional domains. In later sections, we show how state-of-the-art frameworks developed over the years in the rendering literature leverage these two special cases to gain theoretical and practical insights for rendering, even when more complex distributions are needed.



(a) (b) (c)

Figure 3: *(a) A general point process generates point distributions with spatially varying characteristics. (b) A stationary point process has translation invariant characteristics. (c) An isotropic point process further generates rotation invariant point distributions so that we may translate or rotate the distributions without affecting their properties.*

2.4. Error in Integral Estimation

We utilize the theory and statistics above to study the error in the sampling-based estimator in Eqn.(1). The error in this estimator consists of a bias $bias_{\mathcal{P}}[\hat{I}] := I - \mathbb{E}_{\mathcal{P}}[\hat{I}]$, and a variance $var_{\mathcal{P}}[\hat{I}] := \mathbb{E}_{\mathcal{P}}[\hat{I}^2] - (\mathbb{E}_{\mathcal{P}}[\hat{I}])^2$ term, where the expectations $\mathbb{E}_{\mathcal{P}}[\cdot]$ are over point distributions generated by the underlying point process $\mathcal{P}$. For brevity, we drop the subscript $\mathcal{P}$ and represent the expectation operator as $\mathbb{E}[\cdot]$ and the variance operator as $var[\cdot]$.

The goal is then to minimize both of these terms simultaneously via point distributions generated by point processes with optimized characteristics. This can be done both qualitatively, e.g. by looking at the graphs of statistics, or quantitatively by steering the characteristics so as to minimize the terms of this error. For both cases, we first need to have meaningful and informative analysis tools, which we review in Section 3.

2.5. Frequency Space

Analyzing random processes in the Fourier realm has a long history [CM67,Uns00]. However, only recently researchers have started emphasizing the importance of Fourier statistics in predicting error during estimation. There exists an intimate relation between the spatial statistics and the frequency domain transforms. For example, spatial statistics are best captured by first and second order correlations. The second order correlations are directly computed from the pair-wise distances (differentials) between each pair of samples. It is

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

straightforward to show that the cosine transform of the differentials gives the corresponding *Fourier power spectrum*. In this section, we briefly introduce the Fourier transform and its periodic counterpart the Fourier series. Later in Section 3.2, we use these basic entities to represent spectral tools like the *power spectrum*.

2.5.1. Fourier Transform

One of the foundations of frequency analysis is the Fourier transform. For any given function $h(\vec{x})$ defined over an infinite (*real*) domain, the continuous Fourier transform:

$$\mathcal{F}_h(\vec{\nu}) := \int_{\mathbb{R}^d} h(\vec{x}) \mathrm{e}^{-i2\pi(\vec{\nu}\cdot\vec{x})} \mathrm{d}\vec{x}, \tag{2}$$

decomposes that function into a continuous sum (integral) of sinusoids of continuous and unbounded set of frequencies $\vec{\nu}$. The set of continuous complex coefficients arising from this decomposition is called the *Fourier spectrum* of the function.

2.5.2. Fourier series

In practice, however, the sampling/integration domain is always finite. In the Fourier domain, these finite domains can be handled by treating them as periodic over a toroidal domain. Under this assumption, the integrand and sampling function may be seen as periodic functions with periodicity given by the extent of the domain (which is unity for a unit hypercube). This leads to a discrete spectrum with the Fourier series coefficients:

$$\mathbf{f}_{\vec{m}} := \int_{[0,1]^d} f(\vec{x})\, \mathrm{e}^{-i2\pi(\vec{m}\cdot\vec{x})} \mathrm{d}\vec{x}, \tag{3}$$

defined over integer frequencies $\vec{m} \in \mathbb{Z}^d$ with $\mathbf{f}_m \in \mathbb{C}^d$ (complex space). The corresponding d-dimensional, unit-periodic function $f(\vec{x})$ can be expanded in terms of its Fourier series coefficients as:

$$f(\vec{x}) = \sum_{\vec{m}\in\mathbb{Z}^d} \mathbf{f}_{\vec{m}}\, \mathrm{e}^{i2\pi(\vec{m}\cdot\vec{x})}. \tag{4}$$

These Fourier series coefficients are translation invariant over the period of the domain, which is unity $[0,1]^d$. Following Singh et al. [SSC*19], we represent all upcoming frequency space formulations using the Fourier series. For brevity sake, we safely omit the vector notation $(\vec{\cdot})$, whenever possible.

3. Equidistribution Measures

In order to study error in integral estimation with respect to correlations, we need to define practical measures that capture the theoretical statistics as introduced in the last section. We expect that the resulting measures provide valuable insights into the sampling patterns and thus allow us to understand, and intelligently choose and adapt sampling patterns according to the particular rendering problem. Several measures for this purpose have been proposed in the last decades. We group these measures into two broad categories: spatial measures, and spectral domain measures. We explain that these are actually fundamentally related (Section 3.2), and that one needs to be careful about the assumptions on the point patterns considered when choosing and interpreting these measures (Section 3.3).

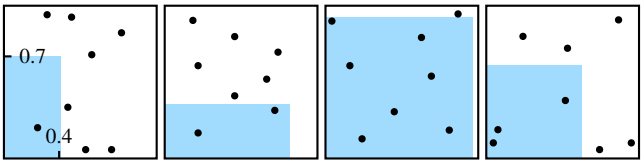


Figure 4: *Discrepancy of a shaded box (in blue) estimates the uniformity of a sample distribution over multiple realizations (5). Four such realizations are shown for $N = 16$ samples. Each sample estimates the area to be $1/N = 0.0625$. In (a), the shaded blue rectangle has a true area of $0.7 \times 0.4 = 0.028$ with one sample within the shaded region. Its discrepancy is the difference between these area values $|0.0625 - 0.028| = 0.0345$. For the star discrepancy, each box is generated with one corner fixed to the origin.*

3.1. Spatial Measures

Spatial measures refer to statistics that directly capture spatial characteristics of point patterns such as coverage over the domain [KN74], or their spatial statistics (e.g. pairwise distance distributions). All spatial measures considered so far in the literature can be easily interpreted in terms of point process statistics as we will see below, with one notable exception: discrepancy. We start with this measure, as historically it has been an important measure in developing sampling patterns for rendering, and provides interesting insights.

3.1.1. Discrepancy

Discrepancy stems from number theory with an interesting statistical background [KN74]. It measures how uniformly a pointset is distributed over a region (which is usually a hypercube) and was introduced to computer graphics by Shirley [Shi91]. Discrepancy measures the worst error that would be made in estimating the area of a subregion of the hypercube by simply noting the fraction of the pointset contained in the subregion. We focus on *star* discrepancy [Zar68] of a sample distribution, which uses only rectangular axis-aligned subregions, $B = \{[0,v_1] \times [0,v_2] \times [0,v_3] \times \cdots \times [0,v_d]\}$ where $0 \le v_i \le 1$, with one of their corners fixed to the origin (see Figure 4). Given a sequence of N sample points $s = x_1, \cdots, x_k, \cdots, x_N$, the discrepancy of s w.r.t. B is:

$$D_N^*(B,s) := \sup_{b\in B} \left| \frac{\mathrm{card}(\vec{x}_k \in b)}{N} - Vol(b) \right|, \tag{5}$$

where $|\cdot|$ is the absolute operator, $\mathrm{card}(\cdot)$ represents the cardinality of a set, $Vol(b)$ is the volume of the d-dimensional box $b \in B$ and sup represents *supremum*, the continuous analog of a max function. For a given point set, the *lower* the discrepancy value the more uniformly the samples are distributed over the domain. In terms of sampling-based estimation, the *star* discrepancy gives a worst-case integration error bound—the Koksma-Hlawka inequality [KN74,Nie92]—when the integrand has bounded variation.

Historically, discrepancy is used to develop low discrepancy patterns, which are still used in practice today to sample high dimensional domains. Although discrepancy is a simple and effective measure for such point distributions, it cannot fully capture first

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

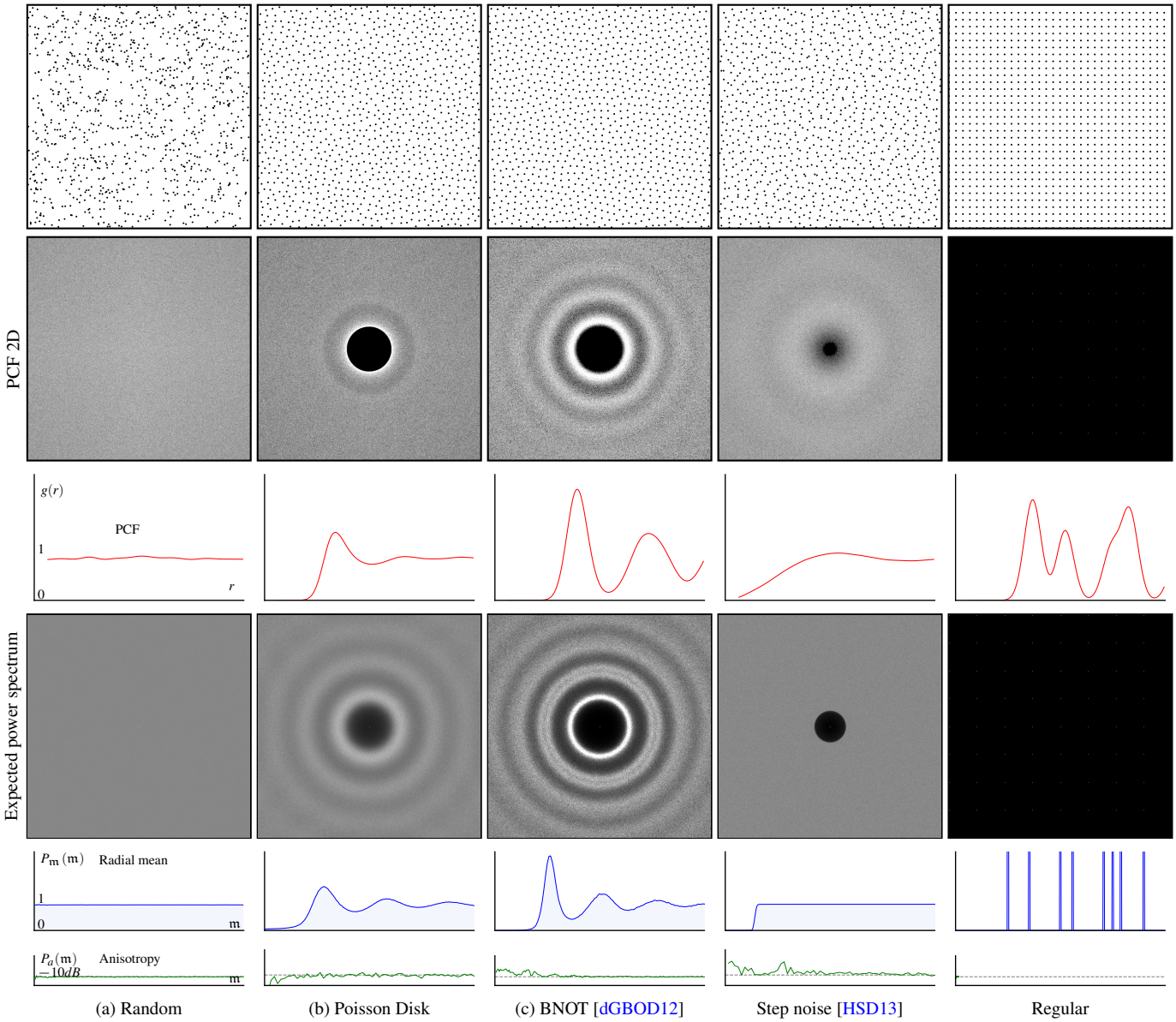


Figure 5: *An example distribution generated by different underlying point processes (top row) and its corresponding 2D PCF (second row) are shown for distributions varying from purely random to a regular grid pattern. Third row shows the 1D PCF version where the r axis is normalized with the maximum possible distance between pairs of points for this square domain and the number of points. The expected power spectrum (details in Section 3.2.1) of each sampling pattern is shown in the fourth row. The corresponding radial statistics include radial mean (fifth row) and radial anisotropy (sixth row) along the radial frequency* $\mathfrak{m}$.

and second order correlations, which are essential to analyze and develop point patterns for further error reduction.

3.1.2. Pair Correlation Function

Point process statistics provide provably informative measures for point patterns, i.e. the characteristics of a point pattern are uniquely and fully defined once these statistics are provided. As mentioned in Section 2, for our purpose of analyzing error in integral estimation, we only need first and second order product densities. We will see why this is the case in Section 4. Historically, such spatial measures of correlations have come quite late into the rendering literature, but have turned out to be very powerful, as they contain all information on a point pattern via the underlying point process.

For general point patterns, in particular if we do not assume stationarity, the second order correlation captured by $\rho(\vec{x},\vec{y})$ is a high dimensional function that is hard to visualize or analyze. Instead, the literature on rendering focuses on *stationary* patterns and hence the

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

pair correlation function (PCF) $g(\vec{r})$ with $\vec{r} = \vec{x} - \vec{y}$, and a constant intensity λ.

Given K point distributions $\mathcal{P}_k$ generated by an underlying point process, an unbiased estimator for the PCF is given by:

$$\hat{g}(\vec{r}) = \frac{1}{K} \sum_{\mathcal{P}_k} \frac{1}{\lambda^2} \sum_{\substack{i \neq j \\ \vec{x}_i, \vec{x}_j \in \mathcal{P}_k}} \delta(\vec{r} - (\vec{x}_i - \vec{x}_j)), \tag{6}$$

where δ is the Dirac delta. Note that if we assume a fixed sample count for each realization $\mathcal{P}_k$, we can set $\lambda = N$. In practice, this means for each point distribution $\mathcal{P}_k$, we compute a histogram of difference vectors $\vec{x}_i - \vec{x}_j$. The outer summation is over different K realizations of a distribution which helps smooth out the noisy histogram. A smoothed version of this histogram can also be computed by using a Gaussian kernel instead of the Dirac function $\delta(\cdot)$, and a single point distribution [WW11, OG12].

This estimator also reveals the fundamental intuition behind the PCF: it is simply a normalized measure of the distribution of difference vectors between pairs of points in point distributions generated by an underlying point process. We can work with just the difference vectors, as we assumed stationarity and thus translation invariant point distributions.

The original estimator in Eqn.(6) implicitly assumes an infinite window, i.e. the whole Euclidean space in which the point process is defined. For a practical estimator, we get points in a finite domain, typically a square. One way to avoid inaccuracies due to this finite observation window is using a toroidal domain with periodic boundary conditions. If this cannot be assumed, we may miss difference vectors that have length larger than half of the length of the edges of this window. These missing difference vectors can be compensated for by carefully normalizing the estimated $\hat{g}(\vec{r})$ for each difference vector $\vec{r}$ with the autocorrelation $a_{\mathbb{I}_V}(\vec{r})$ of the indicator function $\mathbb{I}_V$ of the domain V [Ö16]. The autocorrelation of the indicator function for a square domain in $\mathbb{R}^d$ is given by $a_{\mathbb{I}_V}(\vec{r}) := \prod_{l=1}^{d} (|V|^{1/d} - |\vec{r}_l|)$ when $|V|^{1/d} > |\vec{r}_l|$ for all l, and 0 otherwise, where $\vec{r}_l$ is the l-th component of the vector $\vec{r}$. In general, this autocorrelation can be computed numerically for a given domain.

For isotropic point processes, a similar PCF estimator can be derived using a one-dimensional kernel κ such as the Gaussian [OG12], and *distances* between points in a point distribution:

$$\hat{g}(r) = \frac{1}{\lambda^2 r^{d-1} |\mathcal{S}|^d} \sum_{i \neq j} \kappa(r - ||\vec{x}_i - \vec{x}_j||) \tag{7}$$

Hence, for the isotropic case where the generated point distributions are translation and rotation invariant, the distribution of distances is sufficient to describe a point pattern. This is a vast simplification of statistics, as it reduces the PCF to be a one-dimensional function regardless of the original dimensionality of the point pattern.

We plot example PCFs for several point processes in Figure 5, along with one example point distribution generated by the corresponding point process. We can clearly see how differences/distances can be read from the PCF graphs, e.g. as points become further apart, PCFs get lower for low r. Note that the right-most point process that generates regular point distributions is actually not a stationary or isotropic point process, and hence the PCF in this case does not uniquely describe the point pattern, i.e. there are other point patterns that share the same PCF.

3.2. Spectral Measures

The Fourier domain provides another set of tools to analyze and understand the sample distributions. We start with the definition of the power spectrum of a point process, and move on to describe these relations.

3.2.1. Power Spectrum

The power spectrum of a function describes the distribution of power into frequency components composing that signal.

In order to define the power spectrum, we first represent a point set with a so-called sampling function $s(x)$, which is a sum of Dirac impulses at the sampling points. Since the sampling-based estimators (1) are weighted by $w(x_j)$, we can also explicitly write these weights as a part of the sampling function:

$$s(\vec{x}) := \sum_{j=1}^{N} w(\vec{x}_j)\delta(\vec{x} - \vec{x}_j), \text{ with } \mathbf{s}_m := \sum_{j=1}^{N} w(\vec{x}_j) \mathrm{e}^{-i2\pi(\vec{m}\cdot\vec{x}_j)} \tag{8}$$

representing the Fourier series coefficients of the sampling function $s(\vec{x})$. The power spectrum, $\mathbf{s}_m^* \mathbf{s}_m$, gives the squared amplitude values of the Fourier series coefficients $\mathbf{s}_m$ at a given frequency $\vec{m}$ and is therefore, a *real* entity.

Sampling analysis using the power spectrum is usually performed for equally weighted uniform samples, and for multiple sampling functions $s(\vec{x})$ from multiple point distributions. Hence, similar to PCF and other point process statistics, we are interested in the expected power spectrum of a point pattern. This expected power spectrum ($\mathbb{E}[\mathbf{s}_m^* \mathbf{s}_m]$) is fundamentally related to the Fourier transform of the PCF ($\mathbf{g}_m$) which is given by the relation [HSD13]:

$$\mathbb{E}[\mathbf{s}_m^* \mathbf{s}_m] = \lambda \mathbf{g}_m + 1, \tag{9}$$

at a given frequency m. Note again that, this is true for *uniform* samples where the sampling function $s(\vec{x}) := \sum_{j=1}^{N} \delta(\vec{x} - \vec{x}_j)$ is defined as simply a sum of Dirac delta positions (with unit weights $w(x_j) = 1$, see [OS18] Section 3.2.1 for the derivation). In the language of point processes, such distributions are generated by stationary point processes. This explains that both the expected power spectrum and PCF are translation invariant, as the former lacks the phase in the Fourier domain, and the latter depends only on difference vectors.

The expected power spectrum and PCF thus carry exactly the same information, and the choice of using one or the other depends on what is desired to be qualitatively analyzed, and the familiarity of the user with the charts. For analytical analysis of error in rendering, both have been considered in the literature as we elaborate in Section 4. We plot power spectra for different samplers in Figure 5. As compared to PCF graphs, they reveal interesting spectral properties of the sampling function, which is useful in understanding how the frequencies of the sampled integrand will be affected.

3.2.2. Radial Mean

We have already seen that for isotropic point processes, the PCF is a one-dimensional function representing the distribution of distances

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

between pairs of points. Similarly, the expected power spectrum can also be summarized with a 1-dimensional radial average for isotropic point processes. This radially averaged Fourier power spectrum [Uli87] has been perhaps the most widely used tool to analyze point samples, characterizing various stochastic sampling patterns ranging from white noise to blue noise, and more recently being used to derive variance convergence rates of various stochastic samplers [PSC*15]. See Figure 5 for examples of radial mean graphs.

3.2.3. Radial Anisotropy

While radial averaging is appropriate for analyzing *isotropic* power spectra, many of the stochastic point sampling strategies used in rendering—such as N-rooks [Shi91]—are in fact anisotropic. To capture such anisotropy, the idea is to compute the radial variance [Uli87, LD08], which is the squared deviation of the expected power spectrum from its radial average at each radius. This is then used to define the *radial anisotropy*, capturing how radially symmetric the expected power spectrum is. We plot radial anisotropy graphs in Figure 5 for the shown power spectra. Oztireli and Gross [OG12] presented a similar anisotropy measure for PCFs. Such anisotropy measures are useful in spotting radii where anisotropy happens, but fall short of explaining which directions are more stretched than others, i.e. the magnitude of anisotropy in different directions.

3.3. Discussion

For qualitative analysis, the d-dimensional PCF and expected power spectrum, and their radially averaged versions are typically utilized. Of course, d-dimensional diagrams become a non-viable option for higher dimensional domains, and hence we are typically left with radial averages and anisotropies. The d-dimensional PCF and expected power spectrum are, however, essential to analytically study error in integral estimation.

A fundamental limitation of these measures is that they are defined for uniform point patterns, where the density of points is constant throughout the space. This is certainly not the case for the commonly used importance sampling in rendering. Fortunately, importance sampling is often implemented as a warping function (details in Section 5.4), where the generated samples in a unit hypercube are warped to match a certain density. Hence, the discussed tools apply to such cases if we want to study second order correlations, i.e. how points are distributed, with clustering due to density changes factored out. For quantitative analysis, the integrand can be warped back and analyzed with the measures, as we will explain in the next section.

4. Theoretical Error Analysis of Monte Carlo Integration

In this section, we relate the equidistribution measures presented in Section 3 to error in sampling-based integral estimation (1), and show that for cases commonly encountered in rendering, the error formulation simplifies dramatically allowing a tractable error analysis of different sampling patterns with complex correlations.

Error in the integral estimator $\hat{I}$ is the deviation from the true integral value:

$$\Delta := \hat{I} - I. \tag{10}$$

We would like to analyze this error over multiple (theoretically infinitely many) realizations, i.e., point distributions generated by the point process. We separate it into bias and variance terms, where the expectations are computed over different point distributions generated by the same point process, as in previous sections:

$$bias[\hat{I}] := \mathbb{E}[\Delta] = \mathbb{E}[\hat{I}] - I\ ,\ var[\hat{I}] := \mathbb{E}[\hat{I}^2] - \mathbb{E}[\hat{I}]^2. \tag{11}$$

The total squared error is then given by $\mathbb{E}[\Delta^2] = bias[\hat{I}]^2 + var[\hat{I}]$. We start with a deterministic bound for Δ arising from discrepancy theory, and move on to exact expressions for bias and variance, derived utilizing the point processes theory.

4.1. Spatial Domain

4.1.1. Error with Discrepancy

For a given point set or a sequence (stochastic or deterministic), the error of a sampling-based estimation is directly related to the discrepancy of the point set following the Koksma-Hlawka inequality:

$$|\Delta| \leq D_N^*(B,s)\mathcal{V}(f). \tag{12}$$

See Niederreiter [Nie92, Section 2.2] for a 1D proof and Kuipers and Neiderreiter [KN74, Theorem 5.5] for a detailed d-dimensional proof. In this inequality $D_N^*(B,s)$ is the *star* discrepancy of the points $0 \leq x_i < 1$, as defined in Section 3.1.1, and $\mathcal{V}(f)$ is the total variation of the function f:

$$\mathcal{V}(f) := \sup_{0=x_0<x_1<\ldots x_n=1} \sum_{j=1}^{n} |f(x_j) - f(x_{j-1})|, \tag{13}$$

which depends on the differentiability of the function. As this bound applies to all point distributions generated from a point process, the expected value of the error is the same as itself. Note that unlike the rest of the analysis tools we describe below, we cannot differentiate between bias and variance in this case, and this inequality provides a bound, as opposed to an exact error expression. Finally, it only applies to error for uniform (non-adaptive) point distributions. However, if we have minimal information about the behavior of an integrand f, this expression provides a guideline that to reduce error we need a point sampling pattern with *low* discrepancy.

4.1.2. Error with Stochastic Point Processes

Exact expressions for bias and variance can be derived starting from the theory of point processes [Ö16].

Bias. By computing $\mathbb{E}[\hat{I}]$ for the estimator in Eqn.(1) using first principles, we can simply write the bias in this form:

$$bias[\hat{I}] = \int_V w(\vec{x}) f(\vec{x}) \lambda(\vec{x}) \mathrm{d}\vec{x} - I \tag{14}$$

Clearly, when $w(\vec{x}) := 1/\lambda(\vec{x})$, bias is zero and we get an unbiased estimator.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

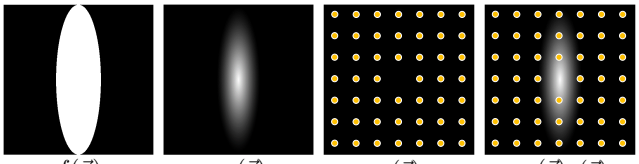


Figure 6: *An integrand f, its autocorrelation a_f, PCF g of a stationary jittering pattern [RAMN12, Ö16] that is a sum of Dirac impulses, and the corresponding variance term $a_f(\cdot)g(\cdot)$ are illustrated. The integral of this multiplication, which in this particular case is a sum due to Dirac impulses, increases the variance for a stationary point process. Figure reproduced from [Ö16].*

Variance. The expression for variance can be derived in the form:

$$\begin{aligned} var[\hat{I}] = &\int_V w^2(\vec{x}) f^2(\vec{x}) \lambda(\vec{x}) \mathrm{d}\vec{x} \\ &+ \int_{V\times V} w(\vec{x}) f(\vec{x}) w(\vec{y}) f(\vec{y}) \rho(\vec{x},\vec{y}) \mathrm{d}\vec{x}\mathrm{d}\vec{y} \\ &- \int_{V\times V} w(\vec{x}) f(\vec{x}) w(\vec{y}) f(\vec{y}) \lambda(\vec{x}) \lambda(\vec{y}) \mathrm{d}\vec{x}\mathrm{d}\vec{y}, \end{aligned} \tag{15}$$

where $\lambda(\vec{x})$ and $\rho(\vec{x},\vec{y})$ capture first and second order correlations, respectively, see Section 2.3. The full derivation can be found in Oztireli [Ö16] (Section 4.1). This expression makes it clear that error only depends on first and second order correlations, and thus higher order correlations are not needed for the analysis. It is also interesting that the expression decomposes the influence of first and second order correlations on the variance into separate terms, and hence they can be studied separately. However, one practical challenge here is that density of points (e.g. in importance sampling) not only affects $\lambda(\vec{x})$, but also $\rho(\vec{x},\vec{y})$. This problem can be solved by considering that importance sampling is obtained via warping from a primal domain, where the point process is stationary.

Stationary Processes. Oztireli [Ö16] further derived simplified expressions for the stationary case. As before, bias vanishes with uniform weights $w = 1/\lambda$, but variance simplifies to

$$var[\hat{I}] = \frac{1}{\lambda}\int_V f^2(\vec{x})\mathrm{d}\vec{x} + \int_{\mathbb{R}^d} a_f(\vec{r}) g(\vec{r}) \mathrm{d}\vec{r} - \left(\int_V f(\vec{x})\mathrm{d}\vec{x}\right)^2, \tag{16}$$

where $a_f(\vec{r})$ denotes the autocorrelation of the integrand $f(\vec{x})$, and $g(\vec{r})$ is the PCF. A closer look at Eqn.(16) reveals that the third summand simply represents the square of the true integral and, therefore, does not affect the variance of the estimator. Since the first and the second summands are positive, the only way to reduce variance is by reducing each summand. The first summand only considers first-order statistics (λ) and therefore, cannot be reduced by introducing pairwise correlations. The second summand, however, suggests that given minimal information about the integrand f, we can reduce variance by reducing the PCF g, especially when the autocorrelation of the integrand is high.

An interesting case where we can show using Eqn.(16) that variance can *increase* due to the directional nature of an integrand and sampling pattern is shown in Figure 6. For this integral estimator, we used a jittering pattern generated by randomly perturbing a regular grid [RAMN12, Ö16], which is stationary and has the PCF $g(\vec{r})$ as an impulse train except at $\vec{r} = 0$ [Ö16]. For this case, the second term in Eqn.(16) turns into a sum of the values of $a_f(\vec{r})$ at a set of $\vec{r}$ that are the locations of the Dirac impulses in $g(\vec{r})$. The impulses along the central vertical line capture high values of the autocorrelation as shown in the figure, leading to a larger second term in Eqn.(16) and hence larger variance.

Importance Sampling. The above expression for variance can be extended to importance sampling by assuming that an invertible warping function $t(\vec{x})$ is applied to the original domain to induce adaptive density. In this setting, a uniform point distribution by a stationary point process is first generated and warped for adaptivity. The bias can always be canceled out by seting $w(\vec{x}) := 1/\lambda(\vec{x})$ as above. For computing the variance, we can warp back the integrand to the primal domain where the stationary point process is defined by $f(t^{-1}(\vec{x}))$. Using this into the above expression for variance of stationary point processes in Eqn.(16) with the second term expanded, and with a change of integration variables, we get:

$$\begin{aligned} var[\hat{I}] = &\frac{1}{\lambda}\int_V f_t^2(\vec{x})\mathrm{d}\vec{x} + \int_{V\times V} f_t(\vec{x}) f_t(\vec{y}) g(t(\vec{x}) - t(\vec{y}))\mathrm{d}\vec{x}\mathrm{d}\vec{y} \\ &- \left(\int_V f_t(\vec{x})\mathrm{d}\vec{x}\right)^2, \end{aligned} \tag{17}$$

where we defined $f_t(\vec{x}) := f(\vec{x}) J_t(\vec{x})$ with J_t denoting the determinant of the Jacobian of the warp t, and assumed that $t^{-1}(V)$ is a subset of V.

4.2. Fourier Domain

To perform error analysis in the Fourier domain, Durand [Dur11] proposed to rewrite the Monte Carlo estimator in Eqn.(1) in the continuous integral form as a product of the sampling function and the integrand $f(x)$:

$$\hat{I} = \int_0^1 f(x) s(x) \mathrm{d}x. \tag{18}$$

Estimator in Fourier Domain. Following the convolution theorem [Bra00], MC estimation can be seen as a convolution at the DC ($m = 0$) in the Fourier domain:

$$\hat{I} = \sum_{m=-\infty}^{\infty} \mathbf{f}_m^* \mathbf{s}_m. \tag{19}$$

Error in Fourier Domain. In the Fourier domain, error from Eqn.(10) can be rewritten in the form:

$$\Delta = \sum_{\substack{m=-\infty \\ m\in\mathbb{Z}}}^{\infty} \mathbf{f}_m^* \mathbf{s}_m - \mathbf{f}_0, \tag{20}$$

using Eqn.(19) and the fact that the true integral ($I = \mathbf{f}_0$) represents the DC ($m = 0$) component in the Fourier domain. This implies that error is introduced when the DC component ($\mathbf{f}_0$) gets polluted by the spectral replicas (aliases). This observation led Durand to derive a revised sampling theorem which states that a function can be perfectly integrated if it is sampled at at least its maximum frequency. This is a factor of two compared to Nyquist and Shannon [Sha49, Uns00].

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

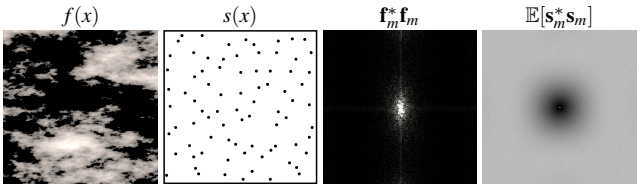


Figure 7: *For a given integrand $f(x)$ that is sampled with randomly jittered samples generated by a stationary point process (e.g. homogenization), the corresponding variance $var[\hat{I}]$ in Eqn.(23) involves only the product of the integrand's power spectrum ($\mathbf{f}_m^*\mathbf{f}_m$) and the samples'expected power spectrum ($\mathbb{E}[\mathbf{s}_m^*\mathbf{s}_m]$). Figure reproduced from [SJ17].*

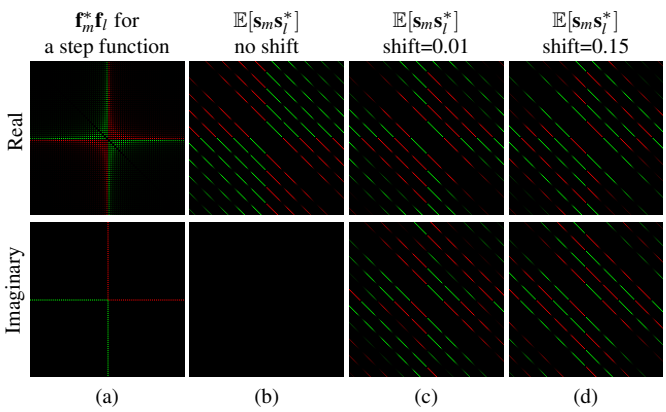


Figure 8: *To emphasize the role of phase in variance reduction, we visualize the third term from Eqn.(22) for a 1D step function (**a**) and 1D jittered samples (for $N = 16$ in **b,c,d**) when all the strata are shifted as indicated above each column. Green, black, and red indicate positive, zero, and negative values, respectively. Figure reproduced from [SSC*19].*

This was later employed to estimate 5D integrals for rendering distribution effects [BSS*13].

Bias. Since over multiple realizations, error consists of a bias and a variance term, we rewrite bias in the form :

$$bias[\hat{I}] = \mathbb{E}[\Delta] = \sum_{m=-\infty}^{\infty} \mathbf{f}_m^*\mathbb{E}[\mathbf{s}_m] - \mathbf{f}_0^*. \tag{21}$$

Following this formulation, Subr and Kautz [SK13] observed that an unbiased estimator can be obtained if either both the summands in Eqn.(21) are equal or $\mathbb{E}[\mathbf{s}_m] = \delta(m)$. The latter implies the expected Fourier spectrum to be non-zero and unity only at the DC frequency ($m = 0$) which can be obtained when $w(x) = 1/\lambda(x)$, similar to Eqn.(14) in the spatial domain. For fixed N, the weights are $w(x) = 1/(Np(x))$ for a given density $p(x)$, which we commonly use to weight the samples in rendering. Consequently, sampling-based estimation with properly weighted stochastic samples is unbiased [SK13, Ö16, SSC*19], which yields error only in terms of variance.

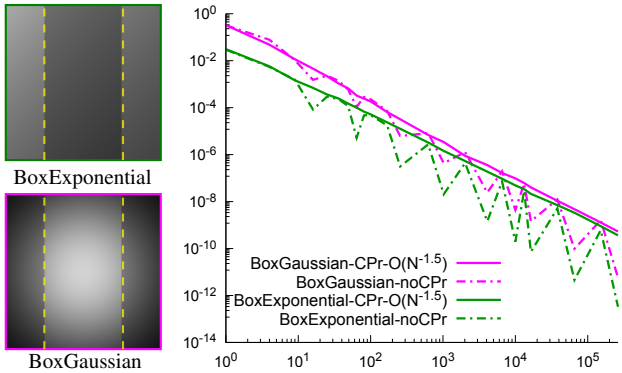


Figure 9: *Jittered sampling shows good convergence (dotted curve) when the strata boundaries align with the step discontinuities introduced at $x = 0.25$ and $x = 0.75$ in an otherwise smooth gaussian and an exponential function. However, homogenization (CPr) destroys this alignment resulting in bad convergence (shown in log-log scale) behavior overall (solid curves). Figure reproduced from [SSC*19].*

Variance. Variance in the Fourier domain has the following form [SSC*19]:

$$var[\hat{I}] = I^2 var[\mathbf{s}_0] + \sum_{\substack{m\in\mathbb{Z}\\ m\neq 0}} \mathbf{f}_m^*\mathbf{f}_m\mathbb{E}[\mathbf{s}_m^*\mathbf{s}_m] + \sum_{m=\mathbb{Z}}\sum_{\substack{l=\mathbb{Z}\\ l\neq m}} \mathbf{f}_m^*\mathbf{f}_l\mathbb{E}[\mathbf{s}_m\mathbf{s}_l^*]. \tag{22}$$

This is a generalized formulation which works for both uniform and importance sampling strategies. Here the first summand contains the DC contribution, the second term is a product of the integrand power spectrum ($\mathbf{f}_m^*\mathbf{f}_m$) and the samples' expected power spectrum ($\mathbb{E}[\mathbf{s}_m^*\mathbf{s}_m]$), visualized in Figure 7. The third summand is crucial as it quantifies both the variance reduction due to importance sampling and the alignment of the samples with the integrand, which accounts for the phase information (see Figure 8). Eqn.(22) can be seen as a covariance matrix where the first and the second terms are along the diagonal whereas the third term represents the non-diagonal elements of the matrix.

Stationary Processes. For stationary point processes, the first and third summand in Eqn.(22) goes to zero. This simplifies the variance formulation to the form:

$$var[\hat{I}] = \sum_{\substack{m\in\mathbb{Z}\\ m\neq 0}} \mathbf{f}_m^*\mathbf{f}_m\mathbb{E}[\mathbf{s}_m^*\mathbf{s}_m]. \tag{23}$$

This result was first derived by Durand [Dur11] and later by Subr and Kautz [SK13], but it was only valid for MC random samples as they are stationary by construction. Pilleboue et al. [PSC*15] generalized this result to all samplers by *homogenizing* each sampling pattern. The idea behind homogenization (Cranley-Patterson rotation [CP76]) is to randomly and uniformly translate each realization of a point set before computing the expectation operator in Eqn.(23). Consequently, the underlying point set becomes stationary (shift-invariant).

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

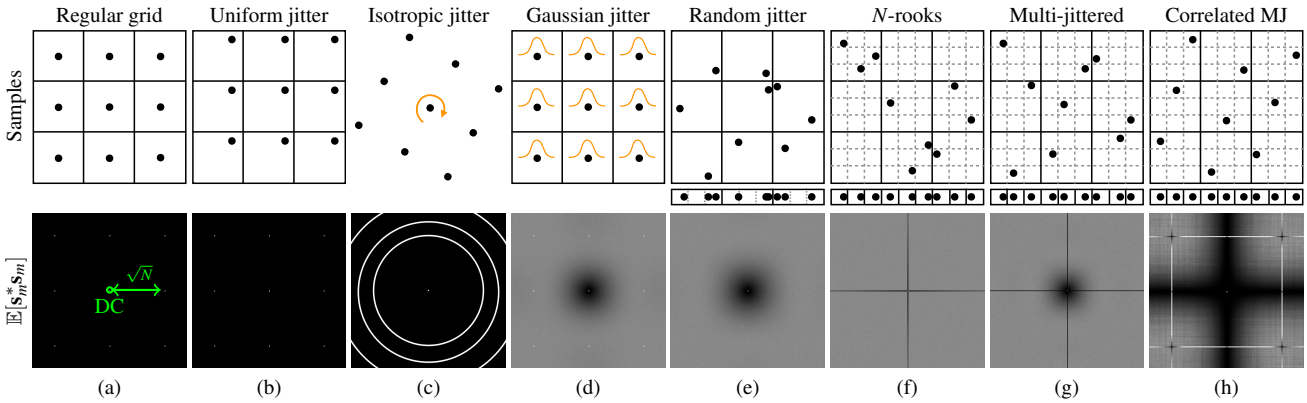


Figure 10: *Different stratification techniques are summarized in 2D with their 1D projection along the vertical y-axis, where applicable. (a) Placing samples at the center of a grid gives visual banding artifacts during rendering. (b) By offsetting all samples within a pixel (uniform jittering) by the same random amount, these artifacts can be avoided. (c) Random translation followed by a random rotation further helps to reduce variance. (d) Another variant involves perturbing each sample within a stratum by a random offset from a radially symmetric Normal distribution. (e) Random (and Gaussian) jittering does not have dense 1D stratification (1D projection contains more than one sample per stratum). (f) LH (or N-rooks) sampling does not preserve 2D stratification (empty strata in the top row and leftmost column). (g) Multi-jittered (MJ) sampling preserves both 1D and 2D stratifications, thereby combining LH and random jittering properties. (h) Correlated multi-jittered sampling further enforces points to be far apart from each other while preserving multi-jittered behavior. Bottom row shows their expected power spectra (exposure adjusted to highlight the spikes) with a peak at the center representing the DC frequency ($m = 0$).*

4.3. Discussion

The variance expressions derived in the spectral domain can be obtained by applying a Fourier transform to the spatial counterpart in Eqn.(16), as the PCF is fundamentally related to the expected power spectrum of a point pattern by Eqn.(9), and the Fourier transform of the autocorrelation a_f of the integrand f is its power spectrum. Eqn.(16) suggests that variance can be reduced if the PCF g is low where a_f is high. This intuition carries over to the spectral domain: the spectral variance expressions suggest samplers with low energy in the spectrum where the integrand spectrum has high energy. Deriving similar connections to the Koksma-Hlawka inequality in Eqn.(12) requires further investigation.

To simplify variance expressions in Eqn.(15) and (22) to their stationary counterpart (Eqn.(16) and (23)), one can perform homogenization or Cranley-Patterson rotation (CPr). It involves adding a random global offset to all sampling point locations for each realization. However, such operations on the point pattern can also destroy important correlations between point samples and the integrand [SSC*19]. A simple example is when step discontinuities align with the strata boundaries during jittered sampling. Homogenization would destroy this alignment, resulting in bad convergence behavior as shown in Figure 9 (solid curves).

If we assume a stationary point process to start with, and an invertible warp on the primal domain for importance sampling, the integrand can be warped back to the primal domain to utilize the standard spatial expression for variance of stationary point processes in Eqn.(16) or its slightly modified version with a change of variables in Eqn.(17). For other sampling techniques for adaptive density, even more simplified expressions can be derived [Ö16]. However, for a general algorithm that introduces adaptive density and correlations, we have to fall back to the more complex expressions for general point processes in Eqn.(15) or (22).

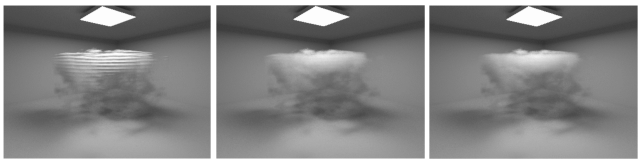
Although utilized in recent works [PSC*15, Ö16, SSC*19], these exact bias and variance expressions have so far not been extensively studied for sampling patterns and integrands encountered in rendering. They, nonetheless, already provide valuable insights and guidance when choosing sampling patterns, which we elaborate on in the next section.

5. Error Analysis for Common Sampling Techniques

The closed-form error formulations summarized in Section 4 provide several insights on what kind of correlations help in error reduction and convergence improvements. In this section, we investigate properties of commonly used sampling strategies in the light of these expressions.

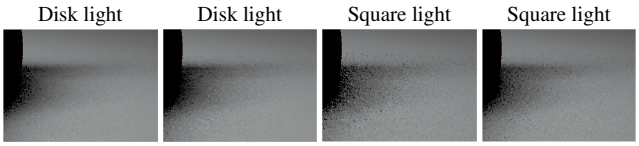
We first look into several stratification techniques that are quite commonly used in rendering due to their simple constructions (see Figure 10 and appendix A for code snippets). We go over recent developments that analyze their impact on the rendering quality under simplified light transport scenarios, which are empirically shown to work in complex cases. In Section 5.2, we briefly review quasi-random samplers that overcome limitations of grid-based stratification techniques. We then dive into sophisticated blue noise algorithms and the improvements they brought in terms of both variance reduction and convergence in Section 5.3. This is followed by recent ideas on combining stratification with blue noise algorithms to obtain the so called *low* discrepancy blue noise samples. In the last Section 5.4, we review how the error or variance behaves when these correlated sampling strategies are combined with importance sampling. Throughout this section, we assume a canonical d-dimensional unit cube $[0,1)^d$ space as the sampling domain.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.



(a) Regular grid (b) Random jitter (c) Uniform jitter

Figure 11: *Regular grid sampling clearly reveals aliasing artifacts in (a) which are no longer visible in the randomized versions (b and c) of the ray marching algorithm [AW87, PH89]. Figure reproduced from [PKK00].*



(a) Uniform jitter (RMS 6.59%) (b) Random jitter (RMS 8.32%) (c) Uniform jitter (RMS 13.4%) (d) Random jitter (RMS 10.4%)

Figure 12: *For (circular) disk lights, uniform jittering (single offset per pixel applied to each stratum) has less noise than random jittering (each stratum is independently perturbed). For square lights, the converse is true, with random jittering per stratum the performance is better. Figure reproduced from [RAMN12].*

5.1. Stratification based sampling

Random sampling is usually suboptimal for Monte Carlo rendering. Mitchell [Mit87, Mit91] conducted some notable work on optimal sampling patterns and argued that better results can be obtained with stratified sampling [Mit96]. Stratification is a simple approach to introduce correlations within samples. The idea is to subdivide the sampling domain into a Cartesian product of intervals [Ney34] and placing one sample[‡] within each stratum either randomly or deterministically. In numerical integration, stratification has provably shown to reduce error/variance and improve convergence rates during sampling-based estimation [Vea97].

Error due to stratification varies depending on where a sample is placed within a stratum [RAMN12]. Regular grid sampling places samples at the center of each stratum. The underlying point process is non-stationary, and thus a significant bias may result, which may manifest itself as visual banding artifacts during rendering (Figure 11a). By simply offsetting each sample by the same random offset (called *uniform jitter*), these artifacts can be avoided [PKK00], as this makes the underlying point process stationary and hence cancels the bias. It is also possible to independently offset each sample within a stratum to obtain *random jitter* (Figure 10e) which is quite commonly used in estimating low-dimensional light transport integrals with difficult visibility functions (multiple discontinuities) [Mit96]. Although random jitter sampling does not result from a stationary process, it is still unbiased as long as the random offsets cover each stratum.

‡ More samples can be placed within each stratum but that can hinder the quality and deteriorate convergence during integral estimation.

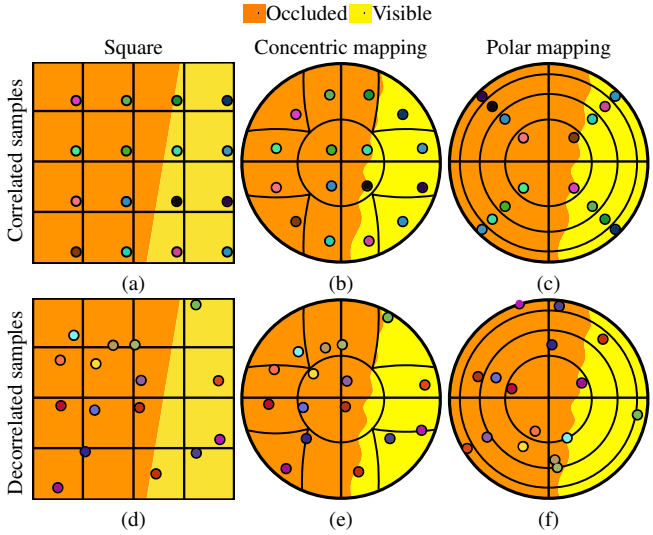


Figure 13: *Canonical visibility configuration for uniform jitter (**top row**) and random jitter (**bottom row**) samples is shown for square and disk lights. In (a), the correlation among different "light scanlines" for uniform jitter can lead to poor results with square lights, since each "light scanline" obtains the same result (unlike in d), without further decreasing variance. However, after polar mapping (c), samples from each column in (a, color-coded) are pushed all over the circle at fixed radius from center, resulting in less variance. (b) Concentric mapping can also achieve some perturbation across the discontinuity but the samples stay in almost the same vicinity. After mapping to disk, the perceived occluded boundary might also transform (shown as wiggled edge). Figure inspired from [RAMN12].*

Ramamoorthi et al. [RAMN12] made an interesting observation that in some cases uniform jitter can perform better than random jitter despite the regular structure present in the uniformly jittered samples. By performing a 1D visibility analysis with different jittering techniques, they found that the improvements due to uniform jitter depends on the geometry of the discontinuity. For example, when a scene is illuminated with a circular area light source, uniform jitter light sampling shows significant improvement over randomly jittered samples (Figure 12). However, for square area light sources, random jittering is more effective compared to uniform jittering. The latter can be explained by the positive correlations which uniformly jittered samples impose w.r.t. discontinuities observed by the square area light source (Figure 13a), whereas, random jittering decorrelates the samples w.r.t. any discontinuity (Figure 13d).

For circular area lights, samples are usually mapped from a unit square domain using either a polar or concentric mapping [SC97]. We visualize these mappings in Figure 13 for a 4×4 grid. Polar mapping $(x, y) \rightarrow (u = \sqrt{x}, \phi = 2\pi y)$ is an area preserving map which results in stretching each grid column to a circle of radius u. As a result, samples sharing the same x-coordinate (e.g., Figure 13a, first column) are placed along the same circle of radius u (e.g., Figure 13c innermost circle, samples are color coded for better visualization). As a consequence, for uniform jitter, the samples in Figure 13a

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

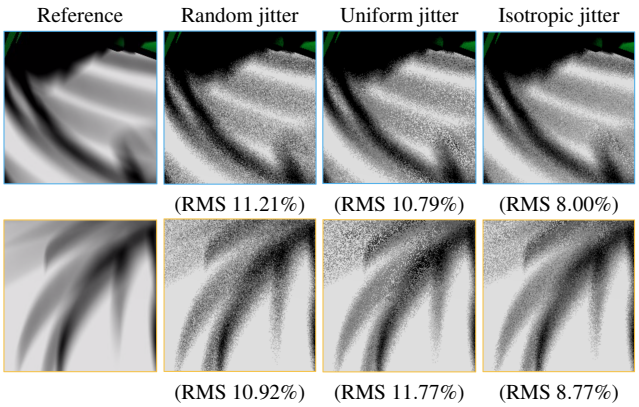


Figure 14: *Isotropic jittering can result in errors lower than uniform and random jitter samples under certain conditions on the scene geometry and light types. The scene is rendered with a disk area light source and an average of 7 visibility samples per pixel. Figure reproduced from [Ö16].*

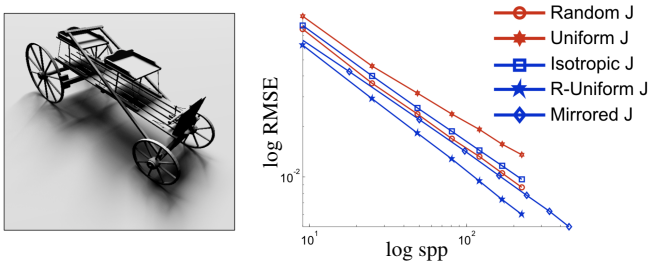


Figure 15: *RMS error is plotted for a scene lit by a square area light source. The correlations among the strata make uniform and isotropic sampling disadvantageous. Among all the sampling strategies shown (excluding R-Uniform J), mirrored jitter sampling results in the least error. Figure reproduced from [Ö16].*

falling along any column are now evenly spread across the domain in a circle (in Figure 13c). Christensen [Chr18] made similar observation which could explain why uniform jitter works so well for circular area light sources. Note that, this hypothesis does not take into the boundary edge transformation that might happen after polar or concentric mapping. Interestingly, random jitter goes through a same transformation (Figure 13d→f) but performs worse than uniform jitter. This can be explained by comparing their expected power spectra from Figure 10 (bottom row). Since the magnitude of variance depends on the product of the sampling and the integrand power spectra (23), uniform jitter would show less variance due to no energy in the low frequency region compared to random jitter spectrum.

Öztireli [Ö16] points out some cases where uniform jittering w.r.t. a particular scene geometry (e.g. circular area light source) might not always be beneficial (bottom row in Figure 14). To further harness the benefits of uniform jittering, Öztireli proposes to rotate the uniformly jittered samples w.r.t. the integrand orientation. For integrands with no prior knowledge about their orientations a random rotation can be performed on uniformly jittered samples (*isotropic uniform jitter*[§] in Figure 10c). He also derived closed-form variance formulations for these different variants of jittered sampling with global inter-strata correlations as special cases of the formulas in Section 4.1, and argued that adapting sample correlations w.r.t. the underlying integrand can show improvements in variance. Following this, the rotated uniform jittering pattern with the minimum variance is termed as *r(rotated)-uniform* jittering.

Some integrands can exhibit negatively correlated parts, which can be exploited to reduce the error in integration. Antithetic sampling [HM56] is a classical strategy to exploit these negative correlations. The idea is to generate a pair of samples (e.g. x and $1-x$), one of which is a mirror copy of another. Öztireli [Ö16] combined this idea with jittered sampling to obtain *mirror jitter* sampling. It involves one randomly jittered sample and its mirror copy within each strartum. These negative correlations work well to reduce variance for soft shadows with not too complex occluders (Figure 15). The corresponding variance expressions were derived by considering mirror jittering as superpositions of uniform jittering.

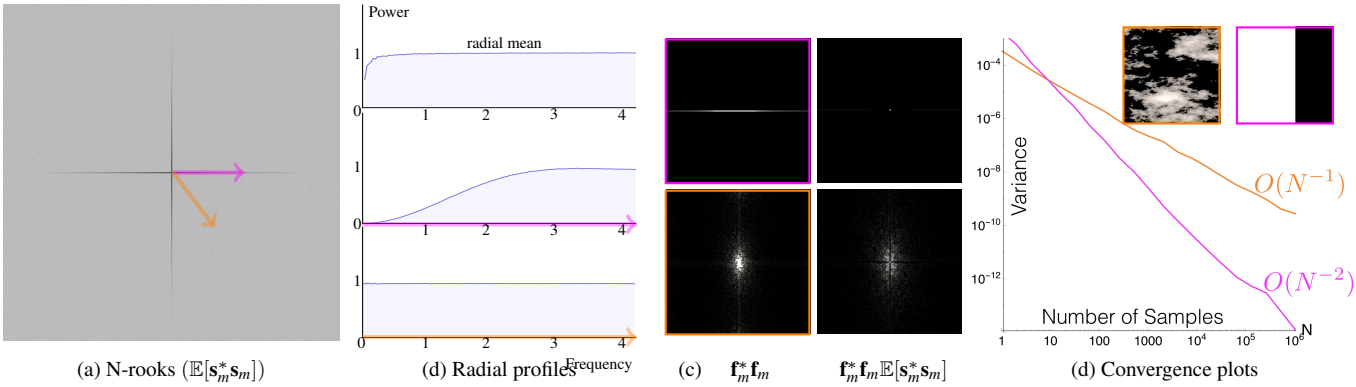
Subr and Kautz [SK13] propose another variant called *Gaussian random jittering*—inline with Cook et al. [CPC84, Coo86]—that involves perturbing a sample within each stratum by a random variable from a Normal distribution ($\mathcal{N}(\mu,\sigma^2)$). The resulting 2D power spectrum looks much like randomly jittered samples but with high energy peaks at $\sqrt{N}$ distance far apart (Figure 10d). This is a biased sampling strategy as the samples cover the whole domain non-uniformly (without weights), as opposed to random jittering. The bias was controlled to trade off variance.

Pilleboue et al. [PSC*15] performed a Fourier domain variance analysis to understand which spectral characteristics of samplers are important to improve convergence rate for integral estimation. Based on a variant of the spectral variance expression in Eqn.(23), they proposed a convergence analysis tool that characterizes radial profiles of different radially averaged sampling power spectra. They fixed the best (no C_0 discontinuites) and the worst case (with C_0 discontinuites) from a given class of functions [BCT01] and propose theoretical error bounds (lower and upper) for different samplers based on their radial profiles (radially averaged expected power spectra). In general, their analysis shows that functions with no C_0 discontinuity seems to always follow the proposed best case convergence rate whereas functions with C_0 discontinuities follow the worst case behavior. Their analysis further reveals that the point processes with no low frequency energy in their power spectrum (e.g., blue noise samples, see Section 5.3) or with spectrum having energy going to zero at DC (e.g., random jitter) would give better variance reduction and convergence rates. However, their convergence tools are only valid for samplers with *isotropic* expected power spectra.

Some of the stratification techniques listed in Figure 10 have anisotropic structures present in their expected power spectra (bottom row). This is because the underlying point processes are neither isotropic, nor stationary [SJ17]. It is thus not sufficient to rely on

§ Uniform jitter (including isotropic one) is also known under the name of systematic sampling [CO89] and has been studied by Sbert et al. [SRFN06] for image analysis.

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.



(a) N-rooks ($\mathbb{E}[\mathbf{s}_m^*\mathbf{s}_m]$) (d) Radial profiles (c) $\mathbf{f}_m^*\mathbf{f}_m$ $\mathbf{f}_m^*\mathbf{f}_m\mathbb{E}[\mathbf{s}_m^*\mathbf{s}_m]$ (d) Convergence plots

Figure 16: *The expected power spectrum in (a) of N-rooks (shown for $N = 256$) samples is highly anisotropic, with different radial behavior along different directions (orange vs. magenta arrows). Radial averaging (radial mean) masks the good anisotropic properties of the sampler along the canonical axes. (c) Two different integrand spectra (first column) and their product with the N-rooks expected power spectrum (second column) are shown. The hairline anisotropy along the axes in (a) cancels out the integrand spectral energy in magenta box resulting in much better convergence as shown in (d) with the two integrands shown in the top-right. Figure reproduced from [SJ17].*

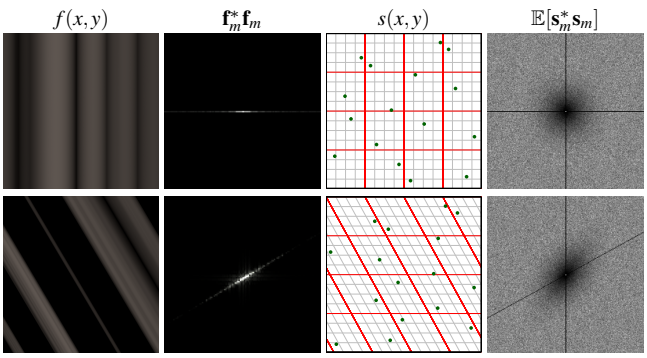


Figure 17: ***Top row*** *illustrates an integrand with spectrum energy only along the horizontal direction. Existing anisotropic samplers can be directly used to integrate this function. However, for an integrand with an arbitrarily oriented power spectrum (**bottom row**), samples can be sheared to match the low energy region of the sampling power spectrum with the high energy regions of the integrand spectrum. Figure reproduced from [SJ17].*

radially averaged power spectra to predict error or convergence rates. An example is shown for an N-rooks sampler in Figure 16, which has hairline anisotropic structures along the canonical axes due to the dense 1D stratification samples along the axes.

When these anisotropic structures completely align with the integrand spectrum, N-rooks shows significantly improved convergence rates (Figure 16d, magenta). The reason for this improvement is depicted in Figure 16c: the error, which depends on the multiplication of the power spectra of the point pattern and integrand, is significantly lower for the integrand in magenta. However, functions encountered in practice can have arbitrary orientations and thus anisotropies. Earlier works [Kel06, DK08] demonstrate sample transformations that reduces aliasing and noise for rendering purposes directly in the spatial domain. However, these techniques are agnostic to the underlying integrand orientation. Singh and Jarosz [SJ17] perform an in-depth Fourier analysis and propose to shear (or rotate) the samples (Figure 17) which dramatically improves the variance convergence rates. They leverage the light transport frequency analysis [DHS*05] to obtain the shear parameters. However, since most existing samplers have only hairline anisotropic structures in their spectra, the improvements could only be shown after a very high sample count, e.g. $N > 16K$ or more. Correlated MJ (Figure 10h) has wider anisotropic structures but they are polluted by high energy streaks which delimit their ability to improve convergence when aligned with an integrand spectrum.

In order to use stratification in higher dimensions, usually samples are generated by *padding* [Coo86]: samples are jittered in lower dimensions (in 1D or 2D for image plane XY, lens UV, time T,...) and are randomly permuted to form *uncorrelated jittered* samples in higher dimensions. This strategy only preserves spatial and spectral characteristics in the original subspaces, whereas, across subspaces (e.g. XU, YV, ...) the good properties (e.g., spectrum with no energy in the low frequency region) are lost resulting in an overall bad convergence behavior [SJ17].

Discussion. Stratifying the sampling domain has many merits in terms of convergence improvement and variance reduction. The analysis shown so far reveals that the choice of the stratification strategy should depend on the correlations present in the integrand, e.g. its spectral energy distribution, or anisotropy.

The spatial domain analysis emphasizes both local and global correlations. For example, if a local patch in a stratum and its mirrored version are negatively correlated, mirrored jitter sampling would reduce variance. Greater reductions can also be obtained when there are global correlations. As examples, negatively correlated strata leads to considerable reduction in variance when using uniform jitter sampling, whereas, for functions with positively correlated strata it is beneficial to use isotropic jitter sampling instead.

The Fourier domain analysis encodes these correlations in the ex-

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

pected power spectrum and emphasizes that for variance reduction it is important to have samplers with no or low energy in their spectrum along the frequencies where the integrand spectrum has higher energy. This can be achieved by performing simple transformations (like shearing or rotation) to the samples w.r.t. the underlying integrand.

While several point sampling constructions and transformations are proposed to improve stratification, MC estimation need not always be a point sampling process. Over the past couple of decades researchers have started employing MC-like estimators using *line (segment) samples* for rendering problems ranging from anti-aliasing [JP00], distribution effects [GDAM10, TPD*12], hair rendering [BGAM12] to path sampling [NNDJ12, KGH*14] or density estimation [JNSJ11]. Billen and Dutré [BD16] demonstrated improvements with line samples for direct illumination. Singh et al. [SMJ17] analyzed the line and line segment based MC estimators and derive closed-form error formulations in the spectral domain. Their analysis reveals that with certain correlations (e.g., jittering or blue noise [SZG*13]), line and line segment samples can show dramatic improvements in convergence and variance reduction. This is primarily because line (segment) samples pre-filter the underlying function leading to smoothing out of C_0 discontinuities. Furthermore, since line samples spread all across the domain, they further reduce the dimensionality of the problem leading to faster convergence and potentially denser stratification.

5.2. Quasi-Monte Carlo sampling

One common limitation to all stratification techniques presented so far is that they cannot generate samples progressively. This is crucial in rendering since it is impossible to know the sampling budget beforehand to render a light transport effect. Recently, Christensen et al. [CKK18] proposed a simple construction to generate progressive versions of randomly jittered and multi-jittered stochastic samples in 2D. However, due to the underlying grid-based constructions, these stratification techniques do not scale well with dimensionality. In contrast, quasi-Monte Carlo samplers do not have this limitation. There are various other quasi-Monte Carlo sample constructions available in the literature [Nie92, KPR12], which would merit its own survey report.

The commonly used *low discrepancy* quasi-Monte Carlo samplers such as Halton [Hal64] or Sobol [Sob67] are known to give better convergence rates and low error when used in integral estimation, as compared to random sampling. This is due to the regularity they induce into the sampling pattern. Hence, the resulting distributions exhibit similar structures to stratification. However, the 2D projections might not have well-distributed samples and many systematic patterns, stripes of points aligned along diagonals, and pairs and triplets of clumped points can be observed. This could give unpleasant artifacts during rendering.

To avoid such non-uniform distributions and the related artifacts, these quasi-random sequences are often randomized [Kel13, Owe03]. One such randomization technique is called Cranley-Patterson rotation [CP76], which represents random toroidal shifts. Another common randomization is random digit scrambling [Owe97, KK02]. Hence, in the end we again get stochastic point patterns with a well-defined underlying point process.

When evaluating and comparing these sample patterns, it is common to use the star discrepancy (5) as a measure of quality [RWCS05]. However, for rendering purposes, discrepancy might not be appropriate for measuring sampling pattern quality due to the presence of randomly oriented edges present within a pixel [Mit92, DEM96]. Recently, Christensen et al. [CKK18] compare different scrambling variants of Halton and Sobol, and show that despite having good low discrepancy properties, some of these variants do not always give good convergence behavior even for simple functions like a triangle, Gaussian, or bilinear functions. We thus believe that a thorough analysis of these sampling patterns from the perspective of point processes would reveal interesting insights.

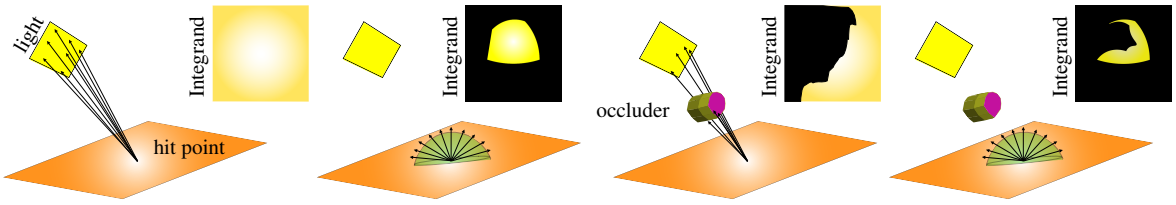
5.3. Blue-Noise Sampling

The frequency analysis by Durand [Dur11] and Pilleboue et al. [PSC*15] suggest that sampling patterns with a blue-noise spectrum [Uli88], i.e. "no low frequency," are advantageous for Monte Carlo integration, especially when the integrand has less energy in the high-frequency range. The spatial domain analysis by Öztireli [Ö16] also suggests that distributions that maximize the spacing between sample points reduce the integration error, confirming the classic intuition of using *Poisson-disk* distributions for Monte Carlo sampling in computer graphics [DW85, Coo86]. These analyses all follow from the error expressions as detailed in Section 4. Similarly, Mitchell [Mit91] shows that blue-noise power spectrum on the image plane plays an important role in the overall evaluation of the visual quality of renderings, which was recently demonstrated by Georgiev et al. [GF16, GIF*18] in realistic settings.

There are, however, many obstacles to the actual adoption of such so-called blue-noise sampling patterns, where the points are randomly distributed with a certain distance among them. First, the underlying assumptions of "no low frequency" is difficult to obtain by many of the blue-noise generation algorithms. For example, blue-noise samplers such as Poisson-disk sampling, FPO [SHD11] and KDM [Fat11] are reported to perform worse than *randomly jittered* samples at high sampling rates [PSC*15] due to the presence of low-frequency energy in their expected power spectra. Even though there exist well performing blue-noise patterns (such as CCVT [BSD09] and BNOT [dGBOD12]) and algorithms [HSD13, JZW*15, KTBV16, WW17] that allows better control to the shape of the spectrum, such high quality blue-noise samplers are only available in low dimensions [GF16]. For high dimensions, samples can be generated by *padding*: sampling each one- or two-dimensional sub-domain (lens, lights, etc) and randomly joining the samples. This will, however, destroy the nice statistics of the distributions. Reinert et al. [RRSG16] proposed a purely optimization based algorithm that can generate blue noise samples in multiple projections (1D, 2D or higher) but it does not scale well to higher dimensions or with higher sample count. Finally, all known blue-noise sampling algorithms are too slow to cope with the high sampling rates in rendering applications. Thus, only lookup techniques for distributing pre-computed blue-noise patterns can be used in rendering.

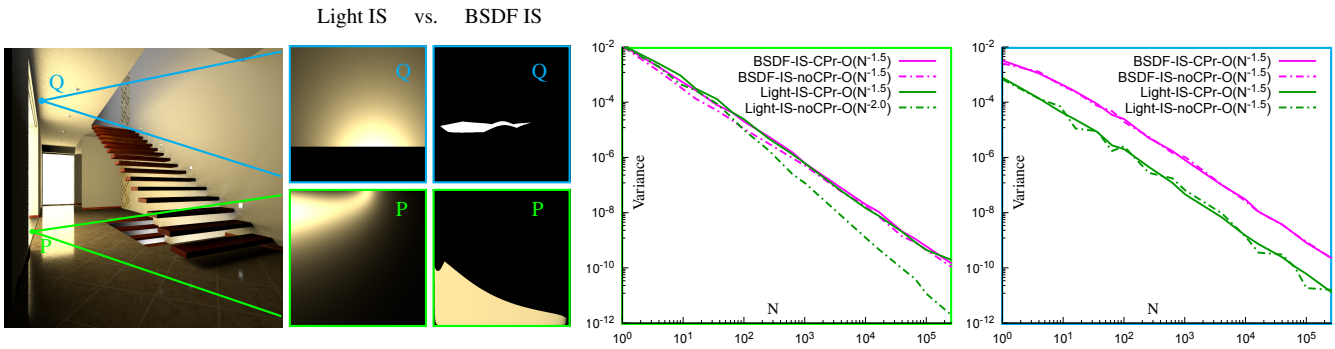
Among the lookup techniques, tiling methods with multiple samples per tile [CSHD03, KCODL06, LD06] are of inferior quality for contemporary needs. Fixed-density approaches like AA Pat-

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.



(a) Light IS (b) BSDF IS (c) Light IS with occluder (d) BSDF IS with occluder

Figure 18: *We illustrate the integrands (top-right in each column) for a shade point directly illuminated by an area light source for light IS and BSDF IS. (a) Light IS has a smooth underlying integrand when the light source is fully visible, unlike BSDF sampling (b), which sees light boundary as a C0 discontinuity. Partially occluded shade points, on the other hand, always have a C_0 discontinuity (c,d). Figure reproduced from [SSC*19].*



(a) Reference (b) Underlying pixel functions (c) Pixel P (d) Pixel Q

Figure 19: *The convergence rate of MC integration with correlated (jittered) samples depends on the importance function (light vs. BSDF). We analyze two Pixels P & Q directly illuminated by a square area light source in (a). Pixel P is fully visible from the light source, which results in a smooth integrand when performing light source surface area sampling (visualized in bottom-left b) and a convergence rate of $\mathcal{O}(N^{-2})$ (c: dot-dashed green curve, all plots are in log-log scale). C_0 discontinuities in the integrand result in $\mathcal{O}(N^{-1.5})$ convergence, which can happen: when using homogenization or Cranley-Patterson rotation (CPr, solid green curve, more details in [SSC*19]); when the light source is partially occluded (Pixel Q); or when sampling the BSDF (in magenta) since this treats the boundary of the light as a C_0 discontinuity even when the light is fully visible (visualized in the second column of b). Figure reproduced from [SSC*19].*

terns [AHD15] and LDBN [APC*16] do not meet the need for progressive sampling in some Monte Carlo integration scenarios. Some adaptive sampling patterns like Penrose tiling [ODJ04], Polyominoes [Ost07] and Polyhexes [WPC*14] need correction vectors that depend on the number of samples, so they are not progressive in the strict sense [CKK18]. However, both Polyominoes and Polyhexes can be easily adapted for progressive sampling as they rely on local subdivision rules. One could also implement the snapping procedure from Ahmed et al. [ANHD17, Section 4.2] to adapt progressivity.

Thus, among the many blue noise generation and distribution algorithms, only a few, such as ART [ANHD17] and the approach by Perrier et al. [PCX*18], seem relevant for rendering. The first one provides adaptive sampling with a stratification property, while the later uses a complex hierarchical scrambling principle to obtain a higher dimensional sequential sampling pattern with blue noise and low discrepancy properties on 2D projections. As discussed by Perrier et al. [PCX*18], enforcing the sequential property in higher dimensions leads to lower quality in the desired 2D projections, as compared to the one obtained by 2D only samplers such as BNOT or LDBN. Christensen et al. [CKK18, CFS*18] considered both samplers in recent comparisons for Monte Carlo integration. They report that both techniques perform competently, with scrambled LD sequences performing the best. In contrast, Kulla et al. [KCSG18] report that ART [ANHD17] gives favorable results for the *Arnold* rendering framework.

Overall, blue noise sampling is thus still an important alternative to low-discrepancy sequences in rendering, especially if it can be demonstrated to eliminate aliasing-like artifacts. We believe that with improved performance of generating algorithms in high dimensions, blue-noise patterns can significantly improve error and visual quality of renderings, as already predicted by the point process theory, and practice.

5.4. Importance Sampling

Importance sampling (IS) [Coc63] is a widely adopted variance reduction technique in rendering. It works by placing more samples where the integrand is high, and thus introduces adaptive intensity. For integrands with highly varying values across parts of the domain,

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

uniform sampling might take a prohibitively long time to explore all intricate regions (e.g. regions having nearly zero value everywhere except a small range). IS is beneficial in making sure all important regions are adequately sampled, given that such behavior is known a priori.

The improvement in error via importance sampling is very common in rendering [Vea97, Owe13, SNJ*14]. A typical example involves importance sampling a *bidirectional scattering distribution function* (BSDF) vs. a light source. For a square area light source, if we importance sample the light plane for pixels with no occluders in between, we get to integrate a smooth function, resulting in lower error (see Figure 18). In contrast, importance sampling the BSDF would lead to an integrand with a discontinuity, which might increase variance and thus error.

However, the benefits of IS—when performed with pure MC samples—is constant and diminishes as more samples are drawn, i.e. IS does not affect the convergence rate. However, when performed in conjunction with correlated samples, IS can significantly improve the convergence behavior (see Figure 19). Singh et al. [SSC*19] shows that importance sampling combined with such correlated point patterns can indeed provide improved convergence behavior for different parameterizations, i.e. change of variables. As elaborated on in Section 4.1, importance sampling is typically implemented by warping an initial domain where a point distribution generated by a stationary point process is defined, to a new domain. Hence, by warping back the integrand, we can easily get exact and tractable error expressions (17) for these cases.

6. Conclusions and Future Directions

In this work, we provide a survey of techniques developed for analysis of point distributions for rendering based on correlations of point locations. After decades of research, we are now at a stage where error in integral estimators for rendering can be written down exactly in useful forms, which are amenable to analysis to extract insights. But as the recent works show, we do not have to stop there, but instead work on new patterns specifically designed for the integrands we encounter in practice, with the help of the error expressions as summarized in the previous sections.

A very important consideration here is the efficiency of sampling algorithms. Various toolkits (Table 2) have been developed over the years to facilitate their comparisons. Although there exist algorithms that match to any given statistic, e.g. the PCF or expected power spectrum, these are typically very slow to be used in practice for rendering, and do not scale well with the number of dimensions. In contrast, classical algorithms such as low discrepancy sampling work efficiently for any dimensions, but lack the flexibility we need for adaptations based on integrands and error. We thus believe that better understanding the nature of the integrands in the light of the error expressions, and designing efficient sampling algorithms that generate optimal and scalable point sequences while adapting correlations w.r.t. the underlying integrand could be a very important future effort for the rendering community. We hope that our work will serve as an entry point for such efforts.

7. Acknowledgements

We are grateful to all the anonymous reviewers for their constructive remarks. This work was partially supported by the Fraunhofer and Max Planck cooperation program within the German pact for research and innovation (PFI). Kartic Subr was supported by a Royal Society University Research Fellowship, Ravi Ramamoorthi was supported by NSF grant 1451830 and Wojciech Jarosz was partially supported by NSF grant ISS-1812796.

Appendix A: Code Snippets for Stratification Strategies

In this section, we provide code snippets for some commonly used 2D stratification sampling strategies. We have borrowed the multi-jittered and correlated MJ code snippets from Kensler [Ken13].

```
regularGrid2D(samples){
  for(row=0; row<√N;row++){
    for(col=0;col<√N;col++){
      p.x = (col + 0.5)/√N;
      p.y = (row + 0.5)/√N;
      samples[count] = p;
      count++;
    }
  }
}

uniformJitter2D(samples){
  randomShift = drand48();
  for(row=0; row< √N;row++){
    for(col=0;col< √N;col++){
      p.x = ((col + 0.5)/√N) + randomShift;
      p.y = ((row + 0.5)/√N) + randomShift;
      samples[count] = p;
      count++;
    }
  }
}

isotropicJitter2D(samples){
  randomShift = drand48(), randomRotate = 90*drand48();
  for(row=0; row< √N;row++){
    for(col=0;col< √N;col++){
      // x and y in [0,1) range
      x = ((col + 0.5)/√N) + randomShift;
      y = ((row + 0.5)/√N) + randomShift;
      //Bring x and y in [-0.5,0.5) range
      xt = x - 0.5;
      yt = y - 0.5;
      //Rotate the samples
      cosval = cos(randomRotate * π/180.0 );
      sinval = sin(randomRotate * π/180.0 );
      p.x = xt * cosval - yt * sinval;
      p.y = xt * sinval + yt * cosval;
      //Bring samples back in the domain
      toroidalWrap(p);
      samples[count] = p;
      count++;
    }
  }
}

randomJitter2D(samples){
  for(row=0; row<√N;row++){
    for(col=0;col<√N;col++){
      p.x = (col + drand48())/√N;
      p.y = (row + drand48())/√N;
      samples[count] = p;
      count++;
    }
  }
}

gaussianJitter2D(samples){
  for(row=0; row<√N;row++){
    for(col=0;col<√N;col++){
      limit=0.5, randX=drand48(), randY=drand48();
      xcenter = randX-limit;
      ycenter = randY-limit;
```

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

BNOT	[dGBOD12]	http://www.geometry.caltech.edu/BlueNoise/
PSA	[HSD13]	https://code.google.com/archive/p/psa/
Polyhex	[WPC*14]	https://github.com/polyhex-sampling/sampler
STK	[PSC*15]	https://github.com/stk-team/stk
LDBN	[APC*16]	https://projet.liris.cnrs.fr/ldbn/
EEA	[SSJ16]	https://github.com/sinbag/EmpiricalErrorAnalysis
UTK	[PCX*18]	https://github.com/utk-team/utk/

Table 2: *Various implementations have been released over the past few years that provide tools to analyze different sampling patterns in the spatial and spectral domain. We are listing a few of them that have active online footprint. This list is by no means exhaustive.*

```
      envelop = exp(-limit*limit)/(1-exp(-limit*limit));
      xgaussjitter = exp(-xcenter*xcenter) - envelop;
      ygaussjitter = exp(-ycenter*ycenter) - envelop;
      p.x = (col + xgaussjitter)/√N;
      p.y = (row + ygaussjitter)/√N;
      samples[count] = p;
      count++;
    }
  }
}

NrooksSampler2D(samples){
  //generate 1D stratified x-coordinates
  for(col=1;col<=N;col++){
    p.x = (col + drand48())/N;
    xarray[count] = p.x;
  }
  //generate 1D stratified y-coordinates
  for(row=1;row<=N;row++){
    p.y = (row + drand48())/N;
    yarray[count] = p.y;
  }
  //in-place random shuffling
  random_shuffle(xarray);
  random_shuffle(yarray); // optional

  samples = (xarray,yarray);
}

multiJitter2D(samples){
  m = √N, n = √N;
  //Producing a canonical arrangement
  for (int j = 0; j < n; ++j) {
    for (int i = 0; i < m; ++i) {
      samples[j * m + i].x = (i + (j + drand48()) / n) / m;
      samples[j * m + i].y = (j + (i + drand48()) / m) / n;
    }
  }
  //Shuffling the canonical arrangement
  for (int j = 0; j < n; ++j) {
    for (int i = 0; i < m; ++i) {
      int k = j + drand48() * (n - j);
      std::swap(samples[j * m + i].x, samples[k * m + i].x);
    }
  }

  for (int i = 0; i < m; ++i) {
    for (int j = 0; j < n; ++j) {
      int k = i + drand48() * (m - i);
      std::swap(samples[j * m + i].y,samples[j * m + k].y);
    }
  }
}

///Correlated MultiJitter is a simple variant of MJ
correlatedMJ2D(samples){
  m = √N, n = √N;
  //Producing a canonical arrangement
  for (int j = 0; j < n; ++j) {
    for (int i = 0; i < m; ++i) {
      samples[j * m + i].x = (i + (j + drand48()) / n) / m;
      samples[j * m + i].y = (j + (i + drand48()) / m) / n;
    }
  }
  //Uniform shuffling of x-coordinates
  for (int j = 0; j < n; ++j) {
    int k = j + drand48() * (n - j);
    for (int i = 0; i < m; ++i) {
      std::swap(samples[j * m + i].x, samples[k * m + i].x);
    }
  }

  //Uniform shuffling of y-coordinates
  for (int i = 0; i < m; ++i) {
    int k = i + drand48() * (m - i);
    for (int j = 0; j < n; ++j) {
      std::swap(samples[j * m + i].y,samples[j * m + k].y);
    }
  }
}
```

Listing 1: *Psuedo code is provided for common stratification strategies: regular grid, uniform jitter, random jitter, N-rooks, multi-jitter, correlated multi-jitter*

References

[AHD15] AHMED A. G. M., HUANG H., DEUSSEN O.: AA patterns for point sets with controlled spectral properties. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 34*, 6 (Oct. 2015), 212:1–212:8. 15

[ANHD17] AHMED A. G. M., NIESE T., HUANG H., DEUSSEN O.: An adaptive point sampler on a regular lattice. *ACM Transactions on Graphics (Proc. SIGGRAPH) 36*, 4 (July 2017), 138:1–138:13. 15

[APC*16] AHMED A. G., PERRIER H., COEURJOLLY D., OSTROMOUKHOV V., GUO J., YAN D.-M., HUANG H., DEUSSEN O.: Low-discrepancy blue noise sampling. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 35*, 6 (2016). 15, 17

[AW87] AMANATIDES J., WOO A.: A fast voxel traversal algorithm for ray tracing. In *EG 1987-Technical Papers* (1987), Eurographics Association. 11

[BCT01] BRANDOLINI L., COLZANI L., TORLASCHI A.: Mean square decay of Fourier transforms in Euclidean and non-Euclidean spaces. *Tohoku Mathematical Journal, Second Series 53*, 3 (2001), 467–478. 12

[BD16] BILLEN N., DUTRÉ P.: Line sampling for direct illumination. *Computer Graphics Forum (EGSR) 35*, 4 (2016), 45–55. 14

[BGAM12] BARRINGER R., GRIBEL C. J., AKENINE-MÖLLER T.: High-quality curve rendering using line sampled visibility. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Nov. 2012), 162:1–162:10. 14

[Bra00] BRACEWELL R.: *The Fourier Transform and its Applications*. Electrical engineering series. McGraw Hill, 2000. 8

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (July 2009), 86:1–86:8. 14

[BSS*13] BELCOUR L., SOLER C., SUBR K., HOLZSCHUCH N., DURAND F.: 5D covariance tracing for efficient defocus and motion blur. *ACM Transactions on Graphics 32*, 3 (July 2013), 31:1–31:18. 9

[CFS*18] CHRISTENSEN P., FONG J., SHADE J., WOOTEN W., SCHUBERT B., KENSLER A., FRIEDMAN S., KILPATRICK C., RAMSHAW

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

C., BANNISTER M., RAYNER B., BROUILLAT J., LIANI M.: Renderman: An advanced path-tracing architecture for movie rendering. *ACM Transactions on Graphics 37*, 3 (Aug. 2018), 30:1–30:21. 1, 15

[Chr18] CHRISTENSEN P.: *Progressive sampling strategies for disk light sources*. Tech. rep., 18-02, Pixar Animation Studios, 2018. 9, 2018. 12

[CKK18] CHRISTENSEN P., KENSLER A., KILPATRICK C.: Progressive multi-jittered sample sequences. *Computer Graphics Forum (EGSR) 37*, 4 (Jun 2018). 14, 15

[CM67] COVEYOU R. R., MACPHERSON R. D.: Fourier analysis of uniform random number generators. *J. ACM 14*, 1 (Jan. 1967), 100–119. 3

[CO89] CRUZ-ORIVE L. M.: On the precision of systematic sampling: a review of matheron's transitive methods. *Journal of Microscopy 153*, 3 (1989), 315–333. 12

[Coc63] COCHRAN W. G.: *Sampling Techniques, 2nd Edition*. John Wiley, 1963. 15

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *Computer Graphics (SIGGRAPH) 5*, 1 (1986), 51–72. 12, 13, 14

[CP76] CRANLEY R., PATTERSON T. N. L.: Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis 13*, 6 (1976), 904–914. 9, 14

[CPC84] COOK R. L., PORTER T., CARPENTER L.: Distributed ray tracing. *Computer Graphics (SIGGRAPH) 18*, 3 (Jan. 1984), 137–145. 12

[CSHD03] COHEN M. F., SHADE J., HILLER S., DEUSSEN O.: Wang tiles for image and texture generation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (July 2003), 287–294. 14

[DEM96] DOBKIN D. P., EPPSTEIN D., MITCHELL D. P.: Computing the discrepancy with applications to supersampling patterns. *ACM Transactions on Graphics 15*, 4 (Oct. 1996), 354–376. 14

[dGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Nov. 2012), 171:1–171:11. 5, 14, 17

[DHS*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F. X.: A frequency analysis of light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH) 24*, 3 (July 2005), 1115–1126. 13

[DK08] DAMMERTZ S., KELLER A.: Image synthesis by rank-1 lattices. In *Monte Carlo and Quasi-Monte Carlo Methods 2006* (2008), Keller A., Heinrich S., Niederreiter H., (Eds.), Springer Berlin Heidelberg, pp. 217–236. 13

[Dur11] DURAND F.: *A frequency analysis of Monte-Carlo and other numerical integration schemes*. Tech. Rep. MIT-CSAILTR-2011-052, CSAIL, MIT,, February 2011. 8, 9, 14

[DW85] DIPPÉ M. A. Z., WOLD E. H.: Antialiasing through stochastic sampling. *Computer Graphics (SIGGRAPH) 19*, 3 (July 1985), 69–78. 14

[Fat11] FATTAL R.: Blue-noise point sampling using kernel density model. *ACM Transactions on Graphics (Proc. SIGGRAPH) 30*, 4 (July 2011), 48:1–48:12. 14

[GDAM10] GRIBEL C. J., DOGGETT M., AKENINE-MÖLLER T.: Analytical motion blur rasterization with compression. In *High-Performance Graphics* (2010), pp. 163–172. 14

[GF16] GEORGIEV I., FAJARDO M.: Blue-noise dithered sampling. *ACM SIGGRAPH 2016 Talks* (2016), 35:1–35:1. 14

[GIF*18] GEORGIEV I., IZE T., FARNSWORTH M., MONTOYA-VOZMEDIANO R., KING A., LOMMEL B. V., JIMENEZ A., ANSON O., OGAKI S., JOHNSTON E., HERUBEL A., RUSSELL D., SERVANT F., FAJARDO M.: Arnold: A brute-force production path tracer. *ACM Transactions on Graphics 37*, 3 (Aug. 2018), 32:1–32:12. 1, 14

[Hal64] HALTON J. H.: Algorithm 247: Radical-inverse quasi-random point sequence. *Commun. ACM 7*, 12 (Dec. 1964), 701–702. 14

[HM56] HAMMERSLEY J. M., MORTON K. W.: A new Monte Carlo technique: antithetic variates. *Mathematical Proceedings of the Cambridge Philosophical Society 52* (7 1956), 449–475. 12

[HSD13] HECK D., SCHLÖMER T., DEUSSEN O.: Blue noise sampling with controlled aliasing. *ACM Transactions on Graphics 32*, 3 (July 2013), 25:1–25:12. 5, 6, 14, 17

[IPSS08] ILLIAN J., PENTTINEN A., STOYAN H., STOYAN D. (Eds.): *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley and Sons, Ltd., 2008. 2

[JNSJ11] JAROSZ W., NOWROUZEZAHRAI D., SADEGHI I., JENSEN H. W.: A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics 30*, 1 (Feb. 2011), 5:1–5:19. 14

[JP00] JONES T. R., PERRY R. N.: Antialiasing with line samples. In *EGWR* (London, UK, 2000), Springer-Verlag, pp. 197–206. 14

[JZW*15] JIANG M., ZHOU Y., WANG R., SOUTHERN R., ZHANG J. J.: Blue noise sampling using an sph-based method. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 34*, 6 (Oct. 2015), 211:1–211:11. 14

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive Wang tiles for real-time blue noise. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25*, 3 (July 2006), 509–518. 14

[KCSG18] KULLA C., CONTY A., STEIN C., GRITZ L.: Sony pictures imageworks Arnold. *ACM Transactions on Graphics 37*, 3 (Aug. 2018), 29:1–29:18. 15

[Kel06] KELLER A.: Myths of computer graphics. In *Monte Carlo and Quasi-Monte Carlo Methods 2004* (2006), Niederreiter H., Talay D., (Eds.), Springer Berlin Heidelberg, pp. 217–243. 13

[Kel13] KELLER A.: Quasi-Monte Carlo image synthesis in a nutshell. In *Monte Carlo and Quasi-Monte Carlo Methods 2012* (2013), Dick J., Kuo F. Y., Peters G. W., Sloan I. H., (Eds.), Springer Berlin Heidelberg, pp. 213–249. 14

[Ken13] KENSLER A.: *Correlated Multi-Jittered Sampling*. Tech. Rep. Pixar Technical Memo 13-01, Pixar, March 2013. 16

[KGH*14] KŘIVÁNEK J., GEORGIEV I., HACHISUKA T., VÉVODA P., ŠIK M., NOWROUZEZAHRAI D., JAROSZ W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33*, 4 (July 2014), 103:1–103:13. 14

[KK02] KOLLIG T., KELLER A.: Efficient multidimensional sampling. *Computer Graphics Forum (Proc. Eurographics) 21*, 3 (2002), 557–563. 14

[KN74] KUIPERS L., NIEDERREITER H.: *Uniform Distribution of Sequences*. Wiley, New York, USA, 1974. 4, 7

[KPR12] KELLER A., PREMOZE S., RAAB M.: Advanced (quasi) Monte Carlo methods for image synthesis. In *ACM SIGGRAPH 2012 Courses* (2012), pp. 21:1–21:46. 14

[KTBV16] KAILKHURA B., THIAGARAJAN J. J., BREMER P.-T., VARSHNEY P. K.: Stair blue noise sampling. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 35*, 6 (Nov. 2016), 248:1–248:10. 14

[LD06] LAGAE A., DUTRÉ P.: An alternative for Wang tiles: Colored edges versus colored corners. *ACM Transactions on Graphics 25*, 4 (Oct. 2006), 1442–1459. 14

[LD08] LAGAE A., DUTRÉ P.: A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 27*, 1 (March 2008), 114–129. 7

[Mit87] MITCHELL D. P.: Generating antialiased images at low sampling densities. *SIGGRAPH Computer Graphics 21*, 4 (Aug. 1987), 65–72. 11

[Mit91] MITCHELL D. P.: Spectrally optimal sampling for distribution ray tracing. *SIGGRAPH Computer Graphics 25*, 4 (July 1991), 157–164. 11, 14

[Mit92] MITCHELL D. P.: Ray tracing and irregularities of distribution. In *Third Eurographics Workshop on Rendering* (1992), pp. 61–69. 14

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.

[Mit96] MITCHELL D. P.: Consequences of stratified sampling in graphics. pp. 277–280. 11

[MW04] MØLLER J., WAAGEPETERSEN R. P.: *Statistical inference and simulation for spatial point processes*. Chapman & Hall/CRC, 2003, 2004. 1, 2

[Ney34] NEYMAN J.: On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society 97*, 4 (1934), 558–625. 11

[Nie92] NIEDERREITER H.: *Random Number Generation and quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992. 4, 7, 14

[NNDJ12] NOVÁK J., NOWROUZEZAHRAI D., DACHSBACHER C., JAROSZ W.: Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics (Proc. SIGGRAPH) 31*, 4 (July 2012). 14

[Ö16] ÖZTIRELI A. C.: Integration with stochastic point processes. *ACM Transactions on Graphics 35*, 5 (Aug. 2016), 160:1–160:16. 6, 7, 8, 9, 10, 12, 14

[ODJ04] OSTROMOUKHOV V., DONOHUE C., JODOIN P.-M.: Fast hierarchical importance sampling with blue noise properties. *ACM Transactions on Graphics (Proc. SIGGRAPH) 23*, 3 (Aug. 2004), 488–495. 15

[OG12] ÖZTIRELI A. C., GROSS M.: Analysis and synthesis of point distributions based on pair correlation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (Nov. 2012), 170:1–170:10. 6, 7

[OS18] ÖZTIRELI C., SINGH G.: Sampling analysis using correlations for Monte Carlo rendering. In *ACM SIGGRAPH Asia Courses* (Nov. 2018), ACM. 6

[Ost07] OSTROMOUKHOV V.: Sampling with Polyominoes. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (July 2007). 15

[Owe97] OWEN A. B.: Monte Carlo variance of scrambled net quadrature. *SIAM J. Numer. Anal. 34*, 5 (Oct. 1997), 1884–1910. 14

[Owe03] OWEN A.: Quasi-Monte Carlo sampling. In *SIGGRAPH Monte Carlo Ray Tracing Course Notes* (2003). 14

[Owe13] OWEN A. B.: *Monte Carlo Theory, Methods and Examples*. To be published, 2013. 16

[PCX*18] PERRIER H., COEURJOLLY D., XIE F., PHARR M., HANRAHAN P., OSTROMOUKHOV V.: Sequences with low-discrepancy blue-noise 2-D projections. *Computer Graphics Forum 37*, 2 (2018), 339–353. 15, 17

[PH89] PERLIN K., HOFFERT E. M.: Hypertexture. *SIGGRAPH Computer Graphics 23*, 3 (July 1989), 253–262. 11

[PKK00] PAULY M., KOLLIG T., KELLER A.: Metropolis light transport for participating media. Springer-Verlag, pp. 11–22. 11

[PSC*15] PILLEBOUE A., SINGH G., COEURJOLLY D., KAZHDAN M., OSTROMOUKHOV V.: Variance analysis for Monte Carlo integration. *ACM Transactions on Graphics (Proc. SIGGRAPH) 34*, 4 (July 2015), 124:1–124:14. 7, 9, 10, 12, 14, 17

[RAMN12] RAMAMOORTHI R., ANDERSON J., MEYER M., NOWROUZEZAHRAI D.: A theory of Monte Carlo visibility sampling. *ACM Transactions on Graphics 31*, 5 (Sept. 2012), 121:1–121:16. 8, 11

[RRSG16] REINERT B., RITSCHEL T., SEIDEL H., GEORGIEV I.: Projective blue noise sampling. *Computer Graphics Forum* (2016). 14

[RWCS05] ROVIRA J., WONKA P., CASTRO F., SBERT M.: Point sampling with uniformly distributed lines. In *Point-Based Graphics, 2005 - Eurographics/IEEE VGTC Symposium Proceedings* (2005), pp. 109–118. 14

[SC97] SHIRLEY P., CHIU K.: A low distortion map between disk and square. *J. Graph. Tools 2*, 3 (Dec. 1997), 45–52. 11

[Sha49] SHANNON C. E.: Communication in the presence of noise. *Proc. Institute of Radio Engineers 37*, 1 (1949), 10–21. 8

[SHD11] SCHLÖMER T., HECK D., DEUSSEN O.: Farthest-point optimized point sets with maximized minimum distance. In *High-Performance Graphics* (2011), HPG '11, pp. 135–142. 14

[Shi91] SHIRLEY P.: Discrepancy as a quality measure for sample distributions. In *Eurographics '91* (1991), Elsevier Science Publishers, pp. 183–194. 4, 7

[SJ17] SINGH G., JAROSZ W.: Convergence analysis for anisotropic Monte Carlo sampling spectra. *ACM Transactions on Graphics (Proc. SIGGRAPH) 36*, 4 (July 2017), 137:1–137:14. 9, 12, 13

[SK13] SUBR K., KAUTZ J.: Fourier analysis of stochastic sampling strategies for assessing bias and variance in integration. *ACM Transactions on Graphics (Proc. SIGGRAPH) 32*, 4 (July 2013), 128:1–128:12. 9, 12

[SMJ17] SINGH G., MILLER B., JAROSZ W.: Variance and convergence analysis of Monte Carlo line and segment sampling. *Computer Graphics Forum (EGSR) 36*, 4 (Jun 2017). 14

[SNJ*14] SUBR K., NOWROUZEZAHRAI D., JAROSZ W., KAUTZ J., MITCHELL K.: Error analysis of estimators that use combinations of stochastic sampling strategies for direct illumination. *Computer Graphics Forum 33*, 4 (2014), 93–102. 16

[Sob67] SOBOL I. M.: The distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics 7* (1967), 86–112. 14

[SRFN06] SBERT M., RIGAU J., FEIXAS M., NEUMANN L.: Systematic sampling in image-synthesis. In *Proceedings of the 6th International Conference on Computational Science and Its Applications - Volume Part I* (Berlin, Heidelberg, 2006), ICCSA'06, Springer-Verlag, pp. 449–458. 12

[SSC*19] SINGH G., SUBR K., COEURJOLLY D., OSTROMOUKHOV V., JAROSZ W.: Fourier analysis of correlated Monte Carlo importance sampling. *Computer Graphics Forum* (2019). 4, 9, 10, 15, 16

[SSJ16] SUBR K., SINGH G., JAROSZ W.: Fourier analysis of numerical integration in Monte Carlo rendering: Theory and practice. In *ACM SIGGRAPH Courses* (July 2016). 17

[SZG*13] SUN X., ZHOU K., GUO J., XIE G., PAN J., WANG W., GUO B.: Line segment sampling with blue-noise properties. *ACM Transactions on Graphics (Proc. SIGGRAPH) 32*, 4 (July 2013), 127:1–127:14. 14

[TPD*12] TZENG S., PATNEY A., DAVIDSON A., EBEIDA M. S., MITCHELL S. A., OWENS J. D.: High-quality parallel depth-of-field using line samples. In *High-Performance Graphics* (2012), pp. 23–31. 14

[Uli87] ULICHNEY R.: *Digital Halftoning*. MIT Press, 1987. 7

[Uli88] ULICHNEY R.: Dithering with blue noise. *Proceedings of the IEEE 76*, 1 (Jan 1988), 56–79. 14

[Uns00] UNSER M.: Sampling—50 years after Shannon. *Proceedings of the IEEE 88*, 4 (2000), 569–587. 3, 8

[Vea97] VEACH E.: Robust Monte Carlo methods for light transport simulation, December 1997. 11, 16

[WPC*14] WACHTEL F., PILLEBOUE A., COEURJOLLY D., BREEDEN K., SINGH G., CATHELIN G., DE GOES F., DESBRUN M., OSTROMOUKHOV V.: Fast tile-based adaptive sampling with user-specified Fourier spectra. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33*, 4 (July 2014), 56:1–56:11. 15, 17

[WW11] WEI L.-Y., WANG R.: Differential domain analysis for non-uniform sampling. *ACM Transactions on Graphics (Proc. SIGGRAPH) 30*, 4 (July 2011), 50:1–50:10. 6

[WW17] WONG K.-M., WONG T.-T.: Blue noise sampling using an N-body simulation-based method. *Visual Computer 33*, 6-8 (June 2017), 823–832. 14

[Zar68] ZAREMBA S. K.: The mathematical basis of Monte Carlo and quasi-Monte Carlo methods. *SIAM Rev. 10*, 3 (July 1968), 303–314. 4

© 2019 The Author(s)
Computer Graphics Forum © 2019 The Eurographics Association and John Wiley & Sons Ltd.