Stratification 1

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

**quiz:** when will we have perfect reconstruction?
Frequency analysis of Monte Carlo integration

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

**Diagram:***

- Fourier integrand: \( \hat{f} \)
- Fourier sampling pattern: \( \hat{S} \)
- Convolution: \( \hat{S} \otimes \hat{f} \)

**Regular Sampling**

- Fourier integrand
- Fourier sampling pattern
- Convolution

**Monte Carlo Sampling**

- Fourier integrand
- Fourier sampling pattern
- Convolution

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

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

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

Mean square error = bias^2 + variance

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

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

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

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

(slightly simplified)

2013
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 many 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)

2013
Bias-variance analysis in Fourier domain

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

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

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

(slightly simplified)

the expected power spectrum of the sampling pattern \( E[\hat{S}^2] \) is the key!!

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

2013
Variance analysis = multiplication of power spectrums

- natural signals/integrands usually have energy concentrated at low frequencies

- quiz: what $E[\hat{S}^2]$ will lead to low variance?

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

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

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

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

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

**quiz:** pros and cons?
Independent random sampling

```c
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

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

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

Ulichney \cite{Ulichney1988}, who investigated a radially averaged power spectra of various sampling patterns. He

Any sampling pattern with Blue noise characteristics is supposed to be well distributed within the

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

for arbitrary dimensions, but due to the first dimension being a regular sampling, knowledge of the

For independent random random sampling

Many sample set realizations

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

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

\[
\frac{1}{N} \sum_{k=1}^{N} \delta(|\vec{x} - \vec{x}_k|) \quad E \left[ \frac{1}{N} \sum_{k=1}^{N} e^{-2\pi i (\vec{\omega} \cdot \vec{x}_k)} \right]^2
\]
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;
    }

**quiz:** pros and cons?
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;
  }

quiz: pros and cons?
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

Samples | Expected power spectrum | Radial mean
---|---|---
Random | [Image of random samples] | [Image of expected power spectrum] | [Image of radial mean]
Jitter | [Image of jittered samples] | [Image of expected power spectrum] | [Image of radial mean]
Multi-jitter | [Image of multi-jittered samples] | [Image of expected power spectrum] | [Image of radial mean]

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 an even 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

---

**References**

Random sampling vs jittered sampling

Chapter 5. Popular sampling patterns

<table>
<thead>
<tr>
<th>Samples</th>
<th>Power spectrum</th>
<th>Radial mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jitter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-jitter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-rooks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 spectra of various sampling patterns. He advocated three important features for an ideal radial power spectrum; First, its peak should be at...
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>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td></td>
<td></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
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
Latin-Hypercube (N-Rooks) Sampling

[Shirley 91]
Latin-Hypercube (N-Rooks) Sampling

```c
// 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,:));
```
// 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

// 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
Power spectrum of N-Rooks sampling
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]

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

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

Shuffle $\gamma$-coords
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]

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

Shuffle y-coords
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Multi-jittered sampling [Chiu 1994]
Evenly distributed in each individual dimension

Evenly distributed in 2D!

Multi-jittered sampling [Chiu 1994]
Power spectrum of multi-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.

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

**quiz:** what is the difference between jittered & multi-jittered?
Progressive multi-jittered sampling
probably the best sampling pattern we discussed today!

- don’t need to know the number of samples in advance!
- idea: keep track of which strata is occupied by previous samples using trees (O(sqrt(N)))
Progressive multi-jittered sampling

first sample: randomly place in the unit square
Progressive multi-jittered sampling

first sample: randomly place in the unit square
divide the unit square into 4 quadrants
Progressive multi-jittered sampling

first sample: randomly place in the unit square
divide the unit square into 4 quadrants
place the second sample at the diagonally opposite quadrant
Progressive multi-jittered sampling

first sample: randomly place in the unit square

divide the unit square into 4 quadrants

place the second sample at the
diagonally opposite quadrant

divide the unit square into 16 regions
Progressive multi-jittered sampling

first sample: randomly place in the unit square

divide the unit square into 4 quadrants

place the second sample at the diagonally opposite quadrant

divide the unit square into 16 regions

choose an empty quadrant
Progressive multi-jittered sampling

first sample: randomly place in the unit square

divide the unit square into 4 quadrants

place the second sample at the diagonally opposite quadrant

divide the unit square into 16 regions

choose an empty quadrant, place a sample that follows the N-rook rule
Progressive multi-jittered sampling

first sample: randomly place in the unit square
divide the unit square into 4 quadrants
place the second sample at the diagonally opposite quadrant
divide the unit square into 16 regions
choose an empty quadrant, place a sample that follows the N-rook rule
place a sample at the diagonally opposite quadrant, following the N-rook rule
Progressive multi-jittered sampling

first sample: randomly place in the unit square
divide the unit square into 4 quadrants
place the second sample at the diagonally opposite quadrant
divide the unit square into 16 regions
choose an empty quadrant, place a sample that follows the N-rook rule
place a sample at the diagonally opposite quadrant, following the N-rook rule
repeat
Progressive multi-jittered sampling

First sample: randomly place in the unit square

Divide the unit square into 4 quadrants

Place the second sample at the diagonally opposite quadrant

Divide the unit square into 16 regions

Choose an empty quadrant, place a sample that follows the N-rook rule

Place a sample at the diagonally opposite quadrant, following the N-rook rule

Repeat
Progressive multi-jittered sampling

first sample: randomly place in the unit square

divide the unit square into 4 quadrants

place the second sample at the diagonally opposite quadrant

divide the unit square into 16 regions

choose an empty quadrant, place a sample that follows the N-rook rule

place a sample at the diagonally opposite quadrant, following the N-rook rule

repeat
Orthogonal array sampling

- stratify in all 2D projections
- need to know no. of samples in advance currently
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

Samples | Expected power spectrum | Radial mean

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 approach uses a set of precomputed tiles, with each tile composed of multiple samples, and later use these tiles, in a sophisticated way, to pave the sampling domain. Second approach employed tiles with one sample per tile 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, 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 $O(N^{-1})$, 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.
Lloyd relaxation for Poisson disc sampling

developed at ~1957, published at 1982

Least Squares Quantization in PCM

video from
Power spectrum of CCVT sampling
[Balzer et al. 2009]

Samples          Expected power spectrum          Radial mean

5.4 Interpreting and exploiting knowledge of the sampling spectra

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: the first approach uses a set of precomputed tiles with each tile composed of multiple samples, and later uses these tiles in a sophisticated way to pave the sampling domain. The second approach employs tiles with one sample per tile 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, 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 $O(N^{-1})$ 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.
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>

quiz: what is the downside of CCVT compared to jittered sampling?
Curse of dimensionality

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

best possible worst case convergence rate (with C1 continuity)

$$O(n^{-\frac{2}{d}-1})$$

Stochastic Quadrature Formulas
By Seymour Haber
Big picture: numerical integration is about placing samples to measure integrals.

don’t get stuck by things like unbiasedness!
Next: low-discrepancy sampling

https://perso.liris.cnrs.fr/david.coeurjolly/publication/cascaded2021/