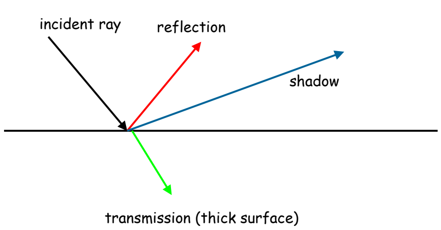
ray tracing

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

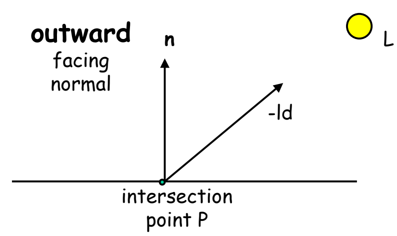
global effects

- shadows
- thick surface transmission (refraction)
- reflections

recursive rays

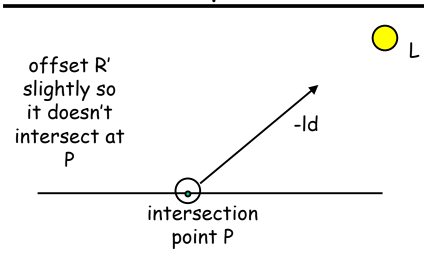


occlusion (shadows)



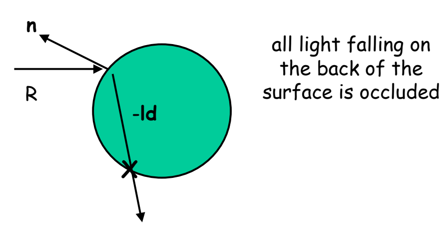
light L is occluded if the ray R'=(P,-Id) intersects some object in the scene

occlusion: implementation

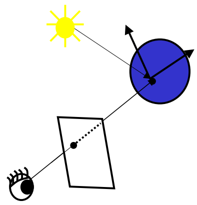


L is occluded if the ray (P,-Id) intersects some object in the scene

note: thick surfaces

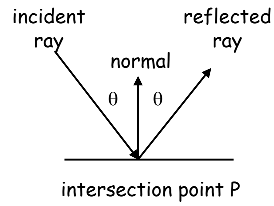


ray tracing



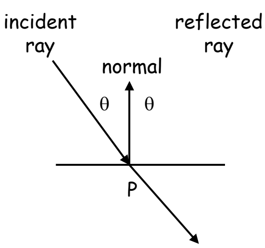
- cast ray through pixel into scene
- find closest intersection (if any)
- compute luminance at intersection
  - direct illumination
  - **reflections**
  - **thick surface transmission**

reflections



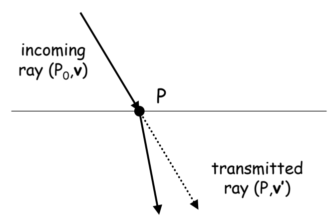
- cast ray reflected at P into scene
- find closest intersection point P′ (if any)
- compute luminance at P′
- scale by msr[g,b] and add to luminance at P

transmission



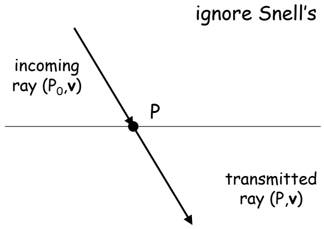
- cast ray transmitted at P into scene
- find closest intersection point P′ (if any)
- compute luminance at P′
- scale by k_{trans}*mdr[g,b] and add to luminance at P

refraction – snell's law

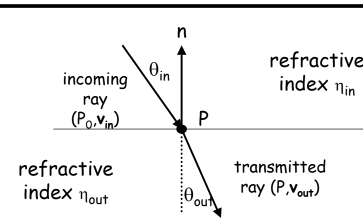


thin surface refraction

ignore Snell's law



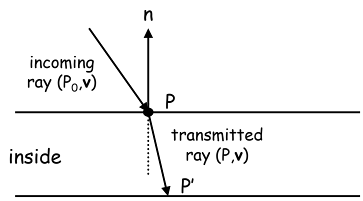
thick surface refraction



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

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

thick surface recursion



What is direct illumination at P'?

stopping conditions

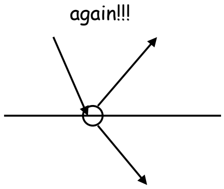
recurse until:

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

- 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

implementation issues

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



- 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

stopping conditions

recurse until:

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

- cast new ray from P into scene
- find closest intersection point P' (if any)
- compute luminance at P'
- scale by msr/g/b or ktrans and add to luminance at P