Announcements

- **Assignment 2**
  - **Theory is graded (Mean: 68%)**
  - **Programming should be graded by Monday**

- **Assignment 3**
  - **Programming** is due **Friday** (Nov 14th)

Advanced Rendering Techniques

Computer Graphics, CSCD18

Fall 2008

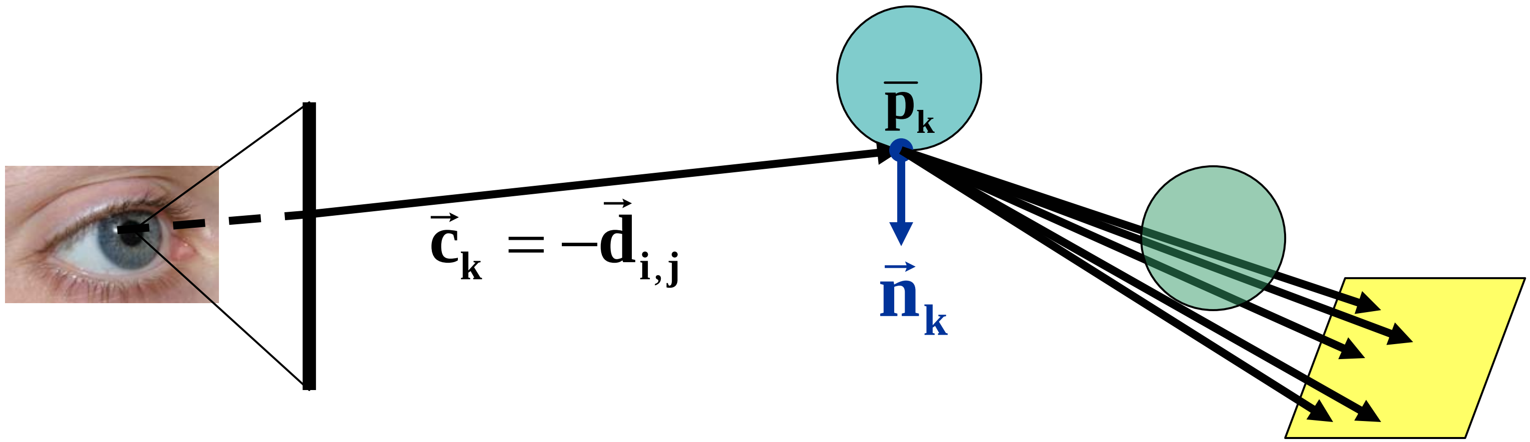
Instructor: Leonid Sigal

Distribution Ray Tracing Review

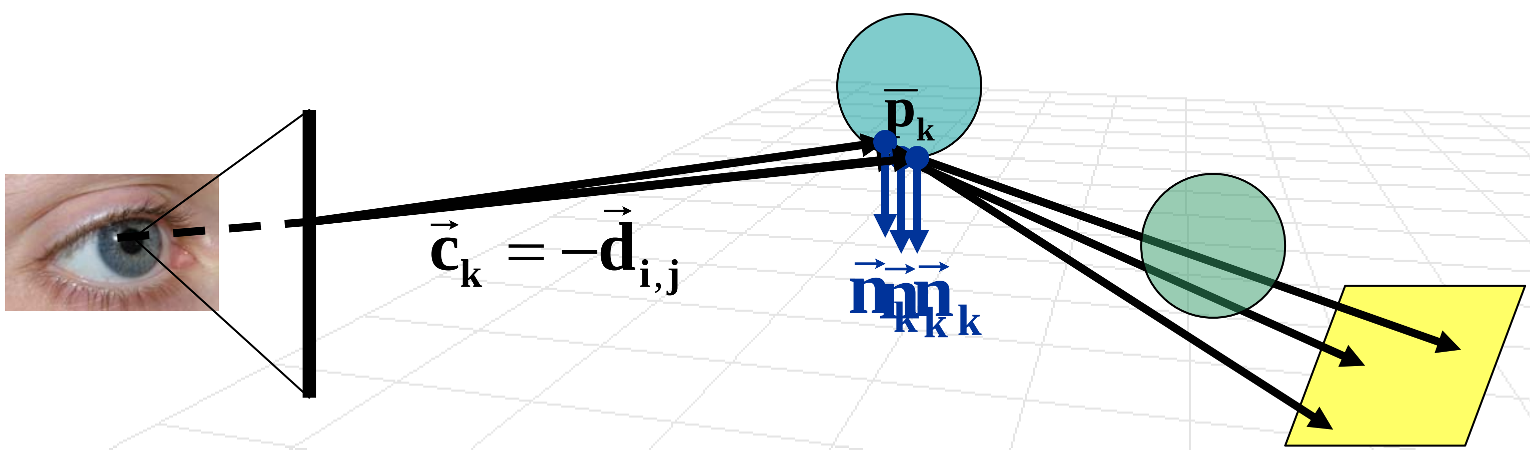
- We can get more complex rendering effects by sending multiple rays per pixel
  - Better global diffuse lighting
    - Color bleeding
    - Bouncing highlights
  - Extended light sources
  - Anti-aliasing
  - Motion blur
  - Depth of field
- Subsurface scattering
- Caustics

One example … Soft Shadows in DTR

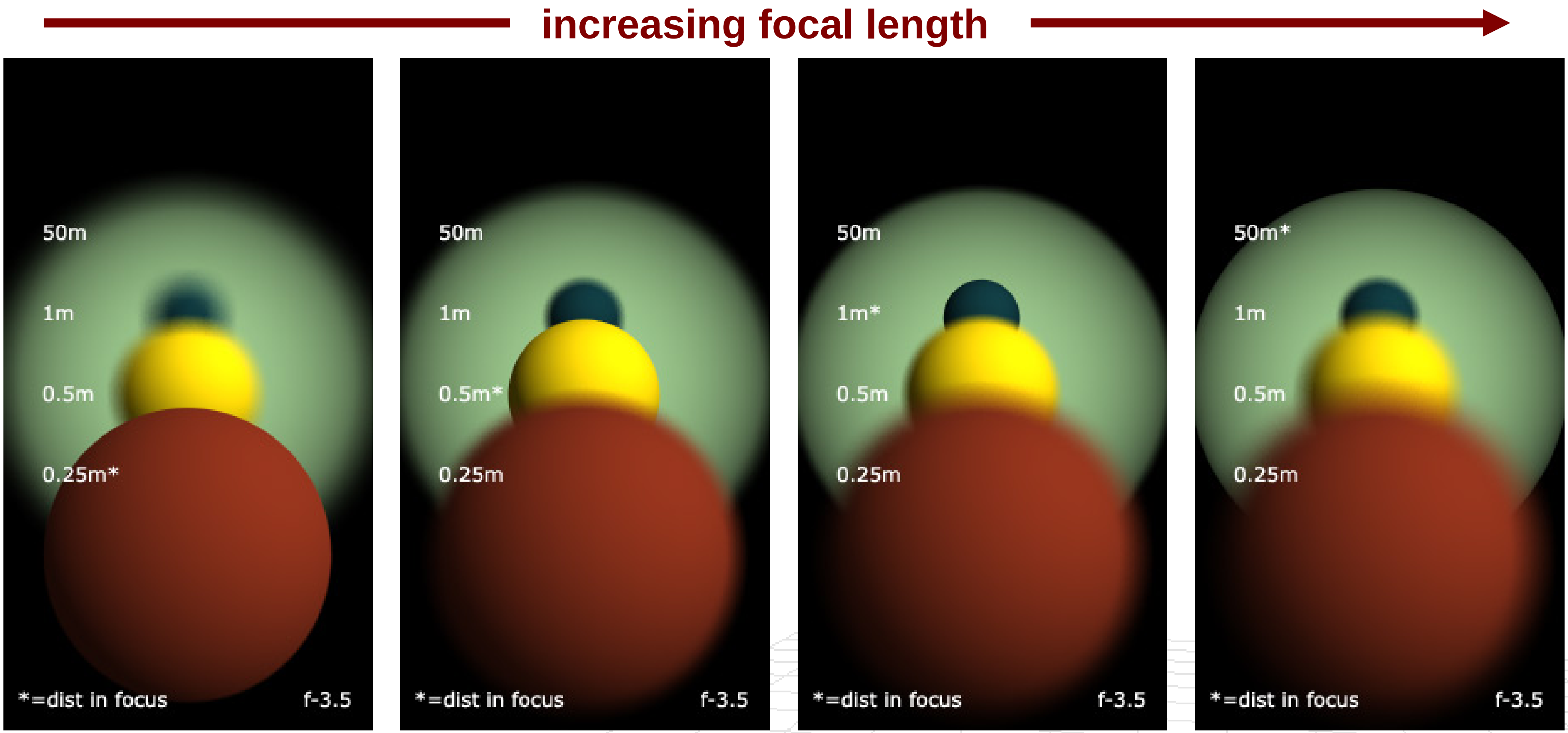
Last class …

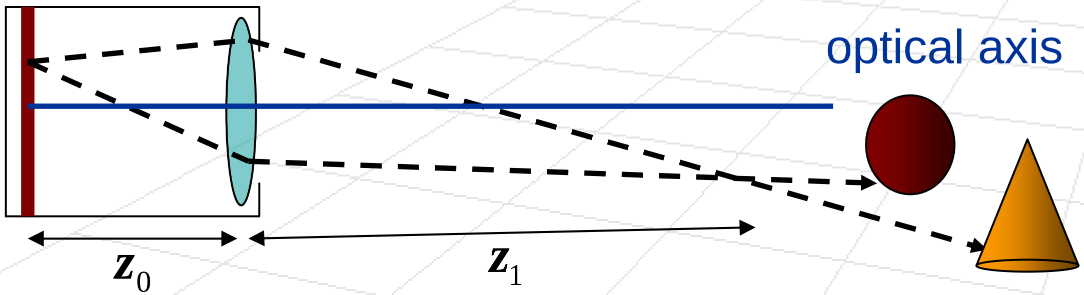


Alternative way of thinking about it



Depth of Field in DRT





220x400 pixels
144 samples per pixel
~4.5 minutes to render

Camera Shutter

- We ignored the fact that **it takes time to form the image**
  - We ignored this for radiometry

- During that time the shutter is open and light is collected
  - We need to **integrate temporally**, not only spatially and account for speed of light transmission

$$\int_t \int_\alpha \int_\beta \mathbf{H}(\alpha, \beta, \mathbf{t})\, \mathbf{d}\alpha\, \mathbf{d}\beta\, \mathbf{dt}$$

Motion Blur

Cook, Porter & Carpenter

Motion Blur

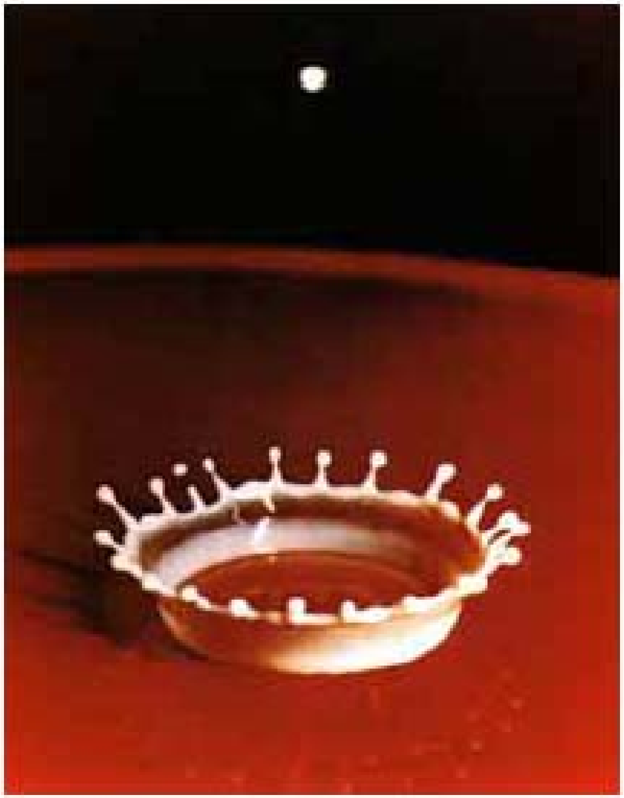
Long Exposure Photography

Motion Blur (long exposures)

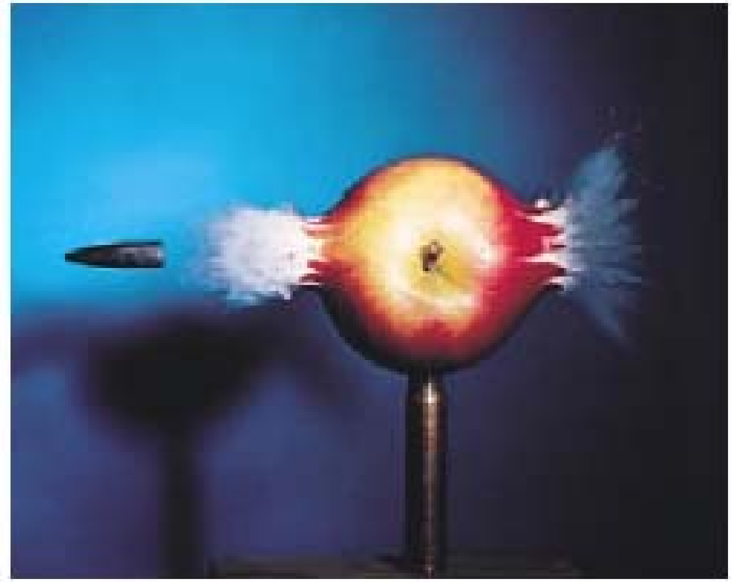
Golden Gate Bridge
30 sec. exposure @ f4

Bodie State Park
30 min. exposure @ f4

Motion Blur (short exposures)



Doc Edgerton, 1936



Sub-surface Scattering

H. W. Jensen

Sub-surface Scattering

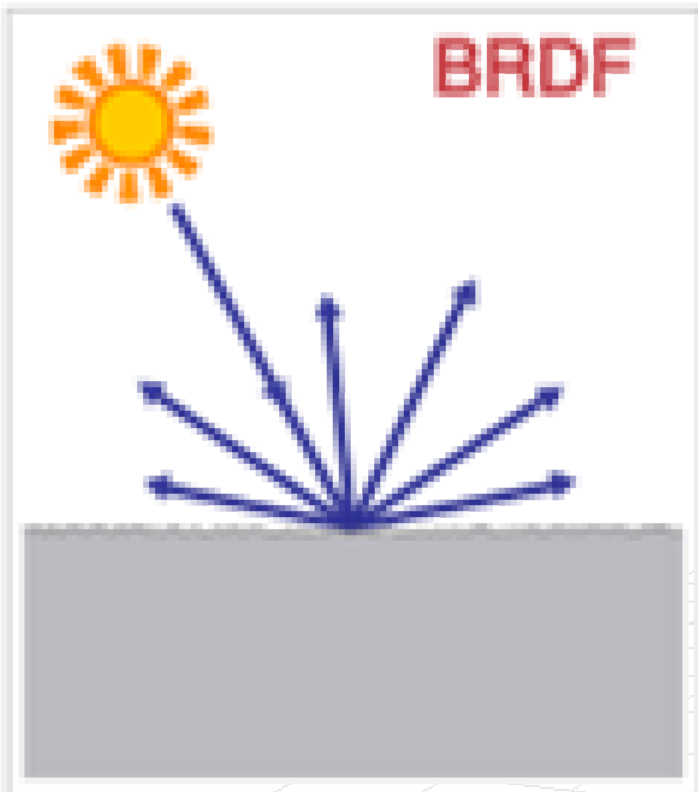
Bidirectional Surface Scattering Reflectance Distribution Function

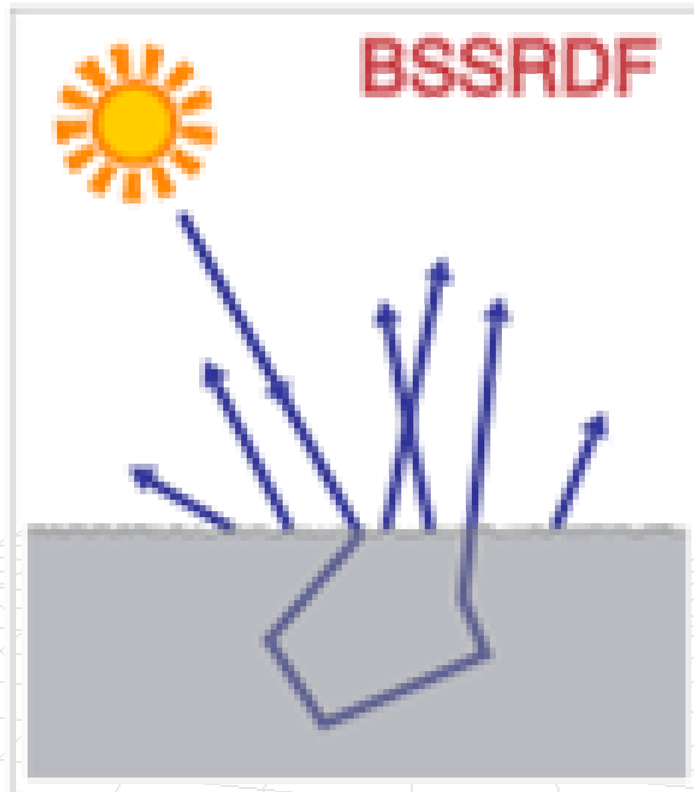
Rendering with BRDF

Rendering with BSSRDF

H. W. Jensen

Bidirectional Surface Scattering Reflectance Distribution Function





[Images taken from Wikipedia]

Semi-Transparencies

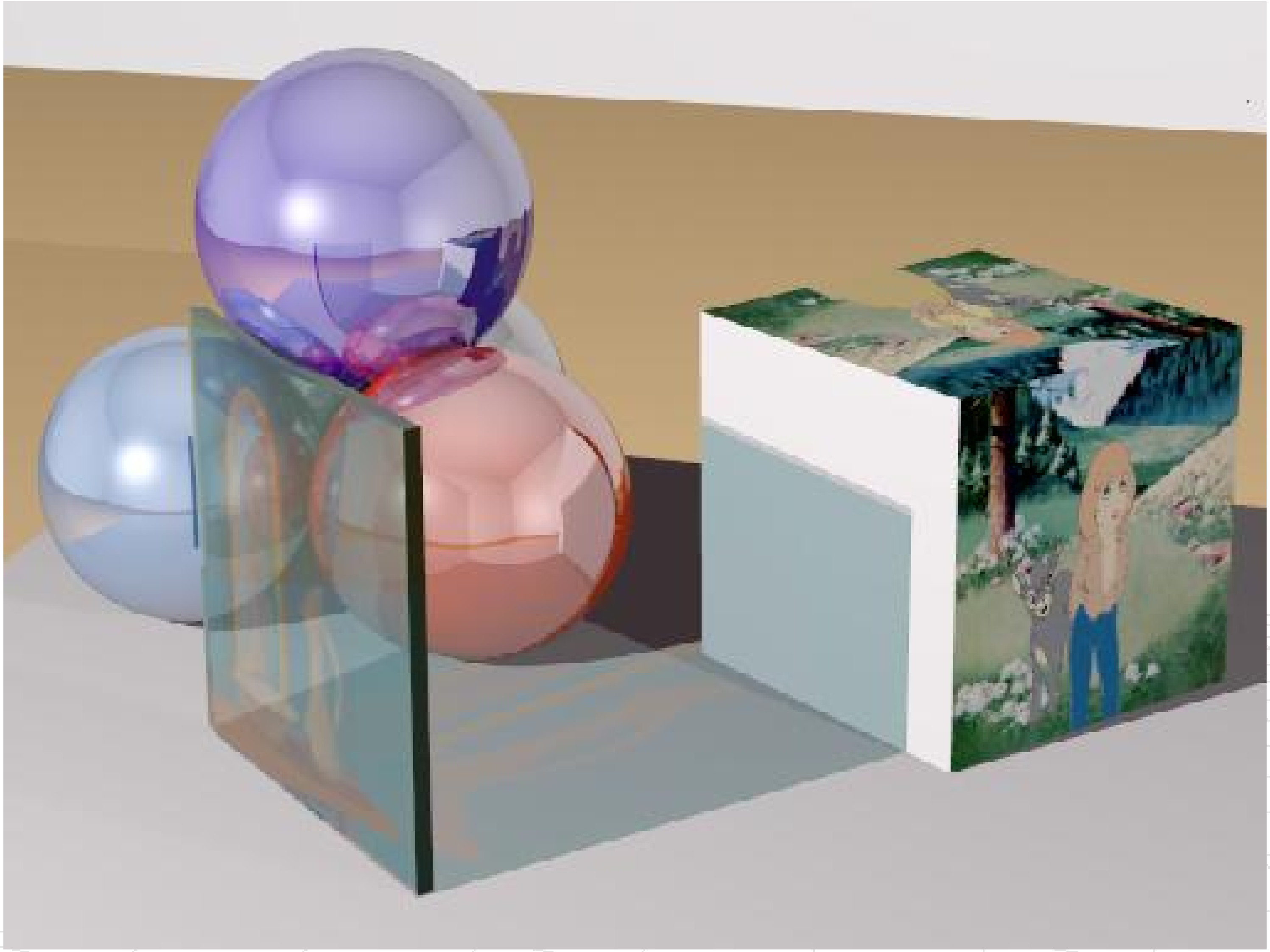


Image form http://www.graphics.cornell.edu/online/tutorial/raytrace/

Texture-mapping and Bump-mapping in Ray Tracer

Image form http://www.graphics.cornell.edu/online/tutorial/raytrace/

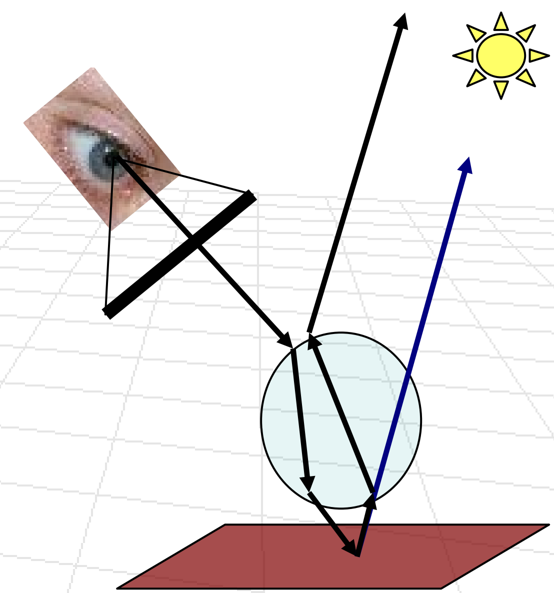
Caustics

- Hard to do in Distribution Ray Tracing
  - **Why?**

Caustics

- Hard to do in Distribution Ray Tracing
  - **Why?**

Hard to come up with a good importance function for sampling,
Hence, VERY VERY slow



Caustics

- Often done using bi-directional ray tracing (a.k.a. **photon mapping**)
  - Shoot light rays from light sources
  - Accumulate the amount of light (radiance) at each surface
  - Shoot rays through image plane pixels to “look-up” the radiance (and integrate irradiance over the area of the pixel)

Photon Mapping

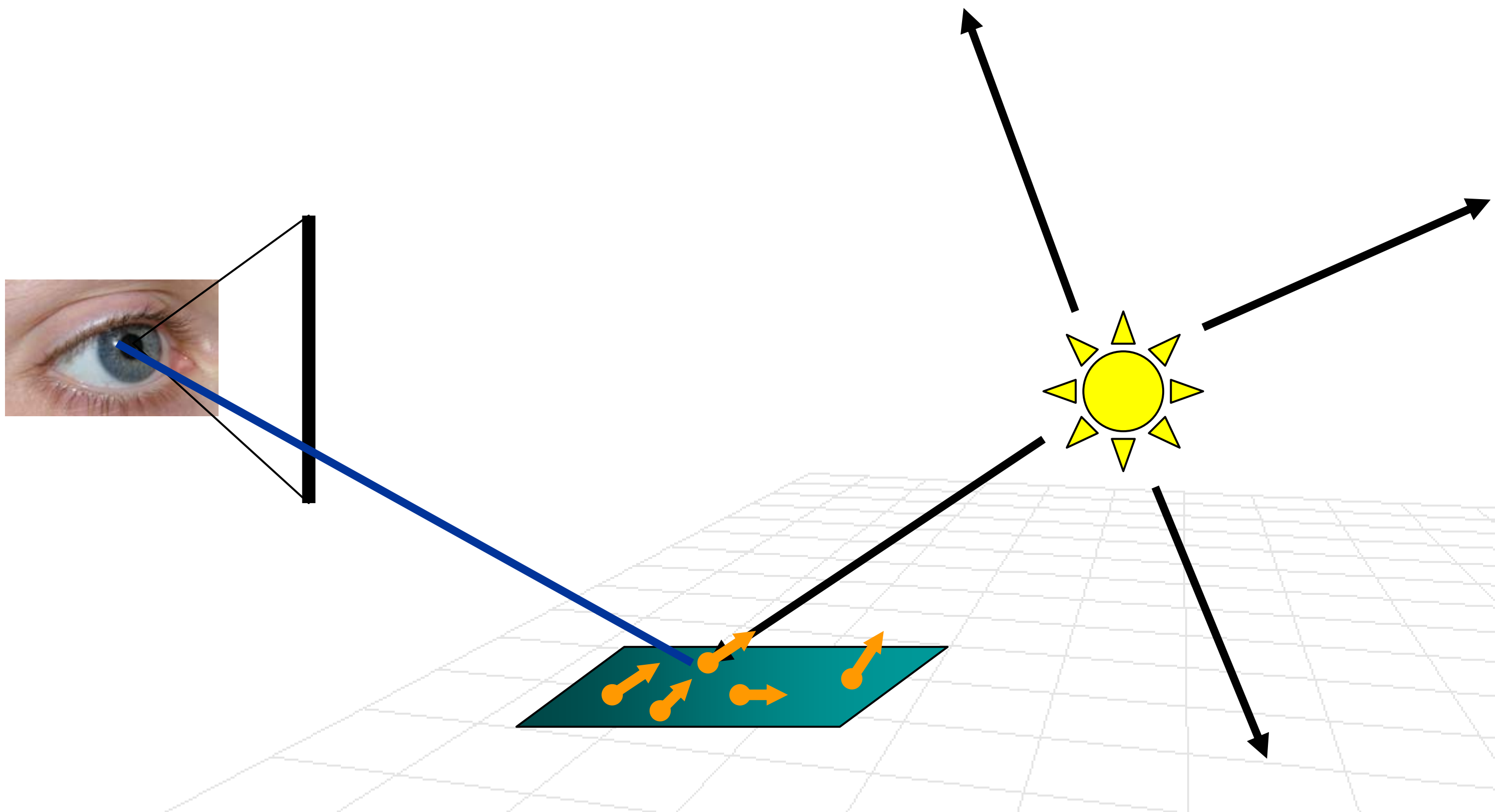
- Simulates **individual photons**
  - In DTR we were simulating radiance (flux)
- Photons are emitted from light sources
- Photons bounce off of specular surfaces
- Photons are **deposited on diffuse surfaces**
  - Held in a 3-D spatial data structure
  - Surfaces need not be parameterized
- Photons **collected by ray tracing** from eye

Photons

- A **photon** is a particle of light that carries flux, which is encoded as follows
  - magnitude (in Watts) and color of the flux it carries, stored as an RGB triple
  - location of the photon (on a diffuse surface)
  - the incident direction (used to compute irradiance)

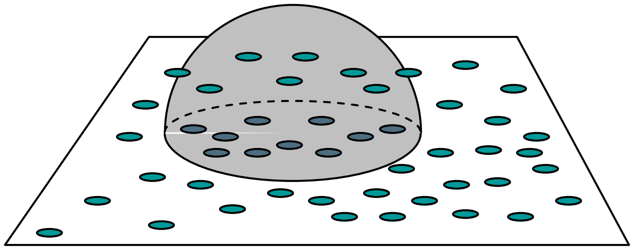
- **Example** (point light source, photons emitted uniformly)
  - Power of source (in Watts) distributed evenly among photons
  - Flux of each photon equal to source power divided by total # of photons
  - 60W light bulb would sending 100 photons, will result in 0.6 W per photon

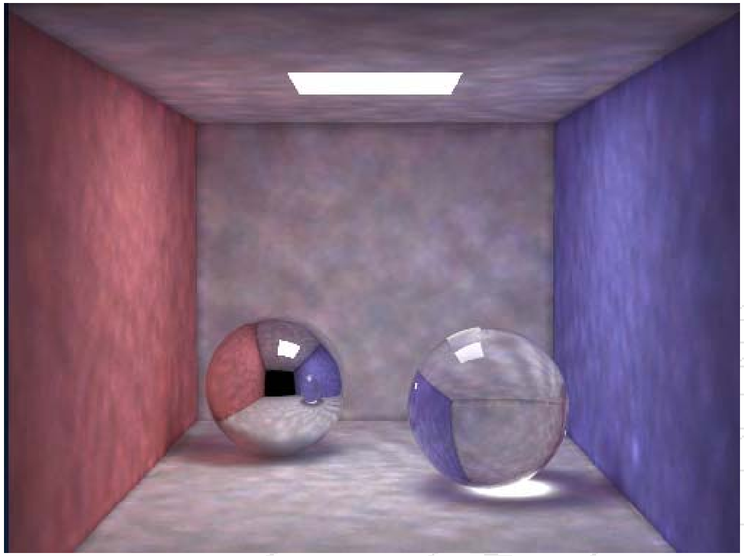
How does this actually work?



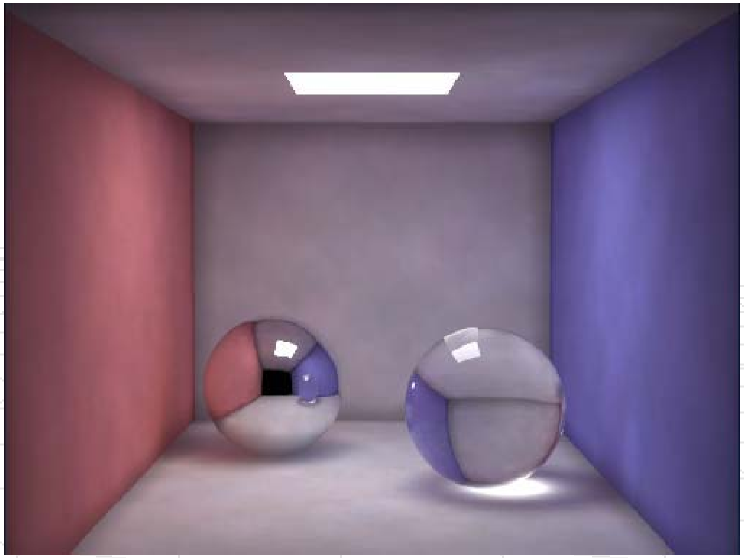
Special data structures are required to do fast look-up (KD-trees)

Photon Mapping Results





Radiance estimate using 50 photons



Radiance estimate using 500 photons