Geometric Sample Reweighting for Monte Carlo Integration

Guo, J.; Eisemann, E.

DOI
10.1111/cgf.14405

Publication date
2021

Document Version
Final published version

Published in
Computer Graphics Forum

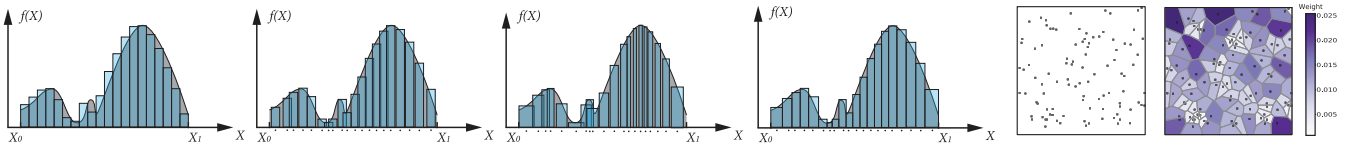
DOI: 10.1111/cgf.14405
Pacific Graphics 2021
E. Eisemann, K. Singh, and F.-L Zhang
(Guest Editors)

Volume 40 (2021), Number 7

Geometric Sample Reweighting for Monte Carlo Integration

J. Guo and E. Eisemann†

Delft University of Technology, the Netherlands



(a) Regular binning (b) MC w/ random samples (c) MC w/ importance sampling (d) MC w/ reweighted random samples (e) 2D sample set (f) Weights for 2D samples

Figure 1: *Top row: Three integration methods using the same amount of function evaluations (i.e., 20 samples): (a) Riemann sum through regular binning (according to right-side value) (b) MC integration using uniform random samples; (c) MC integration using samples distributed according to a pdf w.r.t. function value. Notice that in (a) and (b), the associated bin widths are equal, i.e., $\frac{1}{20}$. Bin widths in (c) are adjusted according to the density linked to the sample generation. Notice the overlaps/gaps between sample bins as illustrated in (b) and (c). Illustrations of our methods: (d) uniform random samples with our reweighting. Notice the absence of gaps/overlaps and bin widths being adjusted according to sample positions. (e) a set of 10^2 i.i.d. random samples in 2D, $(0,1)^2$. Rather than directly averaging all samples (i.e., weight $= 10^{-2}$ for all samples), our consistent estimator uses the Voronoi volume as sample weight during the reconstruction process (f). Notice that the weights here are inversely proportional to the density.*

Abstract

Numerical integration is fundamental in multiple Monte Carlo rendering problems. We present a sample reweighting scheme, including underlying theory, and analysis of numerical performance for the integration of an unknown one-dimensional function. Our method is simple to implement and builds upon the insight to link the weights to a function reconstruction process during integration. We provide proof that our solution is unbiased in one-dimensional cases and consistent in multi-dimensional cases. We illustrate its effectiveness in several use cases.

CCS Concepts

• ***Computing methodologies*** → *Ray tracing;*

Keywords: Sampling and Reconstruction, Monte Carlo Integration, Sample Reweighting, Rendering

1. Introduction

Monte Carlo (MC) techniques form the foundation of realistic image synthesis for decades [CPC84]. The principle is simple: a function is sampled and the samples are combined to approximate its integral. Standard MC is often referred to as *brute-force* because of the easy implementation and guaranteed convergence to the correct solution, but often suffers from high variance and slow convergence. The integral approximation can be improved by reconstructing the underlying function from the samples and previous work focused on several particular cases (e.g., shadows [ETH*09] or depth of field [SSD*09]). In this work, we focus on one-dimensional numeric integration and revisit the function reconstruction process.

Our observation is that standard sample weights are often less accurate for lower sampling rates because they do not reflect the integration domain nor the local sample density well. Our contribution is to propose a weighting scheme that considers all samples of a given set and defines weights based on their Voronoi cells. This results in a consistent estimator that outperforms standard weighting schemes. Additionally, we render the estimator unbiased for sets of independent and identically distributed (i.i.d) uniform random samples. While the proof of unbiasedness is complex, the algorithm is easy to implement.

While our work focuses on the one-dimensional case, the derived

† Author email: J.Guo-3, E.Eisemann@tudelft.nl

theory shows that the bias of the consistent estimator is expected to be low. We can therefore expect also good performance for low-dimensional problems and provide a numerical evaluation.

Our solution can be easily integrated in existing rendering pipeline. It provides an immediate benefit over existing schemes for various rendering problems and builds upon a solid theoretical analysis.

After covering prior work and MC integration (Sec. 2), we motivate our approach (Sec.3) and explain the core of our method (Sec. 4). We then evaluate its numerical performance and show applications to rendering (Sec. 5) before concluding (Sec. 7).

2. Background

MC methods Since the 80s [CPC84], MC integration plays a major role in rendering complex distribution effects, such as motion blur, depth of field, soft shadows and global illumination. The complete light transport is described by the *rendering equation* [Kaj86], which can be solved using *path tracing* as an associated MC solution.

Standard Monte Carlo (MC) methods solve a definite integration $I = \int_\Omega f(x)\mathrm{d}x$ of a function f over a finite support $\Omega \subset \mathbb{R}^d$ by using a random sample set ($\{x_i \in \Omega\}$). The resulting estimator is

$$\hat{I}_{MC} = \frac{1}{N}\sum_{i=1}^{N} f(x_i). \tag{1}$$

Nevertheless, not all samples taken during the evaluation of an integral contribute strongly to the result.

Importance sampling The sampling process can be influenced with a probability distribution function (pdf) $p : \Omega \to \mathbb{R}$ [VG95]. Its unbiased estimator is:

$$\hat{I}_p = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}, \tag{2}$$

which effectively weighs samples differently, given non-uniform sample input.

Importance sampling is interesting when having knowledge about the scene. For instance, importance sampling the light source works better in scenes with small or point light sources [DBB06, Deb08]. Sampling according to the BSDF works better with glossy to highly glossy surfaces [Shi91, LFTG97]. Multiple importance sampling (MIS) combines different sampling strategies [VG95].

One strategy to modify subsequent sample choices is to rely on previous samples, i.e., a Markov process. Metropolis sampling [VG97] can handle complex light-path configurations by extensively exploring contributing paths once they are discovered. Multidimensional k-d trees [HJW*08] can update a global structure with samples, which can then be used as a means to control future sample placement.

Reweighting Specialized weight definitions have proven beneficial for rendering, e.g., derived in Sobolev spaces [MBB18]. However, these previous solutions target hemispherical illumination integrals and are not generally applicable to other problems. A reweighting scheme was also proposed for addressing firefly artifacts [ZHD18] but the solution is biased and limited to narrow application scenarios.

Recently Bitterli *et. al.* present a reweighting process following resampling from reservoir sampling (ReSTIR) [BWP*20]. Reservoir sampling [Vit85] generates fix size samples that follows certain population distribution. The reweighting as proposed in the work is biased but leads to significant variance reduction.

Other specialized reconstruction techniques exist, including solutions for soft shadows [ETH*09], defocus blur [SSD*09], and motion blur [EHDR11], which lead to significant improvements. More complex reconstructions for environment maps illumination [ARBJ03] and light fields [LAC*11] have proven very successful but are biased (though consistent). Our method handles general functions and is independent of the application scenario.

General weighted reconstruction during MC integration has initially been investigated: a trapzoidal rule and a nearest-neighbor rule have been proposed by Yakowitz *et. al.* [YKS78]. Our novelty lies in supporting domains with open boundaries and a new weight definition. This is different from using the Euclidean distance between nearest neighbors [YKS78]. Additionally, our method results in an unbiased estimator. As this approach pursues similar goals, we will analyze the difference in Sec.4.1.

Our approach links sample weights to *Voronoi* cell sizes. The latter has also been studied in the context of anti-aliasing problems [Mit90], for which the two-dimensional area of Voronoi cells bounded by the pixel are directly used as sample weights. This leads to improved anti-aliasing, but the theory has not been further developed for unbiased solutions, nor generalized to other contexts. Voronoi cell size has been used as weights for Monte Carlo integration by Vorechovsk *et. al.* [VSE16]. They treat domain boundaries by either clipping to them or by extending them periodically by adding auxiliary samples next to the opposing domain border. While both solutions improve numerical performance of MC integration, we show that such a solution is biased in Sec.4.

3. Formulation and Problem Statement

Riemann integration approximates an integral using function values $f(x_i)$ times a weight $w(x_i)$:

$$\hat{I} = \sum_{i=1}^{N} w(x_i) f(x_i). \tag{3}$$

The Riemann weights are constant and stem from a partitioning of the support Ω into equally-sized *hypervolumes* - in 1D, these are intervals - each containing exactly one sample at the center. Referring to Eq. 2, the estimator using samples following distribution p is also a sum of function values $f(x_i)$ times a weight $w(x_i) = \frac{1}{Np(x_i)}$, which are typically easy to compute but cannot be associated to a hypervolume partitioning of Ω. They are scaled versions of $\frac{1}{N}$ according to a per-sample density $p(x_i)$; corresponding volumes would overlap or introduce gaps. Only when the sample count is increased to densely cover the support Ω, due to the stochastic nature of the process, the difference in the weight definitions becomes negligible. See Fig. 1 (a), (b) and (c) for an illustration. In consequence,

especially for low sample counts, the weights $\frac{1}{Np(x_i)}$ do not lead to a good function approximation [TCE05].

4. Geometric Sample Reweighting

Our goal is to associate integration weights to samples that relate to an improved function reconstruction. We will first define a consistent solution, inspired by Riemann integration (Sec. 4.1). It is independent of the sampling pattern and can be applied on any sample set in a postprocess to improve the integral approximation. While consistent, it is not unbiased for all sampling strategies (Sec. 4.2). We then propose a modification to obtain an unbiased estimator for 1D cases (Sec. 4.3). We treat uniform random samples that are *independent and identically distributed* (i.i.d.). Sec subsection 4.4 discusses the possibility to generalize our method for samples generated with a non-uniform pdf, and we propose a simple solution for piecewise-constant importance sampling, which is not uncommon in the context of computer graphics.

4.1. Consistent Estimator

Riemann integration typically assumes a regular partitioning of the domain. Using a *Voronoi tessellation* of the sample points, it is possible to partition the domain Ω to take the actual sample density into account. A Voronoi tessellation is a partition into regions such that the points in each region share the same closest sample location. It can be shown that the Voronoi cell corresponds to the intersection of half spaces defined by hyperplanes that are equidistant to two sample points. The theory of Voronoi diagrams is beyond the scope of this paper but details can be found in works by Aurenhammer *et. al.* and De Berg *et. al.* [Aur91, DBVKOS97].

For a D dimensional problem setting, the diagram will be bounded by the hypercube $(0,1)^D$, the domain from which samples are drawn. The volume of each Voronoi cell determines the corresponding sample weight. Given that the cells are intersections of half-spaces, these Voronoi cells are convex and their volume can be easily computed. The resulting estimator of this approach is:

$$\hat{I}_C = \sum_{i=1}^{N} w_C(x_i) f(x_i), \text{where } w_C(x_i) = \frac{|V_i|}{|\Omega|} \tag{4}$$

Such an estimator is consistent, as this construction implicitly approximates the integrand via a piecewise-constant representation. Fig. 1 (d) and (f) show an illustration of this strategy for a one-dimensional and two-dimensional case, respectively. In principle, even more advanced approximations could be used. However, as we will show, it turns out that such weight definitions, while consistent, lead to a biased estimate. In the following, we will show the reasons and derive an unbiased estimator for i.i.d. uniform sampling and stratified sampling for our use case with $D = 1$.

4.2. Bias Analysis

The problem of integrating over a *given* set of samples brings some changes to the classic MC integration process: given a set of samples $\{x_i | x_i \sim \mathcal{U}(0,1)\}$, we want to work with a specific set $\{x_i\}$ such that $0 < x_1 < x_2 < \cdots < x_N < 1$. This will ease the derivation of the weights in our method. From *Order Statistics* [DN04], we know that the probability of a set fulfilling this condition is $P(\{x_i\}) = N! \prod_{i=1}^{N} p(x_i)$. Intuitively, a random set of samples with N elements can be ordered in $N!$ ways.

To ease understanding of our bias analysis, we first illustrate the derivation with classic MC $\hat{I}_{MC}$ (i.e., equal weight for all samples). We denote the composed operation as $\mathbb{F}(\{x_i\}) = \sum_{i=1}^{N} \frac{1}{N} \frac{f(x_i)}{p(x_i)}$. Conveniently, $P(\{x_i\}) = N!$ and $\mathbb{F}(\{x_i\}) = \sum_{i=1}^{N} \frac{1}{N} f(x_i)$ for uniform sampling ($p(x) = 1$). The expectation of integration using such a set of samples is:

$$\begin{aligned}
\mathbb{E}\left[\hat{I}_{MC}\right] &= \int_{(0,1)^N} \mathbb{F}(\{x_i\}) P(\{x_i\}) \mathrm{d}\{x_i\} \\
&= \int_0^1 \int_0^1 \cdots \int_0^1 \left[\sum_{i=1}^{N} \frac{1}{N} \frac{f(x_i)}{p(x_i)}\right] N! \prod_{i=1}^{N} p(x_i) \mathrm{d}\{x_i\} \\
&= \int_0^1 \int_{x_1}^1 \cdots \int_{x_{N-1}}^1 \left[\sum_{i=1}^{N} \frac{1}{N} f(x_i)\right] N! \mathrm{d}\{x_i\} \\
&= N! \int_0^1 \int_{x_1}^1 \cdots \int_{x_{N-1}}^1 \\
&\quad \left[\frac{1}{N} f(x_1) + \frac{1}{N} f(x_2) + \cdots + \frac{1}{N} f(x_N)\right] \mathrm{d}x_N \mathrm{d}x_{N-1} \cdots \mathrm{d}x_1 \\
&= \int_0^1 f(x) dx,
\end{aligned} \tag{5}$$

Line 2 expands from line 1 following the definition. Properties $p(x_i) = 1$ and $0 < x_1 < x_2 < \cdots < x_N < 1$ are used to get line 3 from line 2. The integration domain is denoted as $(0,1)^N$ to indicate domain of the sample vector. The notation $\mathrm{d}\{x_i\}$ means the differential sample set, which expands to $\mathrm{d}x_N \mathrm{d}x_{N-1} \cdots \mathrm{d}x_1$. In Appendix A, we prove the last step, showing that classic MC is indeed unbiased.

A similar derivation for our consistent estimator (Eq.4) yields:

$$\begin{aligned}
&\mathbb{E}\left[\hat{I}_C\right] = \\
&\quad N! \int_0^1 \int_{x_1}^1 \cdots \int_{x_{N-1}}^1 \\
&\quad \left[\frac{x_1 + x_2}{2} f(x_1) + \sum_{i=2}^{N-1} \frac{x_{i+1} - x_{i-1}}{2} f(x_i) + \right. \\
&\quad \left. \left(1 - \frac{x_{N-1} + x_N}{2}\right) f(x_N)\right] \mathrm{d}x_N \mathrm{d}x_{N-1} \cdots \mathrm{d}x_1 \\
&= N! \cdot \int_0^1 \left\{ \frac{1}{2} \cdot \frac{x \cdot (1-x)^{N-1}}{(N-1)!} \right. \\
&\quad + \frac{1}{2} \cdot \frac{(1-x)^{N-1} [(N-1) \cdot x + 1]}{N!} \\
&\quad + \sum_{i=2}^{N-1} \left[\frac{1}{2} \cdot \frac{(1-x)^{N-i} [(N-i) \cdot x + 1] \cdot x^{i-1}}{(N-i+1)!(i-1)!} \right. \\
&\quad \left. - \frac{1}{2} \cdot \frac{(1-x)^{N-i} \cdot x^i}{i(N-i)!(i-2)!} \right] \\
&\quad \left. + \left[\frac{x^{N-1}}{(N-1)!} - \frac{1}{2} \cdot \frac{x^N}{N(N-2)!} - \frac{1}{2} \cdot \frac{x^N}{(N-1)!} \right] \right\} \cdot f(x) \mathrm{d}x
\end{aligned}$$

$$= \int_0^1 \frac{(N-Nx-x)x^{N-1}+(Nx+x-1)(1-x)^{N-1}+2}{2} \cdot f(x)\mathrm{d}x. \tag{6}$$

Unfortunately, the expression is not identical to $\int f(x)\mathrm{d}x$ for $N > 1$. Thus, our consistent estimator as introduced in Eq.4 introduces bias. Similarly, we can show that other correction schemes using augmented periodic samples, as proposed in [VSE16], results in a biased estimate (compare Appendix B). The issue is that a weight adaptation of the boundary samples is insufficient, as inner samples also share a weight expectation different from $1/N$. Nevertheless, as the bias is linked to the expression $(N-Nx-x)x^{N-1}+(Nx+x-1)(1-x)^{N-1}$, we do observe that for the large part of the domain it converges to 1 for larger N, which illustrates that the consistent estimator can be of practical relevance.

4.3. Deriving an Unbiased Estimator

A closer look reveals that Eq.6 takes the form of $\int g(x) \cdot f(x)$, where $g(x) = [(N-Nx-x)x^{N-1}+(Nx+x-1)(1-x)^{N-1}+2]/2$. As $g(x) \neq 0$ for all $x \in (0,1)$, we can define $F(X) = \frac{f(x)}{g(x)}$. Following the derivation of Eq.6 but considering F instead of f, we obtain the expectation of this new estimator $\hat{I}_C^F$:

$$\begin{aligned} \mathbb{E}\left[\hat{I}_C^F\right] &= \int_0^1 g(x) \cdot F(x)\mathrm{d}x = \int_0^1 g(x) \cdot \frac{f(x)}{g(x)}\mathrm{d}x \\ &= \int_0^1 f(x)\mathrm{d}x = \mathbb{E}\left[\hat{I}\right]. \end{aligned} \tag{7}$$

It can be seen that sampling F yields an estimate for the integration of f. Hence, the function g can serve as a weight definition to obtain an unbiased estimator for the integrand $f(x)$ over a set of N i.i.d. uniform samples with reweighting. Consequently, our unbiased estimator is given as:

$$\hat{I}_{GR} = \sum_{i=1}^{N} w_{GR}(x_i) f(x_i) = \sum_{i=1}^{N} \frac{|V_i|}{|\Omega|} \frac{1}{g(x_i)} f(x_i) \tag{8}$$

4.4. Working with Non-uniform Samples

The general form of MC integration follows from Eq.5, where $p(x_i)$ is not necessarily 1 for all x_i. Thus, the analytic expression for when deriving it as in Eq.6 leads to uniform samples depending on p. As p can be arbitrary, it is impossible to develop the expression further.

Following order statistics, we need to know the *joint probability density* of samples following given distribution. For uniform distributions this is convenient as $p(x_i) = 1$. For non-uniform cases, deriving corresponding g_p term would mean the presence of $\prod_{i=1}^{N} p(x_i)$ that can not be easily cancelled out. For this to be unbiased, strictly analytical approach is required. Even for a given p with simple analytical formulation, chances are high that an analytic integration is usually too complex. Nevertheless, it is possible to adapt the uniform approach to the special case of p being a piecewise-constant distribution, which is not uncommon in many practical use cases. Here, we can stratify the domain according to the distribution and reweight samples within each stratum, following the previous solution for the whole domain. In this way, we can estimate the final integration without introducing bias. We provide a numerical evaluation of this strategy in Sec.5.1.

5. Results

In section 4, we developed the unbiased estimator for one-dimensional functions (Eq.8). Nevertheless, we also illustrated that the use of Voronoi volumes remains consistent for arbitrary dimensions. As the computation of a Voronoi diagram can be costly for higher dimensions, the method is best suited for low-dimensional functions. In the following, we will use the unbiased estimator for one-dimensional and the consistent estimator for multi-dimensional problems. The derivation of an unbiased reweighting scheme for arbitrary dimensions is an open challenge for future work.

The result section is organized in two parts, first covering the numerical performance (Sec. 5.1) of our estimator before applying it to several rendering problems (Sec. 5.2). All tests compare multiple different estimators:

1. Standard MC (i.i.d. uniform sampling)
2. Our weighted standard MC (i.i.d. uniform sampling)
3. Stratified MC (stratified sampling)
4. Our weighted stratified MC (stratified sampling)
5. Low discrepency sequences when applicable

5.1. Numerical Performance

We first test the numerical performance of MC integration in 1D and 2D. The 1D functions were chosen to include discontinuities, large-scale variations, and small scale changes. In the supplementary material, the reader can find many more test cases for our algorithm. Here, due to their representative behavior, we analyze:

$$f(x) = 10 \times \begin{cases} \sqrt{-x^2+0.5x} & x \leq 0.25 \\ -\sqrt{-x^2+x-0.1875}+0.25 & 0.25 < x \leq 0.5 \\ 20 \times (x-0.5) & 0.5 < x \leq 0.55 \\ 1.0 & 0.55 < x \leq 0.65 \\ -20 \times (x-0.7) & 0.65 < x \leq 0.7 \\ 0.1 \times \sin(10\pi \cdot (x-0.7)) & 0.7 < x \leq 0.8 \\ 0.25 \times \sin(10\pi \cdot (x-0.8)) & 0.8 < x \leq 0.9 \\ 0.5 \times \sin(10\pi \cdot (x-0.9)) & 0.9 < x \end{cases}$$

For the 2D consistent estimator, we take the *Cameraman* image (see Fig. 3 (d)). The results are shown in Fig. 3(3). Generally, the MSE drops as more samples are added (Column 1). Our solutions outperform standard uniform sampling (and stratified sampling) by several orders of magnitude and converges around 10^3 (10^2) times in 1D and 10^2 (10) times faster in 2D.

We study the numerical performance of non-uniform samples that are importance sampling the piecewise-constant distribution in Fig.2 (actual function $f(x)$ is plotted in dashed lines). We take two reasonable non-uniform piecewise constant distributions. Red: 4-piecewise distribution that does not match $f(x)$ and blue: 10-piecewise distribution that matches $f(x)$ perfectly. It can be

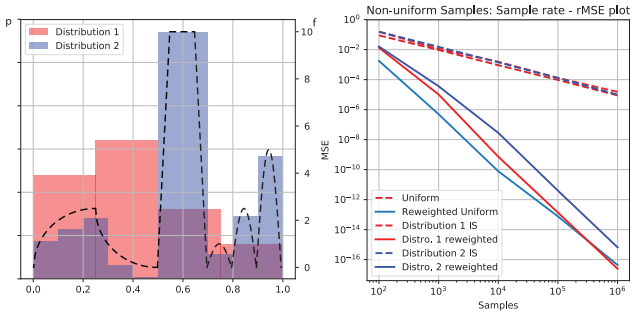


Figure 2: *Numerical performance of reweighting both uniform and two non-uniform distributions. Integrand in test is plotted in dashed line in the left subfigure. Uniform samples are drawn normally while non-uniform samples are drawn from the piecewise constant distributions to the left. Log-log scale convergence performances are plotted in the right subfigure.*

seen from the results that both reweighted estimators show similar/superior converging behaviors while a distribution that does not match function yields no practical benefits comparing to uniform samples. Unexpectedly, the distribution that matches the function perfectly only slightly outperforms uniform and a non-matching distribution. Meanwhile, all three reweighted version show similar convergence behavior. This illustrates not only the effectiveness of our reweighting but also how tricky importance sampling is in real-world scenarios dealing with functions that have significant/sharp variations. It could also be seen that in this specific experiment, a carefully crafted distribution is much less effective when it comes to variance reduction when comparing to our reweighting scheme.

Numerical performance against common 1D low discrepency sequences is presented in Fig. 3 (a). As can be seen from the result, our reweighted uniform/stratified achieves better performance than traditional stratified and *Latin Cube* sequence while slightly worse than *Halton* and *Sobol* sequence. For 2D cases (Fig. 3 (c)), both reweighted estimators perform the best and all low discrepency sequences fail to make any improvement.

We next investigate the impact of distributing samples into batches to show the unbiased convergence behavior of our reweighting scheme in 1D cases. First, we fixate the amount of samples to 10^6 (Fig.3(b)). As expected, the performance of standard uniform sampling remains invariant with respect to the amount of samples per batch, as it is already an averaging process. In general, more batches means a higher overall sample count and all methods improve with the addition of batches. In all cases, the graphs stop after reaching 10^6 samples. Our solution performs best and the graphs illustrate a fast convergence.

In a direct comparison, our estimator asymptotically achieves the same convergence rate (with a constant offset) as the estimator proposed in [YKS78], which is proven to be $O(N^{-4/D})$, where D is the dimension of the function. The interested reader is referred to the original work for the proof. Yet, in contrast, our work has the advantage of being unbiased. We used the same functions and configurations as in their original work. The results are plotted in Fig. 4.

5.2. Application to Rendering

We implemented our method in Mitsuba [Jak10], targeting one and two-dimensional integration problems, namely motion blur (Sec. 5.2.1), dispersion (Sec. 5.2.2), depth of field (Sec. 5.2.3) and illumination integrals (Sec. 5.2.4). We evaluate MSE and visual appearance, as well as convergence behavior. For all implementations, our reweighting operates at a per-pixel level. Notice that some low discrepency sequences are incompatible with certain integrators in Mitsuba. For example, only (0,2) sequence can be used in motion blur tests and Hammersley can not be used for direct illumination integrator.

5.2.1. Motion Blur

To simulate motion blur, distribution rendering samples the time domain: For a pixel (i, j), the luminance $L_{(i,j)}$ is given by:

$$L_{(i,j)} = \int_{t_{\text{open}}}^{t_{\text{close}}} f_{(i,j)}(t)\mathrm{d}t,$$

with t_{open} and t_{close} being the shutter opening and closing time and f incorporating the shutter function. The problem is thus one-dimensional and we tested our implementation in two animated scenes (Fig.5 and 6).

All temporal samples are drawn uniformly from $[t_{\text{close}}, t_{\text{open}}]$. In both scenes, objects are moving at high speed. The implementation takes minimum effort as only one outer loop in the program needs modification.

5.2.2. Spectral Rendering

Light dispersion can happen at reflective or refractive dielectric materials, leading to rainbows, resulting from different wavelengths traveling in different directions. It requires spectral sampling to simulate multiple wavelengths. To reduce the complexity of the additional spectral dimension [BDM09], *hero wavelength spectral sampling* [WND*14] can be used as an approximation:

$$\hat{I}_{(i,j)} = \frac{1}{N}\sum_{k=1}^{N}\frac{f_{(i,j)}(\lambda_k)}{p_{(i,j)}(\lambda_k)}$$

Our implementation of spectral sampling uses 15-bin wavelengths. Hero wavelength sampling is used with 3 shifted additional wavelength samples [WND*14]. We tested our method with two scenes configured with dispersive dielectric materials (Fig. 7 and 8). As shown in the results, our method brings down color noise significantly and dispersive regions look much smoother at a low sampling rate.

5.2.3. Defocus Blur

A camera with aperture leads to defocus blur/depth-of-field effects. The aperture is usually modelled as a 2D shape, e.g., a square, a circle, or a star. The shaper is sampled to determine the origin of each primary sample ray. These rays pass through the positions on the focal plane corresponding to the pixels. For lens aperture $\mathbb{A} \subset \mathbb{R}^2$, we obtain:

$$L_{(i,j)} = \int_{\mathbb{A}} f_{(i,j)}(s)\mathrm{d}s.$$

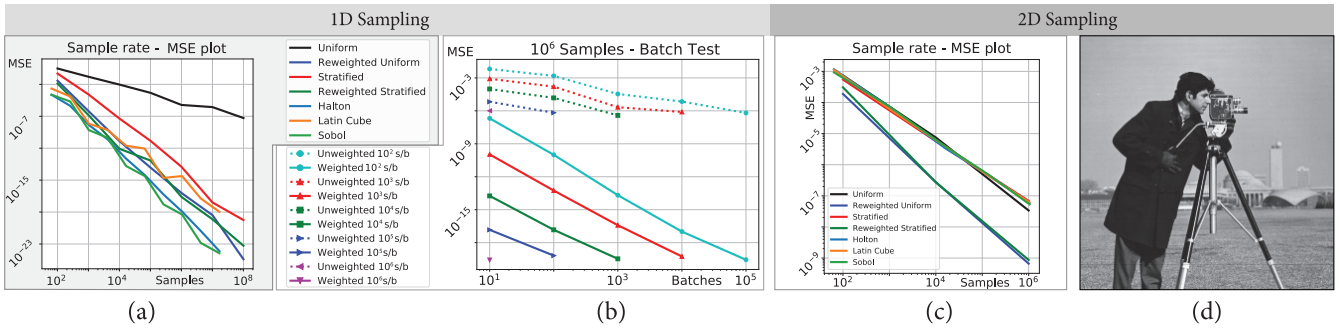


Figure 3: *We apply our geometric sample reweighting to one and two dimensional MC integration problems. Subfigures (a) & (b) show results for 1D case, and (c) shows a 2D integration problem where the underlying function is an image shown in (d). Performance gain of our reweighting scheme can be observed from the faster convergence rate comparing to unweighted ones in 1D cases and all methods in 2D cases. In 1D cases, our unbiased reweighting achieves performance that is close to low discrepency sequence (better than latin hypercube). In 2D cases, our consistent estimator outperforms all methods in test, including all low discrepency sequences. All results are plotted in log-log scale.*

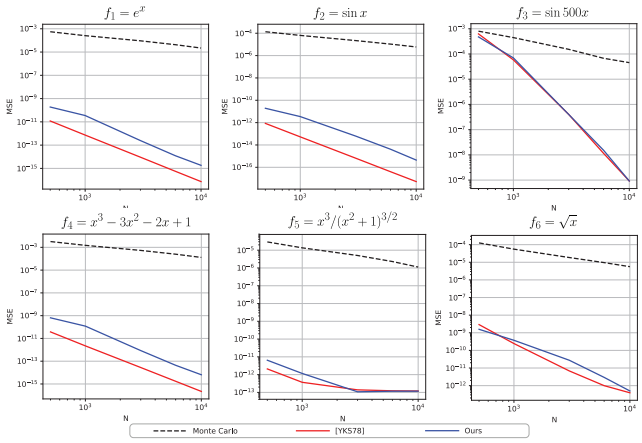


Figure 4: *Log scale MSE plot of three estimators: Monte Carlo, [YKS78] and ours. As can be seen from the slope, our estimator is asymtotically parallel to [YKS78] and is much superior to Monte Carlo. In some cases, our estimator performs less as good, but that is extra cost of remaining unbiased.*

To determine our weights, we use a 2D Voronoi diagram based on the aperture samples. We tested a simple glossy sphere illuminated by an environment map (Fig. 9).

5.2.4. Direct Illumination

Leaving out irrelevant terms, the luminance $L_{\mathbf{x}}$ at a scattering point $\mathbf{x}$ with one bounce is given by:

$$L_{\mathbf{x}} = L_e(\mathbf{x}) + L_{\text{direct}} + L_{\text{indirect}}$$
$$= L_e(\mathbf{x}) + \int_{\mathbb{L}} f_s(\mathbf{x}) L_e(l \to \mathrm{x}) \mathrm{d}l + \int_{\Omega} f_s(\mathbf{x}, \omega) L_i(\omega) \mathrm{d}\omega,$$

where L_e denotes light emission and $l \in \mathbb{L}$ denotes all light sources. In this application, we use light sampling instead of random rays to ensure that the light source is always sampled. Our reweighting achieves the best convergence and, as shown in the insets, also the smoothest results (Fig.10∼12). Each image uses 4 primary rays per pixel and at each scattering event 256 light samples. Our solution is applied to these 256 samples.

6. Discussion

In all cases, our solution leads to smoother visual result and less black pixels in the falloff regions. It improves uniform sampling and stratified sampling comparing to unweighted ones. From the MSE plots, we can see that standard MC with uniform sampling has the worst performance, while our weighted stratified sampling generally performs best. In some cases where the integrand is smooth, stratified sampling alone achieves good results. We can also see that even with uniform sampling as input, our weighted uniform sampling not only improves upon the unweighted version but also has a performance comparable to our weighted stratified sampling.

When comparing tests with low discrepency sequences (Halton, (0,2), Hammersley and Sobol sequences), we observe the general trend of seemingly biased convergence, as is evidently shown in Fig.5, Fig.6, Fig.9 and Fig.10∼Fig.12, where the MSE plot curve shows flattening trend towards higher sample rate. This is partly due to the fact that the correlation patterns (errors) usually take huge amount of samples to converge, as such low discrepency sequences (*quasi Monte Carlo, qmc*) work by trading off bias and variance [Caf98]. Moreover, most renderings with Halton, Hammersley and Sobol sequences show strong correlation/aliasing in images. In low sample rate cases, lower error could be achieved with low discrepency sequences, at the cost of obvious correlation/aliasing.

Due to the fact that we pre-compute samples and weights, the runtime for unweighted and weighted rendering are identical and therefore not reported. Pre-computing sample weights has a negligible overhead (e.g., on a 8-core CPU: less than 2 seconds for 1024 one dimensional samples in Fig. 7, which takes few hours to render at a resolution of 512×512; around 5 milliseconds overhead per pixel for 1024 two dimensional samples in Fig. 10∼Fig.12. One

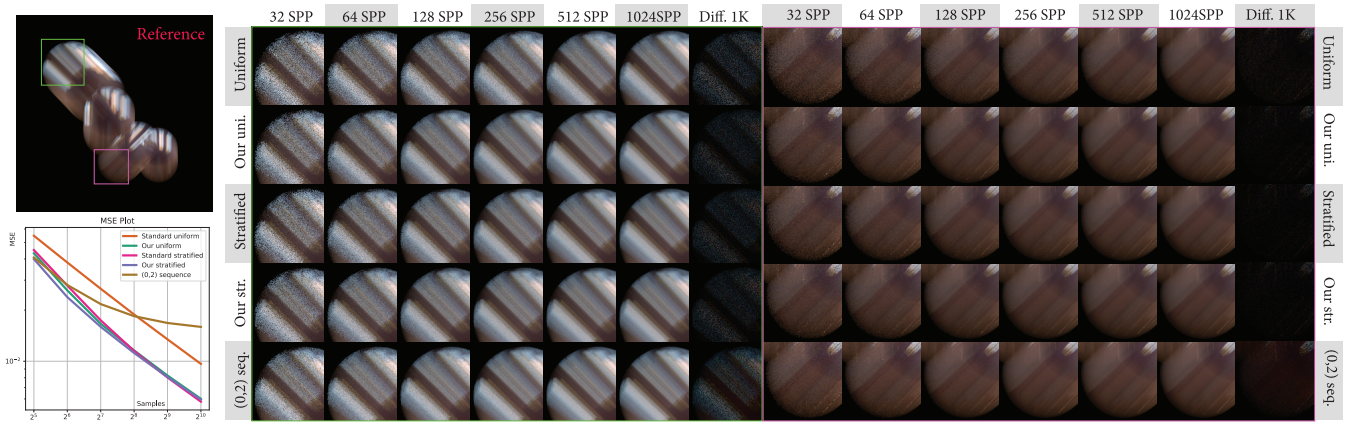


Figure 5: *Four highly glossy spheres moving in different directions. In each subfigure: corresponding renderings of increasing sample rate, difference of* 1*K sample rate with reference. A much brighter difference (with reference) can be observed for (0,2) sequences, which confirms the MSE plot at the bottom left. All results are plotted in log-log scale.*

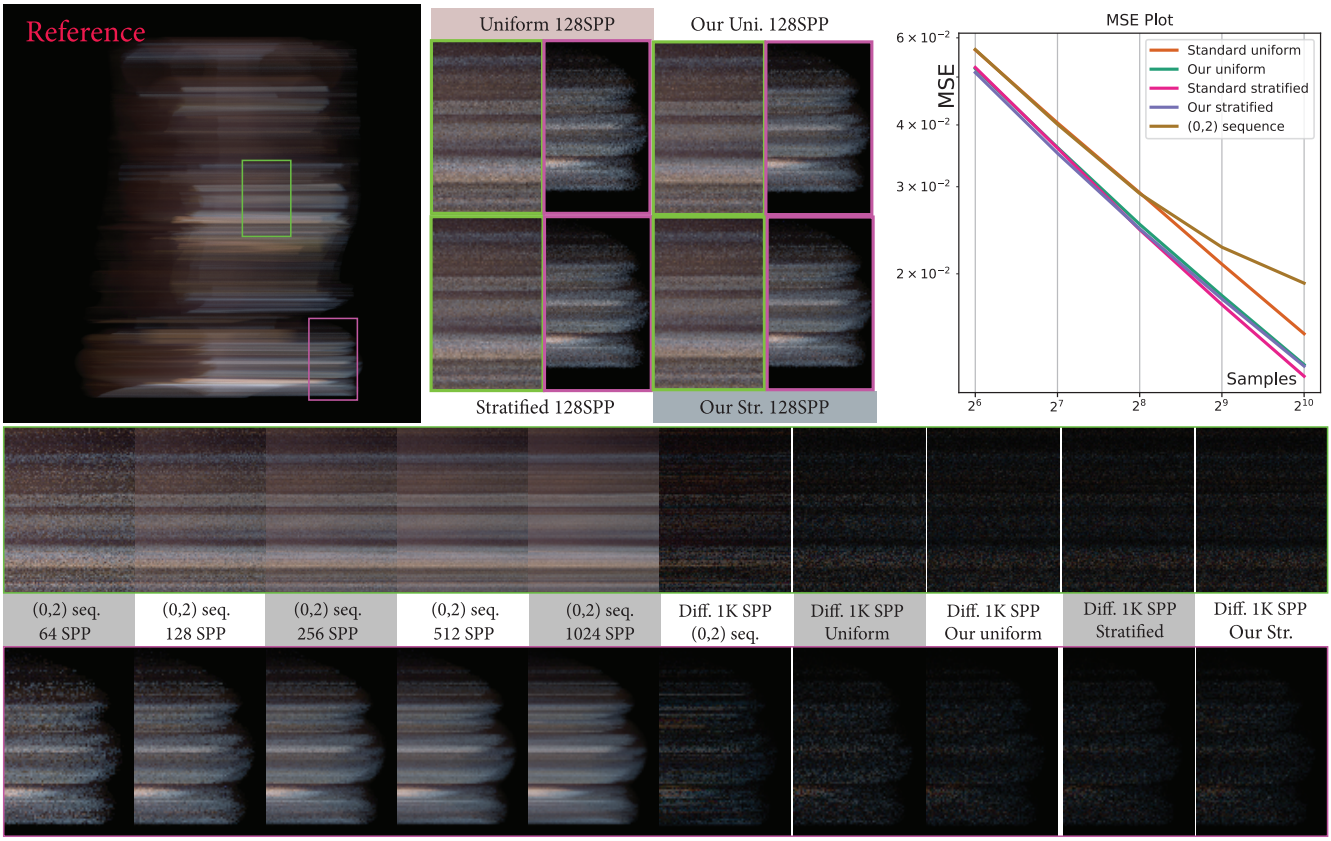


Figure 6: *Highly glossy Buddha moving horizontally with* 128 *samples per pixel. Bottom two rows: renderings of corresponding regions for (0,2) sequences and the* 1*K sample rate difference of all methods. Similar behavior is observed for (0,2) sequences as in Fig.5. All results are plotted in log-log scale.*

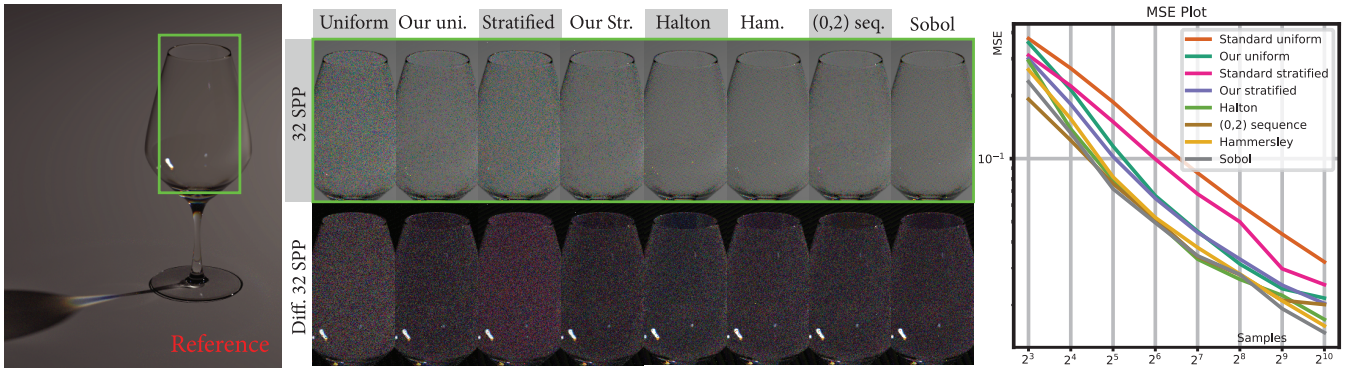


Figure 7: *Wineglass with dispersive dielectric materials with* 32 *samples per pixel. As can be seen, our reweighted schemes improve upon uniform and stratified samples and produces much smoother result in terms of color noise. Significant improvements can be observed in low discrepency sequences. All results are plotted in log-log scale.*

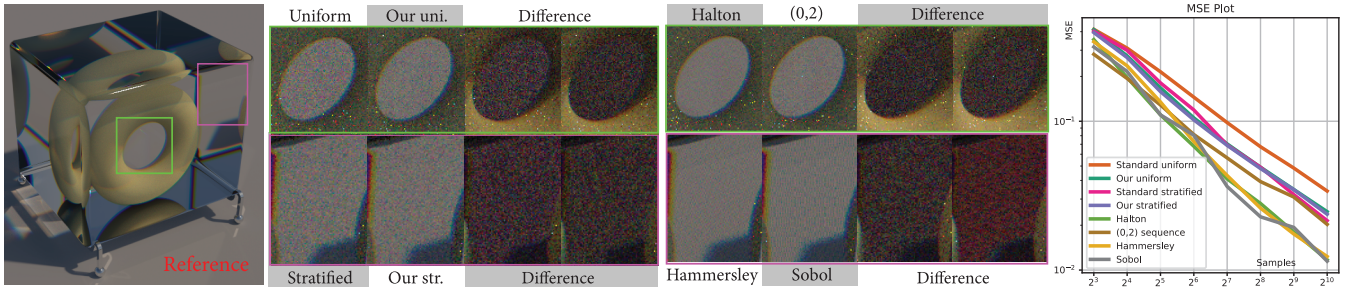


Figure 8: *Torus with dielectric materials with* 64 *samples per pixel. Our reweighted schemes produce smoother renderings while avoiding aliasing that is obvious in low discreoency renderings. All results are plotted in log-log scale.*

can always precompute weights and reuse for different integrals, e.g., pixels in different render tiles.

7. Conclusion

Our reweighting scheme enables a better approximation than standard MC weights. It is general and does not require any prior knowledge about the Riemann integrable function. Implicitly, it approximates the integrand via a reconstruction from the samples. We presented an analysis of the introduced bias of a consistent estimator based on Voronoi cell volumes and propose an unbiased 1D estimator. We showed the practical benefit of our solution for various rendering problems.

Despite the system error introduced for high-dimensional integration, we observe huge improvement in terms of variance reduction. Nevertheless, it requires the computation of high-dimensional Voronoi cell volumes, which can be costly. Still, we demonstrate several low-dimensional application cases in this paper.

As our method makes no assumptions about the function to be integrated, it is an attractive approach for many purposes and can be easily integrated in existing solutions (e.g., ReSTIR [BWP*20]). We believe our method could be of interest beyond graphics community.

8. Acknowledgments

This work was supported by the NWO Vernieuwingsimpuls VIDI grant *Next View*. The authors would like to thank the reviewers for the valuable comments and suggestions. Special thanks for Dr. Bauszat, Dr. Billeter, Dr. Ruszel and Chongchuo Li for fruitful discussions. The majority of the test scenes used in this work are based on rendering resources released by Bitterli.

References

[ARBJ03] AGARWAL S., RAMAMOORTHI R., BELONGIE S., JENSEN H. W.: Structured importance sampling of environment maps. SIGGRAPH '03, Association for Computing Machinery, p. 605–612. doi:10.1145/1201775.882314. 2

[Aur91] AURENHAMMER F.: Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR) 23*, 3 (1991), 345–405. doi:10.1145/116873.116880. 3

[BDM09] BERGNER S., DREW M. S., MÖLLER T.: A tool to create illuminant and reflectance spectra for light-driven graphics and visualization. *ACM Transactions on Graphics (TOG) 28*, 1 (2009), 5. doi:10.1145/1477926.1477931. 5

[BWP*20] BITTERLI B., WYMAN C., PHARR M., SHIRLEY P., LEFOHN A., JAROSZ W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 39*, 4 (July 2020). doi:10/gg8xc7. 2, 8

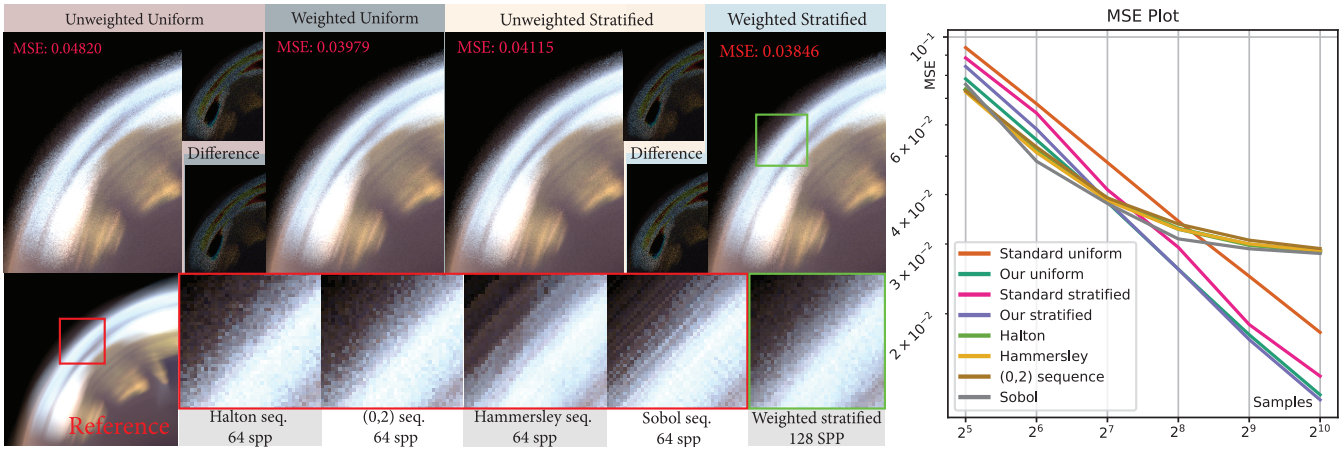


Figure 9: *Glossy sphere crossing the focal plane of the camera. Top:* 128 *samples per pixel; bottom: equal error renderings of low discrepency sequences. Lower error can be observed in low discrepency sequence renderings, at the cost of noticeable correlation patterns. All results are plotted in log-log scale.*

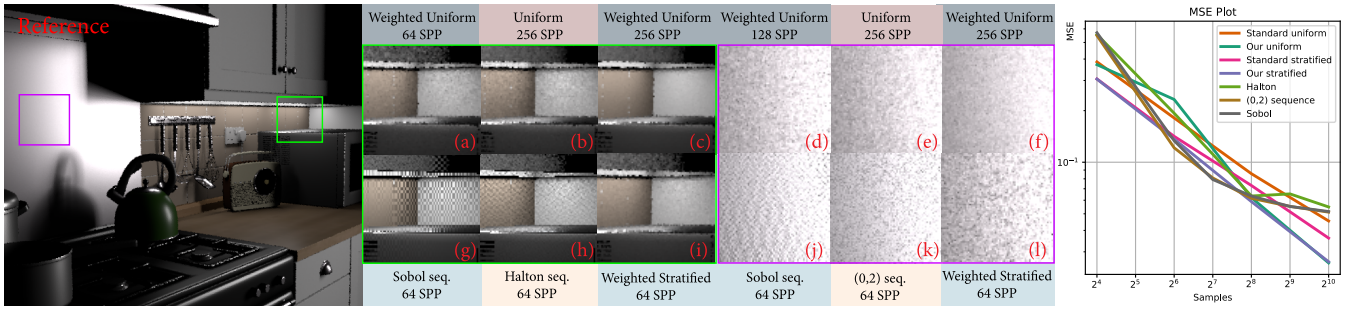


Figure 10: *An equal quality and sample rate comparison of applying our reweighting in the* Kitchen *scene. In subfigures (b)&(e):* 256 *samples per pixel with unweighted uniform; in (a)&(d): equal error rendering (comparing to (b)&(e)) is achieved with weighted uniform at* 64&128 *samples per pixel; in (c)&(f) weighted uniform results in a much smoother rendering at the equal sample rate of* 256 *samples per pixel (comparing to (b)&(e)); subfigures (g)-(l): equal error renderings of low discrepency sequences and the weighted stratified rendering. As can be observed from the results, our weighted schemes delivers results close in quality with lower sample rates and better converged results at equal sample rate. Despite the low error renderings, low discrepency sequences result in obvious patterns of correlation. All results are plotted in log-log scale.*

[Caf98] CAFLISCH R. E.: Monte carlo and quasi-monte carlo methods. *Acta Numerica 7* (1998), 1–49. doi:10.1017/S0962492900002804. 6

[CPC84] COOK R. L., PORTER T., CARPENTER L.: Distributed ray tracing. In *ACM SIGGRAPH Computer Graphics* (1984), vol. 18, ACM, pp. 137–145. doi:10.1145/800031.808590. 1, 2

[DBB06] DUTRE P., BEKAERT P., BALA K.: *Advanced global illumination*. AK Peters/CRC Press, 2006. 2

[DBVKOS97] DE BERG M., VAN KREVELD M., OVERMARS M., SCHWARZKOPF O.: Computational geometry. In *Computational geometry*. Springer, 1997, pp. 1–17. 3

[Deb08] DEBEVEC P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes* (2008), ACM, p. 32. doi:10.1145/1401132.1401175. 2

[DN04] DAVID H. A., NAGARAJA H. N.: *Order statistics*, 3rd ed. Wiley Online Library, 2004. doi:10.1002/0471722162. 3

[EHDR11] EGAN K., HECHT F., DURAND F., RAMAMOORTHI R.: Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Transactions on Graphics (TOG) 30*, 2 (2011), 9. doi:10.1145/1944846.1944849. 2

[ETH*09] EGAN K., TSENG Y.-T., HOLZSCHUCH N., DURAND F., RAMAMOORTHI R.: Frequency analysis and sheared reconstruction for rendering motion blur. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 93. doi:10.1145/1540000. 1, 2

[HJW*08] HACHISUKA T., JAROSZ W., WEISTROFFER R. P., DALE K., HUMPHREYS G., ZWICKER M., JENSEN H. W.: Multidimensional adaptive sampling and reconstruction for ray tracing. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 33. doi:10.1145/1399504.1360632. 2

[Jak10] JAKOB W.: Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 5

[Kaj86] KAJIYA J. T.: The rendering equation. In *ACM Siggraph Computer Graphics* (1986), vol. 20, ACM, pp. 143–150. doi:10.1145/280811.280987. 2

[LAC*11] LEHTINEN J., AILA T., CHEN J., LAINE S., DURAND F.:

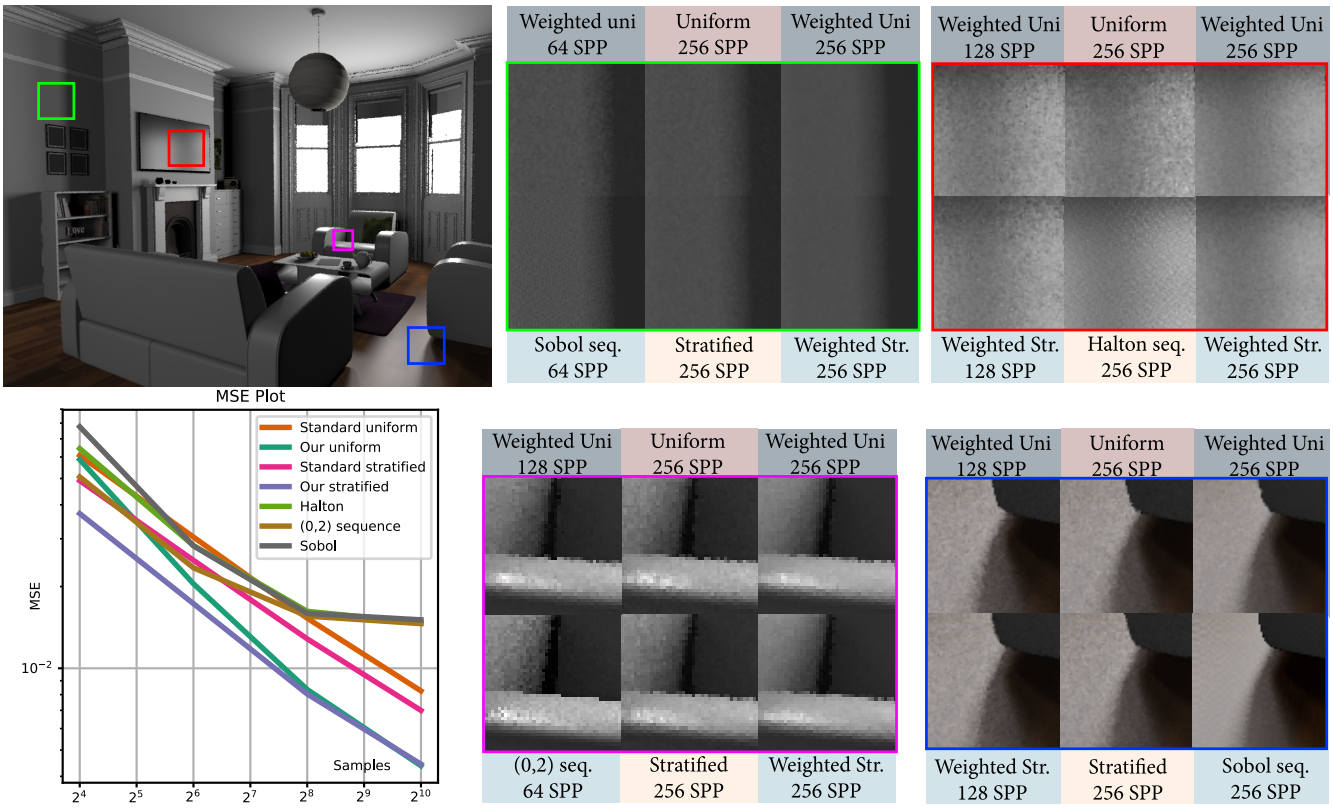


Figure 11: *An equal quality and sample rate comparison of applying our reweighting in the* Livingroom *scene. As can be observed from the results, our weighted renderings achieve better convergence at equal sample rate comparing to unweighted ones and use less samples reaching similar error level. Noticeable patterns of correlation are observed in low discrepency sequence renderings. All results are plotted in log-log scale.*

Temporal light field reconstruction for rendering distribution effects. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 55. doi:10.1145/2010324.1964950. 2

[LFTG97] LAFORTUNE E. P., FOO S.-C., TORRANCE K. E., GREENBERG D. P.: Non-linear approximation of reflectance functions. In *Proceedings of Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 117–126. doi:10.1145/258734.258801. 2

[MBB18] MARQUES R., BOUVILLE C., BOUATOUCH K.: Optimal sample weights for hemispherical integral quadratures. In *Computer Graphics Forum* (2018), Wiley Online Library. doi:10.1111/cgf.13392. 2

[Mit90] MITCHELL D. P.: The antialiasing problem in ray tracing. *Advanced Topics in Ray Tracing, SIGGRAPH 1990 Course Notes* (1990). 2

[Shi91] SHIRLEY P. S.: *Physically based lighting calculations for computer graphics*. PhD thesis, University of Illinois at Urbana-Champaign, 1991. 2

[SSD*09] SOLER C., SUBR K., DURAND F., HOLZSCHUCH N., SILLION F.: Fourier depth of field. *ACM Transactions on Graphics (TOG) 28*, 2 (2009), 18. doi:10.1145/1516522.1516529. 1, 2

[TCE05] TALBOT J., CLINE D., EGBERT P.: Importance Resampling for Global Illumination. In *Eurographics Symposium on Rendering (2005)* (2005), Bala K., Dutre P., (Eds.), The Eurographics Association. doi:10.2312/EGWR/EGSR05/139-146. 3

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of Computer graphics and interactive techniques* (1995), ACM, pp. 419–428. doi:10.1145/218380.218498. 2

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 65–76. doi:10.1145/258734.258775. 2

[Vit85] VITTER J. S.: Random sampling with a reservoir. *ACM Trans. Math. Softw. 11*, 1 (Mar. 1985), 37–57. URL: https://doi.org/10.1145/3147.3165, doi:10.1145/3147.3165. 2

[VSE16] VORECHOVSK M., SADILEK V., ELIAS J.: Application of voronoi weights in monte carlo integration with a given sampling plan. In *Proceedings of REC2016* (2016), Inderscience Publishers, pp. 441–452. 2, 4

[WND*14] WILKIE A., NAWAZ S., DROSKE M., WEIDLICH A., HANIKA J.: Hero wavelength spectral sampling. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 123–131. doi:10.1111/cgf.12419. 5

[YKS78] YAKOWITZ S., KRIMMEL J., SZIDAROVSZKY F.: Weighted monte carlo integration. *SIAM Journal on Numerical Analysis 15*, 6 (1978), 1289–1300. 2, 5, 6

[ZHD18] ZIRR T., HANIKA J., DACHSBACHER C.: Re-weighting firefly samples for improved finite-sample monte carlo estimates. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 410–421. doi:10.1111/cgf.13335. 2

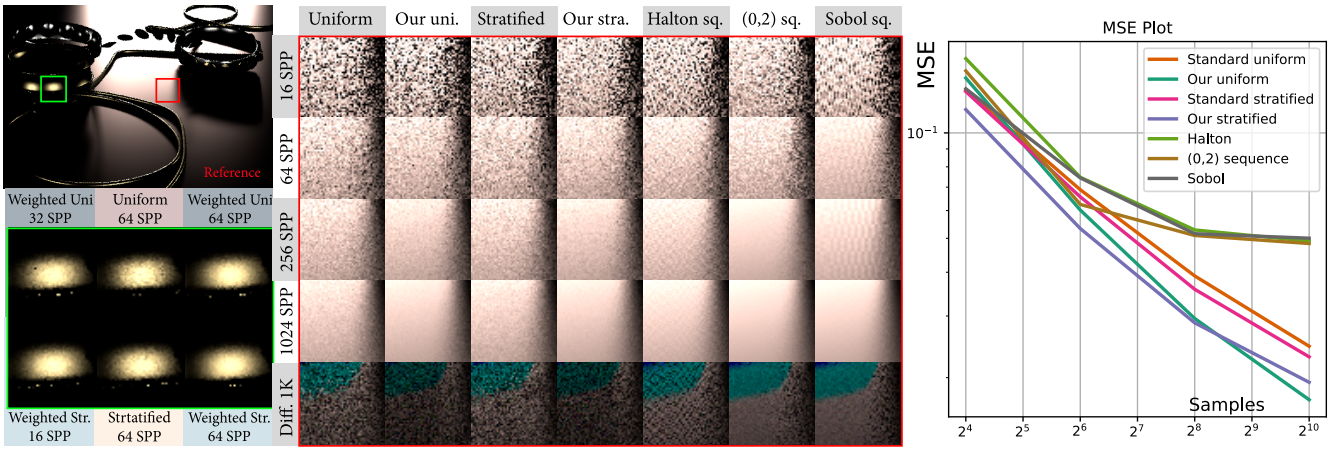


Figure 12: *An equal quality and sample rate comparison of applying our reweighting in the* Necklace *scene. Similar behavior can be observed as in Fig.11. (0,2) sequences manage to produce alias-free renderings. However, brighter difference comparing to reference is observed, which confirms the MSE plot in the right subfigure. All results are plotted in log-log scale.*

Appendix A: Expectation of integration over given sample set

Here, we demonstrate the process of deriving expectation of a classic MC estimator. We make use of the following identity, common for multiple integration:

$$\int_0^1\int_{x_1}^1\cdots\int_{x_{N-1}}^1 \mathrm{d}x_N\mathrm{d}x_{N-1}\cdots\mathrm{d}x_1 = \int_0^1\int_0^{x_N}\cdots\int_0^{x_2}\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_N.$$

It can be shown by induction that:

$$\int_0^{x_{i+1}}\int_0^{x_i}\cdots\int_0^{x_2}\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_i = \frac{x_{i+1}^i}{i!},$$
$$\int_{x_{i-1}}^1\int_{x_i}^1\cdots\int_{x_{N-1}}^1\mathrm{d}x_N\mathrm{d}x_{N-1}\cdots\mathrm{d}x_i = \frac{(1-x_{i-1})^{N-i+1}}{(N-i+1)!}.$$

Plugging these results into Eq.5 gives:

$$\begin{aligned}
\mathbb{E}\left[\hat{I}_{MC}\right] &= N!\int_0^1\int_{x_1}^1\cdots\int_{x_{N-1}}^1 \\
&\quad\left[\frac{1}{N}f(x_1)+\frac{1}{N}f(x_2)+\cdots+\frac{1}{N}f(x_N)\right]\mathrm{d}x_N\mathrm{d}x_{N-1}\cdots\mathrm{d}x_1 \\
&= N!\cdot\frac{1}{N}\left[\int_0^1 f(x_1)\int_{x_1}^1\cdots\int_{x_{N-1}}^1\mathrm{d}x_N\cdots\mathrm{d}x_2\mathrm{d}x_1+\cdots\right. \\
&\quad+\int_0^1\cdots\int_{x_{i-1}}^1 f(x_i)\cdots\int_{x_{N-1}}^1\mathrm{d}x_N\mathrm{d}x_{N-1}\cdots\mathrm{d}x_1+\cdots \\
&\quad\left.+\int_0^1\int_{x_1}^1\cdots\int_{x_{N-1}}^1 f(x_N)\mathrm{d}x_N\mathrm{d}x_{N-1}\cdots\mathrm{d}x_1\right] \\
&= N!\cdot\frac{1}{N}\left[\int_0^1\frac{(1-x_1)^{N-1}}{(N-1)!}f(x_1)dx_1+\cdots\right. \\
&\quad+\int_{x_{i-1}}^1\frac{x_i^{i-1}}{(i-1)!}\frac{(1-x_i)^{N-i}}{(N-i)!}f(x_i)\mathrm{d}x_i+\cdots \\
&\quad\left.+\int_{x_{N-1}}^1\frac{x_N^{N-1}}{(N-1)!}f(x_N)\mathrm{d}x_N\right] \\
&= N!\cdot\frac{1}{N}\left[\int_0^1\frac{(1-x)^{N-1}}{(N-1)!}f(x)\mathrm{d}x+\cdots+\right. \\
&\quad\int_0^1\frac{x^{i-1}}{(i-1)!}\frac{(1-x)^{N-i}}{(N-i)!}f(x)\mathrm{d}x+\cdots \\
&\quad\left.+\int_0^1\frac{x^{N-1}}{(N-1)!}f(x)\mathrm{d}x\right] \\
&= N!\cdot\frac{1}{N}\int_0^1\sum_{i=1}^N\frac{x^{i-1}}{(i-1)!}\frac{(1-x)^{N-i}}{(N-i)!}f(x)\mathrm{d}x \\
&= \int_0^1\sum_{i=1}^N\frac{(N-1)!}{(i-1)!(N-i)!}x^{i-1}(1-x)^{N-i}f(x)\mathrm{d}x \\
&= \int_0^1\cdot[x+(1-x)]^{N-1}\cdot f(x)\mathrm{d}x = \int_0^1 f(x)\mathrm{d}x.
\end{aligned}$$

Appendix B: Expectation of integration with augmented periodic samples

By augmenting samples to get a periodic sample set (one can imagine this process as creating a circular domain, where the last and first sample are hereby influencing each other's weight):

$$\begin{aligned}
&\mathbb{E}\left[\hat{I}_P\right] = \\
&\quad N!\int_0^1\int_{x_1}^1\cdots\int_{x_{N-1}}^1 \\
&\quad\left[\frac{1+x_2-x_N}{2}f(x_1)+\sum_{i=2}^{N-1}\frac{x_{i+1}-x_{i-1}}{2}f(x_i)+\right. \\
&\quad\left.\left(\frac{1+x_1-x_{N-1}}{2}\right)f(x_N)\right]\mathrm{d}x_N\mathrm{d}x_{N-1}\cdots\mathrm{d}x_1,
\end{aligned}$$

which integrates to $\int_0^1((-1)^{1+n}n(x-1)^n(n+x-2)+(x-1)(nx+2x-1-(1-x)^n(n+x-1+(1/(1-x))^n(nx+1))))/(2(x-1)^2)f(x)\mathrm{d}x$. Again, this process results in a biased estimator.