Path Tracing & Microfacet BSDFs

Gurprit Singh
Ray Tracing
Ray Tracing

Scene designed by David Coeurjolly

Image Plane
Ray Tracing

Scene designed by David Coeurjolly
Direct Illumination

Image rendered using PBRT
Direct Illumination

Image rendered using PBRT
Direct and Indirect Illumination
Path Tracing
Path Tracing
Path Tracing
Path Tracing
Direct and Indirect Illumination

Image rendered using PBRT
Variance Reduction Techniques

- Correlated Sampling
- Importance Sampling
- Perceptual Error Distribution
Correlated Sampling: Jittered Sampling
Variance reduction: Stratified Sampling

Random 2D

Jittered 2D
Variance reduction: Stratified Sampling

Random 2D

Jittered 2D
Variance reduction: Stratified Sampling

Random 2D

Jittered 2D
Variance reduction: Stratified Sampling

Random 2D

Jittered 2D
Random vs. Stratified Sampling
Random vs. Stratified Sampling

Random Samples

Jittered Samples

N = 64 spp

Stratified sampling suffers from the curse of dimensionality
Correlated Sampling: Latin Hypercube Sampling
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)

Initialize
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)

Shuffle columns
Latin Hypercube Sampler (N-rooks)
Latin Hypercube Sampler (N-rooks)
Variants of stratified sampling

Figure 2.25: Stratification of $I^2$ with Voronoi diagrams. (a) 64-element Hammersley point set; (b) Voronoi diagram implied through (a); (c) 64-element hexagonal grid; (d) Voronoi diagram implied through (c).

Slide from Philipp Slusallek
Correlated Sampling: Quasi-Monte Carlo Integration
Quasi-Monte Carlo Integration

- Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible.

Quasi-Monte Carlo Integration
Quasi-Monte Carlo Integration

- Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible
Quasi-Monte Carlo Integration

- Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible.

- The success of any Monte Carlo procedure stands or falls with the quality of these random samples.
Quasi-Monte Carlo Integration

• Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible

• The success of any Monte Carlo procedure stands or falls with the quality of these random samples
Quasi-Monte Carlo Integration

- Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible.

- The success of any Monte Carlo procedure stands or falls with the quality of these random samples.

- If the distribution of the sample points is not uniform then there are large regions where there are no samples at all, which can increases the error.
Quasi-Monte Carlo Integration

- Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible.

- The success of any Monte Carlo procedure stands or falls with the quality of these random samples.

- If the distribution of the sample points is not uniform then there are large regions where there are no samples at all, which can increases the error.
Quasi-Monte Carlo Integration

- Monte Carlo integration suffers, apart from the slow convergence rate, from the disadvantages that only probabilistic statements on convergence and error boundaries are possible.

- The success of any Monte Carlo procedure stands or falls with the quality of these random samples.

- If the distribution of the sample points is not uniform then there are large regions where there are no samples at all, which can increase the error.

- Closely related to this is the fact that a smooth function is evaluated at unnecessary many locations if samples are clumped.
Quasi-Monte Carlo Integration

• Deterministic generation of samples, while making sure uniform distributions
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions
- Based on number-theoretic approaches
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions

- Based on number-theoretic approaches
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions

- Based on number-theoretic approaches
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions

- Based on number-theoretic approaches

- Samples with good uniform properties can be generated in very high dimensions.
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions
- Based on number-theoretic approaches
- Samples with good uniform properties can be generated in very high dimensions.
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions

- Based on number-theoretic approaches

- Samples with good uniform properties can be generated in very high dimensions.
Quasi-Monte Carlo Integration

- Deterministic generation of samples, while making sure uniform distributions
- Based on number-theoretic approaches
- Samples with good uniform properties can be generated in very high dimensions.
- Sample generation is pretty fast: (almost) no pre-processing
Quasi-Monte Carlo Integration

• Low discrepancy sequences
Quasi-Monte Carlo Integration

- Low discrepancy sequences
  - Halton and Hammersley sequences
Quasi-Monte Carlo Integration

• Low discrepancy sequences
  • Halton and Hammerslay sequences
  • Scrambled sequences
Quasi-Monte Carlo Integration

- Low discrepancy sequences
  - Halton and Hammersley sequences
  - Scrambled sequences
- Discrepancy
Discrepancy: Basic idea

- The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution
The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution.
The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution.
The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution.
The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution.

Area of the blue box: 0.09
Area associated to each sample: 0.25
Discrepancy: Basic idea

- The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution.

Area of the blue box: 0.09
Area associated to each sample: 0.25
Discrepancy:
Discrepancy: Basic idea

- The concept of discrepancy can be viewed as a quantitative measure for the deviation of a given point set from a uniform distribution.

Area of the blue box: 0.09
Area associated to each sample: 0.25
Discrepancy: 0.25 - 0.09 = 0.16
Spatial Statistics: Discrepancy

Discrepancy = BoxArea - FractionSamples

Random

Jitter

Poisson Disk

Star Discrepancy
Spatial Statistics: Discrepancy

Discrepancy = BoxArea - FractionSamples

Random

Jitter

Poisson Disk

Star Discrepancy

Realistic Image Synthesis SS2021
Spatial Statistics: Discrepancy

Discrepancy = BoxArea - FractionSamples

Star Discrepancy
**Discrepancy**

**Definition 2.1 (Discrepancy)** Let \( P = \{x_1, x_2, \ldots, x_N\} \) with \( x_i \in I^s \), \( i = 1, \ldots, N \) be a point set. The discrepancy of \( P \), denoted as \( D_N(P) \), is a measure for the deviation of a point set from its ideal distribution. The discrepancy of \( P \) is defined as

\[
D_N(P) \equiv D_N(P, \mathcal{B}) \quad \text{def} \quad \sup_{B \in \mathcal{B}} \left| \frac{\#(P \cap B)}{N} - \mu^s(B) \right|
\]

where \( \mathcal{B} \) corresponds to a Lebesgue measurable family of subsets of \( I^s \), \( \# \) corresponds to the counting measure over \( \mathcal{B} \) with respect to \( P \), \( \mu^s \) is, as usual, the Lebesgue measure and \( \mathcal{B} \) refers to a non-empty subset of \( \mathcal{B} \).
Radical Inverse

Techniques based on a construction called as radical inverse

Any integer can be represented in the form:

\[ n = \sum_{i=1}^{\infty} d_i b^{i-1} \]

<table>
<thead>
<tr>
<th>n</th>
<th>Binary</th>
<th>( \Phi_b(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>001</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td></td>
</tr>
</tbody>
</table>
Radical Inverse

Any integer can be represented in the form:

\[ n = \sum_{i=1}^{\infty} d_i b^{i-1} \]

Radical inverse:

\[ \Phi_b(n) = 0.d_1 d_2 \ldots d_m \]

<table>
<thead>
<tr>
<th>n</th>
<th>Binary</th>
<th>( \Phi_b(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0,1</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>0,01</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>0,11</td>
</tr>
<tr>
<td>4</td>
<td>001</td>
<td>0,001</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>0,101</td>
</tr>
</tbody>
</table>
Radical Inverse

Techniques based on a construction called as **radical inverse**

Radical inverse:

\[ \Phi_b(n) = 0.d_1d_2...d_m \]

<table>
<thead>
<tr>
<th>n</th>
<th>Binary</th>
<th>( \Phi_b(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.1 = 1/2</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>0.01 = 1/4</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>0.11 = 3/4</td>
</tr>
<tr>
<td>4</td>
<td>001</td>
<td>0.001 = 1/8</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>0.101 = 5/8</td>
</tr>
</tbody>
</table>
Techniques based on a construction called as radical inverse

Radical inverse:

\[ \Phi_b(n) = 0.d_1d_2...d_m \]

<table>
<thead>
<tr>
<th>n</th>
<th>Binary</th>
<th>( \Phi_b(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.1 = 1/2</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>0.01 = 1/4</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>0.11 = 3/4</td>
</tr>
<tr>
<td>4</td>
<td>001</td>
<td>0.001 = 1/8</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>0.101 = 5/8</td>
</tr>
</tbody>
</table>
Halton and Hammerslay Sequence

Techniques based on a construction called as radical inverse

Radical inverse: \( \Phi_b(n) = 0.d_1d_2...d_m \)

Halton Sequence: For n-dimensional sequence, we use different base b for each dimension

\[ x_i = (\Phi_2(i), \Phi_3(i), \Phi_5(i), \ldots, \Phi_{p_n}(i)) \]
Halton and Hammerslay Sequence

Techniques based on a construction called as **radical inverse**

Radical inverse: \( \Phi_b(n) = 0.d_1d_2...d_m \)

Halton Sequence: For n-dimensional sequence, we use different base b for each dimension

\[
x_i = (\Phi_2(i), \Phi_3(i), \Phi_5(i), \ldots, \Phi_{p_n}(i))
\]

Hammerslay Sequence: All except the first dimension has co-prime bases

\[
x_i = \left( \frac{i}{N}, \Phi_{b_1}(i), \Phi_{b_2}(i), \ldots, \Phi_{b_n}(i) \right)
\]
Halton and Hammersley Sequence

Techniques based on a construction called as **radical inverse**

Radical inverse: \( \Phi_b(n) = 0.d_1d_2\ldots d_m \)

Halton Sequence:
\[
x_i = (\Phi_2(i), \Phi_3(i), \Phi_5(i), \ldots, \Phi_{p_n}(i))
\]

Hammerslay Sequence:
\[
x_i = \left( \frac{i}{N}, \Phi_{b_1}(i), \Phi_{b_2}(i), \ldots, \Phi_{b_n}(i) \right)
\]

Hammerslay has slightly **lower** discrepancy than Halton
Low discrepancy samplers

Halton 4spp
Low discrepancy samplers

Halton 8spp
Low discrepancy samplers

Sobol 4spp
Low discrepancy samplers

Random 8spp
Visualizing samples

Figure 2.7: Hammersley Point Set on the 2D Plane. Three 2-dimensional Hammersley point sets $P^2_{HAM} = \left( \frac{i}{N}, \Phi_2(i) \right)_{i \in \{0, ..., N-1\}}$ of sizes $N = 64$-element, $N = 256$-element and $N = 512$-element.
Visualizing samples

Figure 2.5: Halton sequence. The first 64, 256, and 512 points of the 2-dimensional Halton Sequence $P_{HAL}^2 = (\Phi_2(i), \Phi_3(i))_{i \in \mathbb{N}_0}$. 

Slide from Philipp Slusallek
Visualizing samples

Projection: (9,10)  Projection: (19,20)  Projection: (29,30)

Halton Sequence

Slide from Philipp Slusallek
Faure's permutation

Figure 2.12: Halton Sequence and Scrambled Halton Sequence, Dimensions 7 and 8. (a) The first 256 elements of the 2-dimensional Halton sequence $P^{2}_{HAL} = (\Phi_7(i), \Phi_8(i))$ and the scrambled versions of dimension 7 and 8 generated according to procedure of Faure.
Importance Sampling
Importance Sampling

\[ L_o(p, \omega) = \int_{H^2} f_r(x, \omega_0, \omega_i) L_i(x, \omega_i) \left| \cos \theta_i \right| d\omega_i \]

What terms can we importance sample?

- BSDF
- Incident radiance
- cosine term
Importance Sampling: Cosine term

\[ L_o(p, \omega) = \int_{H^2} f_r(x, \omega_0, \omega_i) L_i(x, \omega_i) |\cos \theta_i| \, d\omega_i \]

What terms can we importance sample?

- BSDF
- Incident radiance
- cosine term
Example: Ambient Occlusion
Example: Ambient Occlusion

\[
L_o(p, \omega) = \int_{\mathcal{H}^2} f_r(x, \omega_0, \omega_i) L_i(x, \omega_i) |\cos \theta_i| d\omega_i
\]

\[
L_o(p, \omega) = \frac{\rho}{\pi} \int_{\mathcal{H}^2} V(x, \omega_i) |\cos \theta_i| d\omega_i
\]
Example: Ambient Occlusion

\[ L_o(p, \omega) = \frac{\rho}{\pi} \int_{\mathcal{H}^2} V(x, \omega_i) |\cos \theta_i| d\omega_i \]
Example: Ambient Occlusion

\[ L_o(p, \omega) = \frac{\rho}{\pi} \int_{H^2} V(x, \omega_i) |\cos \theta_i| d\omega_i \]

\[ L_o(p, \omega) = \frac{\rho}{\pi} \frac{1}{N} \sum_{k=1}^{N} \frac{V(x, \omega_{i,k}) |\cos \theta_{i,k}|}{p(x, \omega_{i,k})} \]
Example: Ambient Occlusion

\[ L_o(p, \omega) = \frac{\rho}{\pi} \frac{1}{N} \sum_{k=1}^{N} \frac{V(x, \omega_{i,k}) | \cos \theta_{i,k} |}{p(x, \omega_{i,k})} \]

\[ p(x, \omega_{i,k}) \propto ??? \]
Hemispherical Sampling: Constant PDF

Uniform Hemispherical Sampling

(1 Sample)
Hemispherical Sampling: Constant PDF

Uniform Hemispherical Sampling

\[ p(x, \omega_i) = \frac{1}{2\pi} \]
Uniform Hemispherical Sampling

$$p(x, \omega_i) = \frac{1}{2\pi}$$

(256 Samples)
Importance Sampling: Cosine term

Uniform Hemispherical Sampling

\[ p(x, \omega_i) = \frac{1}{2\pi} \]

Cosine-weighted Importance Sampling

\[ p(x, \omega_i) = \cos \theta_i \]
Uniform hemispherical sampling  1 sample/pixel  Cosine-weighted importance sampling
Uniform hemispherical sampling 4 sample/pixel

Cosine-weighted importance sampling
Uniform hemispherical sampling 16 sample/pixel

Cosine-weighted importance sampling

Slide from Wojciech Jarosz
Importance Sampling: Incident Radiance

\[ L_o(p, \omega) = \int_{\mathcal{H}^2} f(p, \omega_0, \omega_i) L_i(x, \omega_i) |\cos \theta_i| d\omega_i \]

What terms can we importance sample?

- BSDF
- Incident radiance
- cosine term
Example: Environment Lighting
Example: Environment Lighting
Environment Lighting

Environment map (distant light source)

Scene

Slide after Wojciech Jarosz
Importance function

\[ \phi \]

\[ \theta \]

Slide after Wojciech Jarosz
Importance function

Scalar version e.g., luminance channel only
Importance function: Scalar function

Multiplication with $\sin \theta$
Importance function: Marginalization

Slide after Wojciech Jarosz
Importance function: Conditional PDFs

Once normalized, each row can serve as the conditional PDF.
Importance function: Sampling
Importance function: Sampling

Slide after Wojciech Jarosz
Importance function: Sampling

Slide after Wojciech Jarosz
Importance function: Sampling
Importance Sampling

For more details, see PBRTv3: 13.2 and 13.6.7
Importance Sampling

\[ L_o(p, \omega) = \int_{\mathcal{H}^2} f(p, \omega_0, \omega_i) L_i(x, \omega_i) |\cos \theta_i| d\omega_i \]

What terms can we importance sample?

- BSDF
- Incident radiance
- cosine term

To handle this, we will introduce Microfacet BSDF theory in the later part of the lecture.
Light vs. BSDF Importance Sampling

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

Light PDF Sampling

BSDF PDF Sampling

Sensor
Light vs. BSDF Importance Sampling

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

- Light PDF Sampling
- BSDF PDF Sampling

Sensor
$I_N = \frac{1}{N} \sum_{k=1}^{N} \frac{f(\vec{x}_k)}{p(\vec{x}_k)}$

Light vs. BSDF Importance Sampling

Light PDF Sampling

BSDF PDF Sampling
Light vs. BSDF Importance Sampling

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

Light PDF Sampling

BSDF PDF Sampling

Sensor
Light vs. BSDF Importance Sampling

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

Light PDF Sampling

BSDF PDF Sampling

Sensor

Light IS

BSDF IS
Variance reduction: Importance sampling

Reference image
N = 1024 spp

BSDF importance sampling
N = 4 spp

Light importance sampling
N = 4 spp
Variance reduction: Importance sampling

Reference image
N = 1024 spp

BSDF importance sampling
N = 4 spp

Light importance sampling
N = 4 spp

BSDF sampling is better in some regions
Variance reduction: Importance sampling

Reference image
N = 1024 spp

BSDF importance sampling
N = 4 spp

Light importance sampling
N = 4 spp

Light sampling is better in other regions
Variance reduction: Importance sampling

Can we combine the benefits of different PDFs?
Variance reduction: Importance sampling

Can we combine the benefits of different PDFs?
Variance reduction: Importance sampling

Can we combine the benefits of different PDFs? Yes!
Variance reduction: Importance sampling

Can we combine the benefits of different PDFs? **Yes!**
Variance reduction: Multiple Importance sampling

Multiple Importance Sampling

\[ I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x)g(x)}{p(x)} \]
Variance reduction: Multiple Importance sampling

Multiple Importance Sampling

\[ I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x)g(x)}{p(x)} \]

\[ p(x) \propto ??? \]
Variance reduction: Multiple Importance sampling

Multiple Importance Sampling

\[ I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x)g(x)}{p(x)} \]

\[ p(x) \propto ??? \]

\[ I_N = \frac{1}{n_f} \sum_{i=1}^{n_f} \frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)} + \frac{1}{n_g} \sum_{j=1}^{n_g} \frac{f(Y_j)g(Y_j)w_g(Y_j)}{p_g(Y_j)} \]
Variance reduction: Multiple Importance sampling

Multiple Importance Sampling

\[ I_N = \frac{1}{n_f} \sum_{i=1}^{n_f} \frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)} + \frac{1}{n_g} \sum_{j=1}^{n_g} \frac{f(Y_j)g(Y_j)w_g(Y_j)}{p_g(Y_j)} \]

Balance heuristic:

\[ w_s(x) = \frac{n_s p_s(x)}{\sum_i n_i p_i(x)} \]

Power heuristic:

\[ w_s(x) = \frac{(n_s p_s(x))^\beta}{\sum_i (n_i p_i(x))^\beta} \quad \beta = 2 \]
Rendering Equation
Rendering Equation

\[ L_o(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o) \]

Outgoing  emitted  reflected

James Kajiya, The Rendering Equation, SIGGRAPH 1986
Rendering Equation

\[ L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\mathcal{H}^2} f_r(x, \omega_0, \omega_i) L_i(x, \omega_i) \cos \theta_i \, d\omega_i \]

Outgoing  emitted  reflected
Rendering Equation: Light Transport

In vacuum, radiance is constant along rays

We can relate out-going radiance to the incoming radiance

\[ L_i(x, \bar{\omega}) = L_0(r(x, \bar{\omega}), -\bar{\omega}) \]
Rendering Equation: Light Transport

In vacuum, radiance is constant along rays.

We can relate out-going radiance to the incoming radiance.

\[ L_i(x, \bar{\omega}) \rightarrow L_o(r(x, \bar{\omega}), -\bar{\omega}) \]

\[ x \rightarrow \bar{\omega} \quad \text{ray tracing function} \]
Rendering Equation

\[ L_o(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L_i(x, \omega) |\cos \theta'| d\omega' \]
Rendering Equation

\[ L(x, \omega) = L_e(x, \omega) + \int_{H^2} f(x, \omega', \omega) L(r(x, \omega), -\omega') | \cos \theta' | d\omega' \]

Only outgoing radiance on both sides

- we drop the "o" subscript
- Becomes Fredholm equation of the second kind (recursive)
Rendering Equation

\[
L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') |\cos \theta'| d\bar{\omega}'
\]
Rendering Equation

\[ L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') | \cos \theta' | d\bar{\omega}' \]
Rendering Equation

\[ L(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{\mathcal{H}^2} f(x, \vec{\omega}', \vec{\omega}) L(r(x, \vec{\omega}'), -\vec{\omega}') | \cos \theta' | d\vec{\omega}' \]
$L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \bar{\omega}) L(r(x, \omega'), -\bar{\omega'}) | \cos \theta' | d\bar{\omega'}$

- **Light source**
- **Integrate over the hemisphere**
Rendering Equation

\[
L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') |\cos \theta'| d\omega'
\]
\[ L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') | \cos \theta' | d\bar{\omega}' \]
Rendering Equation

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') |\cos \theta'| d\omega' \]
Rendering Equation

\[
L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') |\cos \theta'| \, d\bar{\omega}'
\]
Rendering Equation

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') |\cos \theta'| d\omega' \]
\[ L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') \cos \theta' |d\bar{\omega}' \]
Rendering Equation

\[ L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \omega') L(r(x, \bar{\omega}'), -\bar{\omega}') \mid \cos \theta' \mid d\bar{\omega}' \]
Rendering Equation

\[ L(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_{\mathcal{H}^2} f(x, \vec{\omega}', \vec{\omega}) L(r(x, \vec{\omega}'), -\vec{\omega}') \cos \theta' \, d\vec{\omega}' \]
The rendering equation is given by:

\[ L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') \left| \cos \theta' \right| d\bar{\omega}' \]
\[ L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') |\cos \theta' | d\bar{\omega}' \]
Rendering Equation

\[
L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') \cos \theta' \, d\bar{\omega}'
\]
Rendering Equation

\[
L(x, \bar{\omega}) = L_e(x, \bar{\omega}) + \int_{\mathcal{H}^2} f(x, \bar{\omega}', \bar{\omega}) L(r(x, \bar{\omega}'), -\bar{\omega}') \left| \cos \theta' \right| d\bar{\omega}'
\]
Rendering Equation

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') | \cos \theta' | d\omega' \]
Rendering Equation

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega') L(r(x, \omega'), -\omega') |\cos \theta'| d\omega' \]

Light source

\[ L(r(x, \omega'), -\omega') \]

\[ x \]

\[ \omega \]

\[ \omega' \]

\[ \omega' \]

\[ \omega \]

recursion
Path Tracing
Path Tracing

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega)L(r(x, \omega'), -\omega') \cos \theta' d\omega' \]
Path Tracing

\[
L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' d\omega'
\]

\[
\approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')}
\]
Path Tracing

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing

\[ L(x, \omega) = L_e(x, \omega) + \int_{H^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' \, d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' \, d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' \, d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing Algorithm

\[ L_o(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o) \]

```plaintext
Color color(Point x, Direction \omega, int moreBounces):
    if not moreBounces:
        return L_e(x, -\omega)
    // sample recursive integral
    \omega' = sample from BRDF
    return L_e(x, -\omega) + BRDF * color(trace(x, \omega'), moreBounces-1) * dot(n, \omega') / pdf(\omega')
```
Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces
Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces

Let's estimate direct illumination separately from indirect illumination, then add the two
Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces

Let's estimate direct illumination separately from indirect illumination, then add the two

- i.e., shoot shadow rays (direct) and gather rays (indirect)

- be careful not to double count!
Partitioning the Integrand

Direct Illumination: sometimes better estimated by sampling the emissive surfaces

Let's estimate direct illumination separately from indirect illumination, then add the two

- i.e., shoot shadow rays (direct) and gather rays (indirect)

- be careful not to double count!

Also known as Next Event Estimation (NEE)
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing Algorithm with NEE

\[
L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' d\omega'
\]

\[
\approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')}
\]
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' \, d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + \int_{H^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing Algorithm with NEE

\[
L(x, \omega) = L_e(x, \omega) + \int_{H^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' d\omega'
\]

\[
\approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')}
\]
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]

Light source
Path Tracing Algorithm with NEE

\[
L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' \, d\omega'
\]

\[
\approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')}
\]
Light source

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' \, d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + \int_{\mathcal{H}^2} f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta' \, d\omega' \]

\[ \approx L_e(x, \omega) + \frac{f(x, \omega', \omega) L(r(x, \omega'), -\omega') \cos \theta'}{p(\omega')} \]

Avoid double count!
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + L_{dir}(x, \omega) + L_{ind}(x, \omega) \]

**Color** color(Point x, Direction ω, int moreBounces):

    if not moreBounces:
        return \( L_e \);

    // next-event estimation: compute \( L_{dir} \) by sampling the light
    ω₁ = sample from light
    \( L_{dir} = BRDF \times color(\text{trace}(x, ω₁), 0) \times \text{dot}(n, ω₁) / \text{pdf}(ω₁) \)

    // compute \( L_{ind} \) by sampling the BSDF
    ω₂ = sample from BSDF;
    \( L_{ind} = BSDF \times color(\text{trace}(x, ω₂), \text{moreBounces}-1) \times \text{dot}(n, ω₂) / \text{pdf}(ω₂) \)

    return \( L_e + L_{dir} + L_{ind} \)
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + L_{\text{dir}}(x, \omega) + L_{\text{ind}}(x, \omega) \]

Color color(Point x, Direction \( \omega \), int moreBounces):

\begin{verbatim}
if not moreBounces:
    return \( L_e \);

// next-event estimation: compute \( L_{\text{dir}} \) by sampling the light
\( \omega_1 \) = sample from light
\( L_{\text{dir}} = \text{BRDF} \times \text{color(trace(x, \omega_1), \theta) \times \text{dot}(n, \omega_1) / pdf(\omega_1)} \)

// compute \( L_{\text{ind}} \) by sampling the BSDF
\( \omega_2 \) = sample from BSDF;
\( L_{\text{ind}} = \text{BSDF} \times \text{color(trace(x, \omega_2), moreBounces-1) \times \text{dot}(n, \omega_2) / pdf(\omega_2)} \)

return \( L_e + L_{\text{dir}} + L_{\text{ind}} \)
\end{verbatim}
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + L_{dir}(x, \omega) + L_{ind}(x, \omega) \]

Color color(Point x, Direction \( \omega \), int moreBounces):

    if not moreBounces:
        return \( L_e \);

    // next-event estimation: compute \( L_{dir} \) by sampling the light
    \( \omega_1 \) = sample from light
    \( L_{dir} = \text{BRDF} \ast \text{color}(\text{trace}(x, \omega_1), 0) \ast \text{dot}(n, \omega_1) / \text{pdf}(\omega_1) \)

    // compute \( L_{ind} \) by sampling the BSDF
    \( \omega_2 \) = sample from BSDF;
    \( L_{ind} = \text{BSDF} \ast \text{color}(\text{trace}(x, \omega_2), \text{moreBounces}-1) \ast \text{dot}(n, \omega_2) / \text{pdf}(\omega_2) \)

    return \( L_e + L_{dir} + L_{ind} \)
Path Tracing Algorithm with NEE

\[ L(x, \omega) = L_e(x, \omega) + L_{\text{dir}}(x, \omega) + L_{\text{ind}}(x, \omega) \]

```python
def color(Point x, Direction \omega, int moreBounces, bool includeL_e):
    L_e = includeL_e ? L_e(x, -\omega) : black
    if not moreBounces:
        return L_e

    // next-event estimation: compute \( L_{\text{dir}} \) by sampling the light
    \( \omega_1 = \text{sample from light} \)
    \( L_{\text{dir}} = \text{BRDF} \times \text{color}(\text{trace}(x, \omega_1), \emptyset, \text{true}) \times \text{dot}(n, \omega_1) / \text{pdf}(\omega_1) \)

    // compute \( L_{\text{ind}} \) by sampling the BSDF
    \( \omega_2 = \text{sample from BSDF} \)
    \( L_{\text{ind}} = \text{BSDF} \times \text{color}(\text{trace}(x, \omega_2), \text{moreBounces}-1, \text{false}) \times \text{dot}(n, \omega_2) / \text{pdf}(\omega_2) \)

    return L_e + L_{\text{dir}} + L_{\text{ind}}
```
Path-wise Visualization

Path: 0
Path-wise Visualization

Path: 0

Path: 1
Path-wise Visualization

Path: 0
Path: 1
Path: 2
Path-wise Visualization

Path: 0
Path: 1
Path: 2
Path: 3
Path-wise Visualization

Path: 0
Path: 1
Path: 2
Path: 3
Path: 4
Path-wise Visualization

Path: 0
Path: 1
Path: 2
Path: 3
Path: 4
Path: 5
Path-wise Visualization

Path: 0

Path: 1

Path: 2

Path: 3

Path: 4

Path: 5

Path: 6
Path-wise Visualization

Path: 0
Path: 1
Path: 2
Path: 3
Path: 4
Path: 5
Path: 6
All Paths added
When we do stop recursion?

Truncating at some fixed depth introducing **bias**

**Solution:** Russian roulette
Russian Roulette

Probabilistically terminate the recursion

New estimator: evaluate original estimator $X$ with probability $P$ (but reweighted), otherwise return zero:

$$X_{rr} = \begin{cases} \frac{X}{P} & \xi < P \\ 0 & \text{otherwise} \end{cases}$$
This will increase variance!

- but it will improve efficiency if $P$ is chosen so that the samples that are expensive, but are likely to make small contribution, are skipped
Microfacet BSDFs
BRDF

Bidirectional Reflectance Distribution Function
BRDF Properties

Real/Physically plausible BRDFs obey:

- Energy conservation:

\[
\int_{\mathcal{H} \in \mathbb{R}} f_r(x, \tilde{\omega}_i, \tilde{\omega}_r) \cos \theta_i \, d\tilde{\omega}_i \leq 1, \quad \forall \tilde{\omega}_r
\]
BRDF Properties

Real/Physically plausible BRDFs obey:

- Energy conservation:
  \[ \int_{\mathcal{H}} f_r(x, \bar{\omega}_i, \bar{\omega}_r) \cos \theta_i d\omega_i \leq 1, \quad \forall \bar{\omega}_r \]

- Helmholtz reciprocity:
  \[ f_r(x, \bar{\omega}_i, \bar{\omega}_r) = f_r(x, \bar{\omega}_r, \bar{\omega}_i) \]
  \[ f_r(x, \bar{\omega}_i \leftrightarrow \bar{\omega}_r) \]
Conductors vs. Dielectrics

Conductors: Materials that conduct electricity, e.g. metal

Dielectrics: Materials that does not conduct electricity, e.g., water, mineral oil, air

| Crystal rocks | Mercury | Clouds |
Conductors vs. Dielectrics

- Copper
- Iron
- Gold
- Glass
- Crystal rocks
- Mercury
- Clouds
Conductors vs. Dielectrics

Smooth conducting material

Smooth dielectric material
Conductors vs. Dielectrics

Smooth conducting material

Smooth dielectric material

Rough conducting material

Rough dielectric material
Three Levels of Detail

**Key Idea:** transition from individual interactions to statistical averages

Macro Scale

Scene geometry
Three Levels of Detail

**Key Idea:** transition from individual interactions to statistical averages

Macro Scale

Scene geometry
Three Levels of Detail

**Key Idea:** transition from individual interactions to statistical averages
Three Levels of Detail

**Key Idea:** transition from individual interactions to statistical averages
Three Levels of Detail

Key Idea: transition from individual interactions to statistical averages
Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:
Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:

\[ r \sim \cos(\theta) \]

incident direction

mirror-reflection direction

outgoing direction
Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:

\[ f_r(\vec{\omega}_o, \vec{\omega}_i) = \frac{e + 2}{2\pi} (\vec{\omega}_r \cdot \vec{\omega}_o)^e \]

\[ \vec{\omega}_r = (2\vec{n}(\vec{n} \cdot \vec{\omega}_i) - \vec{\omega}_i) \]
Blinn-Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:
Blinn-Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:

\[ f_r(\vec{\omega}_o, \vec{\omega}_i) = \frac{e + 2}{2\pi} (\vec{\omega}_h \cdot \vec{n})^e \]

\[ \vec{\omega}_r = (2\vec{n}(\vec{n} \cdot \vec{\omega}_i) - \vec{\omega}_i) \]
Blinn-Phong BRDF

Reflection direction distributed over an exponentiated cosine lobe:

\[ f_r(\vec{ω}_o, \vec{ω}_i) = \frac{e + 2}{2\pi} (\vec{ω}_h \cdot \vec{n})^e \]

\[ \vec{ω}_r = (2\vec{n}(\vec{n} \cdot \vec{ω}_i) - \vec{ω}_i) \]

\[ \vec{ω}_h = \frac{\vec{ω}_i + \vec{ω}_o}{||\vec{ω}_i + \vec{ω}_o||} \]

incident direction

outgoing direction

half-vector \( \vec{ω}_h \)
Rough Surfaces

Empirical glossy models have limitations:

- not physically-based
- (often) not reciprocal
Rough Surfaces

Empirical glossy models have limitations:

- not physically-based
- (often) not reciprocal
- not energy-preserving (can be normalized): many conflicting normalizations in the literature
- (often) no Fresnel effects
- cannot accurately model appearance of many glossy surfaces
Empirical glossy models have limitations:
- not physically-based
- (often) not reciprocal
- not energy-preserving (can be normalized): many conflicting normalizations in the literature
- (often) no Fresnel effects
- cannot accurately model appearance of many glossy surfaces

Blinn-Phong was first step in the right direction

Can do Better
Microfacet Theory
In geometric-optics-based approaches, rough surfaces can be modeled as a collection of small microfacets.

Surfaces comprised of microfacets are often modeled as heightfields, where the distribution of facet orientations is described statistically.
Assume surface consists of tiny facets

Assume that the differential area being viewed/illuminated is relatively large compared to the size of microfacets

A facet can be perfectly specular or diffuse
Torrance-Sparrow Model

Developed by Torrance & Sparrow in 1967

Originally used in the physics community
Torrance-Sparrow Model

Developed by Torrance & Sparrow in 1967

Originally used in the physics community

Adapted by Cook & Torrance and Blinn for graphics

- added ambient and diffuse terms
Torrance-Sparrow Model

Developed by Torrance & Sparrow in 1967

Originally used in the physics community

Adapted by Cook & Torrance and Blinn for graphics

- added ambient and diffuse terms

Explain off-specular peaks

Assumes surface is composed of many micro-grooves, each of which is a perfect mirror
Cook-Torrance (1981)

Copper-colored plastic  Copper
General Microfacet Model

\[ f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|\vec{\omega}_i \cdot \vec{n}|(\vec{\omega}_o \cdot \vec{n})} \]

\[ \vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{||\vec{\omega}_i + \vec{\omega}_o||} \]
General Microfacet Model

Fresnel coefficient

\[ f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|\vec{\omega}_i \cdot \vec{n}|(\vec{\omega}_o \cdot \vec{n})} \]

\[ \vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{||\vec{\omega}_i + \vec{\omega}_o||} \]
General Microfacet Model

\[ f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|\vec{\omega}_i \cdot \vec{n}|(\vec{\omega}_o \cdot \vec{n})} \]

Fresnel coefficient

Microfacet distribution

\[ \vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{||\vec{\omega}_i + \vec{\omega}_o||} \]
General Microfacet Model

\[ f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|\langle \vec{\omega}_i \cdot \vec{n} \rangle | \langle \vec{\omega}_o \cdot \vec{n} \rangle |} \]

- **Fresnel coefficient**
- **Microfacet distribution**
- **Shadowing/masking**

\[ \vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{||\vec{\omega}_i + \vec{\omega}_o||} \]
General Microfacet Model

\[ f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|\vec{\omega}_i \cdot \vec{n}|(\vec{\omega}_o \cdot \vec{n})|} \]

Fresnel coefficient

\[ \vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{||\vec{\omega}_i + \vec{\omega}_o||} \]
Fresnel Term
General Microfacet Model

\[ f(\bar{\omega}_i, \bar{\omega}_o) = \frac{F(\bar{\omega}_h, \bar{\omega}_o) \cdot D(\bar{\omega}_h) \cdot G(\bar{\omega}_i, \bar{\omega}_o)}{4|\bar{\omega}_i \cdot \bar{n}|(\bar{\omega}_o \cdot \bar{n})|} \]

Microfacet distribution

\[ \bar{\omega}_h = \frac{\bar{\omega}_i + \bar{\omega}_o}{||\bar{\omega}_i + \bar{\omega}_o||} \]
Microfacet Distribution

\[ \mathbf{m}_i \quad \text{and} \quad \mathbf{m}_o \]
Microfacet Distribution

\( \vec{\omega}_i \)

\( \vec{\omega}_o \)
Microfacet Distribution

$\vec{\omega}_i$

$\vec{\omega}_o$
Microfacet Distribution

\( \vec{\omega}_i \)

\( \vec{\omega}_o \)
Microfacet Distribution

How much of the surface reflects?

\[ \vec{\omega}_i \]

\[ \vec{\omega}_o \]
What fraction of the surface participates in the reflection?

1) difficult to say (need an actual micro surface to compute this, tedious..)

2) Solve using principles of statistical physics

  - Is there anything general we can say about the surface when there are many bumps?
Microfacet Distribution

Fraction of facets facing each direction

Probability density function over projected solid angle (must be normalized):

$$\int_{H^2} D(\omega_h) \cos \theta_h d\omega_h = 1$$
Beckmann-Spizzichino Model

The slopes follow a Gaussian distribution

Let's express slope distribution w.r.t. directions

$$D(\mathbf{\omega}_h) = \exp \left( -\frac{\tan^2 \theta_h}{\alpha^2} \right)$$
Beckmann-Spizzichino Model

The slopes follow a Gaussian distribution

Let's express slope distribution w.r.t. directions

\[
D(\vec{\omega}_h) = \frac{1}{\pi \alpha^2 \cos^4 \theta_h} \exp -\frac{\tan^2 \theta_h}{\alpha^2}
\]
General Microfacet Model

\[ f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|\vec{\omega}_i \cdot \vec{n})(\vec{\omega}_o \cdot \vec{n})|} \]

shadowing/masking

\[ \vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{|| \vec{\omega}_i + \vec{\omega}_o ||} \]
Microfacet Distribution: Masking effect
Microfacet Distribution: Masking effect
Microfacet Distribution: Masking effect
Microfacet Distribution: Masking effect
Microfacet Distribution: Masking effect

The microfacet of interest not visible to the viewer due to occlusions

Masking effect:
Microfacet Distribution: Masking effect

The microfacet of interest not visible to the viewer due to occlusions

Masking effect:
Microfacet Distribution: Masking effect

The microfacet of interest not visible to the viewer due to occlusions

Masking effect:
Microfacet Distribution: Masking effect

The microfacet of interest not visible to the viewer due to occlusions

Masking effect:
Microfacet Distribution: Shadowing effect

Light does not reach the microfacet

Shadowing effect:
Microfacet Distribution: Shadowing/Masking

Light bounces among the facets before reaching the viewer
Microfacet Distribution: Shadowing/Masking

Light bounces among the facets before reaching the viewer
Microfacet Distribution: Shadowing/Masking

Light bounces among the facets before reaching the viewer
Microfacet Distribution: Interreflection

Light bounces among the facets before reaching the viewer

Interreflection
Microfacet Distribution: Interreflection

Light bounces among the facets before reaching the viewer
Reading

- PBRT Section 8.4
- GGX Distribution, Walter et al. (EGSR 2007)
- Isotropic and anisotropic microfacet distributions
- Oren–Nayar model, a "directed-diffuse" microfacet model, with perfectly diffuse (rather than specular) microfacets.
- Ashikhmin-Shirley model, allowing for anisotropic reflectance, along with a diffuse substrate under a specular surface
Isotropic microfacet distribution

Anisotropic microfacet distribution
Acknowledgements

Slides material borrowed from multiple resources.

Special thanks to Wojciech Jarosz for making his rendering lectures available online.