

Ray Tracing

CSCI-6962 Advanced Computer Graphics Cutler

Last Time?

CSCI-6962 Advanced Computer Graphics Cutler

Quiz!

- Tuesday Oct 18th
- In class
- 1 page of notes

CSCI-6962 Advanced Computer Graphics Cutler

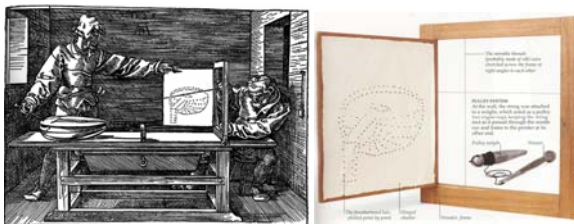
Today

- Ray Casting
 - Ray-Plane Intersection
 - Ray-Sphere Intersection
 - Point in Polygon
- Ray Tracing
 - Shadows
 - Reflection
 - Refraction
- Recursive Ray Tracing

CSCI-6962 Advanced Computer Graphics Cutler

Durer's Ray Casting Machine

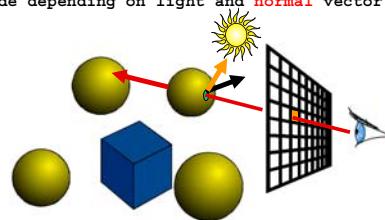
- Albrecht Durer, 16th century



CSCI-6962 Advanced Computer Graphics Cutler

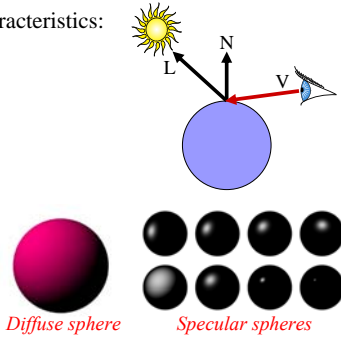
Ray Casting

For every pixel
Construct a ray from the eye
For every object in the scene
Find **intersection** with the ray
Keep if closest
Shade depending on light and **normal** vector



A Note on Shading

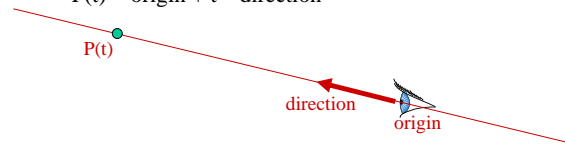
- Surface/Scene Characteristics:
 - surface normal
 - direction to light
 - viewpoint
- Material Properties
 - Diffuse (matte)
 - Specular (shiny)
 - ...
- More later!



CSCI-6962 Advanced Computer Graphics Cutler

Ray Representation?

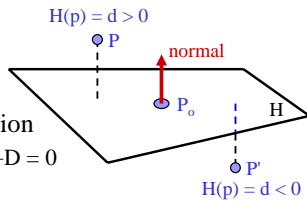
- Two vectors:
 - Origin
 - Direction (normalized is better)
- Parametric line (*explicit* representation)
 - $P(t) = \text{origin} + t * \text{direction}$



CSCI-6962 Advanced Computer Graphics Cutler

3D Plane Representation?

- Plane defined by
 - $P_o = (x, y, z)$
 - $n = (A, B, C)$
- Implicit plane equation
 - $H(P) = Ax + By + Cz + D = 0$
 - $= n \cdot P + D = 0$
- Point-Plane distance?
 - If n is normalized, distance to plane, $d = H(P)$
 - d is the *signed distance*!



CSCI-6962 Advanced Computer Graphics Cutler

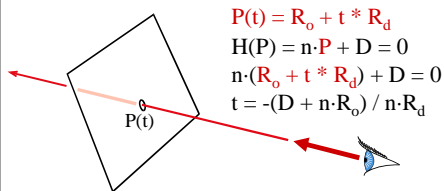
Explicit vs. Implicit?

- Ray equation is explicit $P(t) = R_o + t * R_d$
 - Parametric
 - Generates points
 - Hard to verify that a point is on the ray
- Plane equation is implicit $H(P) = n \cdot P + D = 0$
 - Solution of an equation
 - Does not generate points
 - Verifies that a point is on the plane
- Exercise: Explicit plane and implicit ray

CSCI-6962 Advanced Computer Graphics Cutler

Ray-Plane Intersection

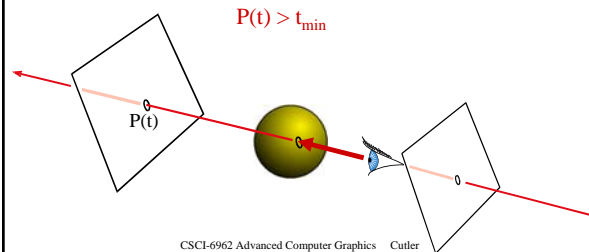
- Intersection means both are satisfied
- So, insert explicit equation of ray into implicit equation of plane & solve for t



CSCI-6962 Advanced Computer Graphics Cutler

Additional Housekeeping

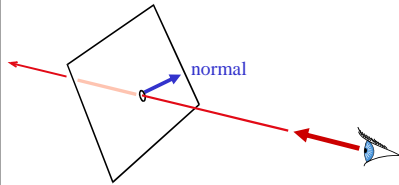
- Verify that intersection is closer than previous
 - $P(t) < t_{\text{current}}$
- Verify that it is not out of range (behind eye)
 - $P(t) > t_{\text{min}}$



CSCI-6962 Advanced Computer Graphics Cutler

Normal

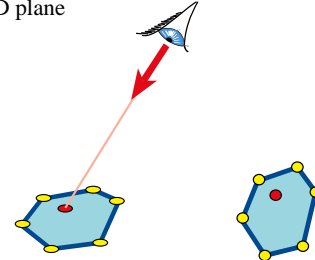
- For shading
 - diffuse: dot product between light and normal
- Normal is constant



CSCI-6962 Advanced Computer Graphics Cutler

Ray-Polygon Intersection

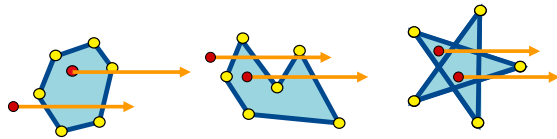
- Ray-plane intersection
- Test if intersection is in the polygon
 - Solve in the 2D plane



CSCI-6962 Advanced Computer Graphics Cutler

Point Inside/Outside Polygon

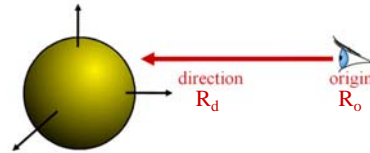
- Ray intersection definition:
 - Cast a ray in any direction
 - (axis-aligned is smarter)
 - Count intersections
 - If odd number, point is inside
- Works for concave and star-shaped
- Special case for triangle...



CSCI-6962 Advanced Computer Graphics Cutler

Sphere Representation?

- Implicit sphere equation
 - Assume centered at origin (easy to translate)
 - $H(P) = P \cdot P - r^2 = 0$



CSCI-6962 Advanced Computer Graphics Cutler

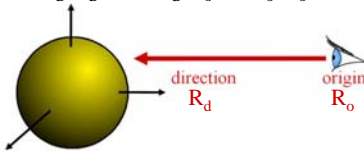
Ray-Sphere Intersection

- Insert explicit equation of ray into implicit equation of sphere & solve for t

$$P(t) = R_o + t \cdot R_d \quad H(P) = P \cdot P - r^2 = 0$$

$$(R_o + tR_d) \cdot (R_o + tR_d) - r^2 = 0$$

$$R_d \cdot R_d t^2 + 2R_d \cdot R_o t + R_o \cdot R_o - r^2 = 0$$



CSCI-6962 Advanced Computer Graphics Cutler

Ray-Sphere Intersection

- Quadratic: $at^2 + bt + c = 0$
 - $a = 1$ (remember, $\|R_d\| = 1$)
 - $b = 2R_d \cdot R_o$
 - $c = R_o \cdot R_o - r^2$
- with discriminant $d = \sqrt{b^2 - 4ac}$
- and solutions $t_{\pm} = \frac{-b \pm d}{2a}$

CSCI-6962 Advanced Computer Graphics Cutler

Questions?

CSCI-6962 Advanced Computer Graphics Cutler

Today

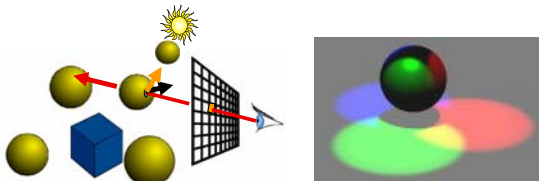
- Ray Casting
 - Ray-Plane Intersection
 - Ray-Sphere Intersection
 - Point in Polygon
- Ray Tracing
 - Shadows
 - Reflection
 - Refraction
- Recursive Ray Tracing

CSCI-6962 Advanced Computer Graphics Cutler

How Can We Add Shadows?

```

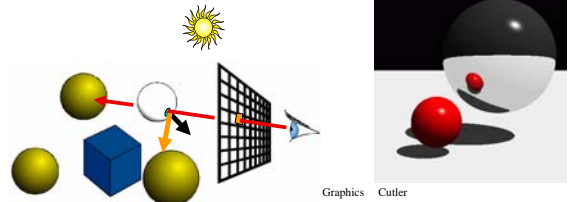
Find the point to be shaded
For every light,
  Construct ray from point to light
For every object
  find intersection of ray with object
  If no objects between point and light
    Add contribution from light
    
```



CSCI-6962 Advanced Computer Graphics Cutler

Mirror Reflection

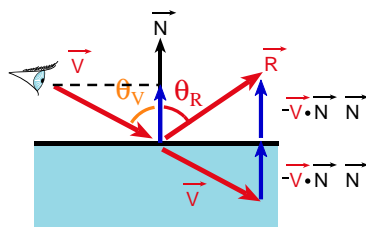
- Cast ray symmetric with respect to the normal
- Multiply by reflection coefficient (color)



Graphics Cutler

Reflection

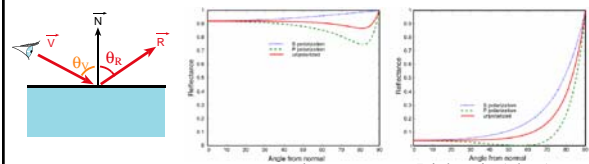
- Reflection angle = view angle
- $\mathbf{R} = \mathbf{V} - 2(\mathbf{V} \cdot \mathbf{N})\mathbf{N}$



CSCI-6962 Advanced Computer Graphics Cutler

Amount of Reflection

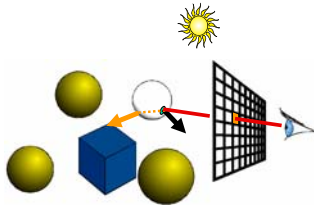
- Traditional ray tracing (hack)
 - Constant reflectionColor
- More realistic:
 - Fresnel reflection term (more reflection at grazing angle)
 - Schlick's approximation: $R(\theta) = R_0 + (1 - R_0)(1 - \cos \theta)^5$



CSCI-6962 Advanced Computer Graphics Cutler

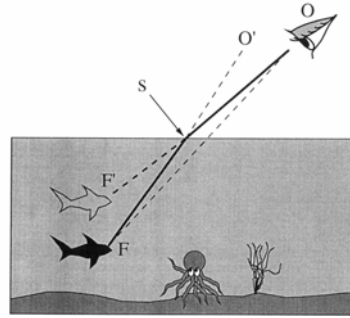
Transparency

- Cast ray in refracted direction
- Multiply by transparency coefficient (color)



CSCI-6962 Advanced Computer Graphics Cutler

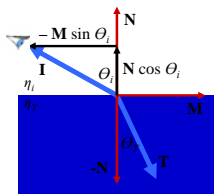
Qualitative Refraction



From "Color and Light in Nature" by Lynch and Livingston

CSCI-6962 Advanced Computer Graphics Cutler

Refraction



$$\begin{aligned} \mathbf{I} &= \mathbf{N} \cos \theta_i - \mathbf{M} \sin \theta_i \\ \mathbf{M} &= (\mathbf{N} \cos \theta_i - \mathbf{I}) / \sin \theta_i \\ \mathbf{T} &= -\mathbf{N} \cos \theta_r + \mathbf{M} \sin \theta_r \\ &= -\mathbf{N} \cos \theta_r + (\mathbf{N} \cos \theta_i - \mathbf{I}) \sin \theta_r / \sin \theta_i \\ &= -\mathbf{N} \cos \theta_r + (\mathbf{N} \cos \theta_i - \mathbf{I}) \eta_r \\ &= [\eta_r \cos \theta_i - \cos \theta_r] \mathbf{N} - \eta_r \mathbf{I} \\ &= [\eta_r \cos \theta_i - \sqrt{1 - \sin^2 \theta_r}] \mathbf{N} - \eta_r \mathbf{I} \\ &= [\eta_r \cos \theta_i - \sqrt{1 - \eta_r^2 \sin^2 \theta_i}] \mathbf{N} - \eta_r \mathbf{I} \\ &= [\eta_r \cos \theta_i - \sqrt{1 - \eta_r^2 (1 - \cos^2 \theta_i)}] \mathbf{N} - \eta_r \mathbf{I} \\ &= [\eta_r (\mathbf{N} \cdot \mathbf{I}) - \sqrt{1 - \eta_r^2 (1 - (\mathbf{N} \cdot \mathbf{I})^2)}] \mathbf{N} - \eta_r \mathbf{I} \end{aligned}$$

Snell-Descartes Law:
 $\eta_i \sin \theta_i = \eta_r \sin \theta_r$

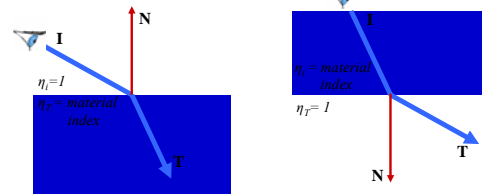
$$\frac{\sin \theta_r}{\sin \theta_i} = \frac{\eta_i}{\eta_r} = \eta_r$$

- Total internal reflection when the square root is imaginary
- Don't forget to normalize!

CSCI-6962 Advanced Computer Graphics Cutler

Refraction & the Sidedness of Objects

- Make sure you know whether you're entering or leaving the transmissive material:



- What about intersecting transparent objects?

CSCI-6962 Advanced Computer Graphics Cutler

Total Internal Reflection



Fig. 3.7A The optical manhole. From under water, the entire celestial hemisphere is compressed into a circle only 97.2° across. The dark boundary defining the edges of the manhole is not sharp due to surface waves. The rays are analogous to the cephalopod type seen in hazy air. Section 1.5. (Photo by B. Grainger)



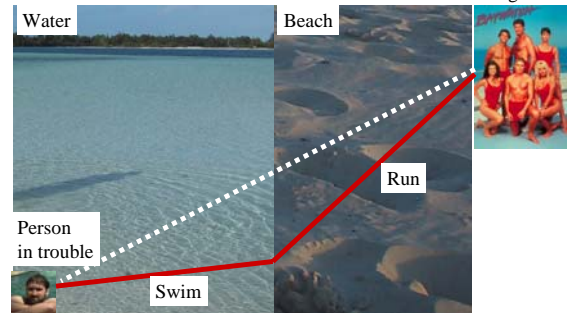
Fig. 3.7B The optical manhole. Light from the horizon (angle of incidence = 90°) is refracted downward at an angle of 48.6°. This compresses the sky into a circle with a diameter of 97.2° instead of its usual 180°.

From "Color and Light in Nature" by Lynch and Livingston

CSCI-6962 Advanced Computer Graphics Cutler

Refraction and the Lifeguard Problem

- Running is faster than swimming



CSCI-6962 Advanced Computer Graphics Cutler

Questions?

CSCI-6962 Advanced Computer Graphics Cutler

Today

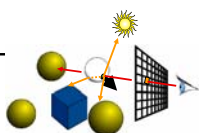
- Ray Casting
 - Ray-Plane Intersection
 - Ray-Sphere Intersection
 - Point in Polygon
- Ray Tracing
 - Shadows
 - Reflection
 - Refraction
- **Recursive Ray Tracing**

CSCI-6962 Advanced Computer Graphics Cutler

Recap: Ray Tracing

```

trace ray
Intersect all objects
color = ambient term
For every light
  cast shadow ray
  color += local shading term
If mirror
  color += color_refl *
  trace reflected ray
If transparent
  color += color_trans *
  trace transmitted ray
    
```



Stopping criteria:

- Recursion depth
 - Stop after a number of bounces
- Ray contribution
 - Stop if reflected/transmitted contribution becomes too small

- *Does it ever end?*

CSCI-6962 Advanced Computer Graphics Cutler

Recursion For Reflection



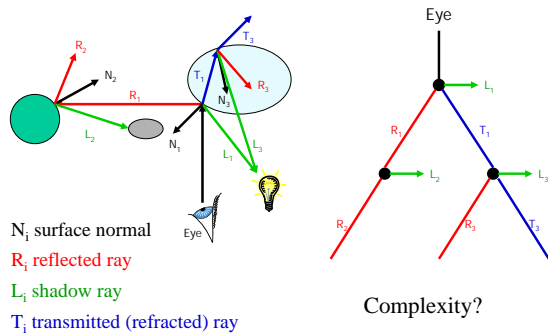
0 recursion

1 recursion

2 recursions

CSCI-6962 Advanced Computer Graphics Cutler

The Ray Tree

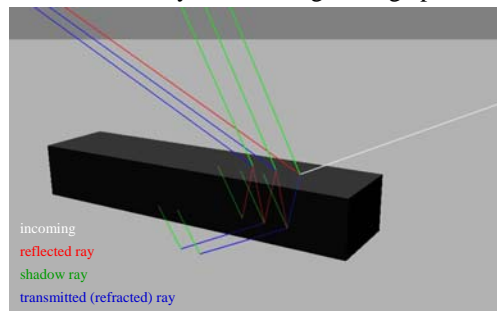


Complexity?

CSCI-6962 Advanced Computer Graphics Cutler

Ray Debugging

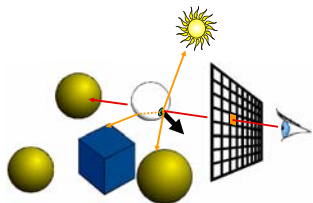
- Visualize the ray tree for single image pixel



CSCI-6962 Advanced Computer Graphics Cutler

Does Ray Tracing Simulate Physics?

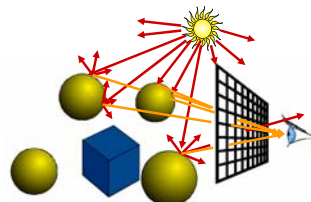
- Photons go from the light to the eye, not the other way
- What we do is *backward ray tracing*



CSCI-6962 Advanced Computer Graphics Cutler

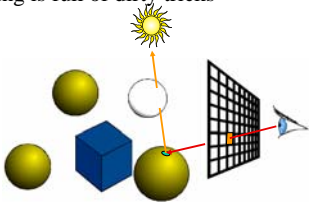
Forward Ray Tracing

- Start from the light source
 - But low probability to reach the eye
- What can we do about it?
 - Always send a ray to the eye.... still not efficient



Transparent Shadows?

- What to do if ray to light source intersects a transparent object?
 - Pretend it's opaque?
 - multiply by transparency color? (ignores refraction & does not produce caustics)
- Ray Tracing is full of dirty tricks



Traditional Ray Tracing?



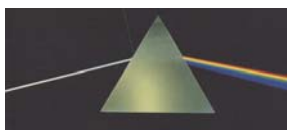
Images by Henrik Wann Jensen

- No, Refraction and complex reflection for illumination are not handled properly in traditional (backward) ray tracing

CSCI-6962 Advanced Computer Graphics Cutler

What makes a Rainbow?

- Refraction is wavelength-dependent
 - Refraction increases as the wavelength of light decreases
 - violet and blue experience more bending than orange and red
- Usually ignored in graphics
- Rainbow is caused by refraction + internal reflection + refraction



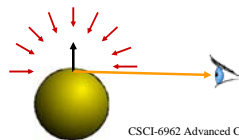
Pink Floyd, *The Dark Side of the Moon*

From "Color and Light in Nature" by Lynch and Livingstone



The Rendering Equation

- Clean mathematical framework for light-transport simulation
- At each point, outgoing **light in one direction** is the integral of **incoming light in all directions** multiplied by reflectance property



CSCI-6962 Advanced Computer Graphics Cutler

Questions?

CSCI-6962 Advanced Computer Graphics Cutler