Lecture 16:

Global Illumination 1

Computer Graphics and Imaging
UC Berkeley CS184/284A, Spring 2017
Direct illumination + reflection + refraction

Image credit: Henrik Wann Jensen
Global illumination

Image credit: Henrik Wann Jensen
Cornell Box – Photograph vs Rendering

Photograph (CCD) vs. 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]
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:

\[ dE(\omega_i) = dL(\omega_i) \cos \theta_i \]

Differential radiance exiting (due to \( dE(\omega_i) \)):

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

$$dL_r(x, \omega_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)} = \frac{dL_r(\omega_r)}{dL_i(\omega_i) \cos \theta_i} \left[ \frac{1}{\text{sr}} \right]$$

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

\[ L_r(p, \omega_r) = \int_{H^2} f_r(p, \omega_i \rightarrow \omega_r) L_i(p, \omega_i) \cos \theta_i \, d\omega_i \]
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 \( p(\omega) \)
- To reduce variance \( p(\omega) \) should match BRDF (easier) or incident radiance function
- Compute the estimator

\[
\frac{1}{N} \sum_{j=1}^{N} \frac{f_r(p, \omega_j \rightarrow \omega_r) \, L_i(p, \omega_j) \, \cos \theta_j}{p(\omega_j)}
\]

CS184/284A

Ren Ng
/ Assume:
/ Ray ray hits surface at point hit_p
/ Normal of surface at hit point is hit_n

Vector3D wr = -ray.d; // outgoing direction
Spectrum Lr = 0.;
for (int i = 0; i < N; ++i) {
    Vector3D wi; // sample incident light from this direction
    float pdf; // p(wi)

    generate_sample(brdf, &wi, &pdf); // generate sample according to brdf

    Spectrum f = brdf->f(wr, wi);
    Spectrum Li = trace_ray(Ray(hit_p, wi)); // compute incoming Li
    Lr += f * Li * fabs(dot(wi, hit_n)) / pdf;
} return Lr / N;
Global Illumination
Deriving the Rendering Equation
Recall: Reflection Equation

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

Reflected radiance depends on incoming radiance

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

Incoming radiance depends on the reflected radiance (at another point in the scene)
Acknowledgments

Thanks to Matt Pharr, Pat Hanrahan and Kayvon Fatahalian for these slides. Thanks also to Steve Marschner for the path tracer code progression sequence.

Thanks to Weilun Sun for rendering the Cornell Box with successive bounces of light.

Thanks to Ben Mildenhall for suggestions to improve many slides.