Ray Tracing

Jehee Lee

Seoul National University

With lots of slides stolen from Doug James, Steve Seitz, Shree Nayar, Alexei Efros, Fredo Durand and others
Local vs. Global Illumination Models

• Local illumination models
  – Object illuminations are independent
  – No light scattering between objects
  – No real shadows, reflection, transmission

• Global illumination models
  – Ray tracing (highlights, reflection, transmission)
  – Radiosity (Surface interreflections)
  – Photon mapping
Forward Ray Tracing

- Rays as paths of photons in world space
- Follow photon from light sources to viewer
- Problem: Many rays will not contribute to image
Backward Ray Tracing

- Trace rays backward from viewer to light sources
- One ray from center of projection through each pixel in image plane
- Ray casting
  - Simplest form of ray tracing
  - No recursion
Backward Ray Tracing

- Illumination
  - Phong illumination
  - Shadow rays
  - Specular reflection
  - Specular refraction

- Specular reflection and refraction are recursive
Shadow Rays

- Determine if light “really” hits surface point
- Cast shadow ray from surface point to light
- If shadow ray hits opaque object, no contribution
Specular Refraction (Snell’s law)

\[
\sin \theta_r = \frac{\eta_i}{\eta_r} \sin \theta_i
\]

\(\eta_i, \eta_r\) : index of refraction of each material
(averaged over wavelengths and temperature)
Specular Refraction

- Path shifts are ignored for thin objects
- From Snell’s law, we can obtain the unit transmission vector $T$ in the direction $\theta_r$

\[
\cos \theta_r = \sqrt{1 - \left(\frac{\eta_i}{\eta_r}\right)^2 (1 - \cos^2 \theta_i)}
\]

\[
T = \left(\frac{\eta_i}{\eta_r} \cos \theta_i - \cos \theta_r\right)N - \frac{\eta_i}{\eta_r}L
\]
Interpolated transparency

\[ I = (1 - k)I_1 + kI_2 \]

\( k \) : transmission coefficient
- 0 for opaque objects,
- 1 for totally transparent objects
Binary Ray-Tracing Tree
Ray-surface Intersections

• Specialized algorithm for most commonly occurring shapes
  – Sphere
  – Polygon
  – Quadric
  – Splines

• Many shapes are represented in either implicit or parametric form
Ray-Implicit Surface Intersections

- Parametric ray equation \( \mathbf{P} = \mathbf{P}_0 + s \mathbf{u} \)
  - Initial position \( \mathbf{P}_0 \)
  - Unit direction vector \( \mathbf{u} \)

- Implicit surface
  - Consists of all points \( \mathbf{P} \) such that \( f(\mathbf{P}) = 0 \)
  - Substitute ray equation for \( \mathbf{P} \)
    \[
    f(\mathbf{P}_0 + s \mathbf{u}) = 0
    \]
  - Solve for \( s \) (univariate root finding)
Ray-Sphere Intersections

• Sphere equation \[ \left\| \mathbf{P} - \mathbf{P}_c \right\|^2 - r^2 = 0 \]

• Substitution \[ \left\| \mathbf{P}_0 + s \mathbf{u} - \mathbf{P}_c \right\|^2 - r^2 = 0 \]

• Letting \[ \Delta \mathbf{P} = \mathbf{P}_c - \mathbf{P}_0 \]

• Solution \[ s^2 - 2(\mathbf{u} \cdot \Delta \mathbf{P})s + (\left\| \Delta \mathbf{P} \right\|^2 - r^2) = 0 \]
Ray-Polygon Intersections

- Bounding sphere and back face culling is useful

- Ray-Plane Intersections
  - Plane equation containing the polygon
    \[ Ax + By + Cz + D = N \cdot P + D = 0 \]
  - Substitution
    \[ N \cdot (P_0 + su) + D = 0 \]
  - Solution
    \[ s = -\frac{D + N \cdot P_0}{N \cdot u} \]

- Perform inside-outside test to determine whether the intersection is inside the polygon
Acceleration Techniques

- Space-subdivision
  - Uniform subdivision
  - Adaptive subdivision (Octrees)
  - BSP trees

- Ray classification
  - Classify 5D ray space
  - Can be reduced to 4D ray space
How do we see the world?

- In computer graphics, we assumed that rays are projected onto the image plane.
How do we see the world?

- In physics, that is not true
- Rays are scattered in all directions
Pinhole camera

• Add a barrier to block off most of the rays
  – This reduces blurring
  – The opening known as the aperture
  – How does this transform the image?
Pinhole camera model

- Pinhole model:
  - Captures **pencil of rays** – all rays through a single point
  - The point is called **Center of Projection (COP)**
  - The image is formed on the **Image Plane**
  - **Effective focal length** $f$ is distance from COP to Image Plane
Camera Obscura

- The first camera
  - Known to Aristotle
  - Depth of the room is the effective focal length
Home-made pinhole camera

Why so blurry?

http://www.debevec.org/Pinhole/
Shrinking the aperture

- Why not make the aperture as small as possible?
  - Less light gets through
  - Diffraction effects…
Shrinking the aperture
The reason for lenses
Small vs. Large Pinholes

Photograph made with small pinhole

Photograph made with larger pinhole
Pinhole vs. Lens

Photograph made with small pinhole

Photograph made with lens
Ideal Lens: Same projection as pinhole but gathers more light!

Lens Formula: \[ \frac{1}{i} + \frac{1}{o} = \frac{1}{f} \]

- \( f \) is the focal length of the lens – determines the lens’s ability to bend (refract) light
- \( f \) different from the effective focal length \( f \) discussed before!
Aperture

The size of the lens opening—the aperture or f-stop—controls the amount of light that passes through the lens. The lens shown here has apertures from f/2.8 to f/22. Each setting is one stop from the next; that is, each lets in twice as much light as the next smaller opening, half as much light as the next larger opening.

The higher the f-stop number, the smaller the lens opening and the less light that is let in. On this lens, f/2.8 is the largest opening and lets in the most light. As the numbers get bigger (4, 5.6, 8), the aperture size gets smaller and the amount of light admitted decreases.

**NOTE:** Some lens barrels no longer display the apertures as shown above. Rather than twisting a ring on the lens to set the aperture, you dial in the setting on the camera body. Regardless of how you change the aperture, though, you are still changing the size of the lens opening and thus the intensity of light that strikes your film or computer chip.
Equivalent exposures. Each combination here of f-stop and shutter speed produces the equivalent exposure (lets in the same amount of light) but produces differences in depth of field and motion.
Focus and Defocus

- A lens focuses light onto the film
  - There is a specific distance at which objects are “in focus”
  - other points project to a “circle of confusion” in the image
  - The diameter is
    \[ 2r_c = \frac{|d' - d| f}{n d} \]
  - How can we change focus distance?

“circle of confusion”
Depth of Field

http://www.cambridgeincolour.com/tutorials/depth-of-field.htm
Changing the aperture size affects depth of field

- A smaller aperture increases the range in which the object is approximately in focus
- But small aperture reduces amount of light – need to increase exposure
Depth-of-Field Scale

- Typical prime lens design
Varying the aperture

LARGE APERTURE, LESS DEPTH OF FIELD
Varying the aperture

**SMALL APERTURE, MORE DEPTH OF FIELD**

- **a**
- **b**
- **c**

Depth of field: 8 feet
Nice Depth of Field effect
Distributed Ray Tracing

- Stochastic sampling that randomly distribute rays according to the various parameters

- Monte Carlo evaluation of the multiple integrals that occur in an accurate physical description of surface lighting
Distributed Ray Tracing

- Antialiasing
  - Oversampling rays in each pixel
  - Regular vs. jittered sampling
Illumination (glossy, diffuse, translucency)
Soft shadow (distributing shadow rays)
Distributed Ray Tracing

• Depth of Field
  – Distributing rays over the circle of confusion
• Motion blur
  – Distributing rays over time
Caustics

- Caustics represents some of the most visually striking patterns of light in nature
- Caustics are formed by light that is reflected or transmitted by a number of specular surfaces before interacting with a diffuse surface
- Examples of caustics are the light patterns on the bottom of a swimming pool and light focused onto a table through a glass of cognac.
Caustics
Translucency

Jensen et al.,
SIGGRAPH 2001
(Photon Mapping)

11/7/2003
Programming Assignment #5

- Simple ray tracer
  - You are required to implement a simple ray tracer

- Required features
  - Ray tracing spheres [10 points]
  - Ray tracing polygons [10 points]
  - Recursive reflection [10 points]
  - Recursive refraction [10 points]
  - Phong illumination [10 points]
  - Importing geometry files such as OBJ [5 points]
  - Export image files [5 points]
  - Texture mapped spheres and polygons [10 points]
  - Report [15 points]
  - Representative pictures [15 points]
Programming Assignment #5

- Hand in all “nice” images you generated

- Your report should explain
  - What features you implemented
  - Which image is demonstrating which features
  - Instructions to render submitted images

- Features that are not demonstrated in images will receive little or no credit
Programming Assignment #5

- Extra features (up to 30 points)
  - Distributed ray tracing
    - Soft shadows, depth of field, and motion blur
  - Spatial partitioning
    - Uniform cell subdivision, octrees, or BSP trees
    - Need to demonstrate performance improvements
  - Bump mapping