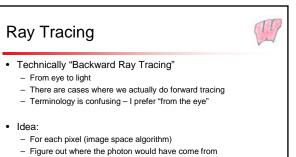


# How the real world "renders"

- Photons (Rays) from source
- Bounce paths
- Some lucky photons make it to the eye (very few)
- Not a practical strategy too inefficient



- Note: get projective transform from ray fan out
- Note: could use real model of lens to determine ray directions
- Note: Sampling Issue

# **Ray Tracing Pieces**

- 1. Figure out what ray is
- 2. Figure out what ray hits (ray-object intersection)
- 3. Figure out where it could have come from
  - Recursive since outgoing ray must have come from someplace

## • Ray / Object Intersection

- Straightforward mathematical calculation (root finding)
- Tricky part: making it go fast
- Accelleration structures:
  - · Simplified models (bounding spheres/boxes)
  - · Hierarchical models (check rough stuff first)
  - Spatial Data structures

# Where did the ray come from?

- · We know: outgoing direction, local surface geometry
- Specular bounce
  - Good for mirror reflection
- Real surfaces are diffuse could come from any direction
  Distribution of likelihoods
  - Different surfaces distribute light differently
  - Really requires an integral over incoming ray directions
  - Bi-directional Reflectance Distribution Function
  - Ideal case: sample all incoming directions

### Hack ray-tracing Shadows • Try to model the rays most likely to be important · Shadows of point lights give hard edges - Even in the real world! - Quite ugly • Mirror reflection bounce (or refraction bounce) · Soft shadows are nicer • Direction towards light sources Come from area light sources - Probably important since they are bright - Umbra / penumbra - Check to see if path is clear (hit something = shadow) - Use local lighting model · How to achieve? - More than one ray towards the light source • What does this give us? - Sampling of directions - Everything from local lighting Shadows

TI.

### - Reflections and Refractions

# More than one ray towards the light source Sampling of directions

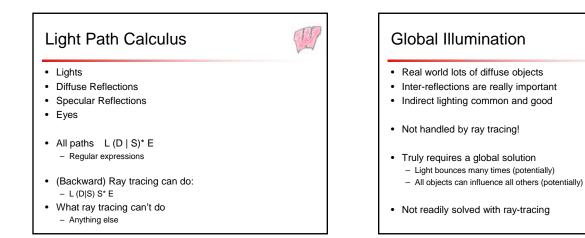
# Distributed Ray Tracing

- Need to sample a distribution of ray directions
- · Some uses:
  - Soft shadows (distribution of directions towards area light)
  - Anti-Aliasing (distribution of rays within the pixel)
  - Imperfect reflections (distribution of outgoing rays)
  - Motion Blur (distribution of times)
  - Depth of Field
  - All indirect light directions (for diffuse surfaces)
    Get inter-object color transfer
  - Notice how quickly this becomes impractical

# What can we do with Ray-Tracing?

- · Given infinite rays, just about anything
- Realistically:
  - Can be clever about how to sample
  - But ultimately, limited in number of rays
- · To understand limits, need to talk about light paths

T



TI

Examples of other things

- Light bounces off mirror (or through lens) to light a diffuse object

• Caustics

– L S\* D E

