Quasi-Monte Carlo Image Synthesis in a Nutshell

Alexander Keller

Abstract This self-contained tutorial surveys the state of the art in quasi-Monte Carlo rendering algorithms as used for image synthesis in the product design and movie industry. Based on the number theoretic constructions of low discrepancy sequences, it explains techniques to generate light transport paths to connect cameras and light sources. Summing up their contributions on the image plane results in a consistent numerical algorithm, which due to the superior uniformity of low discrepancy sequences often converges faster than its (pseudo-) random counterparts. In addition, its deterministic nature allows for simple and efficient parallelization while guaranteeing exact reproducibility. The underlying techniques of parallel quasi-Monte Carlo integro-approximation, the high speed generation of quasi-Monte Carlo points, treating weak singularities in a robust way, and high performance ray tracing have many applications outside computer graphics, too.

1 Introduction

"One look is worth a thousand words" characterizes best the expressive power of images. Being able to visualize a product in a way that cannot be distinguished from a real photograph before realization can greatly help to win an audience. As ubiquitous in many movies, a sequence of such images can tell whole stories in a captive and convincing way. As a consequence of the growing demand and benefit of synthetic images, a substantial amount of research has been dedicated to finding more efficient rendering algorithms.

The achievable degree of realism depends on the physical correctness of the model and the consistency of the simulation algorithms. While modeling is beyond the focus of this tutorial, we review the fundamentals in Sec. 2. The paradigm of consistency is discussed in the next Sec. 1.1 as it is key to the quasi-Monte Carlo

Alexander Keller
NVIDIA, Fasanenstraße 81, 10623 Berlin, Germany, e-mail: keller.alexander@gmail.com

techniques in Sec. 3 that are at the heart of the deterministic rendering algorithms explored in Sec. 4.

On a historical note, the investigation of quasi-Monte Carlo methods in computer graphics goes back to Shirley [69] and Niederreiter [54], and received early industrial attention [60]. This comprehensive tutorial surveys the state of the art, includes new results, and is applicable far beyond computer graphics, as for example in financial mathematics and general radiation transport simulation.

1.1 Why Consistency matters most

Analytic solutions in light transport simulation are only available for problems too simple to be of practical relevance, although some of these settings are useful in understanding and testing algorithms [31]. In practical applications, functions are high-dimensional and contain discontinuities that cannot be located efficiently. Therefore approximate solutions are computed using numerical algorithms. In the following paragraphs, we clarify the most important notions, as they are often confused, especially in marketing.

Consistency

Numerical algorithms, whose approximation error vanishes as the sample size increases, are called consistent. Note that consistency is not a statement with respect to the speed of convergence. Within computer graphics, consistency guarantees image synthesis without persistent artifacts such as discretization artifacts introduced by a rendering algorithm; the results are consistent with the input model and in that sense the notion of consistency is understandable without any mathematical background. While many commercial implementations of rendering algorithms required expert knowledge to tweak a big set of parameters until artifacts due to intermediate approximations become invisible, the design of many recent rendering algorithms follows the paradigm of consistency. As a result, users can concentrate on content creation, because light transport simulation has become as simple as pushing the "render"-button in an application.

Unbiased Monte Carlo Algorithms

The bias of an algorithm using random numbers is the difference between the mathematical object and the expectation of the estimator of the mathematical object to be approximated. If this difference is zero, the algorithm is called unbiased. However, this property alone is not sufficient, because an estimator can be unbiased but not consistent, thus even lacking convergence. In addition, biased but consistent algorithms can handle problems that unbiased algorithms cannot handle: For example,

density estimation allows for efficiently handling the problem of "insufficient techniques" (for the details see Sec. 4.4.1).

The theory of many unbiased Monte Carlo algorithms is based on independent random sampling, which is used at the core of many proofs in probability theory and allows for simple parallelization and for estimating the variance as a measure of error.

Physically Based Modeling

Physically based modeling subsumes the creation of input for image synthesis algorithms, where physical entities such as measured data for light sources and optical properties of matter or analytic models thereof are used for the input specification. Modeling with such entities and relying on consistent light transport simulation to many users is much more natural as compared to tweaking lights and materials in order to deliver photorealistic results.

Although often confused in computer graphics, physically correct rendering is not equivalent to unbiased Monte Carlo algorithms: Even non-photorealistic images can be rendered using unbiased Monte Carlo algorithms. In addition, so far none of the physically based algorithms can claim to comply with all the laws of physics, because they are simply not able to efficiently simulate all effects of light transport and therefore cannot be physically correct.

Deterministic Consistent Numerical Algorithms

While independence and unpredictability characterize random numbers, these properties often are undesirable for computer simulations: Independence compromises the speed of convergence and unpredictability disallows the exact repetition of a computer simulation. Mimicking random numbers by pseudo-random numbers generated by deterministic algorithms, computations become exactly repeatable, however, arbitrarily jumping ahead in such sequences as required in scalable parallelization often is inefficient due to the goal of emulating unpredictability.

In fact, deterministic algorithms can produce samples that approximate a given distribution much better than random numbers can. By their deterministic nature, such samples must be correlated and predictable. The lack of independence is not an issue, because independence is not visible in an average anyhow and consistency can be shown using number theoretic arguments instead of probabilistic ones. In addition, partitioning such sets of samples and leaping in such sequences of samples can be highly efficient.

As it will be shown throughout the article, advantages of such deterministic consistent numerical algorithms are improved convergence, exact reproducibility, and simple communication-avoiding parallelization. Besides rendering physically based models, these methods also apply to rendering non-physical models that often are chosen to access artistic freedom or to speed up the rendering process. The illustra-

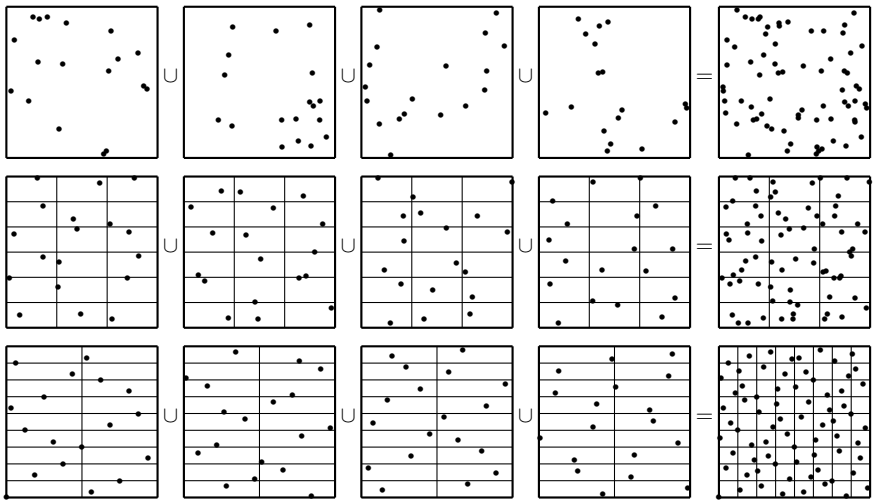

Fig. 1 Illustration of the difference between unbiased and deterministic consistent uniform sampling: The top row shows four independent sets of 18 points each and their union as generated by a pseudo-random number generator. The middle row shows independent realizations of so-called stratified samples with their union that result from uniformly partitioning the domain and independently sampling inside each resulting interval in order to increase uniformity. However, points can come arbitrarily close together along interval boundaries and there is no guarantee for their union to improve upon uniformity. The bottom row shows the union of four contiguous blocks of 18 points of the Halton sequence. As opposed to the pseudorandom number generator and stratified sampling, the samples of the Halton sequence are more uniform, nicely complement each other in the union, and provide a guaranteed minimum distance and intrinsic stratification along the sequence.

tion in Fig. 1 provides some initial intuition of the concepts and facts discussed in this section.

2 Principles of Light Transport Simulation

Implementing the process of taking a photo on a computer involves the simulation of light transport. This in turn requires a mathematical model of the world: A boundary representation with attached optical properties describes the surfaces of the objects to be visualized. Such a model may be augmented by the optical properties of volumes, spectral properties, consideration of interference, and many more physical phenomena. Once the optical properties of the camera system and the light sources are provided, the problem specification is complete.

The principles of light transport simulation are well covered in classic textbooks on computer graphics: Currently, [66] is the most updated standard reference, [16] is a classic reference available for free on the internet, and [70] can be considered

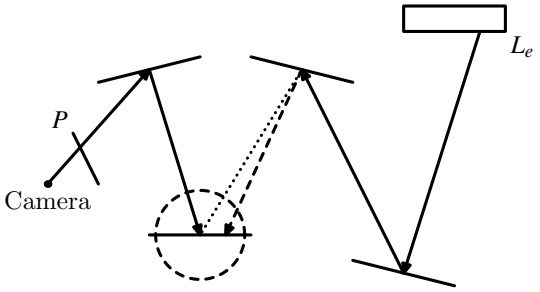

Fig. 2 Bidirectional generation of light transport paths: A path segment started from the camera and a path segment started from a light source L_e can be connected by a shadow ray (dotted line, see Sec. 4.4.2), which checks whether the vertices to connect are mutually visible. Alternatively, the basic idea of photon mapping (see Sec. 4.4.1) is to relax the precise visibility check by allowing for a connection of both path segments if their end points are sufficiently close as indicated by the dashed circle. Both techniques are illustrated for identical path length, which is the reason for the dashed prolongation of the light path segment for photon mapping.

a primer and kick start. Recent research is well surveyed in [82, 22, 6] along with profound investigations of numerical algorithms and their issues.

2.1 Light Transport along Paths

Light transport simulation consists of identifying all paths that connect cameras and light sources and integrating their contribution to form the synthetic image. Fig. 2 illustrates the principles of exploring path space.

One way of generating light transport paths is to follow the trajectories of photons emitted from the light sources along straight line segments between the interactions with matter. However, no computational device can simulate a number of photons sufficiently large to represent reality and hence the direct simulation often is not efficient.

When applicable, light transport paths can be reversed due to the Helmholtz reciprocity principle and trajectories can be traced starting from the camera sensor or eye. Most efficient algorithms connect such camera and light path segments and therefore are called bidirectional.

Vertices of paths can be connected by checking their mutual visibility with respect to a straight line or by checking their mutual distance with respect to a suitable metric. While checking the mutual visibility is precise, it does not allow for efficiently simulating some important contributions of light caused by surfaces that are highly specular and/or transmissive, which is known as the problem of insufficient techniques [42]. In such cases, connecting paths by merging two vertices that are sufficiently close helps. The resulting bias can be controlled by the maximum distance allowed for merging vertices.

The interactions with matter need to be modeled: Bidirectional scattering distribution functions (BSDFs) describe the properties of optical interfaces, while scattering and absorption cross sections determine when to scatter in volume using the distribution given by a phase function [66]. Similarly, the optical properties of the light sources and sensors have to be mathematically modeled. For cameras, models range from a simple pinhole to complete lenses allowing for the simulation of depth of field and motion blur. Light sources often are characterized by so-called light profiles. All these physical properties can be provided in measured form, too, which in many cases provides quality superior to the current analytic models.

Beyond that, optical properties can be modeled as functions of wavelength across the spectrum of light in order to overcome the restriction of the common approach using only three selected wavelengths to represent red, green, and blue and to enable dispersion and fluorescence. The simulation of effects due to polarization and the wave character of light are possible to a certain extent, however, are subject to active research.

While modeling with real entities is very intuitive, it must be noted that certain violations of physics can greatly help the efficiency of rendering and/or help telling stories at the cost of systematic errors.

2.2 Accelerated Ray Tracing and Visibility

The boundary of the scene often is stored as a directed acyclic graph, which allows for referencing parts of the scene multiple times to instance them at multiple positions in favor of a compact representation. Complex geometry like for example hair, fur, foliage, or crowds often are generated procedurally, in which case the call graph implicitly represents the scene graph. Triangles, quadrangles, or multi-resolution surfaces, which include subdivision surfaces, are the most common geometric primitives used for boundary representation.

The vertices of a light transport path are connected by straight line segments. First, these can be found by tracing rays from a point x into a direction ω to identify the closest point of intersection $h(x,\omega)$ with the scene boundary. A second way to construct paths is to connect two vertices x and y of two different path segments. This can be accomplished by checking the mutual visibility $V(x,y)$, which is zero if the straight line of sight between the points x and y, a so-called shadow ray, is occluded, one otherwise. As a third operation, two vertices can be merged, if their distance with respect to a metric is less than a threshold. Efficient implementations of the three operations all are based on hierarchal culling (see [39, 35] for a very basic primer).

In order to accelerate ray tracing, the list of objects and/or space are recursively partitioned. Given a ray to be traced, traversal is started from the root node descending into a subtree, whenever the ray intersects this part of the scene. Most parts of the scene thus are hierarchically culled and never touched. In case the cost of the construction of such an auxiliary acceleration hierarchy can be amortized over

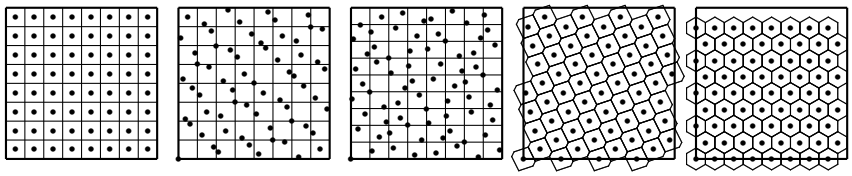

Fig. 3 Examples of $(\mathscr{M},\mu)$-uniform point sets with an outline of their partition $\mathscr{M}$ (from left to right): $n = 8\times 8$ points of a Cartesian grid, the first $n = 64$ points of the Sobol' sequence, the first $n = 72$ points of the Halton sequence, and the maximized minimum distance rank-1 lattice with $n = 64$ points and generator vector $\mathbf{g} = (1,28)$. The hexagonal grid with $n = 72$ point is shown for comparison, as it cannot tile the unit square due to its irrational basis.

tracing many paths, it makes sense to store it partially or completely. Checking the mutual visibility by a shadow ray is even more efficient, since the traversal can be stopped upon any intersection with the boundary, while tracing a ray requires to find the intersection closest to its origin.

Efficiently merging vertices follows the same principle of hierarchical culling [35]: Given two sets of points in space, the points of the one set that are at a maximum given distance from the points of the other set are found by hierarchically subdividing space and pruning the search for partitions of space that cannot overlap within the given distance.

3 Principles of Quasi-Monte Carlo Integro-Approximation

Image synthesis can be considered an integro-approximation problem of the form

$$g(\mathbf{y}) := \int_X f(\mathbf{x},\mathbf{y}) d\mu(\mathbf{x}) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{x}_i,\mathbf{y}), \tag{1}$$

where $f(\mathbf{x},\mathbf{y})$ is the measurement contribution to a location $\mathbf{y}$ by a light transport path identified by $\mathbf{x}$. We will focus on deterministic linear algorithms [78] to consistently determine the whole image function g for all pixels $\mathbf{y}$ using one low discrepancy sequence $\mathbf{x}_i$ of deterministic sample points. The principles of such quasi-Monte Carlo methods have been introduced to a wide audience in [55], which started a series of MCQMC conferences, whose proceedings contain almost all recent developments in quasi-Monte Carlo methods. Many of the results and developments are summarized in recent books [72, 49, 10].

Before reviewing the algorithms to generate low discrepancy sequences in Sec. 3.1 and techniques resulting from their number theoretic construction in Sec. 3.2, error bounds are discussed with respect to measures of uniformity.

Uniform Sampling, Stratification, and Discrete Density Approximation

A common way to generate a discrete approximation of a density comprises the creation of uniformly distributed samples that are transformed [25, 9]. For many such transformations, an improved uniformity results in a better discrete density approximation. Measures of uniformity often follow from proofs of error bounds (see the next paragraph) as a result of the attempt to bound the error by a product of properties of the sampling points and the function as used in for example Thm. 1. For the setting of computer graphics, where X is a domain of integration, $\mathscr{B}$ are the Borel sets over X, and μ the Lebesgue measure, a practical measure of uniformity is given by

Definition 1 (see [56]). Let $(X,\mathscr{B},\mu)$ be an arbitrary probability space and let $\mathscr{M}$ be a nonempty subset of $\mathscr{B}$. A point set P_n of n elements of X is called $(\mathscr{M},\mu)$-uniform if

$$\sum_{i=0}^{n-1}\chi_M(\mathbf{x}_i)=\mu(M)\cdot n \qquad \text{for all } M\in\mathscr{M},$$

where $\chi_M(\mathbf{x}_i)=1$ if $\mathbf{x}_i\in M$, zero otherwise.

Fig. 3 shows examples of $(\mathscr{M},\mu)$-uniform points from $X=[0,1)^2$ that obviously can only exist if the measures $\mu(M)$ are rational numbers with the same denominator n [56]. While the subset $\mathscr{M}$ may consist of the Voronoi regions of a lattice, it also may consist of axis aligned intervals of the form given by

Definition 2 (see [56]). An interval of the form

$$E(p_1,\ldots,p_s):=\prod_{j=1}^{s}\left[\frac{p_j}{b_j^{d_j}},\frac{p_j+1}{b_j^{d_j}}\right)\subseteq[0,1)^s$$

for $0\le p_j<b_j^{d_j}$ and integers $b_j,d_j\ge 0$ is called an *elementary interval*.

As compared to the original definition in [55, p. 48], which considers the special case of b-adic intervals, i.e. $b_j=b$ (for $b_j=2$, the intervals are called dyadic), different bases b_j are allowed for each dimension j to include a wider variety of point sets [52, 56]. Representing numbers in base b_j, d_j can be thought of as the number of digits and fixes the resolution in dimension j, which allows for specifying an elementary interval by its coordinates $p_1,\ldots,p_s$.

Characterizations of uniformity beyond the stratification properties imposed by $(\mathscr{M},\mu)$-uniformity (see Fig. 3) include the maximum minimum distance

$$d_{\min}(P_n):=\min_{0\le i<j<n}\|\mathbf{x}_j-\mathbf{x}_i\|_T$$

of the points on the torus $T=[0,1)^s$ [33], and their deviation from uniformity measured by various kinds of discrepancy [55].

In many applications, uniform points are transformed to approximate a continuous density. The quality of such a discrete density approximation can be judged by the star-discrepancy

$$D^*(p,P_n) := \sup_{A=\prod_{j=1}^{s}[0,a_j)\subset[0,1)^s} \left| \int_{[0,1)^s} \chi_A(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} - \frac{1}{n}\sum_{i=0}^{n-1} \chi_A(\mathbf{x}_i) \right|$$

with respect to the density p [25]. Discrepancies can be understood as integration errors, where the exact measure of a test set A with respect to p is compared to the average number of points in that set. For $p \equiv 1$, we have the so-called star-discrepancy $D^*(P_n) := D^*(1,P_n,)$ [55], which is of central importance for quasi-Monte Carlo methods: Low discrepancy point sequences have $D^*(P_n) \in \mathscr{O}\left(\frac{\log^s n}{n}\right)$, while uniform random numbers can only achieve an order of $\mathscr{O}\left(\sqrt{\frac{\log\log n}{n}}\right)$ manifesting the asymptotic inferiority of random sampling with respect to discrete density approximation.

Error Bounds

Using $(\mathscr{M},\mu)$-uniformity, an error bound for the integro-approximation problem in Eq. 1 is given by

Theorem 1 (see [33]). *Let $(X,\mathscr{B},\mu)$ be an arbitrary probability space and let $\mathscr{M} = \{M_1,\ldots,M_k\}$ be a partition of X with $M_j \in \mathscr{B}$ for $1 \le j \le k$. Then for any $(\mathscr{M},\mu)$-uniform point set $\mathscr{P} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ and any bounded function f, which restricted to X is μ-integrable, we have*

$$\left\| \frac{1}{n}\sum_{i=0}^{n-1} f(\mathbf{x}_i,\mathbf{y}) - \int_X f(\mathbf{x},\mathbf{y}) d\mu(\mathbf{x}) \right\| \le \sum_{j=1}^{k} \mu(M_j) \left\| \sup_{\mathbf{x}\in M_j} f(\mathbf{x},\mathbf{y}) - \inf_{\mathbf{x}\in M_j} f(\mathbf{x},\mathbf{y}) \right\|$$

for any suitable norm $\|\cdot\|$.

In analogy to the Monte Carlo case [15], the above theorem has been derived in order to prove the convergence of quasi-Monte Carlo methods for Eq. 1 in the setting of computer graphics, where the only properties of f that are easily accessible are square integrability and boundedness. By omitting $\mathbf{y}$, the above theorem reduces to an error bound for quasi-Monte Carlo integration as originally developed in [56, Thm. 2], which improved the derivation and results obtained for Riemann integrable functions [24].

Other than trivial worst case bounds, the theorem does not provide a rate of convergence, which is the price for its generality. However, including more knowledge about the function f by restricting the function class allows one to obtain much better error bounds along with measures of uniformity: The Koksma-Hlawka inequality [55] bounds the error by a product of the star discrepancy of the point set and the

variation of the function in the sense of Hardy and Krause, bounds for functions with sufficiently fast decaying Fourier coefficients are found in [72], and the error for integrating Lipschitz functions can be bounded by a product of the Lipschitz constant and the maximum minimum distance of a lattice of points [8].

While quasi-Monte Carlo methods allow for improved convergence rates as compared to Monte Carlo methods, the variance of an estimate cannot be consistently computed due to the lack of independence. As a compromise to this issue, randomized quasi Monte Carlo methods [4, 62] have been introduced that sacrifice some uniformity of the sample points in order to control adaptive termination by unbiased variance estimation. As we focus on deterministic algorithms only, this is not an option and we refer to a deterministic variant of termination by comparing differences of norms of intermediate results as introduced in [65]. In computer graphics such norms should reflect the properties of the human visual system and often the L^2-norm [11, Sec. 3.5] is appropriate to measure error.

3.1 Algorithms for Low Discrepancy Sequences

Most known constructions of low discrepancy sequences imply sequences of $(\mathscr{M},\mu)$-uniform point sets (see [56, Rem. 1] and [33]) that guarantee Eq. 1 to converge. In the following, such mappings from $\mathbb{N}_0$ into the s-dimensional unit cube $[0,1)^s$ are surveyed with respect to their algorithmic principles that enable the techniques reviewed in Sec. 3.2. Note that the Weyl sequence [85] is irrational and as such cannot fulfill the condition of $(\mathscr{M},\mu)$-uniformity. It is therefore excluded from our considerations, since most proofs, and especially computers, rely on rational numbers.

3.1.1 Radical Inversion

A digital radical inverse

$$\begin{aligned}&\Phi_{b,C}:\mathbb{N}_0\to\mathbb{Q}\cap[0,1)\\ i=\sum_{l=0}^{M-1}a_l(i)b^l\mapsto&\left(b^{-1}\cdots b^{-M}\right)\left[C\begin{pmatrix}a_0(i)\\ \vdots\\ a_{M-1}(i)\end{pmatrix}\right]\end{aligned}\tag{2}$$

in a prime power base b is computed using a generator matrix C, where the matrix-vector multiplications are performed in the finite field $\mathbb{F}_b$ (for the theory and mappings from and to $\mathbb{F}_b$ see [55]). While in theory these matrices are infinite-dimensional, in practice they are finite due to the finite precision of computer arithmetic. The inverse mapping $\Phi_{b,C}^{-1}$ exists, if C is regular. M is the number of digits, which allows for generating up to $N=b^M$ points.

The time to compute digital radical inverses is far from negligible in many applications. Efficient implementations use tables of precomputed terms [14], take advantage of bit vector arithmetic in $b = 2$ [82], or enumerate the radical inverses using the Gray-code order [67]. Cancellation errors of floating point arithmetic are avoided by ordering summations and computing in integers as long as possible. Note that already the conversion to floating point numbers by the multiplication with $\left(b^{-1} \cdots b^{-M}\right)$ causes collisions of numbers that were different in integer representation.

Van der Corput Sequence

Selecting the identity matrix I as generator matrix results in the points $\Phi_b(i) := \Phi_{b,I}(i) = \sum_{l=0}^{\infty} a_l(i) b^{-l-1}$ of the van der Corput sequence, which is the simplest radical inverse. The mapping reflects the digits $a_l(i)$ of the index i represented in base b at the decimal point. Obviously the computation is finite, as i has only finitely many digits $a_l(i) \neq 0$.

For $0 \leq i < b^m$, the mapping $b^m \Phi_b(i)$ is a permutation and hence the first b^m points of the sequence $\Phi_b(i)$ are equidistantly spaced with a distance of $\frac{1}{b^m}$. Furthermore, this implies that partitioning the van der Corput sequence into contiguous blocks of length b^m, the integer parts of the points within each block multiplied by b^m must be permutations, too. Many of the techniques described in this article rely on these properties and their generalizations.

Another interesting property of the van der Corput sequence is its intrinsic stratification [33]: For example, $\Phi_2(i) < \frac{1}{2}$ for even i and $\Phi_2(i) \geq \frac{1}{2}$ otherwise. In general,

$$\Phi_b(k + l \cdot b^m) \in [\Phi_b(k), \Phi_b(k) + b^{-m}) \text{ for } l \in \mathbb{N}_0.$$

While this property is very useful, it also is the reason why pseudo-random number generators cannot just be replaced by the van der Corput sequence (and radical inverses in general): Already a two-dimensional vector assembled by subsequent numbers from the van der Corput sequence is not uniformly distributed.

3.1.2 Scrambling

Scrambling a set of points on $H = [0,1)$ comprises the following steps:

1. Partition H into b equal intervals $H_1, H_2, \ldots, H_b$.
2. Permute these intervals.
3. For $h \in \{1, 2, \ldots, b\}$, recursively repeat the procedure starting out with $H = H_h$.

Formalizing the scrambling of the i-th point of a sequence represented in base b as defined in Eq. 2 yields the scrambled digits

$$
\begin{aligned}
a'_{i,0} &:= \pi\left(a_0(i)\right)\\
a'_{i,1} &:= \pi_{a_0(i)}\left(a_1(i)\right)\\
&\vdots\\
a'_{i,l} &:= \pi_{a_0(i),a_1(i),\ldots,a_{l-1}(i)}\left(a_l(i)\right),\\
&\vdots
\end{aligned}
$$

where the l-th permutation $\pi_{a_0(i),a_1(i),\ldots,a_{l-1}(i)} : \{0,\ldots,b-1\} \to \{0,\ldots,b-1\}$ depends on the $l-1$ leading digits $a_0(i), a_1(i), \ldots, a_{l-1}(i)$. The mapping is bijective, because it is based on the sequential application of permutations.

While obviously this procedure becomes finite by the finite precision of computation, uniformly distributed points are mapped to uniformly distributed points. Originally, these properties combined with random permutations were introduced to randomize uniform points sets [62, 63]. However, many deterministic optimizations of low discrepancy sequences in fact can be represented as scramblings with deterministic permutations. Note that using a regular generator matrix $C \neq I$ in Eq. 2 already can be considered a deterministic scrambling of the van der Corput sequence.

3.1.3 Halton Sequence and Hammersley Points

The Halton sequence [21]

$$\mathbf{x}_i = \left(\Phi_{b_1}(i), \ldots, \Phi_{b_s}(i)\right)$$

has been constructed by using one van der Corput sequence for each component, where the bases b_j are relatively prime. Replacing one of the components by $\frac{i}{n}$ results in n points that form the Hammersley point set. As compared to the Halton sequence, where by construction subsequent points fill the largest holes in space, the Hammersley points are even more uniformly distributed, however, at the price of not being extensible.

Although the Halton sequence is of low discrepancy, it has the undesirable property that projections are not as well distributed as they could be: For example, the first $\min\{b_1, b_2\}$ points of a two-dimensional Halton sequence $\left(\Phi_{b_1}(i), \Phi_{b_2}(i)\right)$ lie on a straight line through the origin. Similar linear alignments appear over and over again in the sequence and the b_j can be large for high dimensional projections.

Therefore many improvements of the Halton sequence have been developed. In fact, all of them turn out to be deterministic scramblings (see Sec. 3.1.2): For example, Zaremba [88] used the simple permutation $\pi_{b_j}(a_l(i)) = a_l(i) + l \bmod b_j$ instead of directly using the digits $a_l(i)$ and later Faure [13] developed a set of permutations generalizing and improving Zaremba's results. A very efficient implementation can be found at http://gruenschloss.org/halton/halton.zip. While the modifications improve the constant of the order of discrepancy, they also improve upon the minimum distance [33].

Whenever the number of samples $n = \prod_{j=1}^{s} b_j^{n_j}$ is a product of power of the bases, the Halton sequence (including all its variants) is fully stratified.

3.1.4 Digital (t,s)-Sequences and (t,m,s)-Nets

Low discrepancy sequences can also be constructed from radical inverses using the same base $b_j = b$. They are based on b-adic elementary intervals as covered by Def. 2:

Definition 3 (see [55, Def. 4.1]). For integers $0 \leq t \leq m$, a (t,m,s)-*net in base* b is a point set of b^m points in $[0,1)^s$ such that there are exactly b^t points in each b-adic elementary interval E with volume b^{t-m}.

Definition 4 (see [55, Def. 4.2]). For an integer $t \geq 0$, a sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots$ of points in $[0,1)^s$ is a (t,s)-*sequence in base* b if, for all integers $k \geq 0$ and $m > t$, the point set $\mathbf{x}_{kb^m}, \ldots, \mathbf{x}_{(k+1)b^m-1}$ is a (t,m,s)-net in base b.

The elementary intervals from Def. 2 use a resolution of b^{d_j} along dimension j. For a (t,m,s)-net in base b we then have $\sum_{j=1}^{s} d_j = m - t$, which relates the number of points determined by m and the quality parameter t. Since scrambling (see Sec. 3.1.2) permutes elementary intervals, it does not change the t parameter. Similar to the Halton sequence, any (t,s)-sequence can be transformed into a $(t,m,s+1)$-net by concatenating a component $\frac{i}{b^m}$ [55].

According to Def. 3, a $(0,s)$-sequence is a sequence of $(0,m,s)$-nets, similar to what is illustrated for the Halton sequence in Fig. 1. This especially includes $(0,ms,s)$-nets, where in each hypercube-shaped elementary interval of side length b^{-m}, there is exactly one point. As the number of points of $(0,ms,s)$-nets is exponential in the dimension, this construction is only feasible in small dimensions.

$(0,1)$-Sequences in Base b

The simplest example of a $(0,1)$-sequence in base b is the van der Corput sequence. For regular generator matrices C, the radical inverses in Eq. 2 are $(0,1)$-sequences, too, and Def. 4 guarantees all properties of the van der Corput sequence as described above for the more general $(0,1)$-sequences.

Constructions of (t,s)-Sequences for $s > 1$

The digital construction

$$\mathbf{x}_i = \left(\Phi_{b,C_1}(i), \ldots, \Phi_{b,C_s}(i)\right)$$

of (t,s)-sequence is based on the radical inverses from Eq. 2 with identical base b and consequently different generator matrices C_j for each coordinate j.

The most popular (t,s)-sequence is the Sobol' sequence [73] in base $b=2$, because it can be implemented efficiently using bit-vector operations [43, 82, 17] to compute the radical inverses. The sequence can be constructed for any dimension and in fact each component is a $(0,1)$-sequence in base 2 itself. Due to the properties of $(0,1)$-sequences, the Sobol' sequence at $n=2^m$ samples must be a Latin hypercube sample [61]. Other than Latin hypercube samples based on random permutations that would have to be stored in $\mathcal{O}(sn)$ memory, the permutations generated by the Sobol' sequence are infinite, can be computed on demand without storing them, and are guaranteed to be of low discrepancy. A description of how to compute the binary generator matrices can be found in [29, 30] and one good set of matrices can be downloaded at http://web.maths.unsw.edu.au/~fkuo/sobol/. In [75] Sobol' et al. introduced additional criteria for the selection of the generator matrices.

As the first two components of the Sobol' sequence form a $(0,2)$-sequence in base 2, the first 2^{2m} two-dimensional points must be stratified such that there is exactly one point in each voxel of a $2^m \times 2^m$ regular grid over $[0,1)^2$. This structure is very useful in image synthesis (see Sec. 4.1).

Since $(0,s)$-sequences can only exist for $s \leq b$ [55, Cor. 4.24], Faure [12] generalized Sobol's construction to higher bases. Following Sobol's idea, each component is constructed as $(0,1)$-sequence. In fact both Sobol's and Faure's construction yield upper triangular generator matrices.

The construction of better generator matrices is an ongoing effort and various approaches have been taken [26]. In fact, there exist (t,m,s)-nets, which cannot be generated by radical inverses [18, Sec. 3]. This in connection with the observation that scrambling often improves the uniformity properties [33] of low discrepancy points alludes to conjecture that there are better low discrepancy sequences that are generated by general permutations instead of only generator matrices as in Eq. 2.

3.1.5 Rank-1 Lattice Sequences and Rank-1 Lattices

For a suitable generator vector $\mathbf{g}=(g_1,\ldots,g_s)$, rank-1 lattice sequences [50, 23, 51]

$$\mathbf{x}_i = \Phi_b(i)(g_1,\ldots,g_s) \bmod 1 \in (\mathbb{Q}\cap[0,1))^s$$

provide the simplest algorithm for generating a low discrepancy sequence in s dimensions. While in theory the components $g_j=\sum_{m=0}^{\infty} g_{j,m} b^m$ of the generator vector are represented by infinite sequences of digits $g_{j,m}\in\{0,\ldots,b-1\}$, in practice the components can be represented by positive integers due to the finite precision of computer arithmetic. Yet, there only exists a tiny number of constructions for the generator vectors [84, 3] and usually good generator vectors result from exhaustive computer searches [2]. An implementation of a variety of such methods is described in [48].

Lattice sequences resemble (t,s)-sequences, as contiguous blocks of b^m points form lattices, where the first lattice is anchored in the origin and the subsequent lattices are shifted copies. For $\gcd(g_i,b^m)=1$ rank-1 lattices are instances of a Latin

hypercube sample, which in addition provides a trivial lower bound on the minimum distance, because the one-dimensional projections are equidistantly spaced at $\frac{1}{b^m}$.

By allowing only generator vectors of the form $\mathbf{g} = (a^0, a^1, a^2, \ldots, a^{s-1})$, Korobov restricted the search space to one integer $a \in \mathbb{N}$ [72]. Note that for suitable a and b^m, the generator vector coincides with a multiplicative linear congruential pseudo-random number generator.

Hybrid Sequences

Besides the common properties of especially the Sobol' (t,s)-sequence and rank-1 lattice sequences in base $b = 2$, there even exist rank-1 lattices that are $(0,2,2)$-nets [8, Sec. 2.1]. There is an even closer relationship as stated by

Theorem 2. *Given b and g_j are relatively prime, the component $\Phi_b(i)g_j \bmod 1$ of a rank-1 lattice sequence is a $(0,1)$-sequence in base b.*

Proof. Φ_b is a $(0,1)$-sequence [55] and by Def. 4 each contiguous block of b^m points is a $(0,m,1)$-net in base b. As a consequence, the integer parts of such a $(0,m,1)$-net multiplied by b^m are a permutation. If now b and g_j are relatively prime, then for such a $(0,m,1)$-net the integers $g_j\lfloor b^m \Phi_b(i)\rfloor \bmod b^m$ form a permutation, too. Hence $\Phi_b(i)g_j \bmod 1$ is a $(0,1)$-sequence in base b. □

If now the generator matrix C is regular, a permutation exists that maps the elements of any $(0,m,1)$-net of $\Phi_b(i)g_j \bmod 1$ to $\Phi_{b,C}(i)$ and consequently $\Phi_{b,C}(i)$ and $\Phi_b(i)g_j \bmod 1$ are scrambled (see Sec. 3.1.2) versions of each other.

This close relationship allows one to combine components of (t,s)-sequences in base b with components of rank-1 lattice sequences using a radical inverse in base b. While this is of theoretical interest [45, 46], it also is of practical interest, especially in computer graphics: Rank-1 lattice sequences are cheap to evaluate, while (t,s)-sequences use the structure of b-adic elementary intervals [83].

3.2 Algorithms for Enumerating Low Discrepancy Sequences

The properties of radical inversion allow for enumerating low discrepancy sequences in different ways that are very useful building blocks of quasi-Monte Carlo methods. The enumeration schemes can be derived by equivalence transformations of integrals.

3.2.1 Enumeration in Elementary Intervals

Both the Halton and (t,s)-sequences are stratificatied with respect to elementary intervals (see Def. 2 and [52]). In [19] methods have been developed to efficiently

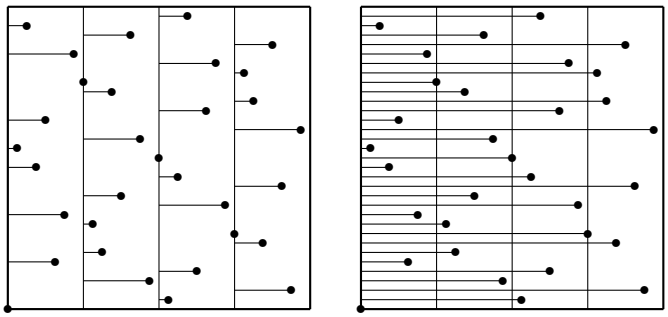

Fig. 4 Illustration of partitioned and nested low discrepancy sequences using the first 32 points of the Sobol' sequence. Left: Partitioning an $s+1$-dimensional low discrepancy sequence by its first component (along the x-axis) results in each one low discrepancy sequence in s dimensions as illustrated by the projections onto the partitioning lines parallel to the y-axis. Right: Thresholding the first component results in nested s-dimensional low discrepancy sequences, where each sequence with a smaller threshold is included in a sequence with a larger threshold.

enumerate the samples in a given elementary interval: Restricting the sequences to a given elementary interval yields a system of equations, whose solution results in an enumeration algorithm.

As the construction of the Halton sequence is based on the Chinese remainder theorem [21], enumerating the Halton sequence restricted to an elementary interval requires to solve a system of congruences. The solution of this system yields the indices $i+t\cdot\prod_{j=1}^{s} b_j^{d_j}, t\in\mathbb{N}_0$ to enumerate the Halton points in a given elementary interval. The initial offset i is uniquely identified by that elementary interval, while the subsequent points are found by jumping along the sequence with a stride that is a product of the prime powers of the bases b_j, where d_j fixes the resolution along dimension j.

For (t,s)-sequences, the system of linear equations is assembled by solving Eq. 2 for each dimension j for $M=d_j$ digits, where the d_j specify the size of the elementary interval as defined in Def. 2. The righthand side of the equation system then is given by each the first d_j digits of the coordinates p_j of the elementary interval and the number q of the point to be computed. In an implementation, the inverse system matrix can be stored and enumerating the points of an elementary interval is as expensive as computing the points of a (t,s)-sequence (see the code at http://gruenschloss.org/sample-enum/sample-enum-src.zip).

Typical applications of enumerating samples per elementary interval are problems, where the structure matches the stratification implied by elementary intervals. Such problems include integro-approximation and adaptive sampling [59, 40, 19], where the number of samples needs to be controlled per elementary interval. Enumerating samples per elementary interval also is a strategy for parallelization [19].

3.2.2 Partitioning Low Discrepancy Sequences

Restricting a low discrepancy sequence to an axis-aligned subinterval does not change its order of discrepancy [55]. Similarly the order of discrepancy is not changed by omitting dimensions, i.e. projecting the points along canonical axis.

Using a characteristic function

$$\chi_j(x') := \begin{cases} 1 & j \le x' < j+1 \\ 0 & \text{otherwise,} \end{cases}$$

the equivalence transformation

$$\int_{[0,1)^s} f(\mathbf{x}) d\mathbf{x} = \sum_{j=0}^{b^m-1} \int_{[0,1)} \int_{[0,1)^s} \chi_j(b^m \cdot x') \cdot f(\mathbf{x}) d\mathbf{x} dx'$$

identifies the point set

$$P_j := \left\{ \mathbf{x}_i : \chi_j(b^m \cdot x_{i,c}) = 1, i \in \mathbb{N}_0 \right\} = \left\{ \mathbf{x}_i : j \le b^m \cdot x_{i,c} < j+1, i \in \mathbb{N}_0 \right\}$$

used to integrate the j-th summand when applying one $s+1$ dimensional quasi-Monte Carlo point sequence $(\mathbf{x}_i)_{i \ge 0}$ for integral estimation, where $x_{i,c}$ is the c-th component of the point $\mathbf{x}_i$. Enumerating the subsequences

$$P_{\Phi_b^{-1}(j/b^m)} = \left\{ \mathbf{x}_{l \cdot b^m + j} : l \in \mathbb{N}_0 \right\}$$

of a sequence partitioned along a component, which is a radical inverse, is as simple as leaping through the sequence with a stride of b^m elements and an offset j [36]. As mentioned before, the P_j must be of low discrepancy, too. The method is illustrated in Fig. 4, where Φ_2 is used to partition the two-dimensional Sobol' sequence.

It is important to note that the partitioning component must not be used to sample the integrand, because computations may diverge, as explained in [36]: This extra dimension is used to partition an $s+1$-dimensional low discrepancy sequence into b^m s-dimensional low discrepancy sequences.

The main application of this scheme are communication-avoiding parallel quasi-Monte Carlo methods: Each thread, process, or job is assigned its own subsequence. Upon the reduction of the partial results, the ensemble of all samples forms the original low discrepancy sequence without any intermediate communication. Even if each thread, process, or job terminates adaptively, on the average the number of points consumed in each thread of process will be similar due to the low discrepancy of each of the subsequences. Due to the partitioning property the result is even independent of the number of processing elements; parallel or sequential execution yield identical results.

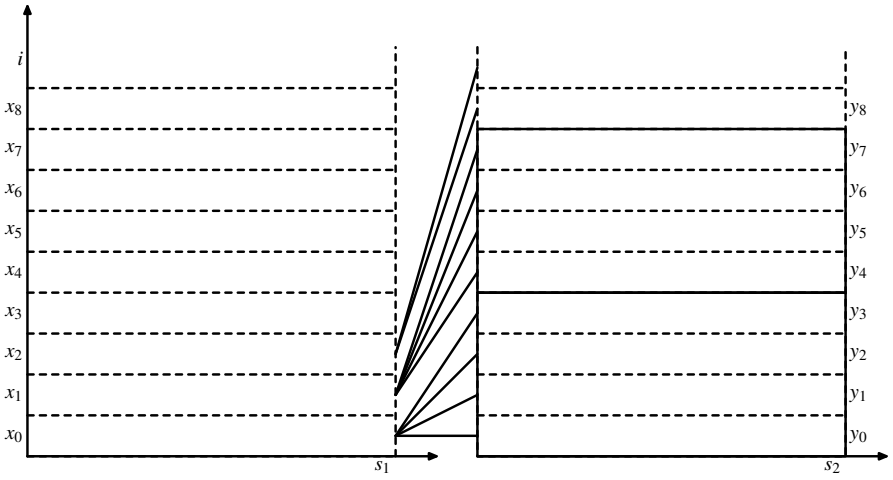

Fig. 5 Splitting can increase the efficiency by sampling the dimensions of y more than the dimensions of x. Using one low discrepancy sequence $(\mathbf{x}_i, \mathbf{y}_i)$, the dimensions of $\mathbf{x}_i$ are enumerated slower by a fixed factor as compared to the dimensions of $\mathbf{y}_i$.

3.2.3 Nested Low Discrepancy Sequences

Similar to partitioning low discrepancy sequences, nested s-dimensional low discrepancy sequences are obtained by thresholding an additional component. As illustrated in Fig. 4, the threshold determines the fraction of samples selected from the original sequence. The sequences are nested in the sense that sequences resulting from a smaller threshold are always included in sequences resulting from a larger threshold.

Nested sequences can be used in consistent algorithms, where several problems use the samples of one sequence. Depending on the single problem, a threshold can be selected to control what fraction of samples is consumed. Similar to the previous section, the nested sequences can be enumerated by leaping with a stride of b^m for a threshold b^{-m}.

3.2.4 Splitting

If a problem is less sensitive in some dimensions as compared to others, efficiency often can be increased by concentrating samples in the more important dimensions of the problem. Trajectory splitting is one such technique that after a certain path length splits one particle into multiple and follows their individual trajectories as illustrated in Fig. 5.

The principle of a very simple and efficient quasi-Monte Carlo algorithm for trajectory splitting is based on rewriting the integral of f

$$\int_{[0,1)^s} f(\mathbf{x},t)dtd\mathbf{x} = \int_{[0,1)^s} \sum_{j=0}^{b^m-1} \chi_{[\frac{j}{b^m},\frac{j+1}{b^m})}(t) f(\mathbf{x}, b^m t - j)dtd\mathbf{x}$$

as an integral of b^m copies of f with respect to the dimension t, where the characteristic function $\chi_A(t)$ is one if $t \in A$ and zero otherwise. Applying a low discrepancy sequence $(\mathbf{x}_i, t_i)$, where the component t_i is a $(0,1)$-sequence generated by an identity matrix scaled by an element from $\mathbb{F}_b \setminus \{0\}$, to compute the righthand side of the equivalence transformation yields:

$$\begin{aligned}
&\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \sum_{j=0}^{b^m-1} \chi_{[\frac{j}{b^m},\frac{j+1}{b^m})}(t_i) f(\mathbf{x}_i, b^m t_i - j) \qquad (3)\\
&= \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \sum_{j=0}^{b^m-1} \chi_{[\frac{j}{b^m},\frac{j+1}{b^m})}(\Phi_b(i)) f(\mathbf{x}_i, b^m \Phi_b(i) - j)\\
&= \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{x}_i, \Phi_b(\lfloor i/b^m \rfloor)) \underbrace{\sum_{j=0}^{b^m-1} \chi_{[\frac{j}{b^m},\frac{j+1}{b^m})}(\Phi_b(i))}_{=1}\\
&= \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{x}_i, t_{\lfloor i/b^m \rfloor}).
\end{aligned}$$

In Eq. 3, the characteristic function χ selects the summands with $t_i \in [\frac{j}{b^m}, \frac{j+1}{b^m})$ and therefore the index j is equal to the integer part of $b^m t_i$. Since $t_i = \Phi_b(i)$ is a $(0,1)$-sequence in base b as defined in Eq. 2 generated by an identity matrix scaled by an element from $\mathbb{F}_b \setminus \{0\}$, the m least significant digits of i can only influence the m most significant digits of $\Phi_b(i)$. Therefore the fraction $b^m \Phi_b(i) - j$ can be computed by just removing the m least significant digits of the index i. In fact, $\Phi_b(\lfloor i/b^m \rfloor) = b^m \Phi_b(i) - j$, which becomes obvious by comparing the digits of both numbers in base b. As the term $\Phi_b(\lfloor i/b^m \rfloor))$ no longer does depend on j, it can be factored out of the sum. The remaining sum of characteristic functions is always one, because the whole unit interval is covered and $\Phi_b(i) \in [0,1) = \cup_{j=0}^{b^m-1} [\frac{j}{b^m}, \frac{j+1}{b^m})$.

As a result, the implementation of the algorithm to sample one dimension at a rate b^m less than others can be as simple as shifting the index i to the right by m digits in base b. Numerical evidence leads to the conjecture that the algorithm also works for the component t_i being generated by an upper triangular generator matrix.

Although multiple splitting along a trajectory creates exponential work, the splitting scheme along one dimension can be applied to multiple dimensions. As a result, each dimension j can have its own sampling rate slow down factor b^{m_j}. Note that this includes components of rank-1 lattice sequences, where the generator and the base are relatively prime. For the Halton sequence, the splitting rate obviously must be a product of prime powers, which grows exponentially with dimension.

The new method replaces and improves upon previous approaches [32, 43] and has many applications in graphics, for example ambient occlusion, scattering, sampling environment maps and area light sources, and simulating motion blur.

4 Deterministic Consistent Image Synthesis

The consistency of quasi-Monte Carlo methods for light transport simulation (see Sec. 2.1) follows from Thm. 1 and allows one to use the building blocks developed in the previous Sec. 3 in deterministic consistent image synthesis algorithms.

The following sections describe how the subsequent components of a vector of a low discrepancy sequence are transformed in order to generate a light transport path: A path is started on the image plane, where the stratification properties of the first two dimensions are used to sample the image plane in Sec. 4.1, while the subsequent dimensions are applied in the simulation of a camera in Sec. 4.2. The path then is continued by repeated scattering as described in Sec. 4.3 and connected to the light sources. Sec. 4.4 considers more aspects of the opposite direction of assembling light transport paths by tracing photon trajectories from the light sources, their use in quasi-Monte Carlo density approximation, and the combination of paths both starting from the image plane and the light sources.

The resulting algorithms in principle all implement Eq. 1 and are progressively refining the results more and more over time. It is simple to interrupt and resume computation at any time, because the current state of the computation is completely described by the index of the last sample taken within the problem domain. Therefore termination can be triggered by any criterion and the computation can be continued unless the result was satisfactory.

4.1 Sampling the Image Plane for Anti-Aliasing

The rectangular picture elements of a display device match the structure of elementary intervals in two dimensions. Simultaneously determining the average color of each pixel is an integro-approximation problem that can be reduced to the form of Eq. 1. As described in Sec. 3.2.1, the computation of all pixels can be realized by one low discrepancy sequence covering the whole screen. Enumerating the samples per pixel offers several advantages:

- The samples in adjacent pixels are never the same, which at low sampling rates hides aliasing artifacts in noise.
- Keeping track of the last sample index per pixel allows for prioritized computation: Pixels of special interest can receive more samples per time, while others progress with relatively fewer samples [40]. Such regions of interest can be specified automatically or by user interaction. Nevertheless, the pixels remain consistent; they only converge at different speeds.
- The computation can be parallelized and load balanced by rendering each pixel in a separate thread. As the parallelization scheme is based on a domain partitioning, the computation is independent of the sequence and timing of the single tasks. Therefore the computation remains strictly deterministic and thus reproducible independent of the parallel execution environment.

The implementation details for the Sobol' $(0,2)$-sequence and the Halton sequence are found in [19], while the code can be downloaded at http://gruenschloss.org/sample-enum/sample-enum-src.zip. After dedicating the first two dimensions of a low discrepancy sequence to sampling the image plane, the use of the subsequent dimensions for the construction of light transport paths is explored.

4.2 Depth of Field, Motion Blur, and Spectral Rendering

Light is colored and reaches the image plane through an optical system. Except for pinhole cameras, light passes a lens with an aperture, which both specify the focal plane and depth of field. With focal plane and aperture selected by the user, the simulation associated to a sample with index i on the image plane continues by using its next dimensions. For the example of a thin lens with the shape of a unit disk, components $x_{i,3}$ and $x_{i,4}$ can be used to uniformly select a point

$$\begin{pmatrix} \sqrt{x_{i,3}} \cos 2\pi x_{i,4} \\ \sqrt{x_{i,3}} \sin 2\pi x_{i,4} \end{pmatrix} \tag{4}$$

on the lens from which a ray is traced through the point in space identified by the thin lens law applied to the sample point $(x_{i,1}, x_{i,2})$ on the image plane. The above mapping from the unit square onto the unit disk has been derived using the multi-dimensional inversion method [74].

The simulation of more advanced camera models including spectral properties [77] follows the same principle: Using another dimension to select a wavelength, the samples are summed up weighted according to spectral response curves, which for example map to the color basis of the display device [22].

However, including the simulation of motion blur caused by camera and/or object motion by just adding another dimension to sample time during the open shutter is not efficient. Each sample with a different time would require to adjust all scene assets to that instant, invoking the loading of temporal data and rebuilding acceleration data structures. On the other hand, interpolating data to increase efficiency may result in inconsistent rendering algorithms: For example, once approximated by an insufficient number of linear spline segments, a rotating propeller will never get round during the course of computation. In addition the memory footprint increases linearly with the number of spline segments.

Unless rendering relativistic effects, the speed of light is much faster than the motion to be rendered and efficiency can be increased by selecting one instant in time for multiple light transport paths. This efficient approach is easily implemented as consistent deterministic algorithm using splitting as introduced in Sec. 3.2.4. A splitting rate in the order of the number of pixels on the display device causes temporal data to be prepared once per accumulated frame. A lower splitting rate results in interleaving and averaging lower resolution images [38].

Phong model silver metallic paint violet rubber

Fig. 6 Simple mathematical models for bidirectional scattering distribution functions (BSDF) $f_s(\omega_o, x, \omega)$ use only a small number of basis functions, like for example the Phong model, and therefore cannot efficiently capture the variety and precision of measured data, like for example the silver metallic paint or the violet rubber. The two cylinders illustrate the direction ω of incidence from the right and the direction of perfect reflection exiting to the left. Images courtesy Ken Dahm partially using data from [53].

4.3 Sampling Light Transport Path Segments

As determined by the sample on the image plane and the camera, a ray is traced into the scene, where it interacts with the scene boundary and media inside the volume.

Given a direction of incidence ω and a direction ω_o of observation, the fraction of light transported in a location x on the boundary is described by the bidirectional scattering distribution function (BSDF) $f_s(\omega_o, x, \omega)$ as illustrated in Fig. 6. Such densities are modeled as linear combinations of basis functions, which can be analytic or tabulated measurements [53].

For the simulation, one of the basis functions can be selected proportional to its linear combination weight. Then, given a direction of incidence, the direction of scattering is determined by transforming two uniformly distributed components of a low discrepancy sequence. If the selected basis function refers to measured data, the table of measurements can be transformed into a discrete cumulative density distribution function and a direction of scattering can be found using binary search. For analytic functions, the inversion method is applied, which of course requires the basis function to have an analytic integral that is invertible in closed form.

The observed radiance

$$L_o(x, \omega_o) = \int_{\mathcal{S}^2_-(x)} f_s(\omega_o, x, \omega) L_{in}(x, \omega) \cos\theta_x \, d\omega \tag{5}$$

results from the incident radiance L_{in} integrated over the hemisphere $\mathcal{S}^2_-(x)$ aligned by the surface normal in x attenuated by the BSDF f_s, where the cosine of the angle θ_x between the normal on the boundary and the direction of incidence accounts for the perpendicular incident radiance, i.e. the effective part.

For the example of a constant basis function, evaluating such integrals requires to sample the hemisphere with respect to the cosine weight. Such unit vectors

$$\begin{pmatrix} \sqrt{x_{i,j}}\cos 2\pi x_{i,j+1} \\ \sqrt{x_{i,j}}\sin 2\pi x_{i,j+1} \\ \sqrt{1-x_{i,j}} \end{pmatrix}$$

are similar to uniform samples on the disk (see Eq. 4), as the z component just results from the constraint of unit norm. Alike transformations exist for many other analytic basis functions of BSDFs [66].

The efficient simulation of scattering within media is subject to active research [68, 87, 57, 58], especially the consistent deterministic simulation of inhomogenous media is an open challenge and beyond the scope of this tutorial.

Controlling Path Length

After a direction of scattering has been determined, the next ray can be traced and the procedure is repeated at each point of interaction to follow a path through the scene. The path length can be controlled by Russian roulette, where an extra dimension of the low discrepancy sequence is compared to the reflectivity/transmissivity of a surface in order to determine whether the path is continued or terminated by absorption [74].

Low discrepancy sequences, like for example the Sobol' sequence, are dimension extensible. Nevertheless, path length must be restricted to a maximum path length in an implementation in order to avoid the possibility of an infinite loop due to numerical issues in scattering and especially ray tracing.

4.3.1 Path Tracing

Path tracing generates samples on the image plane, traces a path through the camera, and determines a scattering direction upon interacting with the scene. Whenever a light source is encountered along the path, its contribution attenuated by the product of BSDFs along the path is recorded on the image plane. This first simple rendering algorithm is deterministic and consistent, as each path is completely determined by one vector of a low discrepancy sequence realizing Eq. 1.

Typical types of light sources are area light sources $L_e(x,\omega)$ and high dynamic range environment maps $L_{e,x}(\omega)$, which describe the light incident in one point x from all directions of the sphere ω. Besides analytic models describing the sky dome, environment maps often contain incident light, for example measured by a high dynamic range photograph of a perfect mirror ball. Similarly, area light sources can be modeled by analytic functions or can be given as measured data. For a given direction ω (and a location x for area light sources), the evaluation of the emission distribution function returns a spectral density.

Path tracing is efficient, as long as hitting a light source is likely as for example in product and car visualization, where objects are rendered enclosed by an environment light source. Whenever a ray is scattered off the object to be visualized

and does not intersect the boundary any more, the light path is terminated and the direction of the ray is used to look up the spectral density in the environment map. In cases where the integrand exposes more variance in the dimensions used for sampling the environment map, it pays off to split (see Sec. 3.2.4) the camera path by sending multiple rays into the hemisphere.

4.3.2 Next Event Estimation

Path tracing is not efficient for small light sources as for example spots as used in interiors. However, as the position of such light sources is known, it is easy to check, whether they are visible. For so-called next event estimation, one component of a low discrepancy vector is used to select a light source, while two more are used to determine a point on the selected light source. The visibility is checked by tracing a so-called shadow ray from the location to be illuminated towards the point on the light source. Unless the ray is occluded, the contribution of the light source is recorded.

If the light sources are small, visible, and at sufficient distance, next event estimation will be efficient. Otherwise, there are issues: Sampling the surface of the light sources, the contribution of the light source must be divided by the squared distance to account for the solid angle subtended by the area of the light source. If the point to be illuminated is close to the point of the light source, the division by the squared distance may result in overmodulation or even a division by zero. The corresponding integral over the area of the light source is therefore called weakly singular. This numerical problem can be robustly treated by combining both techniques of sampling the solid angle and the area by either partitioning the integral [44] or weighting the contributions [47, Sec. 4.1.5]. Both approaches bound the integrand and are simple to implement, while the latter approach performs slightly superior with respect to path tracing with next event estimation (see the example in Sec. 4.4.3).

Applying the splitting technique from Sec. 3.2.4 as illustrated in Fig. 5 overcomes the necessity of randomization as required in [43]. Testing multiple shadow rays for one location to be illuminated may increase efficiency.

With an increasing number of light sources and varying area and distance, the selection of contributing light sources becomes costly. Especially visibility cannot be efficiently predicted in a general way and must be tested. Typical such scenarios include architectural scenes, where light comes through door slits and corridors and many lights can be occluded by walls. Note that shadow rays are testing only geometry and therefore transparent or refractive surfaces report occlusion. For that reason, next event estimation will not transport light through glass.

4.3.3 Light Tracing

Instead of starting light transport paths on the image plane, it appears more natural to follow the trajectories of photons emitted by light sources, which requires the simulation of the emission distribution functions. Similar to the previous section, a light source is to be selected. For area light sources a point of emission needs to be determined in addition. The direction of emission results from transforming two more uniform components according to the emission distribution function. For the special case of environment maps, the procedure is described in detail in [7].

Once emitted, the photon trajectory can be followed as described before in Sec. 4.3. Similar to the issue of small light sources in path tracing, it is not very likely that photons pass the camera to hit the image plane and therefore shadow rays are traced from the light path vertices to the camera (in analogy to next event estimation, see Sec. 4.3.2).

Opposite to path tracing, light tracing with next event estimation can render caustics, which for example are caused by light projected through a glass onto a diffuse surface. Generating such light transport paths starting on the image plane is inefficient due to the low probability of hitting the light source after scattering on the diffuse surface, especially, if the light is small. However, if the caustic is seen through a mirror, light tracing with next event estimation fails, too, because the connection from the mirror to the camera realizes the reflection direction with probability zero.

4.4 Blockwise Sampling of Light Transport Paths

When path tracing and light tracing with or without next event estimation are not efficient due to a small probability of establishing a light transport path, starting path segments from both the light sources and the camera and connecting them can help. This requires two Markov chains to be simulated: One with the emission distribution function as initial distribution and the BSDF as transition probabilities and another one starting on the image plane using the BSDF as transition probabilities as well.

By partitioning one low discrepancy sequence along the components as illustrated in Fig. 7, both Markov chains can be realized by one vector $(\mathbf{x}_i, \mathbf{y}_i)$, where for example the odd components $\mathbf{x}_i$ determine the path segment starting from the image plane and the even components $\mathbf{y}_i$ determine a photon trajectory.

As illustrated in Fig. 2 and introduced in Sec. 2.1 the connections between path segments can be established by checking the mutual visibility of the end points of the path segments (see Sec. 4.4.2) or by proximity (see Sec. 4.4.1). Depending on the kind of the connection and the specific length of each of the two path segments, the same transport path may be generated by multiple such techniques (similar to the special case mentioned in Sec. 4.3.2). Their optimal combination is discussed in Sec. 4.4.3.

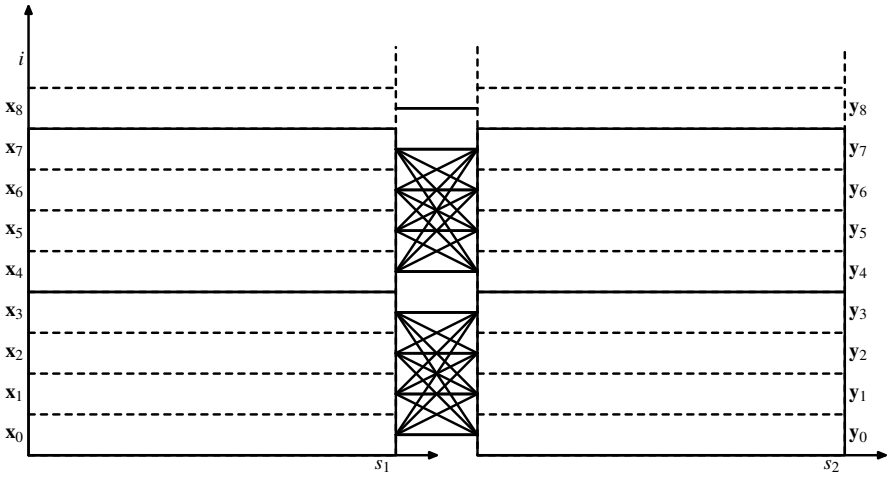

Fig. 7 In order to increase the efficiency of deterministic consistent density estimation (see Sec. 4.4.1), one low discrepancy sequence is partitioned along the components and enumerated in blocks. Within each block, each parameter $\mathbf{x}_i$ is combined with each parameter $\mathbf{y}_i$. It is instructive to compare the blockwise averaging to splitting as shown in Fig. 5.

4.4.1 Connecting Path Segments by Proximity

The basic idea of photon mapping [27, 28] is to compute the transported light using density estimation [71]. The discrete density is stored as a point cloud called photon map, which results from tracing photon trajectories from the light sources and recording the incident energy at each interaction with the scene. In order to compute the radiance as given by Eq. 5, the contribution of the photons within a sphere around the point of interest is averaged.

As has been shown in [34], photon mapping can be realized as a deterministic consistent quasi-Monte Carlo method: Either a (t,s)- or a rank-1 lattice sequence $(\mathbf{x}_i,\mathbf{y}_i)$ in base b is progressively enumerated in blocks of size b^m. For each vector $(\mathbf{x}_i,\mathbf{y}_i)$ of the low discrepancy sequence, a light transport path segment from the camera is constructed using the dimensions of $\mathbf{x}_i$, while a photon trajectory is started from the lights using $\mathbf{y}_i$. Then all camera and light path segments within a block are combined to simultaneously compute the radiance

$$L_P = \lim_{n\to\infty} \frac{|P|}{n} \sum_{i=0}^{n-1} \chi_P(\mathbf{x}_i) W(\mathbf{x}_i) \frac{1}{b^m} \sum_{k=0}^{b^m-1} \frac{\chi_{\mathscr{B}(r(n))}(h(\mathbf{x}_i) - h(\mathbf{y}_{b^m\lfloor i/b^m\rfloor+k}))}{\pi r^2(n)} \cdot f_s(\omega(\mathbf{x}_i), h(\mathbf{x}_i), \omega(\mathbf{y}_{b^m\lfloor i/b^m\rfloor+k}))\phi(\mathbf{y}_{b^m\lfloor i/b^m\rfloor+k}) \tag{6}$$

through each pixel P on the image plane. Fig. 7 illustrates how the low discrepancy sequence is used in the equation and how the sum over k enumerates all $\mathbf{y}_{b^m\lfloor i/b^m\rfloor+k}$ for each $\mathbf{x}_i$. Out of the camera paths, $\chi_P(\mathbf{x}_i)$ selects the ones contributing the pixel P. Their contribution is weighted by W, which is the product of all attenuations by

interactions along the path segment until the query location $h(\mathbf{x}_i)$ is hit. The flux deposited in $h(\mathbf{y}_{b^m\lfloor i/b^m\rfloor+k})$ by a photon is ϕ. If now the difference of both hit points is in a ball $\mathscr{B}$ of radius $r(n)$, both path segments are considered connected by proximity and the product of weight, the flux, and the BSDF f_s is recorded for the pixel P. Assuming a locally planar surface around the query location, the contribution is averaged over the disk area $\pi r^2(n)$ and both ω denote the directions from which the end points of the path segments are hit.

Consistency requires the radius

$$r^2(n) = \frac{r_0^2}{n^\alpha} \text{ for } 0 < \alpha < 1$$

to decrease with a power of n. As shown in [37, Sect. 3.1], the radius vanishes arbitrarily slowly and the influence of the parameter α becomes negligible already after enumerating a few blocks. Consequently, the efficiency is controlled by the initial radius r_0 and the parameter m determining the block size b^m. The initial radius r_0 determines the ratio of how many photons can interact with a query location. Besides choosing r_0 constant, adaptive methods have been discussed in [37, 41].

Once all query locations and photons within a block have been stored as point clouds, space is subdivided hierarchically in order to prune sets of query locations and photons that cannot interact [35] within the distance of $r(n)$. For the light path segments that can be connected by proximity, the contribution is accumulated according to Eq. 6. Obviously, the block size should be chosen as large as memory permits in order to connect as many light path segments as possible within one block.

The algorithm as described, for example, can render caustics seen in a mirror or through a transparent object and thus overcomes the problem of "insufficient techniques" (see [42, Fig. 2] and Sec. 4.3.3). However, this comes at a price: Since connections are not precise but within a certain radius, images may appear blurred and light may leak through the boundary. Although these artifacts vanish due to consistency, they vanish arbitrarily slowly [37, Sect. 3.1], which underlines the importance of the choice of r_0.

4.4.2 Connecting Path Segments by Shadow Rays

Bidirectional path tracing (BDPT) [80, 47, 79] is a generalization of next event estimation (see Sec. 4.3.2), where any vertex of a camera path can be connected to any vertex of a light path segment by a shadow ray as illustrated in Fig. 2. The algorithm is complementary to photon mapping, because it still is limited by the problem of "insufficient techniques" (see previous section), however, lacks the transient artifacts of progressive photon mapping due to precisely testing the mutual visibility of vertices.

The mapping of a low discrepancy sequence to light transport path segments works as described before by partitioning along the dimensions. While the original

approach connected vertices of one camera path segment with the vertices of the corresponding light path segment, now connections can be established within a block of path segments (see Fig. 7). Opposite to photon mapping, where the number of connections can be restricted by proximity, the number of shadow rays is quadratic in the block size multiplied by the product of camera and light path segment length. Besides complexity, the block size also determines the look of the transient artifacts. Using larger block sizes, camera paths in neighboring pixels are illuminated by the same light path vertices. These can be considered point light sources, resulting in sharp shadow contours that of course vanish due to consistency. As an extension, splitting (see Sec. 3.2.4) allows for controlling the ratio of camera and light paths to be connected.

With photon mapping, bidirectional path tracing, and all the variants of sampling, the question of which technique is best to use comes up and will be discussed in the next section.

4.4.3 Optimally Combining Techniques

As described in the previous two sections, connecting camera and light path segments by either shadow rays or proximity, the same path may be generated by multiple sampling techniques. While the mapping of low discrepancy sequences to camera and light path segments is shared over all techniques, their optimal combination is still subject of active research, because the efficiency of the rendering techniques may depend on the scene description. One of the key issues is the lack of efficient algorithms for predicting visibility and especially the discontinuities in the integrands of computer graphics.

Multiple Importance Sampling

Importance sampling aims to improve the efficiency by sampling the function more frequently in important regions of the integration domain. Given a density p representing importance, an integral

$$\int_{[0,1)^s} f(x)dx = \int_{[0,1)^s} f(x)\frac{p(x)}{p(x)}dx = \int_{[0,1)^s} \frac{f(x)}{p(x)}dP(x) = \int_{[0,1)^s} \frac{f(P^{-1}(x))}{p(P^{-1}(x))}dx \quad (7)$$

of a function f is transformed using the substitution $p(x) = \frac{dP(x)}{dx}$ and the formalism of the Riemann-Stieltjes integral. Assuming existence, in many cases the transformation P^{-1} of uniform samples into p-distributed samples can be realized by the multi-dimensional inversion method [25, 74]. Originally developed in Monte Carlo integration [74], the theory has been extended to cover quasi-Monte Carlo integration as well [76].

The observation that in light transport simulation the same path may be generated by different importance sampling techniques led to the idea of multiple importance

sampling [80, 47, 79]: Given a function $f(x)$ to be integrated and m densities $p_i(x)$ that can be evaluated and sampled, defining the weights

$$w_i(x) := \frac{p_i^\beta(x)}{\sum_{j=0}^{m-1} p_j^\beta(x)} \qquad \text{with} \qquad \sum_{i=0}^{m-1} w_i(x) = 1 \tag{8}$$

allows for transforming the integral

$$\int_{[0,1)^s} f(x)dx = \int_{[0,1)^s} \sum_{i=0}^{m-1} w_i(x) f(x)dx = \sum_{i=0}^{m-1} \int_{[0,1)^s} w_i(x) \frac{f(x)}{p_i(x)} dP_i(x). \tag{9}$$

into a sum of integrals, where the i-th integral is evaluated using samples that are distributed according to p_i in analogy to Eq. 7. Hence for $m = 1$ the general formulation of multiple importance sampling coincides with importance sampling.

The convex combination using the set of weights $w_i(x)$ is called the power heuristic [79]. While for $\beta = 0$ the weights $w_i = \frac{1}{m}$ result in a uniform weighting, for $\beta > 0$ higher weights are assigned to techniques with higher density. Special cases are the balance heuristic for $\beta = 1$ with weights $w_i \sim p_i$ and the maximum heuristic for $\beta = \infty$, which selects the technique with the highest density $p_i(x)$. Among these and other heuristics, the power heuristic with $\beta = 2$ is slightly superior [79, Thm. 9.2].

Samples x for which $p_i(x) = 0$ obviously cannot be generated, which requires $w_i(x) = 0$ to make the method work. As a direct consequence, for any x at least one density must be positive. It may well happen that this cannot be guaranteed, which is called the problem of "insufficient techniques" [42]. A related issue is the situation, where the denominator is smaller than the numerator and samples may be overly amplified [64, Sec. 2.2], although their importance actually is small.

Example: Removing the Weak Singularity in Direct Illumination

Given an emission distribution function L_e and a BSDF f_s on the scene surface, the direct illumination

$$\begin{aligned} L_d(x,\omega_o) &= \int_{\mathscr{S}^2_-(x)} f_s(\omega_o,x,\omega) L_e(h(x,\omega),-\omega)\cos\theta_x d\omega \\ &= \int_A f_s(\omega_o,x,\omega) L_e(y,-\omega)\cos\theta_x V(x,y)\frac{\cos\theta_y}{|x-y|^2} dy \end{aligned} \tag{10}$$

is equivalently determined by either integrating over the hemisphere $\mathscr{S}^2_-(x)$ or the surface A of the light source L_e with area $|A|$, where the direction ω points from x towards the respective point y on the light source. The ray tracing function $h(x,\omega)$ and the visibility $V(x,y)$ are introduced in Sec. 2.2. Note that Eq. 10 is weakly singular due to the division by the squared distance $|x-y|^2$, which in this form causes numerical problems whenever x and y are sufficiently close.

Two resulting sampling techniques are simulating scattering directions according to

$$p_1 \equiv f_s(\omega_o, x, \omega)\cos\theta_x \qquad \text{and using} \qquad p_2 \equiv \frac{1}{|A|}$$

to generate uniform samples on the light source. For a given hit point $y := h(x,\omega)$, the visibility $V(x,y)$ is one and changing the measure from solid angle in x to a point y on the area of a light source and vice versa [47, Sec. 4.1.5] results in the densities

$$p_1 \frac{\cos\theta_y}{|x-y|^2} \qquad \text{and} \qquad p_2 \frac{|x-y|^2}{\cos\theta_y}.$$

Then the weights for the balance heuristic in Eq. 8 are

$$w_1 \equiv \frac{p_1}{p_1 + p_2\frac{|x-y|^2}{\cos\theta_y}} = \frac{|A| f_s(\omega_o, x, \omega)\cos\theta_x\cos\theta_y}{|A| f_s(\omega_o, x, \omega)\cos\theta_x\cos\theta_y + |x-y|^2} \qquad \text{and}$$

$$w_2 \equiv \frac{p_2}{p_1\frac{\cos\theta_y}{|x-y|^2} + p_2} = \frac{|x-y|^2}{|A| f_s(\omega_o, x, \omega)\cos\theta_x\cos\theta_y + |x-y|^2}.$$

While w_1 has been derived using densities with respect to the solid angle and w_2 has been using densities with respect to the area measure, the weights are ratios of densities with respect to the same measure and therefore have no unit. Using the transformation in Eq. 9 and the equivalence in Eq. 10, the direct illumination amounts to

$$\begin{aligned} &L_d(x,\omega_o) \qquad (11)\\ &= \int_{[0,1)^2} L_e(h(x,\omega),-\omega)\frac{|A| f_s(\omega_o,x,\omega)\cos\theta_x\cos\theta_y}{|A| f_s(\omega_o,x,\omega)\cos\theta_x\cos\theta_y + |x-y|^2} dP_1(\omega)\\ &\quad + \int_{[0,1)^2} L_e(y,-\omega)V(x,y)\frac{f_s(\omega_o,x,\omega)\cos\theta_x\cos\theta_y}{|A| f_s(\omega_o,x,\omega)\cos\theta_x\cos\theta_y + |x-y|^2} dP_2(y) \end{aligned}$$

in accordance with [47, Eq. 4.7, Sec. 4.1.5].

Assuming that directions can be generated with the density p_1, both integrands in Eq. 11 are bounded, because the weak singularity [44] has been removed by the transformation, which underlines one of the major advantages of multiple importance sampling.

Note that the undefined $\frac{0}{0}$ case needs to be handled explicitly: In order to avoid numerical exceptions, it is sufficient to test $f_s\cos\theta_x\cos\theta_y$ for zero explicitly, since then no radiation is transported. Comparing this term to a small, positive threshold, substantial amounts of transport may be missed for small distances $|x-y|^2$.

A Path Tracer with Next Event Estimation and Multiple Importance Sampling

For the purpose of this tutorial an efficient implementation of a path tracer with next event estimation (see Sec. 4.3.2) and multiple importance sampling is described [47, Sec. 4.1.5]. Light transport is modeled by a Fredholm integral equation of the second kind

$$L = L_e + T_{f_s} L,$$

where the integral operator

$$T_{f_s} L \equiv \int_{\mathscr{S}^2_-(x)} f_s(\omega_o, x, \omega) L(h(x,\omega), -\omega) \cos\theta_x d\omega$$

determines the transported radiance in analogy to Eq. 10. The radiance thus is the source radiance L_e plus transported radiance

$$T_{f_s} L = T_{f_s}(L_e + T_{f_s} L) = T_{f_s}((w_1 + w_2) L_e + T_{f_s} L) = T_{f_s}(w_1 L_e + T_{f_s} L) + T_{f_s} w_2 L_e,$$

which can be computed by first replacing one instance of the radiance by its definition and then inserting a linear combination of weights that always sums up to one as derived in the previous section. As a result the transported radiance is determined by two terms: The first term is evaluated by integration over the hemisphere, which comprises sampling a scattering direction, tracing a ray and summing up the weighted emission L_e and recursively computing the transported radiance. The second term uses a shadow ray towards the support of the light sources and the according weight as derived in Eq. 11 for the example of the balance heuristic.

The implementation can be realized as a simple loop without recursion, terminating the path started from the image plane by Russian roulette (see Sec. 4.3.1), and using the scattering direction both for path tracing and next event estimation with multiple importance sampling. For regions with visibility $V = 1$ the method converges much faster than the path tracer without multiple importance sampling although no additional rays need to be traced.

This simple but already quite powerful algorithm can be extended to bidirectional path tracing, where all vertices of a camera path are connected to all vertices of a light path. Using the principle of implicit importance sampling as before, the implementation is compact [5, 1]. As bidirectional path tracing suffers the problem of insufficient techniques [42], photon mapping can be added [34] by using multiple importance sampling as well [20, and references therein]. Using the quasi-Monte Carlo technique of blockwise enumeration as illustrated in Fig. 7, all image synthesis algorithms can be implemented as progressive, consistent, and deterministic algorithms.

Although multiple importance sampling takes care of optimally combining techniques, it does not consider visibility: For the simple example of a light bulb, all shadow rays will report occlusion by the glass around the glowing wire. Similarly, shadow rays are not efficient, when light enters a room or a car through a window. On the other hand, mostly diffuse scenes do not benefit from photon mapping, which

raises the question, whether for a given scene description the relevant techniques can be determined algorithmically and whether visibility can be efficiently predicted.

5 State of the Art

Consistent quasi-Monte Carlo methods are easily parallelized and are perfectly reproducible due to their deterministic nature. In industry they take advantage of SIMD (single instruction multiple data) architectures and are perfectly suitable for latency hiding architectures, especially GPUs (graphics processing units). Besides computer graphics, other domains like for example finance, will benefit from parallel quasi-Monte Carlo methods as well.

Low discrepancy sequences can be generated at the speed of high-quality pseudo-random numbers and they offer a performance advantage due to better discrete density approximation [25]. On certain restricted function classes [55, 72, 8], quasi-Monte Carlo methods are roughly quadratically faster than Monte Carlo methods and it is known that quasi-Monte Carlo methods outperform Monte Carlo methods on the average [86, 78]. However, due to the deterministic nature of quasi-Monte Carlo methods, it is possible to construct theoretical worst cases, especially for the class of square integrable functions, where a Monte Carlo method can be expected to be better. For this reason the general Thm. 1 cannot provide a good rate of convergence on the class of square integrable functions.

Beyond the state of the art as surveyed in this article, there are still fundamental issues in image synthesis: While (multiple) importance sampling is deeply explored in the context of computer graphics, there are indications that the weights for combining bidirectional path tracing and photon mapping are not optimal and that there is no efficient deterministic method that can incorporate the prediction of visibility (see Sec. 4.4.2), yet. While the Metropolis light transport [81, 79] algorithm can efficiently handle boundaries with complex visibility, there does not exist a deterministic version and it is unknown how to benefit from low discrepancy.

For all known techniques, settings can be constructed that result in inefficient performance: For example, shadow rays do not work with transparent objects like glass and the Metropolis light transport algorithm is not efficient in simple settings. There is a desire to algorithmically determine which techniques are efficient for a given setting.

Besides lighting complexity, the amount of data to be rendered in one frame reaches amounts that require simplification to enable efficient processing. Such approaches relate to multi-level algorithms and function representations and level-of-detail representations. Finding such approximations is still a challenge, because changing visibility often dramatically changes the light transport and consequently the rendered image.

In conclusion, the paradigm of consistency has led to many new developments in quasi-Monte Carlo methods and numerous industrial rendering solutions apply quasi-Monte Carlo methods for light transport simulation.

Acknowledgments

The author likes to thank Ian Sloan, Frances Kuo, Josef Dick, and Gareth Peters for the extraordinary opportunity to present this tutorial at the MCQMC2012 conference and Pierre L'Ecuyer for the invitation to present an initial tutorial on "Monte Carlo and Quasi-Monte Carlo Methods in Computer Graphics" at MCQMC2008. In addition, the author is grateful to the anonymous reviewers, Nikolaus Binder, and Ken Dahm.

References

1. van Antwerpen, D.: Unbiased physically based rendering on the GPU. Master's thesis, Computer Graphics Research Group, Department of Software Technology Faculty EEMCS, Delft University of Technology, the Netherlands (2011)
2. Cools, R., Kuo, F., Nuyens, D.: Constructing embedded lattice rules for multivariate integration. SIAM J. Sci. Comput. **28**, 2162–2188 (2006)
3. Cools, R., Reztsov, A.: Different quality indexes for lattice rules. J. Complexity **13**(2), 235–258 (1997)
4. Cranley, R., Patterson, T.: Randomization of number theoretic methods for multiple integration. SIAM Journal on Numerical Analysis **13**, 904–914 (1976)
5. Dahm, K.: A comparison of light transport algorithms on the GPU. Master's thesis, Computer Graphics Group, Saarland University, Germany (2011)
6. Dammertz, H.: Acceleration Methods for Ray Tracing based Global Illumination. Ph.D. thesis, Universität Ulm (2011)
7. Dammertz, H., Hanika, J.: Plane sampling for light paths from the environment map. Journal of Graphics, GPU, and Game Tools **14**(2), 25–31 (2009)
8. Dammertz, S., Keller, A.: Image Synthesis by Rank-1 Lattices. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 217–236. Springer (2008)
9. Devroye, L.: Non-Uniform Random Variate Generation. Springer, New York (1986)
10. Dick, J., Pillichshammer, F.: Digital Nets and Sequences. Discrepancy Theory and Quasi-Monte Carlo Integration. Cambridge University Press (2010)
11. Edwards, D.: Practical Sampling for Ray-Based Rendering. Ph.D. thesis, The University of Utah (2008)
12. Faure, H.: Discrépance de suites associées à un système de numération (en dimension s). Acta Arith. **41**(4), 337–351 (1982)
13. Faure, H.: Good permutations for extreme discrepancy. J. Number Theory **42**, 47–56 (1992)
14. Friedel, I., Keller, A.: Fast generation of randomized low-discrepancy point sets. In: H. Niederreiter, K. Fang, F. Hickernell (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 257–273. Springer (2002)
15. Frolov, A., Chentsov, N.: On the calculation of certain integrals dependent on a parameter by the Monte Carlo method. Zh. Vychisl. Mat. Fiz. **2**(4), 714 – 717 (1962). (in Russian)
16. Glassner, A.: Principles of Digital Image Synthesis. Morgan Kaufmann (1995)
17. Grünschloß, L.: Motion Blur. Master's thesis, Ulm University (2008)
18. Grünschloß, L., Keller, A.: (t,m,s)-Nets and Maximized Minimum Distance, Part II. In: P. L'Ecuyer, A. Owen (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2008, pp. 395–409. Springer (2009)
19. Grünschloß, L., Raab, M., Keller, A.: Enumerating Quasi-Monte Carlo Point Sequences in Elementary Intervals. In: L. Plaskota, H. Woźniakowski (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 399–408. Springer (2012)

20. Hachisuka, T., Pantaleoni, J., Jensen, H.: A path space extension for robust light transport simulation. ACM Trans. Graph. **31**(6), 191:1–191:10 (2012)
21. Halton, J.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Math. **2**(1), 84–90 (1960)
22. Hanika, J.: Spectral Light Transport Simulation using a Precision-based Ray Tracing Architecture. Ph.D. thesis, Universität Ulm (2010)
23. Hickernell, F., Hong, H., L'Ecuyer, P., Lemieux, C.: Extensible lattice sequences for quasi-Monte Carlo quadrature. SIAM J. Sci. Comput. **22**, 1117–1138 (2001)
24. Hlawka, E.: Discrepancy and Riemann integration. In: L. Mirsky (ed.) Studies in Pure Mathematics, pp. 121–129. Academic Press, New York (1971)
25. Hlawka, E., Mück, R.: Über eine Transformation von gleichverteilten Folgen II. Computing **9**(2), 127–138 (1972)
26. Hong, H.: Digital Nets and Sequences for Quasi-Monte Carlo Methods. Ph.D. thesis, Hong Kong Baptist University (2002)
27. Jensen, H.: Global illumination using photon maps. In: Rendering Techniques 1996 (Proc. 7th Eurographics Workshop on Rendering), pp. 21–30. Springer (1996)
28. Jensen, H.: Realistic Image Synthesis Using Photon Mapping. AK Peters (2001)
29. Joe, S., Kuo, F.: Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator. ACM Trans. Math. Softw. **29**(1), 49–57 (2003)
30. Joe, S., Kuo, F.: Constructing Sobol' sequences with better two-dimensional projections. SIAM Journal on Scientific Computing **30**(5), 2635–2654 (2008)
31. Keller, A.: Quasi-Monte Carlo methods in computer graphics: The global illumination problem. Lectures in App. Math. **32**, 455–469 (1996)
32. Keller, A.: Trajectory splitting by restricted replication. Monte Carlo Methods and Applications **10**(3-4), 321–329 (2004)
33. Keller, A.: Myths of computer graphics. In: H. Niederreiter (ed.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 217–243. Springer (2006)
34. Keller, A., Binder, N.: Deterministic consistent density estimation for light transport simulation. In: J. Dick, F. Kuo, G. Peters, I. Sloan (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2012, p. accepted for publication. Springer (2013)
35. Keller, A., Droske, M., Grünschloß, L., Seibert, D.: A Divide-and-Conquer Algorithm for Simultaneous Photon Map Queries. Poster at High-Performance Graphics in Vancouver (2011)
36. Keller, A., Grünschloß, L.: Parallel Quasi-Monte Carlo Integration by Partitioning Low Discrepancy Sequences. In: L. Plaskota, H. Woźniakowski (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 487–498. Springer (2012)
37. Keller, A., Grünschloß, L., Droske, M.: Quasi-Monte Carlo Progressive Photon Mapping. In: L. Plaskota, H. Woźniakowsi (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 499–509. Springer (2012)
38. Keller, A., Heidrich, W.: Interleaved sampling. In: K. Myszkowski, S. Gortler (eds.) Rendering Techniques 2001 (Proc. 12th Eurographics Workshop on Rendering), pp. 269–276. Springer (2001)
39. Keller, A., Wächter, C.: Efficient Ray Tracing without Auxiliary Acceleration Data Structure. Poster at High-Performance Graphics in Vancouver (2011)
40. Keller, A., Wächter, C., Kaplan, M.: System, method, and computer program product for consistent image synthesis. United States Patent Application 20110025682 (2011)
41. Knaus, C., Zwicker, M.: Progressive photon mapping: A probabilistic approach. ACM Transactions on Graphics (TOG) **30**(3) (2011)
42. Kollig, T., Keller, A.: Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration. In: H. Niederreiter, K. Fang, F. Hickernell (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 290–305. Springer (2002)
43. Kollig, T., Keller, A.: Efficient multidimensional sampling. Computer Graphics Forum (Proc. Eurographics 2002) **21**(3), 557–563 (2002)
44. Kollig, T., Keller, A.: Illumination in the presence of weak singularities. In: D. Talay, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 245–257. Springer (2004)

45. Kritzer, P.: On an example of finite hybrid quasi-Monte Carlo point sets. Monatsh. Math. **168**, 443–459 (2012)
46. Kritzer, P., Leobacher, G., Pillichshammer, F.: Component-by-component construction of hybrid point sets based on Hammersley and lattice point sets. In: J. Dick, F. Kuo, G. Peters, I. Sloan (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2012, p. accepted for publication. Springer (2013)
47. Lafortune, E.: Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering. Ph.D. thesis, Katholieke Universitiet Leuven, Belgium (1996)
48. L'Ecuyer, P., Munger, D.: Latticebuilder: A general software tool for constructing rank-1 lattice rules. ACM Transactions on Math. Software **00**(0), submitted (2012)
49. Lemieux, C.: Monte Carlo and Quasi-Monte Carlo Sampling. Springer (2009)
50. Maize, E.: Contributions to the Theory of Error Reduction in Quasi-Monte Carlo Methods. Ph.D. thesis, Claremont Graduate School (1980)
51. Maize, E., Sepikas, J., Spanier, J.: Accelerating the convergence of lattice methods by importance sampling-based transformations. In: L. Plaskota, H. Woźniakowski (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, *Springer Proceedings in Mathematics & Statistics*, vol. 23, pp. 557–572. Springer (2012)
52. Matoušek, J.: On the L_2-discrepancy for anchored boxes. J. Complexity **14**(4), 527–556 (1998)
53. Matusik, W., Pfister, H., Brand, M., McMillan, L.: A Data-Driven Reflectance Model. ACM Transactions on Graphics (Proc. SIGGRAPH 2003) **22**(3), 759–769 (2003)
54. Niederreiter, H.: Quasirandom sampling in computer graphics. In: Proc. 3rd International Seminar on Digital Image Processing in Medicine, Remote Sensing and Visualization of Information (Riga, Latvia), pp. 29–34 (1992)
55. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia (1992)
56. Niederreiter, H.: Error bounds for quasi-Monte Carlo integration with uniform point sets. J. Comput. Appl. Math. **150**, 283–292 (2003)
57. Novák, J., Nowrouzezahrai, D., Dachsbacher, C., Jarosz, W.: Progressive virtual beam lights. Computer Graphics Forum (Proceedings of EGSR 2012) **31**(4) (2012)
58. Novák, J., Nowrouzezahrai, D., Dachsbacher, C., Jarosz, W.: Virtual ray lights for rendering scenes with participating media. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2012) **31**(4) (2012)
59. Nuyens, D., Waterhouse, B.: A global adaptive quasi-Monte Carlo algorithm for functions of low truncation dimension applied to problems of finance. In: L. Plaskota, H. Woźniakowsi (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 591–609. Springer (2012)
60. Ohbuchi, R., Aono, M.: Quasi-Monte Carlo rendering with adaptive sampling. IBM Tokyo Research Laboratory (1996)
61. Owen, A.: Orthogonal arrays for computer experiments, integration and visualization. Statistica Sinica **2**, 439–452 (1992)
62. Owen, A.: Randomly permuted (t,m,s)-nets and (t,s)-sequences. In: H. Niederreiter, P. Shiue (eds.) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, *Lecture Notes in Statistics*, vol. 106, pp. 299–315. Springer (1995)
63. Owen, A.: Monte Carlo variance of scrambled net quadrature. SIAM Journal on Numerical Analysis **34**(5), 1884–1910 (1997)
64. Owen, A., Zhou, Y.: Safe and effective importance sampling. Journal of the American Statistical Association **95**(449), 135–143 (2000)
65. Paskov, S.: Termination criteria for linear problems. Journal of Complexity **11**, 105–137 (1995)
66. Pharr, M., Humphreys, G.: Physically Based Rendering. Morgan Kaufmann, 2nd Ed. (2011)
67. Press, H., Teukolsky, S., Vetterling, T., Flannery, B.: Numerical Recipes in C. Cambridge University Press (1992)
68. Raab, M., Seibert, D., Keller, A.: Unbiased global illumination with participating media. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 669–684. Springer (2007)

69. Shirley, P.: Discrepancy as a quality measure for sampling distributions. In: Proc. Eurographics 1991, pp. 183–194. Elsevier Science Publishers, Amsterdam, North-Holland (1991)
70. Shirley, P.: Realistic Ray Tracing. AK Peters, Ltd. (2000)
71. Silverman, B.: Density Estimation for Statistics and Data Analysis. Chapman & Hall/CRC (1986)
72. Sloan, I., Joe, S.: Lattice Methods for Multiple Integration. Clarendon Press, Oxford (1994)
73. Sobol', I.: On the Distribution of points in a cube and the approximate evaluation of integrals. Zh. vychisl. Mat. mat. Fiz. **7**(4), 784–802 (1967)
74. Sobol', I.: Die Monte-Carlo-Methode. Deutscher Verlag der Wissenschaften (1991)
75. Sobol', I., Asotsky, D., Kreinin, A., Kucherenko, S.: Construction and comparison of high-dimensional Sobol' generators. WILMOTT magazine pp. 64–79 (2011)
76. Spanier, J., Maize, E.: Quasi-random methods for estimating integrals using relatively small samples. SIAM Review **36**(1), 18–44 (1994)
77. Steinert, B., Dammertz, H., Hanika, J., Lensch, H.: General spectral camera lens simulation. Computer Graphics Forum **30**(6), 1643–1654 (2011)
78. Traub, J., Wasilkowski, G., Woźniakowski, H.: Information-Based Complexity. Academic Press (1988)
79. Veach, E.: Robust Monte Carlo Methods for Light Transport Simulation. Ph.D. thesis, Stanford University (1997)
80. Veach, E., Guibas, L.: Optimally combining sampling techniques for Monte Carlo rendering. In: Proc. SIGGRAPH 1995, Annual Conference Series, pp. 419–428 (1995)
81. Veach, E., Guibas, L.: Metropolis light transport. In: T. Whitted (ed.) Proc. SIGGRAPH 1997, Annual Conference Series, pp. 65–76. ACM SIGGRAPH, Addison Wesley (1997)
82. Wächter, C.: Quasi-Monte Carlo Light Transport Simulation by Efficient Ray Tracing. Ph.D. thesis, Universität Ulm (2008)
83. Wächter, C., Keller, A.: System and process for improved sampling for parallel light transport simulation. ISF MI-12-0006-US0 filed as United States Patent Application (2012)
84. Wang, Y., Hickernell, F.: An historical overview of lattice point sets. In: H. Niederreiter, K. Fang, F. Hickernell (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2000, pp. 158–167. Springer (2001)
85. Weyl, H.: Über die Gleichverteilung von Zahlen mod. Eins. Mathematische Annalen **77**, 313–352 (1916)
86. Woźniakowski, H., Traub, J.: Breaking intractability. Scientific American (270), 102–107 (January 1994)
87. Yue, Y., Iwasaki, K., Chen, B., Dobashi, Y., Nishita, T.: Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. ACM Trans. Graph. **29**(6), 177 (2010)
88. Zaremba, S.: La discrépance isotrope et l'intégration numérique. Ann. Mat. Pura Appl. **87**, 125–136 (1970)