Stratification

UCSD CSE 272
Advanced Image Synthesis

Tzu-Mao Li

with slides & images from Wojciech Jarosz & Gurprit Singh
Sampling pattern matters

which one is better?
Noise v.s. aliasing trade-offs
A middle ground?
Comparison

per pixel (relative) error
Comparison

per pixel (relative) error
Questions

- Are there other ways to stratify?
- How do we generalize this to high-dimensional space?
- What are the mathematical tools we have for analyzing these patterns?
- Pros and cons between different patterns?
Frequency analysis of Monte Carlo integration

\[
\int f(x)\,dx \approx \frac{1}{N} \sum_{i=0}^{N} f(x_i) = \int f(x)S(x)\,dx \quad S(x) = \sum \delta(x - x_i)
\]
Frequency analysis of Monte Carlo integration

- numerical integration = taking DC of the convolution between sampling patterns & integrand in frequency domain

\[ \int f(x)S(x)dx = \hat{f} \otimes \hat{S}(0) \]
Frequency analysis of Monte Carlo integration

- numerical integration = taking DC of the convolution between sampling patterns & integrand in frequency domain

Fourier integrand $\hat{f}$

Fourier sampling pattern $\hat{S}$

$\hat{S} \otimes \hat{f}$ (convolution)

high sampling rate

don't show

low sampling rate

don't show

error in integration
Frequency analysis of Monte Carlo integration

- numerical integration = taking DC of the convolution between sampling patterns & integrand in frequency domain

want to avoid low frequency spikes!
Observation: $S$ is a random variable

$$\int f(x)S(x)dx = \hat{f} \otimes \hat{S}(0)$$
Bias-variance analysis in Fourier domain

\[ F = \int f(x)dx \]

mean square error = bias\(^2\) + variance

\[ F_{est} = \int f(x)S(x)dx = \hat{f} \otimes \hat{S}(0) \]

\[ E \left[ (F_{est} - F)^2 \right] = E \left[ F_{est} - F \right]^2 + \text{Var} \left[ F_{est} - F \right] \]
Bias-variance analysis in Fourier domain

\[ F = \int f(x)S(x)dx = \hat{f} \otimes \hat{S}(0) \]

bias = \( \hat{f}(0) - \int \hat{f}^*(\omega)E[\hat{S}(\omega)]d\omega \)

variance = \( \int |\hat{f}(\omega)|^2 E \left[ |\hat{S}(\omega)|^2 \right] d\omega \)

(slightly simplified)
Bias-variance analysis in Fourier domain

\[ F = \int f(x)S(x)dx = \hat{f} \otimes \hat{S}(0) \]

\[
\text{bias} = \hat{f}(0) - \int \hat{f}^*(\omega)E[\hat{S}(\omega)]d\omega
\]

for most random samplers, \( E[\hat{S}(\omega)] = 0 \) iff \( \omega \neq 0 \)

\[
\text{variance} = \int |\hat{f}(\omega)|^2 E\left[ |\hat{S}(\omega)|^2 \right] d\omega
\]

(slightly simplified)
Bias-variance analysis in Fourier domain

\[ F = \int f(x)S(x)dx = \hat{f} \otimes \hat{S}(0) \]

\[ \text{bias} = \hat{f}(0) - \int \hat{f}^*(\omega)E[\hat{S}(\omega)]d\omega \]

\[ \text{variance} = \int \left| \hat{f}(\omega) \right|^2 E \left[ \left| \hat{S}(\omega) \right|^2 \right] d\omega \]

for most random samplers, \( E[\hat{S}(\omega)] = 0 \) iff \( \omega \neq 0 \)

the expected power spectrum of the sampling pattern is the key!!

(slightly simplified)
Variance analysis = multiplication of power spectrums

- natural signals/integrands usually have energy concentrated at low frequencies
- sampling patterns with small low frequency energy are better!!

\[
E \left[ |\hat{S}(\omega)|^2 \right]
\]

\[
\text{variance} = \int |\hat{f}(\omega)|^2 E \left[ |\hat{S}(\omega)|^2 \right] d\omega
\]
Let’s look at different sampling patterns!

slides heavily borrowed from Wojciech Jarosz
https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier.html
Independent random sampling

for (int k = 0; k < num; k++)
{
    samples(k).x = randf();
    samples(k).y = randf();
}

✔ Trivially extends to higher dimensions
✔ Trivially progressive and memory-less
✘ Big gaps
✘ Clumping
Frequency analysis of independent random sampling

Chapter 5. Popular sampling patterns

Samples

\[
\frac{1}{N} \sum_{k=1}^{N} \delta(|\vec{x} - x_k|)
\]

Power spectrum

\[
\left| \frac{1}{N} \sum_{k=1}^{N} e^{-2 \pi i \langle \tilde{\omega}, \vec{x}_k \rangle} \right|^2
\]
advocated three important features for an ideal radial power spectrum; First, its peak should be at

\( \frac{1}{N} \sum_{k=1}^{N} \delta(|\vec{x} - \vec{x}_k|) \)

sequence is called the Hammersley sequence, which can create a even lower discrepancy point set

\( \text{Expected power spectrum} \)

\( \vec{\omega}_y \)

\( \vec{\omega}_x \)

\( \sqrt{ \left( \frac{1}{N} \sum_{k=1}^{N} e^{-2\pi \imath (\vec{\omega} \cdot \vec{x}_k)} \right)^2} \)
Useful to visualize the radial mean of expected power spectrum

\[
\frac{1}{N} \sum_{k=1}^{N} \delta(|\vec{x} - \vec{x}_k|) \quad E \left[ \left| \frac{1}{N} \sum_{k=1}^{N} e^{-2 \pi i (\vec{\omega} \cdot \vec{x}_k)} \right|^2 \right]
\]
Regular sampling: high bias, zero variance

for (uint i = 0; i < numX; i++)
    for (uint j = 0; j < numY; j++)
    {
        samples(i,j).x = (i + 0.5)/numX;
        samples(i,j).y = (j + 0.5)/numY;
    }

✔ Extends to higher dimensions, but...
✘ Curse of dimensionality
✘ Aliasing
Jittered/stratified sampling: zero bias, low variance

for (uint i = 0; i < numX; i++)
  for (uint j = 0; j < numY; j++)
  {
    samples(i,j).x = (i + randf())/numX;
    samples(i,j).y = (j + randf())/numY;
  }

✔ Provably cannot increase variance
✔ Extends to higher dimensions, but...
✘ Curse of dimensionality
✘ Not progressive
Chapter 5. Popular sampling patterns

- Random
- Jitter
- Multi-jitter
- N-rooks

Figure 5.6: Illustration of random and some stochastic grid-based sampling patterns with the corresponding Fourier expected power spectra and the corresponding radial mean of their expected power spectra.

5.3 Blue noise

Any sampling pattern with Blue noise characteristics is supposed to be well distributed within the spatial domain without containing any regular structures. The term Blue noise was coined by Ulichney [47], who investigated a radially averaged power spectrum of various sampling patterns. He advocated three important features for an ideal radial power spectrum; First, its peak should be at...
Random sampling vs jittered sampling

Figure 5.6: Illustration of random and some stochastic grid-based sampling patterns with the corresponding Fourier expected power spectra and the corresponding radial mean of their expected power spectra.

5.3 Blue noise

Any sampling pattern with Blue noise characteristics is supposed to be well distributed within the spatial domain without containing any regular structures. The term Blue noise was coined by Ulichney [47], who investigated a radially averaged power spectrum of various sampling patterns. He advocated three important features for an ideal radial power spectrum: First, its peak should be at

<table>
<thead>
<tr>
<th>Samples</th>
<th>Power spectrum</th>
<th>Radial mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Random sampling (16 samples per pixel)
Jittered sampling (16 samples per pixel)
High-dimensional stratification is hard

Stratification requires $O(N^d)$ samples

- e.g. pixel (2D) + lens (2D) + time (1D) = 5D
  - splitting 2 times in 5D = $2^5 = 32$ samples
  - splitting 3 times in 5D = $3^5 = 243$ samples!

Inconvenient for large $d$

- cannot select sample count with fine granularity
Uncorrelated Jitter [Cook 1986]

Compute stratified samples in sub-dimensions

- 2D jittered (x,y) for pixel
- 2D jittered (u,v) for lens
- 1D jittered (t) for time
- combine dimensions in random order
Not all dimensions are well stratified with uncorrelated jitter
4D integral with uncorrelated jitter

Reference

Random Sampling

Uncorrelated Jitter
Uncorrelated jitter is a special case of Latin hypercube sampling

Stratify samples in each dimension separately
- for 5D: 5 separate 1D jittered point sets
- combine dimensions in random order

<table>
<thead>
<tr>
<th>x</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>x2</td>
<td>x3</td>
<td>x4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>y2</td>
<td>y3</td>
<td>y4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>u</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>u1</td>
<td>u2</td>
<td>u3</td>
<td>u4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>v</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>v2</td>
<td>v3</td>
<td>v4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>t</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>t2</td>
<td>t3</td>
<td>t4</td>
<td></td>
</tr>
</tbody>
</table>
Uncorrelated jitter is a special case of Latin hypercube sampling

Stratify samples in each dimension separately

- for 5D: 5 separate 1D jittered point sets
- combine dimensions in random order

Shuffle order

<table>
<thead>
<tr>
<th>x</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>x2</td>
<td>x3</td>
<td>x4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y4</td>
<td>y2</td>
<td>y1</td>
<td>y3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>u</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>u3</td>
<td>u4</td>
<td>u2</td>
<td>u1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>v</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>v2</td>
<td>v1</td>
<td>v3</td>
<td>v4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>t</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>t2</td>
<td>t1</td>
<td>t4</td>
<td>t3</td>
<td></td>
</tr>
</tbody>
</table>
N-Rook: 2D version of Latin hypercube

Stratify samples in each dimension separately

- for **2D**: 2 separate 1D jittered point sets
- combine dimensions in random order

<table>
<thead>
<tr>
<th>x</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>x2</td>
<td>x3</td>
<td>x4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y4</td>
<td>y2</td>
<td>y1</td>
<td>y3</td>
<td></td>
</tr>
</tbody>
</table>
Latin-Hypercube (N-Rooks) Sampling

[Shirley 91]
// initialize the diagonal
for (uint d = 0; d < numDimensions; d++)
    for (uint i = 0; i < numS; i++)
        samples(d,i) = (i + randf())/numS;

// shuffle each dimension independently
for (uint d = 0; d < numDimensions; d++)
    shuffle(samples(d,:));
Latin-Hypercube (N-Rooks) Sampling

// initialize the diagonal
for (uint d = 0; d < numDimensions; d++)
    for (uint i = 0; i < numS; i++)
        samples(d,i) = (i + randf())/numS;

// shuffle each dimension independently
for (uint d = 0; d < numDimensions; d++)
    shuffle(samples(d,:));
Latin-Hypercube (N-Rooks) Sampling

```csharp
// initialize the diagonal
for (uint d = 0; d < numDimensions; d++)
    for (uint i = 0; i < numS; i++)
        samples(d,i) = (i + randf())/numS;

// shuffle each dimension independently
for (uint d = 0; d < numDimensions; d++)
    shuffle(samples(d,:));
```

Shuffle columns
// initialize the diagonal
for (uint d = 0; d < numDimensions; d++)
    for (uint i = 0; i < numS; i++)
        samples(d,i) = (i + randf())/numS;

// shuffle each dimension independently
for (uint d = 0; d < numDimensions; d++)
    shuffle(samples(d,:));
Latin-Hypercube (N-Rooks) Sampling:
good 1D projections, gaps in 2D
Latin-Hypercube (N-Rooks) Sampling:
good 1D projections, gaps in 2D
Latin-Hypercube (N-Rooks) Sampling: good 1D projections, gaps in 2D

Unevenly distributed in n-dimensions

Evenly distributed in each individual dimension
Power spectrum of N-Rooks sampling
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]

Shuffle x-coords
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]

Shuffle y-coords
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]

Evenly distributed in 2D!

Evenly distributed in each individual dimension.
Power spectrum of multi-jittered sampling

![Samples](image1.png)  ![Expected power spectrum](image2.png)  ![Radial mean](image3.png)

**Figure 5.6:** Illustration of random and some stochastic grid-based sampling patterns with the corresponding Fourier expected power spectra and the corresponding radial mean of their expected power spectra.

Sequence is called the Hammersley sequence, which can create a lower discrepancy point set for arbitrary dimensions, but due to the first dimension being a regular sampling, knowledge of the number of total samples is necessary. Figure 5.7 illustrates the Hammersley point set with 16 and 64 points in 2D. The corresponding sampling power spectra for Halton and Hammersley samples (first two components) are summarised in Figures 5.8.

### 5.3 Blue noise

Any sampling pattern with Blue noise characteristics is supposed to be well distributed within the spatial domain without containing any regular structures. The term Blue noise was coined by Ulichney [47], who investigated a radially averaged power spectra of various sampling patterns. He advocated three important features for an ideal radial power spectrum: First, its peak should be at
Multi-jittered vs N-Rooks vs jittered

Any sampling pattern with Blue noise characteristics is supposed to be well distributed within the first two components, as summarized in Figures 5.8.

Blue noise advocated three important features for an ideal radial power spectrum: First, its peak should be at the origin in the spatial domain without containing any regular structures. The term Blue noise was coined by Ulichney, who investigated a radially averaged power spectra of various sampling patterns. He found that for arbitrary dimensions, but due to the first dimension being a regular sampling, knowledge of the sequence is called the Hammersley sequence, which can create an even lower discrepancy point set.

Figure 5.6: Illustration of random and some stochastic grid-based sampling patterns with the corresponding Fourier expected power spectra and the corresponding radial mean of their expected power spectra. Figure 5.7 illustrates the Hammersley point set with 16 and 64 points in 2D. The corresponding sampling power spectra for Halton and Hammersley samples for arbitrary dimensions, but due to the first dimension being a regular sampling, knowledge of the sequence is called the Hammersley sequence, which can create an even lower discrepancy point set.
Progressive multi-jittered sampling

- don’t need to know the number of samples in advance!

- idea: keep track of which strata is occupied by previous samples using binary trees (O(sqrt(N)))
Orthogonal array sampling [Jarosz 2019]

- stratify in all 2D projections
- not progressive yet
Poisson-disk/blue noise sampling

• human eyes’ sampling pattern!

https://www.csie.ntu.edu.tw/~cyy/courses/rendering/16fall/lectures/handouts/chap05_color_radiometry.pdf
Dart throwing algorithm [Cook 1986]
Power spectrum of Poisson disk

5.3.3 Tiling-based methods

There are some tile-based approaches that can be used to generate blue noise samples. Tile-based methods overcome the computational complexity of dart-throwing and/or relaxation-based approaches in generating blue noise sampling patterns. In the computer graphics community, two tile-based approaches are well known: First, a set of precomputed tiles [10, 25], with each tile composed of multiple samples, and later use these tiles, in a sophisticated way, to pave the sampling domain. Second, a tile with one sample per tile [34, 33, 49] and uses some relaxation-based schemes, with look-up tables, to improve the overall quality of samples.

Although many blue noise sample generation algorithms exist, none of them are easily extendable to higher dimensions (>3).

5.4 Interpreting and exploiting knowledge of the sampling spectra

Recently [39], it has been shown that the low frequency region of the radial power spectrum (of a given sampling pattern) plays a crucial role in deciding the overall variance convergence rates of sampling patterns used for Monte Carlo integration. Since blue noise sampling patterns contain almost no radial energy in the low frequency region, they are of great interest for future research to obtain fast results in rendering problems. Surprisingly, Poisson Disk samples have shown the convergence rate of $O(N^{1/2})$, which is the same as given by purely random samples. This can be explained by looking at the low frequency region in the radial power spectrum of Poisson Disk samples (Fig. 5.9) which is not zero. The importance of the shape of the radial mean power spectrum in the low frequency region demands methods and algorithms that could eventually allow sample generation directly from a target Fourier spectrum.

5.4.1 Radially-averaged periodograms

Figures 5.6, 5.8 and 5.9 depict radially averaged periodograms of the various sampling strategies described in this chapter. These spectra reveal two important characteristics of estimators built using the corresponding sampling strategies.
Power spectrum of CCVT sampling

[Balzer et al. 2009]

5.4 Interpreting and exploiting knowledge of the sampling spectra

Recently [39], it has been shown that the low frequency region of the radial power spectrum (of a given sampling pattern) plays a crucial role in deciding the overall variance convergence rates of sampling patterns used for Monte Carlo integration. Since blue noise sampling patterns contains almost no radial energy in the low frequency region, they are of great interest for future research to obtain fast results in rendering problems. Surprisingly, Poisson Disk samples have shown the convergence rate of $O(N^{-1})$ which is the same as given by purely random samples. This can be explained by looking at the low frequency region in the radial power spectrum of Poisson Disk samples (Fig. 5.9) which is not zero. The importance of the shape of the radial mean power spectrum in the low frequency region demands methods and algorithms that could eventually allow sample generation directly from a target Fourier spectrum.

5.4.1 Radially-averaged periodograms

Figures 5.6, 5.8 and 5.9 depict radially averaged periodograms of the various sampling strategies described in this chapter. These spectra reveal two important characteristics of estimators built using the corresponding sampling strategies.
Low discrepancy sequences

- deterministic & progressive Latin hypercube samples based on the minimization of discrepancy
- entire field of study called “Quasi-Monte-Carlo”

$$D_n = \max_{\text{all rectangles}} \left| \frac{\text{no. of points in the rectangle}}{n} - \text{area(rectangle)} \right|$$
Koksma-Hlawka inequality

- discrepancy is the upper bound of the absolute estimation error!

\[
\left| \frac{1}{n} \sum_{i=0}^{n} f(x_i) - \int f(x) \, dx \right| \leq D_n^*(x_1, x_2, \ldots, x_n)
\]

star discrepancy: only consider rectangles with one vertex at the origin

\[
D_n = \max_{\text{all rectangles}} \left| \frac{\text{no. of points in the rectangle}}{n} - \text{area(rectangle)} \right|
\]
The Van Der Corput sequence

Radical Inverse $\Phi_b$ in base 2

Subsequent points “fall into biggest holes”

<table>
<thead>
<tr>
<th>$k$</th>
<th>Base 2</th>
<th>$\Phi_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>.1 = 1/2</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>.01 = 1/4</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>.11 = 3/4</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>.001 = 1/8</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>.101 = 5/8</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>.011 = 3/8</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>.111 = 7/8</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Halton & Hammersley sequences

**Halton**: Radical inverse with different base for each dimension:

- The bases should all be relatively prime.
- Progressive generation of samples

**Hammersley**: Same as Halton, but first dimension is $k/N$:

- Not progressive, need to know sample count, $N$, in advance
Hammersley sequence

1 sample in each “elementary interval”
(digital nets property)
Hammersley sequence

1 sample in each “elementary interval”
(digital nets property)
Hammersley sequence

1 sample in each “elementary interval”
(digital nets property)
Hammersley sequence

1 sample in each "elementary interval"
(digital nets property)
Hammersley sequence

1 sample in each "elementary interval"
(digital nets property)
Hammersley sequence

1 sample in each “elementary interval”
(digital nets property)
Scrambling low discrepancy sequences
[Owen 1995]

- randomly swapping two adjacent elementary intervals
Rank-1 lattice

Rank-1 Lattices for Efficient Path Integral Estimation
Hongli Liu$^1$, Honglei Han$^1$, and Min Jiang$^2$

1. State Key Laboratory of Media Convergence and Communication, and School of Animation and Digital Arts, Communication University of China, China
2. Framestore, United Kingdom

Image Synthesis by Rank-1 Lattices
S. Dammertz and A. Keller
Institute of Media Informatics, Ulm University, Germany
Spherical Fibonacci lattice

Spherical Fibonacci Point Sets for Illumination Integrals

R. Marques¹, C. Bouville², M. Ribardière², L. P. Santos³ and K. Bouatouch⁴

¹INRIA Rennes, France
²IRISA Rennes, France
³Universidade do Minho, Braga, Portugal

https://math.stackexchange.com/questions/3291489/can-the-fibonacci-lattice-be-extended-to-dimensions-higher-than-3
Sobol’ sampling: low discrepancy sampling with matrix multiplication

- specifically designed to satisfy the digital nets property
- usually the best performing one among all low discrepancy sequences

\[ x_a = \begin{bmatrix} b^{-1} & b^{-2} & \ldots & b^{-n} \end{bmatrix} \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ c_{n,1} & \cdots & \cdots & c_{n,n} \end{bmatrix} \begin{bmatrix} d_1(a) \\ d_2(a) \\ \vdots \\ d_n(a) \end{bmatrix} \]

https://www.pbr-book.org/3ed-2018/Sampling_and_Reconstruction/(0,2)-Sequence_Sampler
Scrambling Sobol’ sequences needs to be done in care!

Cascaded Sobol’ Sampling

LOÏS PAULIN, Université de Lyon, CNRS, LIRIS, France
DAVID COEURJOLLY, Université de Lyon, CNRS, LIRIS, France
JEAN-CLAUDE IEHL, Université de Lyon, CNRS, LIRIS, France
NICOLAS BONNEEL, Université de Lyon, CNRS, LIRIS, France
ALEXANDER KELLER, NVIDIA, Germany
VICTOR OSTROMOUKHOV, Université de Lyon, CNRS, LIRIS, France
Power spectrum of Sobol’ sampling

Sequences with Low-Discrepancy Blue-Noise 2-D Projections

Helène Perrier¹, David Courjolly¹, Feng Xie², Matt Pharr³, Pat Hanrahan², Victor Ostrovskiǐkh²

¹Université de Lyon, CNRS, LIRIS, France ²Stanford, USA ³Google, USA
Blue-noise Low-Discrepancy Sequences

Owen’s scrambling

[Owe95]

Owen, $K = 4$

Perrier et al.

Sequences with Low-Discrepancy Blue-Noise 2-D Projections

Hélène Perrier$^1$  David Courjolly$^1$  Feng Xie$^2$  Matt Pharr$^3$  Pat Hanrahan$^2$  Victor Ostrovoukob$^3$

$^1$Université de Lyon, CNRS, LIRIS, France  $^2$Stanford, USA  $^3$Google, USA
Theoretical convergence rate (in 2D)

- All sampling sequences work best for low frequency/smooth signals

<table>
<thead>
<tr>
<th>Samplers</th>
<th>Worst Case</th>
<th>Best Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>$O(N^{-1})$</td>
<td>$O(N^{-1})$</td>
</tr>
<tr>
<td>Jitter</td>
<td>$O(N^{-1.5})$</td>
<td>$O(N^{-2})$</td>
</tr>
<tr>
<td>Poisson Disk</td>
<td>$O(N^{-1})$</td>
<td>$O(N^{-1})$</td>
</tr>
<tr>
<td>CCVT</td>
<td>$O(N^{-1.5})$</td>
<td>$O(N^{-3})$</td>
</tr>
</tbody>
</table>

Low-discrepancy sequences (worst case, assuming no discontinuities):

$$O\left(\frac{(\log N)^2}{N}\right)$$

For large N, low discrepancy sequences win
Curse of dimensionality

- in high-dimensional space with high frequency between dimensions, all methods fail

best possible worst case convergence rate (with C1 continuity)

\[ O(n^{-\frac{2}{d}-1}) \]
Connection to optimal transport / Wasserstein distance

- Rubinstein-Kantorovich theorem

\[ \left| \int f(x) \, dx - \frac{1}{n} \sum_{i=1}^{n} f(x^i) \right| \leq \text{Lip}(f) \cdot W_1(X, 1_{\Omega}). \]

instead of using discrepancies, measure the earth mover distance
In practice

- know the number of samples in advance: Sobol’ + extensions (e.g. blue noise) probably works the best

- don’t know the number of samples in advance: progressive multi-jitter probably works the best

- warning: NVIDIA holds the patent of using Sobol’ sampling for rendering
Related topic: blue-noise dithered sampling

- focus on the reconstruction properties of sampling patterns, instead of integration
Next: path-space & Eric Veach

Figure 10.1: A transport path from a light source to the camera lens, created by concatenating two separately generated pieces.

(a) $s = 0, t = 3$  
(b) $s = 1, t = 2$  
(c) $s = 2, t = 1$  
(d) $s = 3, t = 0$

Robust Monte Carlo Methods for Light Transport Simulation

A Dissertation
Submitted to the Department of Computer Science
And the Committee on Graduate Studies
Of Stanford University
In Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy

by
Eric Veach
December 1997