Irradiance Caching & Photon Mapping

Raytracing & Epsilon

Solution: advance the ray start position \( \epsilon \) distance along the ray direction OR ignore all intersections \( < \epsilon \) (rather than \( < 0 \))

What’s a good value for \( \epsilon \)? Depends on hardware precision & scene dimensions

Image from Zachary Lynn
Last Time?

\[ L(x', \omega') = E(x', \omega') + \int \rho_{s}(\omega, \omega') L(x, \omega) G(x, x') V(x, x') \, dA \]

- Rendering Equation
- Monte-Carlo Integration
- Monte Carlo Rendering
- Forward Ray Tracing
- Stratified Sampling

Today

- Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Ray Grammar
Monte-Carlo Ray Tracing

- Cast a ray from the eye through each pixel
- Cast random rays to accumulate radiance contribution
  - Recurse to solve the Rendering Equation

Should also systematically sample the primary light

Monte Carlo Path Tracing

- Trace only one secondary ray per recursion
- But send many primary rays per pixel
  (performs antialiasing as well)
Ray Tracing vs Path Tracing

2 bounces
5 glossy samples
5 shadow samples

How many rays cast per pixel?

1 main ray + 5 shadow rays +
5 glossy rays + 5x5 shadow rays +
5*5 glossy rays + 5x5x5 shadow rays
= 186 rays

How many 3 bounce paths can we trace per pixel for the same cost?

186 rays / 8 ray casts per path
= ~23 paths

Which will probably have less error?

Path Tracing is costly

• Needs tons of rays per pixel
Indirect Illumination: smooth

- The indirect illumination is smooth
- Store the indirect illumination

**Irradiance Cache**

- The indirect illumination is smooth
- Store the indirect illumination
Irradiance Cache

- Interpolate nearby cached values
- But do full calculation for direct lighting
Questions?

• Why do we need “good” random numbers?
  – With a fixed random sequence, we see the structure in the error

Today

• Monte-Carlo Ray Tracing vs. Path Tracing
• Irradiance Caching
• Photon Mapping
• Ray Grammar
Photon Mapping

• Preprocess: cast rays from light sources
  – independent of viewpoint

Photon Mapping

• Store photons
  – position + light power + incoming direction
Photon Map

- Efficiently store photons for fast access
- Use hierarchical spatial structure (kd-tree)

Rendering with Photon Map

- Cast primary rays
- For secondary rays
  - reconstruct irradiance using k closest photons
- Combine with irradiance caching and other techniques
Photon Map Results

Photon Mapping - Caustics

- Special photon map for specular reflection and refraction
Comparison

Path Tracing
1000 paths/pixel

Photon mapping

Closest Photon Details

• Find the tightest sphere that captures $k$ photons
  – NOTE: HW3 code gives you all photons that *might* be in the query bounding box
    (you need to test for exact box and/or exact sphere)
• Divide the energy from those photons by the surface area covered by that sphere
• What about thin surfaces, concave corners, & convex corners?
Today

- Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Ray Grammar

Ray Grammar

- Classify local interaction:
  - E = eye
  - L = light
  - S = perfect specular reflection or refraction
  - G = glossy scattering
  - D = diffuse scattering

From Dutre et al.’s slides
Classic Ray Casting/Tracing

Ray casting: L D E

Ray tracing: L D S* E

“Adaptive Radiosity Textures for Bi-directional Ray Tracing”
Heckbert SIGGRAPH 1990

Photon Tracing

Radiosity: L D* E

Caustics: L S* D E (or worse!)

“Adaptive Radiosity Textures for Bi-directional Ray Tracing”
Heckbert SIGGRAPH 1990
Questions?

Readings for Tuesday (after break!)


Announcements: Final Projects

• *Everyone* should post one or more ideas for a final project on LMS (*“due”* Monday 3/30 @ 11:59pm)

• Connect with potential teammates (teams of 2 strongly recommended)

• Start finding & reading background papers

• Proposal & summary of background research are due April 13th (see webpage for details on proposal/project requirements)