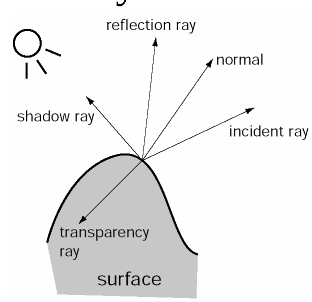
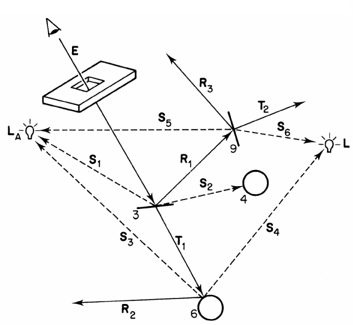
EECS 487 | **Kinds of rays** | Lecture 16

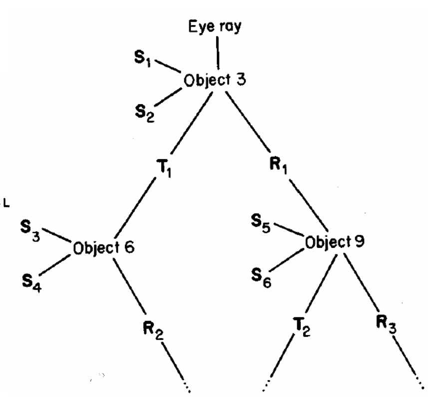
- Primary ray
  - leaves the eye and travels out to the scene
- When hit - spawn three new rays to “collect light”
  - shadow ray
    - towards lights
  - reflection ray
  - transparency ray



EECS 487 | **The ray tree** | Lecture 16

- so we have a tree.

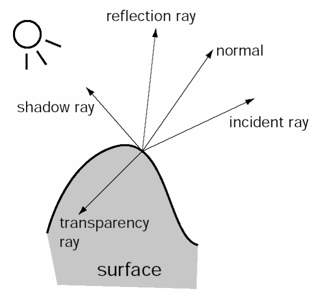




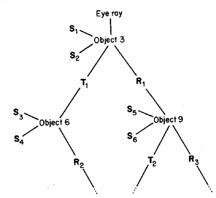
Raytracing is …

Lecture 16

- I(incident-out) =
  I(shadow-local-in)
  + Kr * I(reflection-in)
  + Kt * I(transparent-in)



  - Kr and Kt decrease with distance
  - Evaluate tree bottom-up

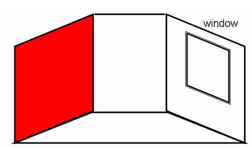


3/23/2004 3

Simple ray tracing is…

Lecture 16

- wrong



  - specular reflection
    - direct-indirect lighting inconsistency
  - aliasing
    - thin rays only work for continuous scenes
      - where are discontinuities?
- can fix all of these though…

3/23/2004 4

EECS 487 | Ray Tracing II | Lecture 16

Shadows

To get shadows in ray tracing, at every intersection we shoot a test ray toward each light source.

If the ray gets to the light source then this light is used in the lighting calculation.

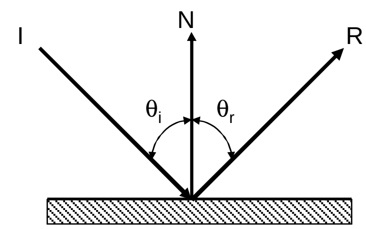
If the ray hits something then the light source is not used in the lighting calculation.

The ray direction used to test the light source can be used in the lighting calculation

3/23/2004 5

EECS 487 | Ray Tracing II | Lecture 16

Ray Reflections



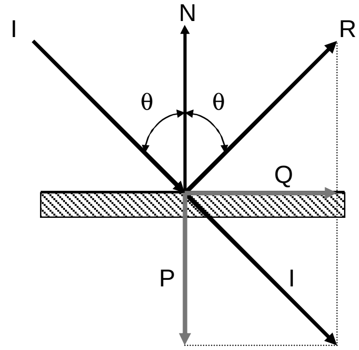
$$\theta_I = \theta_r$$

3/23/2004 6

EECS 487

Ray Tracing II

Lecture 16



Assume |N| = 1 (unit length)
Let P = projection of **I** on N
P = (**I**•N)N
Take Q = **I** – P
So **I** = Q + P

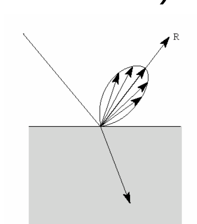
From the diagram:
R = Q – P
= **I** – 2P
= **I** - 2(**I**•N)N

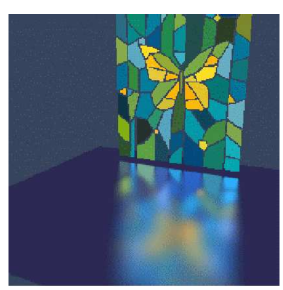
EECS 487

Distribution ray tracing

Lecture 16

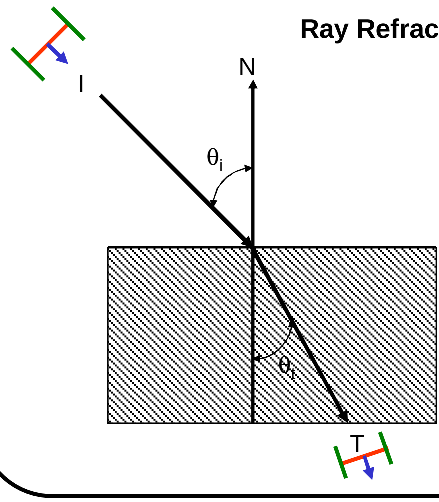
- Sample reflection directions based on a reflection model
  - BRDF (Bidirectional Reflection Distribution Function)





EECS 487 | **Ray Tracing II** | Lecture 16

Ray Refraction



Snell's Law:

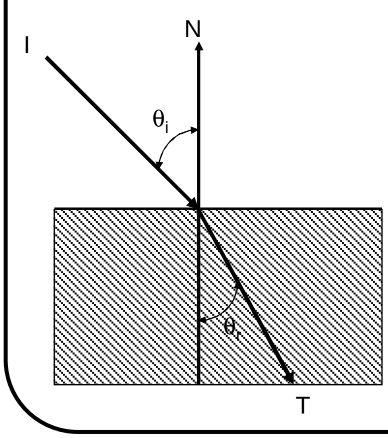
$$\sin\theta_t = \frac{\eta_i}{\eta_t}\sin\theta_i$$

η: index of refraction

ratio of speed of light in a vacuum to speed of light in the material

EECS 487 | **Ray Tracing II** | Lecture 16

Ray Refraction

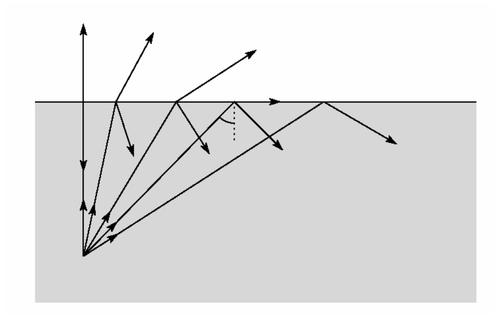


$$T = \left(\frac{\eta_i}{\eta_t}\cos\theta_i - \cos\theta_t\right)N + \frac{\eta_i}{\eta_t}I$$

(assuming I and N are unit length)

EECS 487 | **Watch out for…** | Lecture 16

- Total internal refraction
  – light may not get through the interface



3/23/2004 11

EECS 487 | **Ray Tracing II** | Lecture 16

Anti - Aliasing

- Sample More Rays
- Instead of Vectors use Truncated Cones, Rods, Pyramids
- Stochastic Sampling

3/23/2004 12

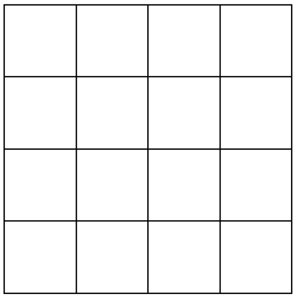
EECS 487

Ray Tracing II

Lecture 16

Anti - Aliasing

- Sample More Rays



EECS 487

Ray Tracing II

Lecture 16

Anti - Aliasing

- Instead of Vectors use Truncated Cones, Rods, Pyramids

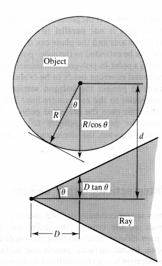


Figure 8.5 Ray/object intersection in cone tracing (a ray is a cone).

$$D \tan \theta + \frac{R}{\cos \theta} > d$$

where:

θ is the spread angle of the cone
R is the radius of the sphere
d is the distance between the point on the cone axis that is nearest to the centre of the sphere
D is the distance between this point on the cone axis and the cone origin

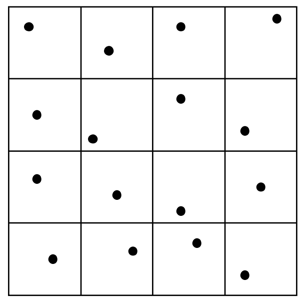
EECS 487

Ray Tracing II

Lecture 16

Anti - Aliasing

• Stochastic Sampling: very easy to do in ray tracing!



EECS 487

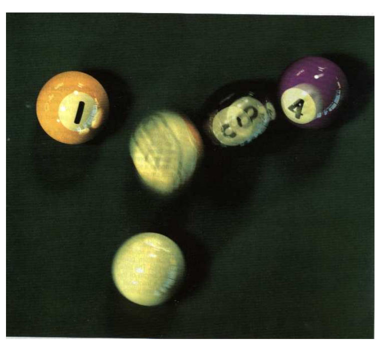
Ray Tracing II

Lecture 16

Temporal Anti - Aliasing

Motion Blur

Sub-pixel samples are stochastically sampled over time and space



EECS 487 | **Ray Tracing II** | Lecture 16

Forward Ray Tracing

This means tracing rays from the light source into the scene.

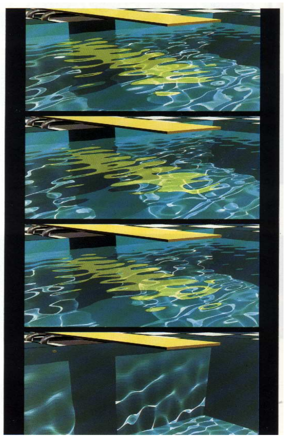
- This can be very expensive
- But adds effect that can only be found if this is done.
- One of the ways to get water caustics

Two passes:
- Trace from the light, depositing illumination on diffuse surfaces
- Trace from the eye, use info from the previous step

3/23/2004 17

EECS 487 | **Ray Tracing II** | Lecture 16

Forward Ray Tracing



3/23/2004 18

EECS 487 | **Depth of field** | Lecture 16

- Pinhole camera has everything in focus
- Real lenses used in real cameras have a finite depth of field
  - Area that is in focus
    - Smaller apertures (larger f-numbers result in greater depth of field)

3/23/2004 19

EECS 487 | **Depth of field** | Lecture 16

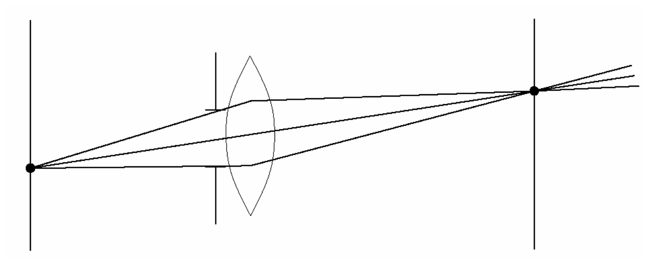
- Large aperture

- Small aperture

3/23/2004 20

EECS 487 | **Depth of field** | Lecture 16

- Jittersample then refract through a lens then proceed as normal



3/23/2004 21

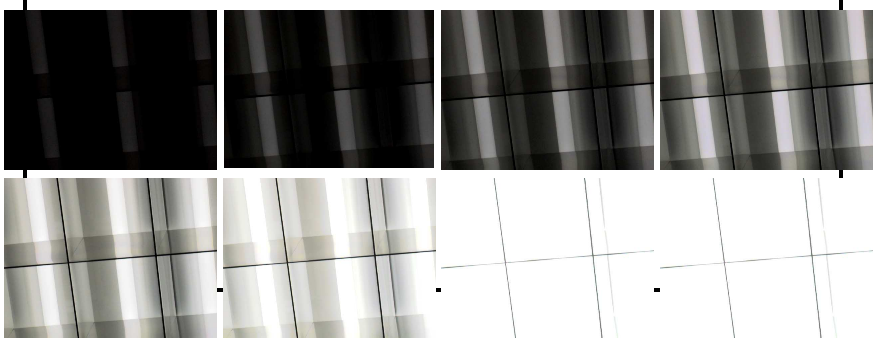
EECS 487 | **Depth of field** | Lecture 16

- Geri's Game from Pixar Inc.

3/23/2004 22

EECS 487 | **Natural Light** | Lecture 16

- HDR imaging
  - High dynamic range
  - Changing exposure from 1/4096 to 1/32 sec



EECS 487 | **HDR motion blur: debevec.org** | Lecture 16

8bit

HDR

3/23/2004

EECS 487

Why do the lights shine through?

Lecture 16

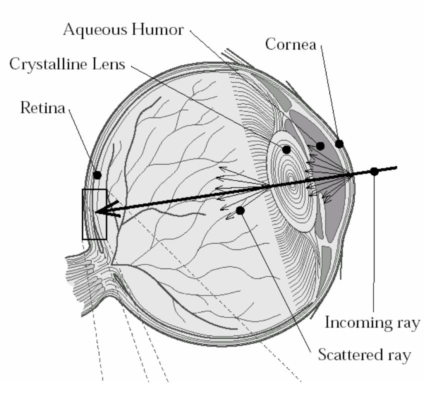
3/23/2004

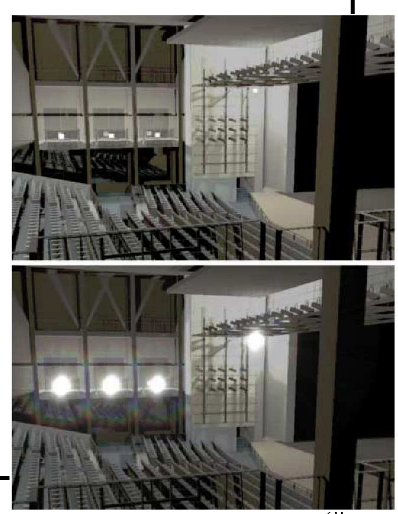
25

EECS 487

Glare/flare/halo

Lecture 16



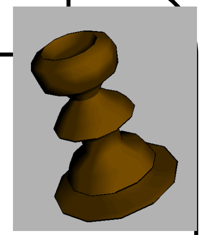


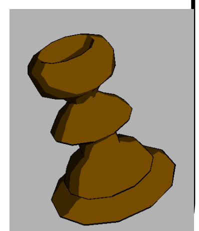
3/23/2004

26

EECS 487

Shading models



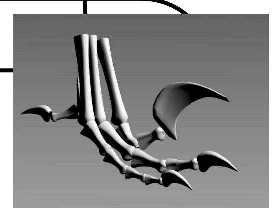


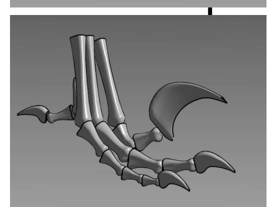
- Toon shading
  - Areas of flat color
    - Integrate 3d with 2d cel animation
  - Fairly easy to do by remapping shades
    - Compute diffuse dot(L,N)
      - Texture coordinate
    - Lookup in 1d texture
      - CPU
      - Vertex shader
      - Three tones on the right
        - » Use material color as well

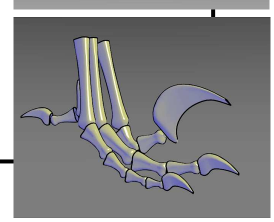
EECS 487

Gooch et al. '98

Technical illustration





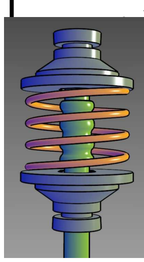


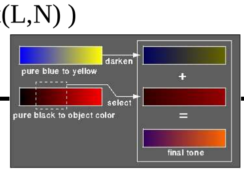
- Photorealistic rendering
  - Phong lighting
    - Some detail is lost
- Illustration shading
  - Luminance and hue change
    - Shape details are visible
    - Feature lines are visible

I = S*Kcool + (1-S)*Kcool

S = 0.5*(1 + dot(L,N))

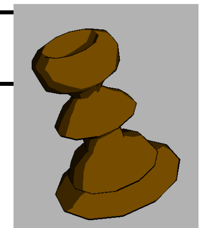
0 < S < 1





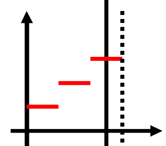
EECS 487

Toon shading details



From Jeff Lander
darwin3d.com

- Texturing
  - Disable lighting
  - Texture has three levels
  - GL_NEAREST minification and magnification filter (no interpolation)
  - Dot(L,N) gives the texture coordinate



- Silhouettes
  - Two passes
    - First
      - Toon shading (DepthFunc = LESS)
    - Second
      - Back face edges drawn in black (DepthFunc = LEQUAL)

3/23/2004 29