My favorite Samples

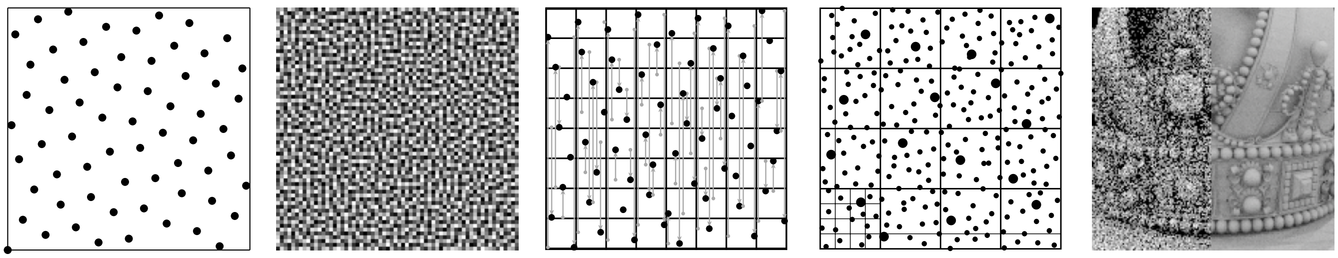
Alexander Keller
NVIDIA
keller.alexander@gmail.com

Iliyan Georgiev
Autodesk
me@iliyan.com

Abdalla Ahmed
King Abdulla University of Science and Technology
abdalla_gafar@hotmail.com

Per Christensen
Pixar
per@pixar.com

Matt Pharr
NVIDIA
matt.pharr@gmail.com



a) b) c) d) e)

Figure 1: What samples are best? Will deterministic samples do the job (a), or are blue noise characteristics required to render images without disturbing artifacts (b)? Can we construct such point sets (c) and sequences (d) in an efficient way? It is important to select the appropriate samples for efficient image synthesis (e).

ABSTRACT

Light transport simulation is ruled by the radiance equation, which is an integral equation. Photorealistic image synthesis consists of computing functionals of the solution of the integral equation, which involves integration, too. However, in meaningful settings, none of the integrals can be computed analytically and, in fact, all these integrals need to be approximated using Monte Carlo and quasi-Monte Carlo methods. Generating uniformly distributed points in the unit-hypercube is at the core of all of these methods. The course teaches the algorithms behind and elaborates on the characteristics of different classes of uniformly distributed points to help selecting the points most efficient for a task.

CCS CONCEPTS

• **Theory of computation → Pseudorandomness and derandomization**; • **Computing methodologies → Ray tracing**.

KEYWORDS

Monte Carlo and quasi-Monte Carlo methods, sampling, rendering.

SIGGRAPH '19 Courses, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6307-5/19/07.
https://doi.org/10.1145/3305366.3328096

ACM Reference Format:
Alexander Keller, Iliyan Georgiev, Abdalla Ahmed, Per Christensen, and Matt Pharr. 2019. My favorite Samples. In *Proceedings of SIGGRAPH '19 Courses.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3305366.3328096

1 INTRODUCTION

Sampling is ubiquitous in graphics. It is used for modeling and rendering. The course will survey the state of the art in sampling and provide advice when which sampling pattern will perform best. Whether you use your samples for modeling, numerical integration, or numerical integro-approximation makes a difference and it will be shown why [3, 4]. Terms like consistency, bias, and unbiased algorithms are very important when evaluating arguments pro and con deterministic and random sampling [2, 5]. In fact, one even needs to distinguish whether one pixel is computed or whether a whole image is reconstructed. It is even more relevant given the recent development in removing noise from synthesized images.

While it is simpler to design point sets in low dimensions, high dimensional point sets are of interest for path tracing in light transport simulation. However, the underlying integrals are of low dimension, which poses the question whether it is more efficient to pad optimized low dimensional samples instead of constructing high-dimensional ones.

The design of such algorithms will be discussed and current research challenges will be addressed. As error bounds are of interest, we will point out that the root-mean square error does not tell the whole story and that error bounds depend on both the set of samples and the function class. In fact, besides the classic Monte Carlo error bounds, error bounds are tricky in computer graphics.

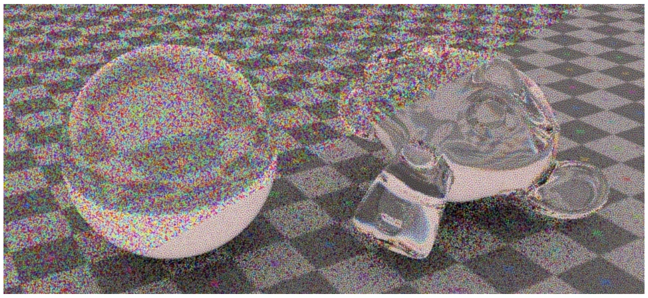


Figure 2: Dithered wavelength sampling correlates the samples between pixels and thus minimizes the low-frequency content in the distribution of the estimation error. Without actually reducing the amount of error, this correlation produces images with higher visual fidelity, especially when using a small number of samples per pixel. The example image uses one sample per pixel with uncorrelated sampling above and correlated sampling below the diagonal.

In particular, the course will share industry experience and new insights. As everything starts with uniformly distributed samples in the s-dimensional unit hypercube $[0, 1)^s$, the course will review the concepts of Monte Carlo and quasi-Monte Carlo methods. It will then address the relevance of samples with blue noise characteristics in rendering in Sec. 1.1 and the construction of low discrepancy point sets and sequences in Sec. 1.2 and Sec. 1.3. The final Sec.1.4 will underline the importance of sampling in rendering algorithms with respect to convergence speed.

1.1 Blue-Noise Dithered Sampling

The visual fidelity of a Monte Carlo rendered image depends not only on the magnitude of the pixel estimation error but also on its distribution over the image. To this end, state-of-the-art methods use high-quality stratified sampling patterns, which are randomly scrambled or shifted to decorrelate the individual pixel estimates. While the white-noise image error distribution produced by random pixel decorrelation is eye-pleasing, it is far from being perceptually optimal. Visual fidelity can be significantly improved by instead *correlating* the pixel estimates in a way that minimizes the low-frequency content in the output noise. One way to achieve this is via blue-noise dithered sampling [3]. This technique, inspired by digital halftoning, can produce substantially more faithful images, especially at low sampling rates as shown for one example in Fig. 2.

1.2 Low-Discrepancy Blue Noise Sampling

A technique is presented that produces two-dimensional low discrepancy blue noise point sets [1]. Using one-dimensional binary van der Corput sequences, we construct two-dimensional low discrepancy point sets, and rearrange them to match a target spectral profile while preserving their low discrepancy. The rearrangement information is stored in a compact lookup table that can be used to produce arbitrarily large point sets. We evaluate the technique and compare it to the state-of-the-art sampling approaches.

1.3 Progressive Multi-Jittered Sequences

Three very useful classes of algorithms to generate randomized progressive uniformly distributed point sequences in two dimensions are introduced. The best sequences have the same low sampling error as a particular randomization of the popular Sobol' (0,2)-sequence in base 2. The sample points are generated using a simple alternating strategy that progressively fills in holes in increasingly fine stratifications. The sequences are progressive and can be considered hierarchical, as any prefix is well distributed, making them suitable for incremental rendering and adaptive sampling. The first family of samples is only jittered in two dimensions; we call it *progressive jittered.* It is nearly identical to existing sample sequences. The second family is multi-jittered: the samples are stratified in both one and two dimensions; we call it *progressive multi-jittered.* The third family is stratified in all elementary intervals in base 2, hence we call it *progressive multi-jittered (0,2)-sequence* [2]. The sequences have been compared to other sequences in numerical experiments. We tested the sequences on function integration and in two settings that are typical for computer graphics: pixel sampling and area light sampling. Within this framework we describe variations that generate visually pleasing samples with blue noise spectra, and well-stratified interleaved multi-class samples.

1.4 The Importance of Sampling

With the recent arrival of ray tracing to the real-time graphics pipeline, sampling well has become important to a much wider range of developers than it has before. Carefully constructed sampling patterns are just the start. Given a limited number of rays that can be traced per frame, one important question to decide is for which lighting effects to trace rays – choices include shadows, reflections, ambient occlusion, and full global illumination. Another important question is how to choose which rays to trace for the chosen effect to maximize results. This part of the course will start by reviewing previous work in this area from offline rendering and discuss how it now applies to real-time.

In the context of real-time, this task is made more challenging by the widespread use of temporal reprojection and denoising algorithms – well-distributed samples at one pixel can easily be thwarted by adding in reprojected values from other pixels that have used completely different samples. Denoisers may as much prefer varied results at a range of nearby pixels as much as accurate results at one of them. Thus, the approaches that initially seem to be the best may not be in practice. We will describe the challenges in this area and outline a few different ways of approaching them.

REFERENCES

[1] A. Ahmed, H. Perrier, D. Coeurjolly, V. Ostromoukhov, J. Guo, D. Yan, H. Huang, and O. Deussen. 2016. Low-discrepancy Blue Noise Sampling. *ACM Transactions on Graphics* 35, 6, Article 247 (2016), 247:1–247:13 pages.
[2] P. Christensen, A. Kensler, and C. Kilpatrick. 2018. Progressive Multi-Jittered Sample Sequences. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering 2018)* 37, 4 (2018), 21–33.
[3] I. Georgiev and M. Fajardo. 2016. Blue-noise Dithered Sampling. *ACM SIGGRAPH 2016 Talks* (2016).
[4] A. Keller. 2006. Myths of Computer Graphics. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*, H. Niederreiter (Ed.). Springer, 217–243.
[5] A. Keller. 2013. Quasi-Monte Carlo Image Synthesis in a Nutshell. In *Monte Carlo and Quasi-Monte Carlo Methods 2012*, J. Dick, F. Kuo, G. Peters, and I. Sloan (Eds.). Springer, 203–238.