Lecture 16:

Global Illumination 1

Computer Graphics and Imaging
UC Berkeley CS184/284A
Direct lighting + recursive specular reflection / refraction
+ random sampling of area light

Image credit: Henrik Wann Jensen
+ all multiple bounces of light = Global Illumination
Cornell Box – Photograph vs Rendering

Photograph (CCD) vs. global illumination rendering
Visual Richness from Complex Lighting

Point Light

Environment Map Lighting
Visual Richness from Indirect Lighting
Visual Richness from Complex Materials

Credit: Bertrand Benoit. “Sweet Feast,” 2009. [Blender /VRay]
Ray Tracer Samples Radiance Along A Ray

The light entering the pixel is the sum total of the light reflected off the surface into the ray’s (reverse) direction.
Mini-Intro To Material Reflection
(Two full lectures next week by Lingqi)
Reflection

Definition: reflection is the process by which light incident on a surface interacts with the surface such that it leaves on the incident (same) side without change in frequency
Categories of Reflection Functions

Ideal specular

- Perfect mirror reflection

Ideal diffuse

- Equal reflection in all directions

Glossy specular

- Majority of light reflected near mirror direction

Retro-reflective

- Light reflected back towards light source

Diagrams illustrate how light from incoming direction is reflected in various outgoing directions.
Materials: Mirror
Materials: Diffuse
Materials: Gold
Materials: Plastic
Materials: Red Semi-Gloss Paint
Materials: Ford Mystic Lacquer Paint
Reflection at a Point

Differential irradiance incoming: \( dE(\omega_i) = L(\omega_i) \cos \theta_i \, d\omega_i \)

Differential radiance exiting (due to \( dE(\omega_i) \)): \( dL_r(\omega_r) \)
**BRDF**

Definition: The bidirectional reflectance distribution function (BRDF) represents how much light is reflected into each outgoing direction $\omega_r$ from each incoming direction $\omega_i$.

\[ f_r(\omega_i \rightarrow \omega_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)} = \frac{dL_r(\omega_r)}{L_i(\omega_i) \cos \theta_i \, d\omega_i} \left[ \frac{1}{\text{sr}} \right] \]

NB: $\omega_i$ points away from surface rather than into surface, by convention.
The Reflection Equation

\[ L_r(x, \omega_r) = \int_{H^2} f_r(p, \omega_i \rightarrow \omega_r) L_i(p, \omega_i) \cos \theta_i \, d\omega_i \]
Solving the Reflection Equation

\[ L_r(p, \omega_r) = \int_{H^2} f_r(p, \omega_i \rightarrow \omega_r) L_i(p, \omega_i) \cos \theta_i \, d\omega_i \]

Monte Carlo estimate:

- Generate directions \( \omega_j \) sampled from some distribution \( p(\omega) \)
- Choices for \( p(\omega) \)
  - Uniformly sample hemisphere
  - Importance sample BRDF (proportional to BRDF)
  - Importance sample lights (sample position on lights)
- Compute the estimator

\[
\frac{1}{N} \sum_{j=1}^{N} \frac{f_r(p, \omega_j \rightarrow \omega_r) L_i(p, \omega_j) \cos \theta_j}{p(\omega_j)}
\]
Recall: Hemisphere vs Light Sampling

Sample hemisphere uniformly

Sample points on light
DirectLightingPseudocode (Uniform Random Sampling)

DirectLightingSampleUniform(x, wo)
    wi = hemisphere.sampleUniform(); // uniform random sampling
    pdf = 1.0 / (2 * pi);

    if (scene.shadowIntersection(x, wi)) // Shadow ray
        return 0;
    else
        L = lights.radiance(intersect(x,wi), -wi);
        return L * x.brdf(wi, wo) * costheta / pdf;
DirectLightingSampleBRDF(x, wo)

\[\omega_i, pdf = x.brdf.sampleDirection(); \quad // \text{Imp. Sample BRDF}\]

\text{if} (\text{scene.shadowIntersection}(x, \omega_i)) \quad // \text{Shadow ray}
\begin{align*}
\text{return } 0;
\end{align*}
\text{else}
\begin{align*}
L &= \text{lights.radiance(intersect}(x, \omega_i), -\omega_i); \\
\text{return } L \times x.brdf(\omega_i, \omega_o) \times \text{costheta} / pdf;
\end{align*}
Direct Lighting Pseudocode (Importance Sampling of Lights)

DirectLightingSampleLights(x, ω₀)
    L, wi, pdf = lights.sampleDirection(x);  // Imp. sampl lights
    if (scene.shadowIntersection(x, wi))  // Shadow ray
        return 0;
    else
        return L * x.brdf(wi, ω₀) * costheta / pdf;

// Note: only one random sample over all lights.
// Assignment 3-1 asks you to, alternatively, loop over
// multiple lights and take multiple samples
Global Illumination
Deriving the Rendering Equation
Recall: Reflection Equation

\[ L_r(p, \omega_r) = \int_{H^2} f_r(p, \omega_i \rightarrow \omega_r) L_i(p, \omega_i) \cos \theta_i \, d\omega_i \]
Challenge: This is Actually A Recursive Equation

Reflected radiance depends on incoming radiance

$$L_r(p, \omega_r) = \int_{H^2} f_r(p, \omega_i \rightarrow \omega_r) L_i(p, \omega_i) \cos \theta_i \, d\omega_i$$

But incoming radiance depends on reflected radiance (at another point in the scene)
Transport Function & Radiance Invariance

Definition: the Transport Function, $tr(p, \omega)$, returns the first surface intersection point in the scene along ray $(p, \omega)$

Radiance invariance along rays: $L_o(tr(p, \omega_i), -\omega_i) = L_i(p, \omega_i)$

“Radiance arriving at $p$ from direction $\omega_i$ is equal to the radiance leaving $p'$ in direction $-\omega_i$”
The Rendering Equation

Re-write the reflection equation:

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

Using the transport function:

\[ L_i(p, \omega_i) = L_o(tr(p, \omega_i), -\omega_i) \]

The Rendering Equation

\[ L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{H^2} f_r(p, \omega_i \rightarrow \omega_o) L_o(tr(p, \omega_i), -\omega_i) \cos \theta_i \, d\omega_i \]

Note: recursion is now explicit

How to solve?
Light Transport Operators
Operators Are Higher-Order Functions

Functions:

\[ f, g : (x, \omega) \rightarrow \mathbb{R} \]

Operators are higher-order functions:

\[ P : ((x, \omega) \rightarrow \mathbb{R}) \rightarrow ((x, \omega) \rightarrow \mathbb{R}) \]

\[ P(f) = g \]

- Take a function and transform it into another function
Linear Operators

• Linear operators act on functions like matrices act on vectors

\[ h(x) = (L(f))(x) \]

• They are linear in that:

\[ L(af + bg) = aL(f) + bL(g) \]

• Examples of linear operators:

\[ H(f)(x) = \int h(x, x') f(x') \, dx' \]

\[ D(f)(x) = \frac{\delta f}{\delta x}(x) \]
**Light Transport Functions & Operators**

- **Emitted radiance function**
  (all surface points & outgoing directions)

- **Incoming/outgoing reflected radiance**
  (all surface points & in/out directions)

- **Transport function** - returns the first scene intersection point along given ray

\[ R(g)(p, \omega_o) \equiv \int_{H^2} f_r(p, \omega_i \rightarrow \omega_o) g(p, \omega_i) \cos \theta_i \, d\omega_i \]

\[ R(L_i) = L_o \]

- **Transport operator**:

\[ T(f)(p, \omega_o) \equiv f(tr(p, \omega), -\omega) \]

\[ T(L_o) = L_i \]
Reflection Operator

Incoming radiance (surface light field)

\[ R(g)(p, \omega_o) \equiv \int_{H^2} f_r(p, \omega_i \rightarrow \omega_o) g(p, \omega_i) \cos \theta_i \, d\omega_i \]

Outgoing radiance (surface light field)

\[ L_i \xrightarrow{R} L_o \]
Transport Operator

Outgoing radiance (surface light field)

\[ T(f)(p, \omega_o) \equiv f(tr(p, \omega), -\omega) \]

Incoming radiance (surface light field)
Define full one-bounce light transport operator: \[ \mathbf{K} = \mathbf{R} \circ \mathbf{T} \]

\[ L_0(p, \omega_o) = L_e(p, \omega_o) + \int_{H^2} f_r(p, \omega_i \to \omega_o) \mathbf{L}_o(tr(p, \omega_i), -\omega_i) \cos \theta_i \, d\omega_i \]

\[ L_0 = L_e + (R \circ T)(L_0) \]
To Be Continued
Acknowledgments

Thanks to Matt Pharr, Pat Hanrahan and Kayvon Fatahalian for many of these slides. Thanks to Ben Mildenhall for suggestions to improve many slides.