Ray Tracing

Recursive Ray Tracing
Gather light from various directions by tracing rays
Each pixel shows light at a surface
  • trace ray from eye to surface
Each surface illuminated by lights and other surfaces
  • trace rays from surface to other surfaces
And so on...

Types of Rays
Eye/pixel rays
Illumination/shadow rays
Reflection rays
Transmission/transparency rays

Eye Rays
Same as in ray casting
Effectively determine visible surfaces
For perspective view, trace from eye through pixel
For orthogonal view, trace parallel rays through pixels in direction of projection
Stop at nearest intersection with surface

Illumination Rays
From surface point towards light source
Intervening surfaces may block or attenuate direct illumination
Light reaching surface applied using local illumination model

Reflection Rays
Gather non-local illumination reflecting (specularly) towards eye (incident ray)
From current point towards reflection direction
Contribute illumination from closest surface intersection
Assume perfect (sharp) specular reflection
Attenuated by specular coefficient
Transmission Rays

- Gather light transmitted through current surface
- Incident ray refracted according to index of refraction and Snell’s law
- May use a single index of refraction (rather than per wavelength)
- Attenuated by transmission coefficient

Trace Algorithm

- **Trace(ray)**
  - Foreach object in scene
  - Intersect(ray, object)
  - If no intersections, return BackgroundColor
  - For each light:
    - Foreach object in scene
    - Intersect(ShadowRay, object)
    - Accumulate local illumination
    - Trace(ReflectionRay)
    - Trace(TransmissionRay)
  - Accumulate global illumination
- Return illumination

No Recursion

1 Bounce (level of recursion)

2 bounces

What’s different here?
**Index of refraction > 1**

**Index of refraction < 1**

*Notice total internal reflection*

**Spheres and Checkerboard**

*Turned Whitted, 1980 (Foley/vanDam III.10)*

**Computing Reflection Ray**

\[
\begin{align*}
R &= (-\mathbf{N})\mathbf{N} + I + (-\mathbf{N})\mathbf{N} = I - 2(\mathbf{N})\mathbf{N}
\end{align*}
\]

**Computing Transmission Ray**

*Given N, I, \( \eta_1 \), and \( \eta_2 \)*

*M is unit vector perpendicular to N*

*I, M, and T all lie in same plane*

*Use Snell's law plus trigonometry to compute T...*

\[
\begin{align*}
\eta_{12} &= \eta_1 / \eta_2 = \sin \theta_2 / \sin \theta_1 \\
T &= (\sin \theta_1 / \sin \theta_2)I - (\mathbf{N})\mathbf{N} - (\cos \theta_2)\mathbf{N}
\end{align*}
\]

*Snell's Law*

\[
\begin{align*}
\eta_{12} = \eta_1 / \eta_2 = \sin \theta_2 / \sin \theta_1 \\
T &= (\sin \theta_1 / \sin \theta_2)I - (\mathbf{N})\mathbf{N} - (\cos \theta_2)\mathbf{N}
\end{align*}
\]
Accumulating Light Contributions at a Point

\[ I = \sum \text{local illumination (attenuated illum. rays)} \]
+ attenuated reflection ray
+ attenuated transmission ray
Illumination rays attenuated by transmission coefficients of light ocluders
Reflection ray attenuated by specular coefficient
Transmission ray attenuated by trans. coefficient
All rays may be attenuated by \(1/r^2\)

Sampling Issues

Currently using only a single sample for each
- Pixel
- Reflection
- Transmission
- Frame time
- Eye point
All of these can cause forms of aliasing

Sampling Theory

Aliasing: a high frequency signal masquerading as a lower frequency
Nyquist limit: maximum frequency signal that may be adequately sampled
(1/2 sampling frequency)
Aliasing may occur when we take regularly-spaced samples of frequencies above the Nyquist limit

Examples of Aliasing

Reverse-rotating wagon wheels in films (temporal aliasing)
http://flowers.oftheneight.org/wagonWheel/wagonWheel.html

“Jaggies” on polygon edges or specular highlights (spatial aliasing)
Creeping effects (jaggies in motion)

Jaggies and Moire pattern

Aliasing Illustration

Figures by George Wolberg, from Foley, vanDam, et al., 
Computer Graphics: Principle and Practice, p. 627-628
**Anti-aliasing - Supersampling**

- Shoot multiple rays through each pixel
- Aim through centers of a regular grid (e.g. 3x3 or 4x4 grid of samples)
- Average resulting intensity values (box filter reconstruction)
- Reduces spatial aliasing effects

**Adaptive Supersampling**

- Sample at corners and center
- Add samples if gradient is more than some threshold
- Add samples only in necessary regions
- Reduces aliasing effects more efficiently
- Still may not sample adequately

**Jittering**

- Apply random perturbations to sample positions of uniform or adaptive grid
- Does not increase sampling rate, but removes regularity
- Converts aliasing to noise, which is perceptually more tolerable

**Aliasing vs. Noise**


**Distributed Ray Tracing**

- Apply distribution-based sampling to many parts of the ray-tracing algorithm
- Fuzzy reflection/transmission
  - perturb directions reflection/transmission, with distribution based on angle from ideal ray
- Motion blur
  - perturb eye ray samples in time
- Depth of field
  - perturb eye position on lens
- Fuzzy shadows/penumbra
  - sample illumination rays across area light

**Importance Sampling**

- Divide sample space into blocks of equal area under weighting function
- Assign each pixel sample a different (random) block
- Perturb randomly within block (ideally, points in block have nearly equal weight…)

possible block arrangement for reflection ray direction
**Motion Blur Illustration**


---

**Depth of Field Illustration**

Fig. 17. Example of depth of field from *Young Sherlock Holmes*. Copyright 1985, Paramount Pictures Corp. from *Young Sherlock Holmes*, Paramount Pictures Corp, 1985. Excerpted from *An Introduction to Ray Tracing*, Andrew Glassner, ed., p. 195

---

**Penumbras and Blurry Reflection**