Advance Sampling: Fourier Analysis of Variance in Monte Carlo Integration

Gurprit Singh
Philipp Slusalek    Karol Myszkowski
Recap: Advanced Sampling
Monte Carlo Integration
Monte Carlo Integration

focal plane / virtual image plane

Aperture

U

V

X

Y
Monte Carlo Integration
Monte Carlo Integration

\[ \int_x \int_y \int_u \int_v f(x, y, u, v) \, dv \, du \, dy \, dx \]
Monte Carlo Integration

\[ \int_{x}^{\infty} \int_{y}^{\infty} \int_{u}^{\infty} \int_{v}^{\infty} f(x, y, u, v) \, dv \, du \, dy \, dx \]
Monte Carlo Integration
Monte Carlo Integration
Monte Carlo Integration

\[ f(\vec{x}) \]
Monte Carlo Integration

\[ I = \int_0^1 f(\vec{x}) d\vec{x} \]
Monte Carlo Integration

\[ f(\bar{x}) \]

\[ I = \int_{0}^{1} f(\bar{x}) d\bar{x} \]
Monte Carlo Integration

\[ I = \int_{0}^{1} f(\bar{x}) d\bar{x} \]
Monte Carlo Integration

\[ f(\vec{x}) \]

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} \frac{f(\vec{x}_k)}{p(\vec{x}_k)} \]
Monte Carlo Integration

\[ f(\vec{x}) \]

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} \frac{f(\vec{x}_k)}{p(\vec{x}_k)} \]
Variance

$$f(x, y, u, v) \, dv \, du \, dy \, dx$$
Random
Random

Jitter
Random vs Jitter

Random distribution compared to Jitter in a grid and pixel distribution.
Random

Jitter

Realistic Image Synthesis SS2020
Random

Jitter

Poisson Disk

Realistic Image Synthesis SS2020
Variance Convergence Rate of Samplers
Variance Convergence Rate of Samplers
Variance Convergence Rate of Samplers

\[ O(N^{-1.25}) \]

Number of Samples

Variance

4D Jittered
Variance Convergence Rate of Samplers

Pilleboue et al. [2015]

Number of Samples

Variance

\[ O(N^{-1.25}) \]

4D Jittered

Poisson Disk
Variance Convergence Rate of Samplers

Pilleboue et al. [2015]

\[ O(N^{-1}) \]

\[ O(N^{-1.25}) \]
Variance Convergence Rate of Samplers

Fredo Durand [2011]
Subr & Kautz [2013]
Pilleboue et al. [2015]

\[ O(N^{-1.25}) \]

\[ O(N^{-1}) \]
Monte Carlo Estimator

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} f(\vec{x}_k) \]

Fredo Durand [2011]
Monte Carlo Estimator

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} f(\vec{x}_k) = \int_0^1 \frac{1}{N} \sum_{k=1}^{N} \delta(\vec{x} - \vec{x}_k) f(\vec{x}) \, d\vec{x} \]

Fredo Durand [2011]
Monte Carlo Estimator

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} f(\vec{x}_k) = \int_{0}^{1} \frac{1}{N} \sum_{k=1}^{N} \delta(\vec{x} - \vec{x}_k) f(\vec{x}) \ d\vec{x} \]

Fredo Durand [2011]
Monte Carlo Estimator

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} f(\vec{x}_k) = \int_0^1 \frac{1}{N} \sum_{k=1}^{N} \delta(\vec{x} - \vec{x}_k) f(\vec{x}) \, d\vec{x} = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

Fredo Durand [2011]
Monte Carlo Estimator

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} f(\tilde{x}_k) = \int_0^1 \frac{1}{N} \sum_{k=1}^{N} \delta(\tilde{x} - \tilde{x}_k) f(\tilde{x}) \, d\tilde{x} = \int_0^1 S_N(\tilde{x}) f(\tilde{x}) \, d\tilde{x} \]

Fredo Durand [2011]
Monte Carlo Estimator

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} f(\vec{x}_k) = \int_{0}^{1} \frac{1}{N} \sum_{k=1}^{N} \delta(\vec{x} - \vec{x}_k) f(\vec{x}) d\vec{x} = \int_{0}^{1} S_N(\vec{x}) f(\vec{x}) d\vec{x} \]

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

Fredo Durand [2011]
Monte Carlo Estimator

\[ I_N = \frac{1}{N} \sum_{k=1}^{N} f(\tilde{x}_k) = \int_0^1 \frac{1}{N} \sum_{k=1}^{N} \delta(\tilde{x} - \tilde{x}_k) f(\tilde{x}) \, d\tilde{x} = \int_0^1 S_N(\tilde{x}) f(\tilde{x}) \, d\tilde{x} \]

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

Fredo Durand [2011]
Samples Power Spectrum

\[ I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

\[ S_N(\vec{x}) = \frac{1}{N} \sum_{k=1}^{N} \delta(\vec{x} - \vec{x}_k) \]
Samples Power Spectrum

\[ I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

Poisson Disk

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

\[ \mathcal{P}_{S_N}(\nu) = \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi \nu \cdot \vec{x}_k} \right|^2 \]
Samples Power Spectrum

\[ I_N = \int_{0}^{1} S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

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

\[ \mathcal{P}_{S_N}(\vec{v}) = \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi\vec{v} \cdot \vec{x}_k} \right|^2 \]
Samples Power Spectrum

\[ I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

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

\[ P_{S_N}(\vec{v}) = \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi\vec{v} \cdot \vec{x}_k} \right|^2 \]
Samples Power Spectrum

\[ I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

Samples

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

Spectrum

\[ P_{S_N}(\vec{v}) = \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi \vec{v} \cdot \vec{x}_k} \right|^2 \]
Samples Power Spectrum

\[ I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

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

\[ P_{S_N}(\nu) = \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi \nu \vec{x}_k} \right|^2 \]

\( \nu = 0 \) DC frequency
Expected Sampling Power Spectra

\[ I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

\[ S_N(\vec{x}) = \frac{1}{N} \sum_{k=1}^{N} \delta(\vec{x} - \vec{x}_k) \]
Expected Sampling Power Spectra

\[
I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x}
\]

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

\[
\mathcal{P}_{S_N}(\nu) = \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi\nu \cdot \vec{x}_k} \right|^2
\]
Expected Sampling Power Spectra

\[ I_N = \int_0^1 S_N(\bar{x}) f(\bar{x}) \, d\bar{x} \]

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

\[ \langle \mathcal{P}_{S_N}(\nu) \rangle = \left\langle \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi\nu \cdot \bar{x}_k} \right|^2 \right\rangle \]
Expected Sampling Power Spectra

\[ I_N = \int_0^1 S_N(\vec{x}) f(\vec{x}) \, d\vec{x} \]

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

\[ \langle P_{S_N}(\nu) \rangle = \left\langle \left| \frac{1}{N} \sum_{k=1}^{N} e^{-i2\pi\nu \cdot \vec{x}_k} \right|^2 \right\rangle \]
Variance of Monte Carlo Estimator

\[ \mathcal{P}_{SN}(\nu) \]

Fredo Durand [2011]
Subr & Kautz [2013]
Variance of Monte Carlo Estimator

\[ \langle P_{SN}(\nu) \rangle \quad P_f(\nu) \]

\[ S_N(\vec{x}) \quad f(\vec{x}) \]

Poisson Disk

Fredo Durand [2011]
Subr & Kautz [2013]
Variance of Monte Carlo Estimator

\[
\text{Var}(I_N) = \int_{\Omega} \left\langle \mathcal{P}_{S_N}(\nu) \right\rangle \times \mathcal{P}_f(\nu) \, d\nu
\]

Fredo Durand [2011]
Subr & Kautz [2013]
Variance of Monte Carlo Estimator

$$\text{Var}(I_N) = \int_{\Omega} \langle P_{S_N}(\nu) \rangle \times P_f(\nu) d\nu$$

- Fredo Durand [2011]
- Subr & Kautz [2013]
- Pilleboue et al. [2015]
Variance of Monte Carlo Estimator in Polar Coordinates

\[
\text{Var}(I_N) = \int_0^\infty \rho^{d-1} \int_{S^{d-1}} \langle \mathcal{P}_{SN}(\rho n) \rangle \times \mathcal{P}_f(\rho n) \, d\rho d\mathbf{n}
\]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator in Polar Coordinates

\[
\text{Var}(I_N) = \int_0^\infty \int_{S^{d-1}} \rho^{d-1} dS^{d-1} \times \langle \mathcal{P}_{SN}(\rho \mathbf{n}) \rangle \times \mathcal{P}_f(\rho \mathbf{n}) d\mathbf{n} d\rho
\]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator in Polar Coordinates

\[ \text{Var}(I_N) = \int_0^\infty \int_{S^{d-1}} \rho^{d-1} \times \langle P_{S_N}(\rho \mathbf{n}) \rangle \times P_f(\rho \mathbf{n}) \, d\mathbf{n} \, d\rho \]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator in Polar Coordinates

\[
\text{Var}(I_N) = \int_0^\infty \rho^{d-1} \int_{S^{d-1}} \langle \mathcal{P}_{S_N}(\rho n) \rangle \mathcal{P}_f(\rho n) \, dn \, d\rho
\]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

\[
\text{Var}(I_N) = \int_0^\infty \rho^{d-1} \int_{S^{d-1}} \langle \mathcal{P}_{SN}(\rho \mathbf{n}) \rangle \times \mathcal{P}_f(\rho \mathbf{n}) \, d\mathbf{n} \, d\rho
\]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

\[ \text{Var}(I_N) = \int_0^\infty \rho^{d-1} \]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

\[
\text{Var}(I_N) = \int_0^\infty \rho^{d-1} \, d\rho
\]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

\[
\text{Var}(I_N) = \int_0^\infty \rho^{d-1} \text{d}\rho
\]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

\[ \text{Var}(I_N) = \int_0^\infty \rho^{d-1} \times \tilde{P}_{SN}(\rho) \times P_f(\rho) \, d\rho \]

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

\[ \text{Var}(I_N) = \int_0^\infty \rho^{d-1} \, d\rho \]

<table>
<thead>
<tr>
<th>Samplers</th>
<th>Worst Case</th>
<th>Best Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poisson Disk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCVT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

$$\text{Var}(I_N) = \int_0^\infty \rho^{d-1} \, d\rho$$

<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>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>

Pilleboue et al. [2015]
Variance of Monte Carlo Estimator for Isotropic Sampling Spectra

\[ \text{Var}(I_N) = \int_0^\infty \rho^{d-1} \]
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)

Shuffle rows
Latin Hypercube Sampler (N-rooks)

Shuffle rows
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)

Shuffle columns
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

N-rooks Spectrum
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

Spectrum
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

N-rooks Spectrum
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

N-rooks Spectrum

Jitter
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

N-rooks Spectrum

Jitter

Jitter Spectrum
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

N-rooks Spectrum

Multi-Jitter Spectrum

Chiu et al. [1993]
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

N-rooks Spectrum

Multi-jitter

Multi-Jitter Spectrum

Chiu et al. [1993]
Anisotropic Sampling Power Spectra

N-rooks / Latin Hypercube

N-rooks Spectrum

Multi-jitter

Multi-Jitter Spectrum

Chiu et al. [1993]
Sampling in Higher Dimensions
4D Sampling

Rob Cook [1986]
4D Sampling

Rob Cook [1986]

$$(x_1, y_1)$$
$$(x_2, y_2)$$
$$(x_3, y_3)$$
$$(x_4, y_4)$$

$$(u_1, v_1)$$
$$(u_2, v_2)$$
$$(u_3, v_3)$$
$$(u_4, v_4)$$
4D Sampling

Rob Cook [1986]

$(x_1, y_1)$
$(x_2, y_2)$
$(x_3, y_3)$
$(x_4, y_4)$

$(u_1, v_1)$
$(u_2, v_2)$
$(u_3, v_3)$
$(u_4, v_4)$

$(x_1, y_1, u_3, v_3)$
$(x_2, y_2, u_1, v_1)$
$(x_3, y_3, u_4, v_4)$
$(x_4, y_4, u_2, v_2)$

...
Uncorrelated Jitter

Rob Cook [1986]

4D Sampling

2D

\[
\begin{align*}
(x_1, y_1) &\rightarrow (u_1, v_1) \\
(x_2, y_2) &\rightarrow (u_2, v_2) \\
(x_3, y_3) &\rightarrow (u_3, v_3) \\
(x_4, y_4) &\rightarrow (u_4, v_4)
\end{align*}
\]

4D

\[
\begin{align*}
(x_1, y_1, u_3, v_3) \\
(x_2, y_2, u_1, v_1) \\
(x_3, y_3, u_4, v_4) \\
(x_4, y_4, u_2, v_2)
\end{align*}
\]
Rob Cook [1986]

Uncorrelated Poisson Disk

4D Sampling

\[
\begin{align*}
(x_1, y_1) &\rightarrow (u_1, v_1) \\
(x_2, y_2) &\rightarrow (u_2, v_2) \\
(x_3, y_3) &\rightarrow (u_3, v_3) \\
(x_4, y_4) &\rightarrow (u_4, v_4) \\
\vdots & \vdots \\
(x_1, y_1, u_3, v_3) & \rightarrow (x_2, y_2, u_1, v_1) \\
(x_2, y_2, u_1, v_1) & \rightarrow (x_3, y_3, u_4, v_4) \\
(x_3, y_3, u_4, v_4) & \rightarrow (x_4, y_4, u_2, v_2) \\
\vdots & \vdots 
\end{align*}
\]
4D Sampling Spectra along Projections

Poisson Disk Spectra

UV

XY

Poisson Disk Samples
4D Sampling Spectra along Projections

Poisson Disk Spectra

Poisson Disk Samples

UV

XY
4D Sampling Spectra along Projections

Poisson Disk Spectra

Poisson Disk Samples
4D Sampling Spectra along Projections

Poisson Disk Spectra

Poisson Disk Samples

UV

XY

XU
4D Sampling Spectra along Projections

Poisson Disk Spectra

UV

XY

XU

YV

Poisson Disk Samples
How can we perform Convergence Analysis for Anisotropic Sampling Spectra?
Variance Formulation for Anisotropic Sampling Spectra

\[ \text{Var}(I_N) = \int_{\Omega} \langle P_{SN}(\nu) \rangle \times P_f(\nu) \, d\nu \]

- N-rooks spectrum
- Integrand spectrum

\( S_N(\vec{x}) \)

\( f(\vec{x}) \)
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \int_0^\infty \rho^{d-1} \int_{S^{d-1}} \langle \mathcal{P}_{S_N}(\rho n) \rangle \times \mathcal{P}_f(\rho n) \, d\mathbf{n} \, d\rho
\]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \int_{S^{d-1}} \int_0^\infty \rho^{d-1} \left\langle \mathcal{P}_{S_N}(\rho \mathbf{n}) \right\rangle \times \mathcal{P}_f(\rho \mathbf{n}) \, d\rho \, d\mathbf{n}
\]
Variance Formulation for Anisotropic Sampling Spectra

$$\text{Var}(I_N) = \int_{S^{d-1}} \int_0^\infty \rho^{d-1} \langle P_{S_N}(\rho \mathbf{n}_k) \rangle \times P_f(\rho \mathbf{n}_k) d\rho d\mathbf{n}$$
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \int_{S^{d-1}} \int_0^\infty \rho^{d-1} \left\langle \mathcal{P}_{S_N}(\rho_k n_k) \right\rangle \times \mathcal{P}_f(\rho_k n_k) \, d\rho \, dn
\]
Variance Formulation for Anisotropic Sampling Spectra

\[ \text{Var}(I_N) = \int_{S^{d-1}} \int_0^\infty \rho^{d-1} \mathbb{E}_{\mathbf{n}_k} \mathcal{P}_f(\rho_k \mathbf{n}_k) \mathcal{P}_S(\rho_k \mathbf{n}_k) \, d\rho \, d\mathbf{n} \]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \lim_{m \to \infty} \sum_{k=1}^{m} \int_{0}^{\infty} \rho^{d-1} \rho_f(\rho_k \mathbf{n}_k) \times \rho_S(\rho_k \mathbf{n}_k) \, d\rho \ \Delta \mathbf{n}_k
\]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \lim_{m \to \infty} \sum_{k=1}^{m} \int_{0}^{\infty} \rho^{d-1} \left\langle P_{SN}(\rho_k n_k) \right\rangle \times P_f(\rho_k n_k) \, d\rho \, \Delta n_k
\]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \lim_{m \to \infty} \sum_{k=1}^{m} \int_{0}^{\infty} \rho^{d-1} \left\langle \mathcal{P}_S(\rho_k \mathbf{n}_k) \right\rangle \mathcal{P}_f(\rho_k \mathbf{n}_k) \, d\rho \, \Delta \mathbf{n}_k
\]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \lim_{m \to \infty} \sum_{k=1}^{m} \int_{0}^{\infty} \rho^{d-1} \left\langle \mathcal{P}_{SN}(\rho \mathbf{n}_k) \right\rangle \mathcal{P}_f(\rho \mathbf{n}_k) \, d\rho \Delta \mathbf{n}_k
\]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \lim_{m \to \infty} \sum_{k=1}^{m} \int_{0}^{\infty} \rho^{d-1} \langle \mathcal{P}_{SN}(\rho_k n_k) \rangle \mathcal{P}_f(\rho_k n_k) d\rho \Delta n_k
\]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \lim_{m \to \infty} \sum_{k=1}^{m} \int_0^{\infty} \rho^{d-1} \langle \mathcal{P}_{SN}(\rho_k \mathbf{n}_k) \rangle \mathcal{P}_f(\rho_k \mathbf{n}_k) \, d\rho \Delta \mathbf{n}_k
\]
Variance Formulation for Anisotropic Sampling Spectra

\[
\text{Var}(I_N) = \lim_{m \to \infty} \sum_{k=1}^{m} \int_0^{\infty} \rho^{d-1} \langle P_{SN}(\rho n_k) \rangle \ P_f(\rho n_k) \ d\rho \Delta n_k
\]
Convergence Analysis for Anisotropic Sampling Spectra

Power Spectrum

Radial Power Spectrum
Convergence Analysis for Anisotropic Sampling Spectra

Power Spectrum

Radial Power Spectrum

Along canonical axes

Jittered Spectrum Profile
Convergence Analysis for Anisotropic Sampling Spectra

Power Spectrum

Radial Power Spectrum

Along canonical axes

Jittered Spectrum Profile

Other directions

Random Spectrum Profile
Convergence Analysis for Anisotropic Sampling Spectra

Power Spectrum

Radial Power Spectrum

Along canonical axes

Jittered Spectrum Profile

Other directions

Random Spectrum Profile
Convergence Analysis for Anisotropic Sampling Spectra

Power Spectrum

Radial Power Spectrum

Along canonical axes

Frequency

Jittered Spectrum Profile

Other directions

Random Spectrum Profile

Frequency
Variance due to N-rooks Sampler

\[ \text{Var}(I_N) = \int_{\Omega} \langle \mathcal{P}_{S_N}(\nu) \rangle \times \mathcal{P}_f(\nu) \, d\nu \]
Variance due to N-rooks Sampler

\[ \text{Var}(I_N) = \int_{\Omega} \langle \mathcal{P}_{SN}(\nu) \rangle \times \mathcal{P}_f(\nu) \, d\nu \]

- N-rooks spectrum
- Integrand spectrum
Variance due to N-rooks Sampler

\[ f(\bar{x}) \]

\[ \text{Var}(I_N) = \int_{\Omega} \langle \mathcal{P}_{SN}(\nu) \rangle \times \mathcal{P}_f(\nu) \, d\nu = \int_{\Omega} d\nu \]

N-rooks spectrum

Integrand spectrum
Variance due to N-rooks Sampler

\[
\text{Var}(I_N) = \int_{\Omega} \nabla f(x) \cdot \nabla f(x) \, d\nu
\]

\[
\text{Var}(I_N) = \int_{\Omega} \nabla f(x) \cdot \nabla f(x) \, d\nu
\]
\[
\text{Var}(I_N) = \int_{\Omega} \langle \mathcal{P}_{SN}(\nu) \rangle \times \mathcal{P}_f(\nu) \, d\nu = \int_{\Omega} \text{N-rooks spectrum} \times \text{Integrand spectrum} \, d\nu
\]
Variance due to N-rooks Sampler

\[ \text{Var}(I_N) = \int_{\Omega} \left( \mathcal{P}_{SN}(\nu) \times \mathcal{P}_f(\nu) \right) \, d\nu \]

N-rooks spectrum \hspace{1cm} \text{Integrand spectrum}

\[ \text{Var}(I_N) = \int_{\Omega} \left( \mathcal{P}_{SN}(\nu) \times \mathcal{P}_f(\nu) \right) \, d\nu \]

N-rooks spectrum \hspace{1cm} \text{Integrand spectrum}
Variance Convergence of Latin Hypercube (N-rooks)

\[ \langle \mathcal{P}_{SN}(\nu) \rangle \quad \mathcal{P}_f(\nu) \]

Pixel A

Pixel B
Variance Convergence of Latin Hypercube (N-rooks)

\( \left\langle P_{SN}(\nu) \right\rangle \quad P_f(\nu) \)

Pixel A

Pixel B

\( O(N^{-1}) \)

\( O(N^{-2}) \)

Number of Samples

Variance

\( N \)
Non-Axis Aligned Integrand Spectra

$P_f(\nu)$

Integrand Spectrum
Non-Axis Aligned Integrand Spectra

Multi-jittered Samples

\[ P_f(\nu) \]

Integrand Spectrum
Non-Axis Aligned Integrand Spectra

Multi-jittered Samples

\[ \langle P_{SN}(\nu) \rangle \]

Sampling Spectrum

\[ P_f(\nu) \]

Integrand Spectrum
Shearing Multi-Jittered Samples

Sheared Samples

\[ \langle \mathcal{P}_{SN}(\nu) \rangle \]

\[ \mathcal{P}_f(\nu) \]

Integrand Spectrum
How can we determine the sample shearing parameters?

\[ \langle \mathcal{P}_{SN}(\nu) \rangle \quad \mathcal{P}_f(\nu) \]
Spatio-temporal Sampling for Reconstructing Distribution Effects

Philipp Slusallek    Karol Myszkowski    Gurprit Singh
Multi-dimensional adaptive sampling of distribution effects

Fourier Analysis of Light Transport

Temporal reconstruction of distribution effects
Multi-dimensional adaptive spatio-temporal sampling

Hachisuka et al. [2008]
Multi-dimensional adaptive sampling of distribution effects

Fourier Analysis of Light Transport
Light Transport
Understanding, manipulating and computing signals

- Discontinuities
  - where things change
- Gradients
  - Useful for interpolation
- **Frequency content (today's main course)**
  - Useful for sampling
  - Useful for inverse problems
    - Sometimes useful as basis functions
  - Statistics

And all these capture perceptual properties
Frequency contents matter in vision

Inverse lighting  Shape from texture  Shape from (de)focus
Illumination effects

Blurry reflections
Illumination effects

Shadow boundaries:
Frequency contents matter in graphics

- Sampling, antialiasing
  - Texture filtering
  - Light Field Sampling
- Fourier-like basis
  - Precomputed radiance transfer
  - Wavelet radiosity
  - Spherical harmonics
- Low frequency assumption
  - Irradiance caching
How does light interactions in a scene explain the frequency content?
How does light interactions in a scene explain the frequency content?

Theoretical framework:

Understanding the frequency content of the radiance function

Mathematical equations of the light transport → Fourier spectrum of the Illumination in the scene
Spatial and Angular frequency

Spatial frequency
(e.g., shadows, textures)

Angular frequency
(e.g., blurry highlights)
Disclaimer: no Fourier optics

Only geometrical optics
Light transport in a scene
Light transport in a scene
Flatland: Light transport in a scene
Flatland: Light transport in a scene
Flatland: Light transport in a scene
Flatland: Light transport
Flatland: Light transport
Flatland: Light transport

Ray Space

angle

space

22

Ray Space

angle

space
Flatland: Light transport
Flatland: Light transport
Flatland: Light transport

Ray Space

angle

diagram

Ray Space

angle

diagram
Flatland: Light transport

Shear in primal

Shear in Fourier, but along the other dimension
Transport --> Shear

Consistent with literature [see Plenoptic Sampling by Chai et al. 2000]

<table>
<thead>
<tr>
<th></th>
<th>(d1) Scene image</th>
<th>(d2) EPI</th>
<th>(d3) Fourier transform of EPI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.png" alt="Scene image" /></td>
<td><img src="image2.png" alt="EPI" /></td>
<td><img src="image3.png" alt="Fourier transform" /></td>
</tr>
</tbody>
</table>
Occlusions

Consider planar occluders

Multiplication by binary function
- mostly in space

Before Occlusion

Blocker function

After occlusion
Occlusions

Multiplication in Primal domain is Convolution in Fourier domain

Before Occlusion

Blocker function

After occlusion

Primal

Fourier

\( \Omega_x (\text{space}) \)

\( \Omega_{\Theta} (\text{angle}) \)

\( \Theta_v (\text{angle}) \)
## Main Transforms: Summary

<table>
<thead>
<tr>
<th>Transformations</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
<td>Shear</td>
</tr>
<tr>
<td>Occlusion</td>
<td>Convolution/Multiplication</td>
</tr>
<tr>
<td>BRDF</td>
<td>Multiplication/Convolution</td>
</tr>
<tr>
<td>Curvature</td>
<td>Shear</td>
</tr>
</tbody>
</table>

**Frequency analysis of light transport** [Durand et al. 2005]
Reconstructing Motion Blur
Motion blur

Objects move while the camera shutter is open

Image is "blurred" over time
Expensive for special effects

Necessary to remove "strobing" in animation
Garfield: A tale of two kitties
Rhythm & Hues Studios
The Incredibles
Pixar Animation Studios
Walt Disney Pictures
Motion blur: Simple approach

\[ t = 0.1 \]
\[ t = 0.3 \]
\[ t = 0.5 \]
\[ t = 0.7 \]
\[ t = 0.9 \]
Motion blur: Simple approach

t = 0:

3.2

3.4

3.6

3.8

4.0
Motion blur: Simple approach

\[ t \in [0, 1] \]
Simple approach

The simple approach is expensive

Can we do better?
Observation

Motion blur is expensive

Motion blur removes spatial complexity
Use axis-aligned pixel filters at each pixel

Requires many samples
Standard Method

Use axis-aligned pixel filters at each pixel

Requires many samples
Filter shearing based on frequency analysis of light transport

We will look at how to reuse nearby pixel samples to reconstruct using filters derived using the frequency analysis of light transport.
Basic Example

No velocity: static scene  \( t \in [0, 1) \)
Basic Example

Low velocity  \( t \in [0, 1) \)
Basic Example

Low velocity \( t \in [0, 1) \)
Basic Example

Low velocity $t \in [0, 1)$
Basic Example

High velocity \( t \in [0, 1) \)
Basic Example

High velocity $t \in [0, 1)$
Shear in space-time

Object moving with low velocity \( t \in [0, 1) \)
Large shear in space-time

Object moving with high velocity  $t \in [0, 1)$
Shear in space-time

Object moving away from the camera $t \in [0, 1)$
Camera shutter filter

Applying shutter blur across time $t \in [0, 1)$
Basic example: Fourier domain

Fourier spectrum, zero velocity $t \in [0, 1)$
Basic example: Fourier domain

Low velocity, small shear in both domains

\[ t \in [0, 1) \]

\[ F(\Omega_x, \Omega_t) \]

Slope = -speed
Basic example: Fourier domain

Low velocity, small shear in both domains

\[ F(\Omega_x, \Omega_t) \]

Slope = -speed

\[ t \in [0, 1) \]
Basic example: Fourier domain

Due to camera motion, slopes are varying

\[ t \in [0, 1) \]

\[ f(x,t) \]

\[ t \]

\[ x \]

\[ F(\Omega_x, \Omega_t) \]

\[ \Omega_x \]

\[ \Omega_t \]

Slope = - max_speed

Slope = - min_speed
Basic example: Fourier domain

When shutter blur is applied, only low frequencies matter

\[ t \in [0, 1) \]

\[ F(\Omega_x, \Omega_t) \]
Basic example: Fourier domain

When shutter blur is applied, only low frequencies matter.
Main Insights

Common case = double wedge spectra

Shutter indirectly removes spatial frequencies
Sampling and Filtering Goals

- Minimal sampling rate to prevent aliasing
- Derive shape of the reconstruction filters
Sampling in Fourier Domain

Sampling produces replicas in the Fourier domain
Sampling in Fourier Domain

Let's say the corresponding image has a Fourier spectrum as shown on the right side.
Sampling in Fourier Domain

Sampling produces replicas in the Fourier domain
Sampling in Fourier Domain

Sampling produces replicas in the Fourier domain

Sparse sampling produces denser replicas
Standard Reconstruction Filtering

Standard filer, dense sampling, slow
Standard Reconstruction Filtering

Standard filter, dense sampling, slow
Standard Reconstruction Filtering

Standard filter, sparse sampling, fast
Sheared Reconstruction Filter

Standard filter, sparse sampling, fast

No aliasing

\[ \Omega_t \quad \Omega_x \]
Sheared Reconstruction Filter

Compact shape in Fourier = wide space-time

\[ \Omega_t \]

\[ \Omega_x \]
Main Insights

Sheared filter allows for many fewer samples
Filters in action: Car example

Static

Motion blurred
Implementation: stage 1

Sparse sampling to compute velocity bounds

min speed

max speed
Implementation: stage 1

Calculate filter widths and sampling rates

min speed

max speed

filter width
Implementation: stage 1

Uniform velocities, wide filter, low samples

min speed

max speed

filter width
Implementation: stage 1

Static surface, small filter, low samples

- min speed
- max speed
Implementation: stage 1

Varying velocities, small filter, high samples

min speed

max speed

filter width
Implementation: stage 2

Then, compute sampling densities

Uniform velocities = low sample count
Implementation: stage 2

Then, compute sampling densities

Varying velocities = high sample count
Implementation: stage 3

Render sample locations in space-time

Apply sheared filters to nearby samples

Sheared filters overlaps samples across multiple pixels
Implementation: stage 3

Filters stretched along the direction of motion
Preserve frequencies orthogonal to the motion
Results

Sheared Filters
4spp

Stratified Sampling
4spp
Multi-dimensional adaptive sampling of distribution effects

Fourier Analysis of Light Transport

Temporal reconstruction of distribution effects
Temporal Light-Field Reconstruction for Rendering Distribution Effects

Lehtinen et al. [2011]

Slides courtesy: Jakko Lehtinen
Requires dense sampling of 5D function:

Pixel area (2D)
Lens aperture (2D)
Time (1D)
Motion blur and depth of field 1 sample per pixel
Our reconstruction
Pinhole camera model

- object
- pinhole
- sensor
- background
Thin lens camera model

- Object
- Lens
- Sensor
- Background
Depth of field
Depth of field
Lens $u$

Screen $x$
Light field [Levoy 1996]

Output: integration over lens
Monte Carlo sampling

Low sample density leads to noise
Monte Carlo sampling

Need many samples to capture the signal:

**computationally expensive**
Temporal light fields

Traditional light field is 4D [Levoy 1996]

x,y over sensor (2D)
u,v over lens (2D)

Add time dimension for moving geometry (5D)
Screen X

Lens u

100
The Integrand is Anisotropic

[Chai00, Durand05, Hachisuka08, Soler09, Egan09, ...]
Multi-dimensional Adaptive Sampling [Hachisuka 08]
Frequency Analysis and Sheared Reconstruction [Egan 09]
Our approach

Start with **sparse** input sampling
Our approach

Start with **sparse** input sampling

Perform **dense** reconstruction using sparse input samples

Standard Monte-Carlo integration using dense reconstruction
Our input has slope information.

For defocus, proportional to inverse depth $1/z$ [Chai00].

For motion, proportional to inverse velocity $1/v$ [Egan09].

Easy to output from any renderer.
What is the radiance at the red location?

Use slope to reproject radiance.
What is the radiance at the red location?

Use slope to **reproject** radiance.

Must account for **occlusion**.
Recap: our approach

Start with **sparse** input sampling

Perform **dense** reconstruction using sparse input samples

Use **slopes** to reproject

Account for **visibility**

Standard Monte-Carlo integration using dense reconstruction
Reprojection and filtering

Simplify visibility by reprojecting into \textit{screen space}.

Reproject to u, v, t of reconstruction location.

Pixel filter over \textit{visible} samples.
Visibility

Cluster samples into **apparent surfaces** to resolve z-order

_SameSurface_ algorithm

Determining **coverage**: Does the apparent surface cover my reconstruction location?
Visibility: SameSurface

Input:
sparse points with slopes
The trajectories of samples originating from a single apparent surface never intersect.
Visibility: SameSurface

Visibility events show up as **intersections**
Visibility: Coverage

Does foreground apparent surface cover reconstruction location?

Search foreground samples for spanning triangle.

foreground surface

background surface

reconstruction location
Recap: our approach

Start with **sparse** input sampling

Perform **dense** reconstruction using sparse input samples

Use **slopes** to reproject

Account for **visibility**

Standard Monte-Carlo integration using dense reconstruction
Observations

We only need sample radiance, depth, and velocity (i.e., slopes). Reconstruction is independent of the original renderer.

We can discard the scene.
Observations

We only need sample radiance, depth, and velocity (i.e., slopes). Reconstruction is independent of the original renderer.

We can discard the scene.

Need efficient sample search:

Fast motion and large defocus can lead to a single sample contributing to hundreds of pixels.

Build a hierarchy over input samples.
Extension to soft shadows

An **area light** is very much like a **lens**.

lens ~ light, sensor ~ virtual plane
Reconstruct **z** instead of radiance

Egan et al. [2010] reconstruct far field **binary visibility** only.

**7D** path-tracing style reconstruction avoiding **combinatorial explosion**

Reconstruct scene point (**5D**)  
Reconstruct shadow **z** shade (**2D**)
Results
Implementation

Multithreaded **CPU**
**GPU**, excluding hierarchy construction

Common sample buffer format accepts outputs from:

- **PBRT**
- **Pixie** (Open source RenderMan)
- Custom ray tracer
Input: 16 spp
1072 sec (PBRT)
Our result: 16 spp + reconstruction at 128spp
1072 sec (PBRT) + 10 sec (reconstruction)
Our result: 16 spp + reconstruction at 128spp
1072 sec (PBRT) + 10 sec (reconstruction)
Input: 16 spp
771 sec (PBRT)
Our result: 16 spp + reconstruction at 128spp
771 sec (PBRT) + 10 sec (reconstruction)
Our result: 16 spp + reconstruction at 128spp
771 sec (PBRT) + 10 sec (reconstruction)

Input: 16 spp
Our result at 128 spp using same input
Reference: 256 spp (16x time)
Comparison to reference

Reconstruction quality (higher is better)

PSNR (dB)

Input samples/pixel

16

Number of reconstruction locations

16 32 64 128 256

Reconstruction quality (higher is better)
Motion blur and depth of field
1 sample per pixel
Proposed reconstruction
Input: 1 spp

Proposed result:
1 spp -> 128 spp

Reference 256 spp (256x time)
Comparison to Egan et al. [2009]

Egan et al. [2009]  
8 samples / pixel

Proposed method  
4 samples / pixel

Reference  
256 samples / pixel
Comparison to Egan et al. [2009]

- Egan et al. [2009]: 8 samples / pixel
- Our method: 4 samples / pixel
- Reference: 256 samples / pixel
Soft shadows, 4 spp
7D soft shadows with motion and defocus, 4 spp
Acknowledgments

Thanks to everyone below for making the slides available online.

Fredo Durand and colleagues [Frequency analysis of light transport 2005]

Toshiya Hachisuka and colleagues [Multi-dimensional adaptive sampling and reconstruction for ray tracing 2008]

Kevin Egan and colleagues [Frequency Analysis and Sheared Reconstruction for Rendering Motion Blur 2009]

Jakko Lehtinen and colleagues [Temporal Light Field Reconstruction for Rendering Distribution Effects 2011]