Ray tracing

- simple ray casting
- recursive ray tracing
- modeling transforms
- cheap tricks
- optimizations

Global effects

- shadows
- thick surface transmission (refraction)
- reflections

Recursive rays

- incident ray
- reflection
- shadow
- transmission (thick surface)

Occlusion (shadows)

- outward normal $n$
- ray $R'=(P, -ld)$
- light $L$ is occluded if the ray $R'= (P, -ld)$ intersects some object in the scene

Occlusion: implementation

- offset $R'$ slightly so it doesn't intersect at $P$
- all light falling on the back of the surface is occluded

Note: thick surfaces

- light $L$ is occluded if the ray $(P, -ld)$ intersects some object in the scene
Ray tracing:
- cast ray through pixel into scene
- find closest intersection (if any)
- compute luminance at intersection
  - direct illumination
  - reflections
  - thick surface transmission

Reflections:
- cast ray reflected at P into scene
- find closest intersection point P' (if any)
- compute luminance at P
  - scale by msr[g,b] and add to luminance at P

Transmission:
- cast ray transmitted at P into scene
- find closest intersection point P' (if any)
- compute luminance at P
  - scale by $k_{trans} \cdot mdr[g,b]$ and add to luminance at P

Reference - Snell's Law:
incoming ray $(P_0, v)$

transmitted ray $(P, v')$

$\theta_{in}$ satisfies: $\eta_{in} \sin \theta_{in} = \eta_{out} \sin \theta_{out}$

$\mathbf{v}_{out} = \left[ (\cos \theta_{in} - (1 - \beta \sin^2 \theta_{in}) ) \mathbf{n} + \beta \mathbf{v}_{in} \right] , \text{ where } \beta = \eta_{in}/\eta_{out}$
thick surface recursion

What is direct illumination at $P'$?

implementation issues

offset new ray slightly to make sure you don’t find $P$ again!!!

stopping conditions

recurse until:

- cast new ray from $P$ into scene
- find closest intersection point $P'$ (if any)
- compute luminance at $P'$
- scale and add to luminance at $P$

a) maximum recursive depth specified by user is reached
b) contribution to luminance is less than user specified bound