Lecture 7:
Sampling and Reconstruction
So Far…

- Graphics Pipeline
- Shadow and Environment Mapping
- Precomputed Radiance Transfer
Today: Sampling and Reconstruction

- Motivation
- Brief History
- Frequency Analysis for Light Transport
- Reflection as Convolution
- Axis-Aligned Filtering for Soft Shadows
  - A follow-up paper will be presented in class
Motivation

- Distribution effects (depth of field, motion blur, global illumination, soft shadows) are slow. (Many dimensions)
- Ray Tracing physically accurate but slow, not real-time
- Can we adaptively sample and filter for fast, real-time?
Outline

- Motivation
- Brief History
- Frequency Analysis for Light Transport
- Reflection as Convolution
- Axis-Aligned Filtering for Soft Shadows
  - A follow-up paper will be presented in class
History

- Adaptive sampling [Mitchell et al. 87, 91,…]

- But not very widely used… artifacts, can miss features

- After seminal papers in 87-91, not much follow on
Directional Coherence Maps

- Allocate samples to edges (Guo 98) Most of variance at those edges in the image
Multi-Dimensional Adaptive Sampling

- Hachisuka, Jarosz, … Zwicker, Jensen [MDAS 2008]
- Sampling in 2D image plane or other dims inadequate
  - Need to consider full joint high-dimensional space
Resurgence (2008 - )

- Eurographics 2015 STAR report by Zwicker et al.
A-Priori Methods

- Egan et al. 2009: Frequency Analysis and Sheared Filtering for Motion Blur; first deep use frequency analysis.
A-Posteriori Methods

- Adaptive Wavelet Rendering (Overbeck et al. 2009)
- Handle general effects. Sample and denoise
- Many more sophisticated methods available now
- [https://www.youtube.com/watch?v=sTUhAfwzsGg](https://www.youtube.com/watch?v=sTUhAfwzsGg)
Near Real-Time

- Axis-Aligned Filtering [Mehta et al. 12,13,14]
  - NVIDIA OptiX plus image-space filtering
  - Could be your Project 2 framework

- Fast Sheared Filtering [Yan et al. 15, Wu et al. 17]

This lecture
- More theoretical
- Reviews basics of frequency analysis based reconstruction

Next lecture
- More practical
- State of the art industrial solutions
Outline

- Motivation
- Brief History
- *Frequency Analysis for Light Transport*
- Reflection as Convolution
- Axis-Aligned Filtering for Soft Shadows
  - A follow-up paper will be presented in class
Frequency Analysis and Sheared Reconstruction for Rendering Motion Blur

Kevin Egan  Columbia University
Yu-Ting Tseng  Columbia University
Nicolas Holzschuch  INRIA -- LJK
Frédo Durand  MIT CSAIL
Ravi Ramamoorthi  University of California, Berkeley
Observation

- Motion blur is expensive
- Motion blur *removes* spatial complexity
Basic Example – Fourier Domain

- Fourier spectrum, zero velocity

\[ f(x, t) \]

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

texture bandwidth

\[ \Omega_x \]

\[ \Omega_t \]
Basic Example – Fourier Domain

- Low velocity, small shear in both domains

\[ f(x, t) \]

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

slope = -speed

\[ \Omega_x \]

\[ \Omega_t \]
Basic Example – Fourier Domain

- Large shear

\[ f(x, t) \]  
\[ F(\Omega_x, \Omega_t) \]
Basic Example – Fourier Domain

- Non-linear motion, wedge shaped spectra

\[ f(x, t) \]

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

- Shutter applies blur across time
- Bandlimits in time - min speed
- Bandlimits in space - max speed
- Indirectly
Sampling in Fourier Domain

- Sampling produces replicas in Fourier domain
- Sparse sampling produces dense replicas

[Diagram showing sampling in both primal and Fourier domains]
Standard Reconstruction Filtering

- Standard filter, dense sampling (slow)

Fourier Domain

- replicas
- no aliasing

\[ \Omega_t \]

\[ \Omega_x \]
Standard Reconstruction Filter

- Standard filter, sparse sampling (fast)
Sheared Reconstruction Filter

- Our sheared filter, sparse sampling (fast)
Sheared Reconstruction Filter

- Compact shape in Fourier = wide in primal
Car Scene

Our Method, 4 samples per pixel

Stratified Sampling, 4 samples per pixel
Teapot Scene

Our Method
8 samples / pix

motion blurred reflection


Outline

- Motivation
- Brief History
- Frequency Analysis for Light Transport
- Reflection as Convolution
- Axis-Aligned Filtering for Soft Shadows
  - A follow-up paper will be presented in class
Reflection as Convolution

- From my PhD advisor’s thesis: A signal-processing framework for forward and inverse rendering [Ramamoorthi 2002]
- Rewrite reflection equation on curved surfaces as a convolution with frequency-space product form
- Theoretical underpinning for much work on relighting (next lecture), limits of inverse problems
- Low-dimensional lighting models for Lambertian...
### Assumptions

- Known geometry
- Convex curved surfaces: no shadows, interreflection
- Distant illumination
- Homogeneous isotropic materials

Later precomputed methods: relax many assumptions
Reflection

\[ B(\theta_o) = \int_{-\pi/2}^{\pi/2} L(\theta_i) \rho(\theta_i, \theta_o) \, d\theta_i \]

Reflecting Light Field  \hspace{1cm} \text{Lighting}  \hspace{1cm} \text{BRDF}
Reflection as Convolution (2D)

\[ B(\theta_o) = \int_{-\pi/2}^{\pi/2} L(\theta_i) \rho(\theta_i, \theta_o) \, d\theta_i \]

Reflected Light Field

Lighting

BRDF
Reflection as Convolution (2D)

\[ B(\theta_o) = \int_{-\pi/2}^{\pi/2} L(\theta_i) \rho(\theta_i, \theta_o) \, d\theta_i \]

Reflected Light Field

\[ B(\alpha, \theta_o) = \int_{-\pi/2}^{\pi/2} L(\theta_i - \alpha) \rho(\theta_i, \theta_o) \, d\theta_i \]

Lighting  BRDF
Reflection as Convolution (2D)

\[ B(\alpha, \theta_o) = \int_{-\pi/2}^{\pi/2} L(\theta_i - \alpha) \rho(\theta_i, \theta_o) \, d\theta_i \]
Convolution

Signal $f(x)$ \hspace{1cm} Filter $g(x)$ = Output $h(u)$
Convolution

\[ h(u_1) = \int g(x - u_1) f(x) \, dx \]
Convolution

\[
h(u_2) = \int g(x - u_2) f(x) \, dx
\]
Convolution

\[ h(u_3) = \int g(x - u_3) f(x) \, dx \]
Convolution

\[ h(u) = \int g(x - u) f(x) \, dx \]

\[ h = f \otimes g = g \otimes f \]

Fourier analysis

\[ h_\omega = f_\omega g_\omega \]
Reflection as Convolution (2D)

\[ B(\alpha, \theta_o) = \int_{-\pi/2}^{\pi/2} L(\theta_i - \alpha) \rho(\theta_i, \theta_o) \, d\theta_i \]

\[ B = L \otimes \rho \]

\[ B_{l,p} = 2\pi L_l \rho_{l,p} \]

Spatial: integral
Frequency: product

Spherical Harmonics

\[ Y_{lm}(\theta, \phi) \]
Spherical Harmonic Analysis

2D:
\[ B(\alpha, \theta_o) = \int_{-\pi/2}^{\pi/2} L(\theta_i - \alpha) \, \rho(\theta_i, \theta_o) \, d\theta_i \]

\[ B_{l,p} = 2\pi L_l \rho_{l,p} \]

3D:
\[ B(\alpha, \beta, \theta_o, \phi_o) = \int_0^{\pi/2} \int_0^{2\pi} L(R_{\alpha, \beta}[\theta_i, \phi_i]) \rho(\theta_i, \phi_i, \theta_o, \phi_o) \, d\theta_i \, d\phi_i \]

\[ B_{lm,pq} = \Lambda_l L_{lm} \rho_{lq,pq} \]
Insights: Signal Processing

Signal processing framework for reflection

- Light is the signal
- BRDF is the filter
- Reflection on a curved surface is convolution
Insights: Signal Processing

Signal processing framework for reflection
- Light is the signal
- BRDF is the filter
- Reflection on a curved surface is convolution

Filter is Delta function : Output = Signal

Mirror BRDF : Image = Lighting

[Miller and Hoffman 84]

Image courtesy Paul Debevec
Insights: Signal Processing

Signal processing framework for reflection

- Light is the signal
- BRDF is the filter
- Reflection on a curved surface is convolution

Signal is Delta function : Output = Filter

Point Light Source : Images = BRDF

[Marschner et al. 00]
Phong, Microfacet Models

Mirror

Roughness

Illumination estimation ill-posed for rough surfaces

Lambertian Incident radiance (mirror sphere)

Irradiance (Lambertian)

\[ A_l = 2\pi \frac{(-1)^{\frac{l}{2}-1}}{(l+2)(l-1)} \left[ \frac{l!}{2^l \left( \frac{l}{2}! \right)^2} \right] \quad l \text{ even} \]


9 Parameter Approximation

Exact image

Order 2
9 terms

RMS Error = 1%

For any illumination, average error < 3% [Basri Jacobs 01]

Ramamoorthi and Hanrahan 01b
Convolution for general materials

\[
B(\vec{N}, \vec{V}) = \int_{\Omega} L \left( R(\vec{N}) \vec{l} \right) \rho(\vec{l}, \vec{V}) \, dl
\]

\[
B = L \otimes \rho
\]

- Spatial: integral
- Frequency: product
- Spherical harmonic analysis

\[
B_{ij} = L_i \rho_{ij}
\]

Ramamoorthi and Hanrahan 01
Real-Time Rendering

Motivation: Interactive rendering with natural illumination and realistic, measured materials
Outline

- Motivation
- Brief History
- Frequency Analysis for Light Transport
- Reflection as Convolution
- **Axis-Aligned Filtering for Soft Shadows**
  - A follow-up paper will be presented in class
Light Occluders Receiver
Shadow Light Field

- Following [Egan2011]
- \( X \) along plane parallel to light
- \( Y \) along light
- Binary Visibility \( f(.,.,.) \) and \( g(.) \)
Shadow Light Field

- Integrate visibility over area light

\[ c(x) = r(x) \times \int f(x, y) l(y) \, dy \]

- To determine filter parameters, need to understand structure of light field \( f \)
Shadow Light Field

\[ \rho = \frac{d_1}{d_2}, \quad \text{slope} = \frac{-1}{\rho - 1} \]
Spectrum of the Shadow Light Field

Shadow Light Field $f$

Fourier Spectrum $F$

\[
\text{slope} \in \left[ \frac{-1}{\rho_{\text{min}} - 1}, \frac{-1}{\rho_{\text{max}} - 1} \right]
\]

\[
\text{slope} \in \left[ \rho_{\text{min}} - 1, \rho_{\text{max}} - 1 \right]
\]
Spectrum of the Shadow Light Field

Frequency Spectrum $F(\Omega_x, \Omega_y)$ is bounded by a double wedge

$$s_{\text{min}} = \rho_{\text{min}} - 1$$
$$s_{\text{max}} = \rho_{\text{max}} - 1$$
Intensity at a pixel

\[ h(x) = \int f(x, y) l(y) \, dy \]

Pixel light intensity \( f(x, y) \) \( l(y) \) visibility light

\[ H(\Omega_x) = \int F(\Omega_x, \Omega_y) L(-\Omega_y) \, d\Omega_y \]
Intensity at a pixel

\[ h(x) = \int f(x, y) I(y) \, dy \]

Pixel light intensity \( \uparrow \) visibility \( \downarrow \) light

Fourier Transform

\[ H(\Omega_x) = \int F(\Omega_x, \Omega_y) L(-\Omega_y) \, d\Omega_y \]
Assume light intensity is gaussian with st. dev.

Fourier st. dev.

\[ \Omega_L^{\text{max}} = \frac{2}{\sigma} \ m^{-1} \]
Filtering the Light Field

So Far: Double Wedge band-limited by light

\[ H(\Omega_x) = \int F(\Omega_x, \Omega_y) L(-\Omega_y) \, d\Omega_y \]
Previous Work: Sheared Filter

- Egan2011: Most compact filter
- Allows close packing of aliases
- Low sampling rate needed
- High overhead due to search in 4D sample space
Outline

- Brief History
- Reflection as Convolution
- Recap of Sheared Filtering
- Axis-Aligned Filtering for Soft Shadows
  - A follow-up paper will be presented in class
Our Method: Axis-Aligned Filter

- Axis-Aligned filter is separable
- Pre-integrate along $y$
  \[ \bar{h}(x) = \int f(x, y) l(y) \, dy \]
- Only spatial filtering needed
  \[ \Omega^r_x = \min \left( \frac{\Omega^\text{max}_L}{s_{\text{min}}}, \Omega^\text{max}_x \right) \]
Image-Space Filtering

- Reduces to screen-space adaptive gaussian blur

\[
h(x) = \int_{x'} N_x (x - x') \overline{h}(x') dx'
\]

\[
N_x (x - x') \propto \exp\left( -8 |x - x'|^2 (\Omega_x^r)^2 \right)
\]

World-space distance projected on light plane
Image-Space Filtering

- Reduces to screen-space adaptive gaussian blur

\[ h(x) = \int_{x'} N_x(x - x') \overline{h}(x') \, dx' \]

- Filtering in 2D space instead of 4D – big speedup!
Image Space Filtering

\[ h(x) = \int_{x'} N_x(x - x') \, \bar{h}(x') \, dx' \]
Packing of Spectra
Packing of Spectra

(a) is tighter than (b)
Adaptive Sampling Rate

- Sampling rate based on compact packing of aliases

\[ n = \left( \Omega_x^* \right)^2 A_p \times \left( \Omega_y^* \right)^2 A_L \]

\[ = \left( 0.5 + l_p \Omega_x^s \right)^2 \left( 1 + l_L s_{\text{max}} \Omega_x^s \right)^2 \]

- When light size \( l_L = 0 \) \( n = 1 \)
Implementation: 1\textsuperscript{st} Pass

- 1\textsuperscript{st} Pass: Sparsely sample (9 spp) to obtain z-values, followed by splatting, and compute spp and filter size.
Implementation: 1\textsuperscript{st} Pass

- 1\textsuperscript{st} Pass: Sparsely sample (9 spp) to obtain z-values, followed by splatting, and compute spp and filter size.

<table>
<thead>
<tr>
<th>Filter width</th>
<th>Samples per pixel $n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.0</td>
</tr>
<tr>
<td>73</td>
<td>100</td>
</tr>
</tbody>
</table>
Implementation: 2\textsuperscript{nd} Pass

- 2\textsuperscript{nd} Pass: Adaptively sample visibility and integrate with light to obtain noisy per-pixel light intensity
Implementation: 3\textsuperscript{rd} Pass

- 3\textsuperscript{rd} Pass: Adaptively filter noisy per-pixel intensity
3rd Pass: Also multiply by BRDF and color

\[ c(x) = r(x) \times \int f(x, y) l(y) \, dy \]
Implementation details

- GPU used: NVIDIA GTX 570

- Use a Gaussian filter separated in the image x and y dimensions:

- No filtering between pixels if angle of normals $> 20^\circ$ since they likely ‘see’ different occluders
Sampling above the minimum

- Converge to ground truth with more samples
- Previous algorithms do not guarantee this
- We adapt sampling rate to filter size, giving convergence
Sampling above the minimum

\[ \Omega_x^* = \Omega_x^r + \Omega_x^{\text{max}} \]

\[ \Omega_y^* = \Omega_L^{\text{max}} + S_{\text{max}} \Omega_x^r \]

Increase filter width by \( \mu \)
Sampling above the minimum

\[ \Omega_x^* = \mu \Omega_x^r + \Omega_x^{\text{max}} \]

\[ \Omega_y^* = \Omega_L^{\text{max}} + \mu S_{\text{max}} \Omega_x^r \]
Error vs. spp: Adaptive Filtering

RMS Error vs. samples per pixel $n$
Our Method 20.5 SPP
22.24 FPS

Unfiltered MC 34 SPP
26.69 FPS
## Speed and Overhead

Resolution 640 x 480

<table>
<thead>
<tr>
<th>Parameter</th>
<th>“Grids”</th>
<th>“Bench”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertices</td>
<td>200</td>
<td>309000</td>
</tr>
<tr>
<td>Total raytracing (ms)</td>
<td>20.4</td>
<td>425</td>
</tr>
<tr>
<td>Total Overhead (ms)</td>
<td>5.01</td>
<td>4.78</td>
</tr>
</tbody>
</table>

Overhead \ll \text{ray-tracing}
## Speed and Overhead

Resolution 640 x 480

<table>
<thead>
<tr>
<th>Parameter</th>
<th>“Grids”</th>
<th>“Bench”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertices</td>
<td>200</td>
<td>309000</td>
</tr>
<tr>
<td>Total raytracing (ms)</td>
<td>20.4</td>
<td>425</td>
</tr>
<tr>
<td>Total Overhead (ms)</td>
<td>5.01</td>
<td>4.78</td>
</tr>
<tr>
<td>FPS w/o overhead</td>
<td>49</td>
<td>2.3</td>
</tr>
<tr>
<td>FPS w/ overhead</td>
<td>39</td>
<td>2.3</td>
</tr>
</tbody>
</table>

**Interactive frame-rates**
Conclusion: Contributions

- Express axis-aligned filtering as integration followed by convolution
- Adaptive sampling rates + filter sizes based on frequency analysis instead of heuristics
- Interactive frame rates: filtering has minimal overhead
- Many subsequent works presented (diffuse GI, multi-effects)