Advanced Ray Tracing Techniques

Course web page:
http://goo.gl/EB3aA

Outline

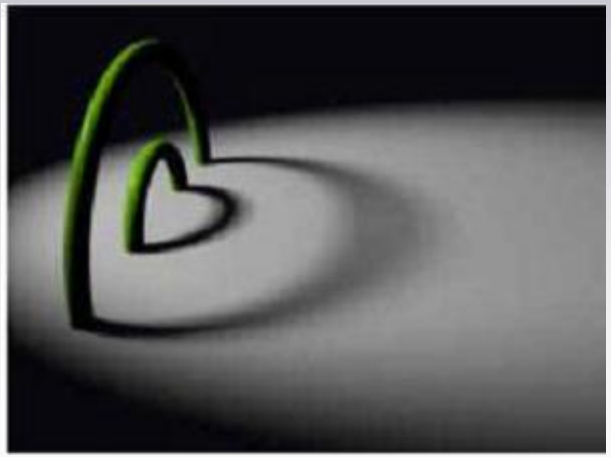
- Distributed/distribution ray tracing & bounding volumes
  - Anti-aliasing
  - Soft shadows, glossy reflections, ambient occlusion
- Light paths & caustics
- HW #3—go over code structure

Basic Ray Tracing: Notes

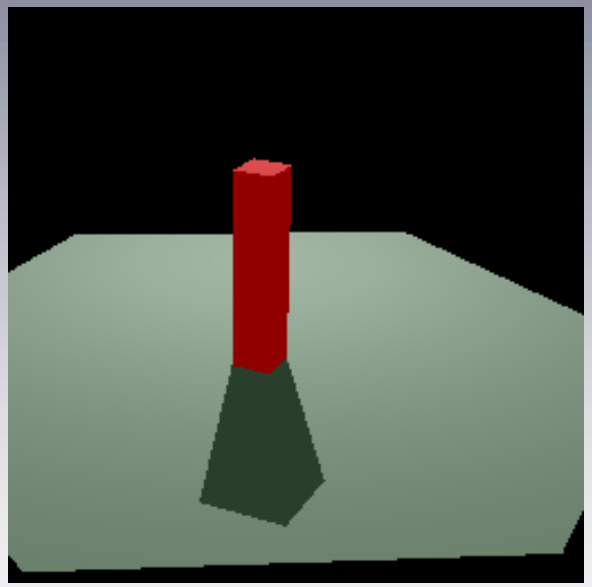
- Global illumination effects simulated by basic algorithm are shadows, purely specular reflection/transmission
- Some outstanding issues
  - Aliasing, aka jaggies
  - Shadows have sharp edges, which is unrealistic
  - No diffuse reflection from other objects
- Intersection calculations are expensive, and even more so for more complex objects
  - Not currently suitable for real-time (i.e., games)

DRT: Soft Shadows

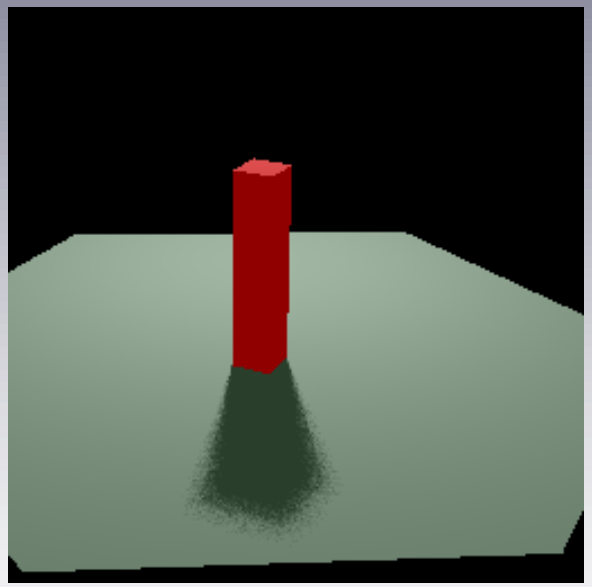
- For point light sources, sending a single shadow ray toward each is reasonable
  - But this gives hard-edged shadows
- Simulating soft shadows
  - Model each light source as sphere
  - Send multiple jittered shadow rays toward a light sphere; use **fraction** that reach it to attenuate color
  - Similar to ambient occlusion, but using list of light sources instead of single hemisphere



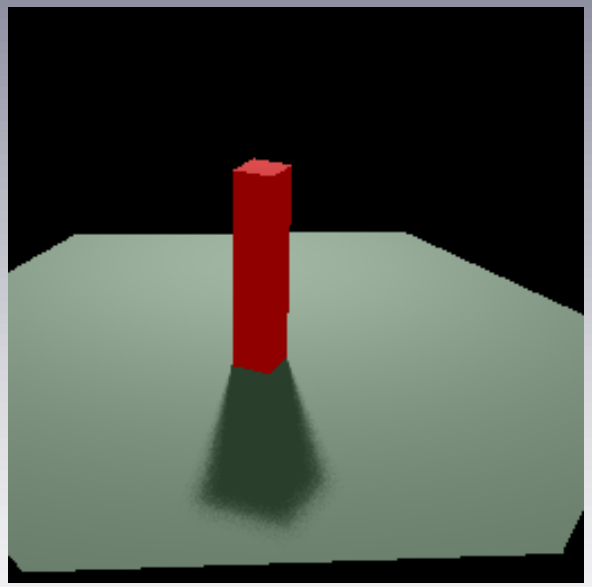
Soft Shadows: Example



1 shadow ray



10 shadow rays



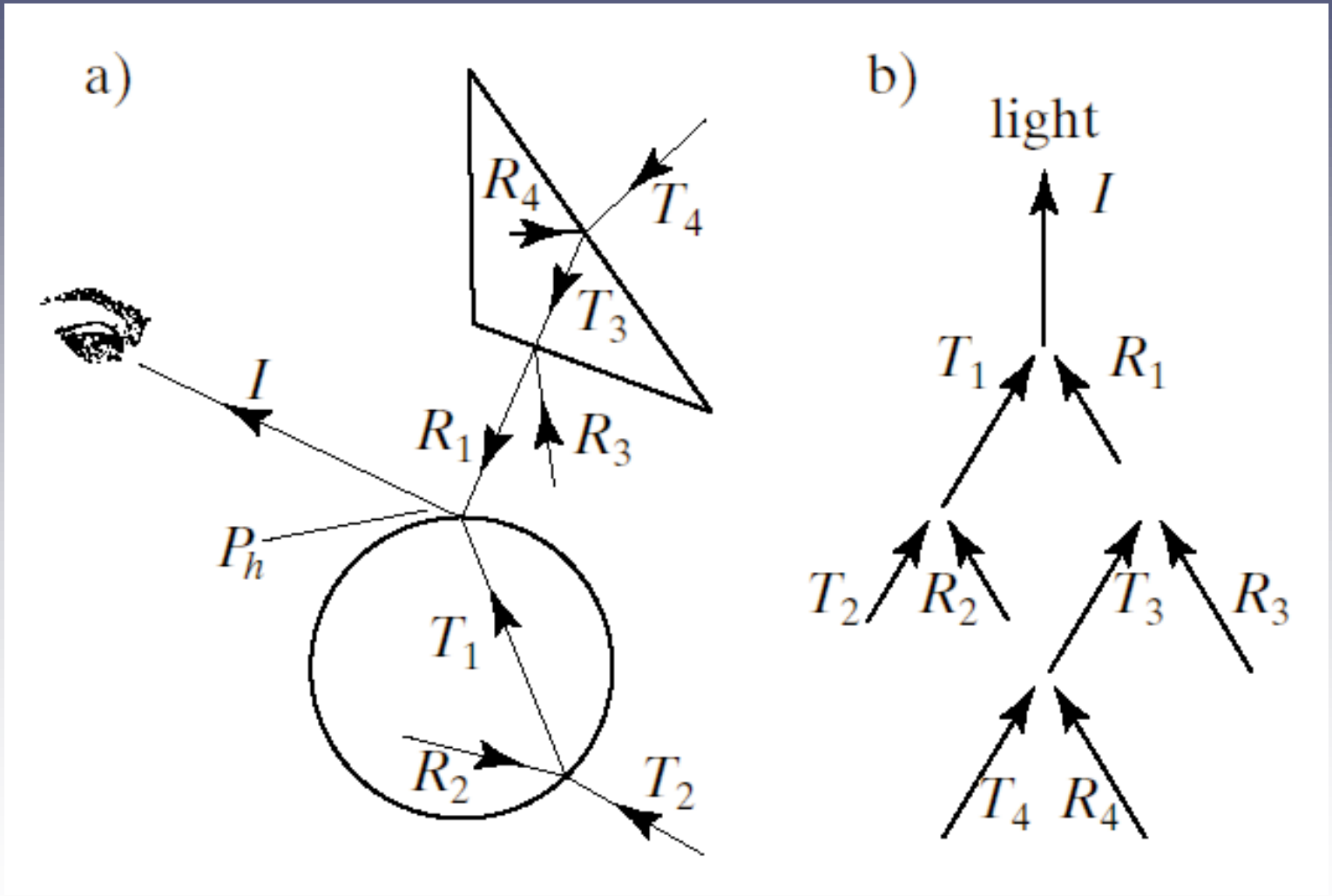
50 shadow rays

DRT: Glossy Reflections

- Analog of hard shadows are "sharp reflections"—every reflective surface acts like a perfect mirror
- To get glossy or blurry reflections, send out multiple jittered reflection rays and average their colors

Why is the reflection sharper at the top?

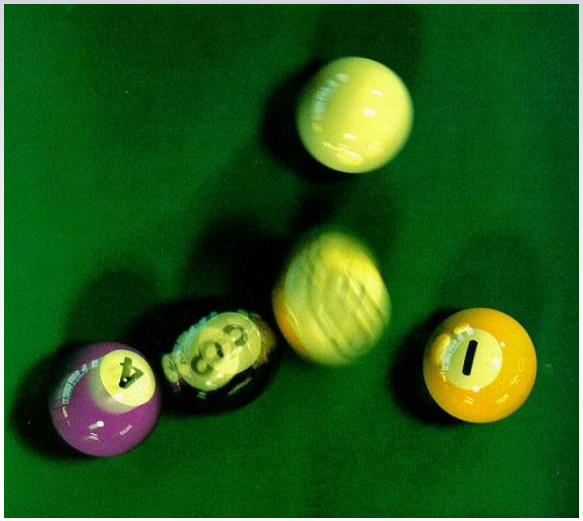
Ray Tracing: Recursion



Other DRT Effects

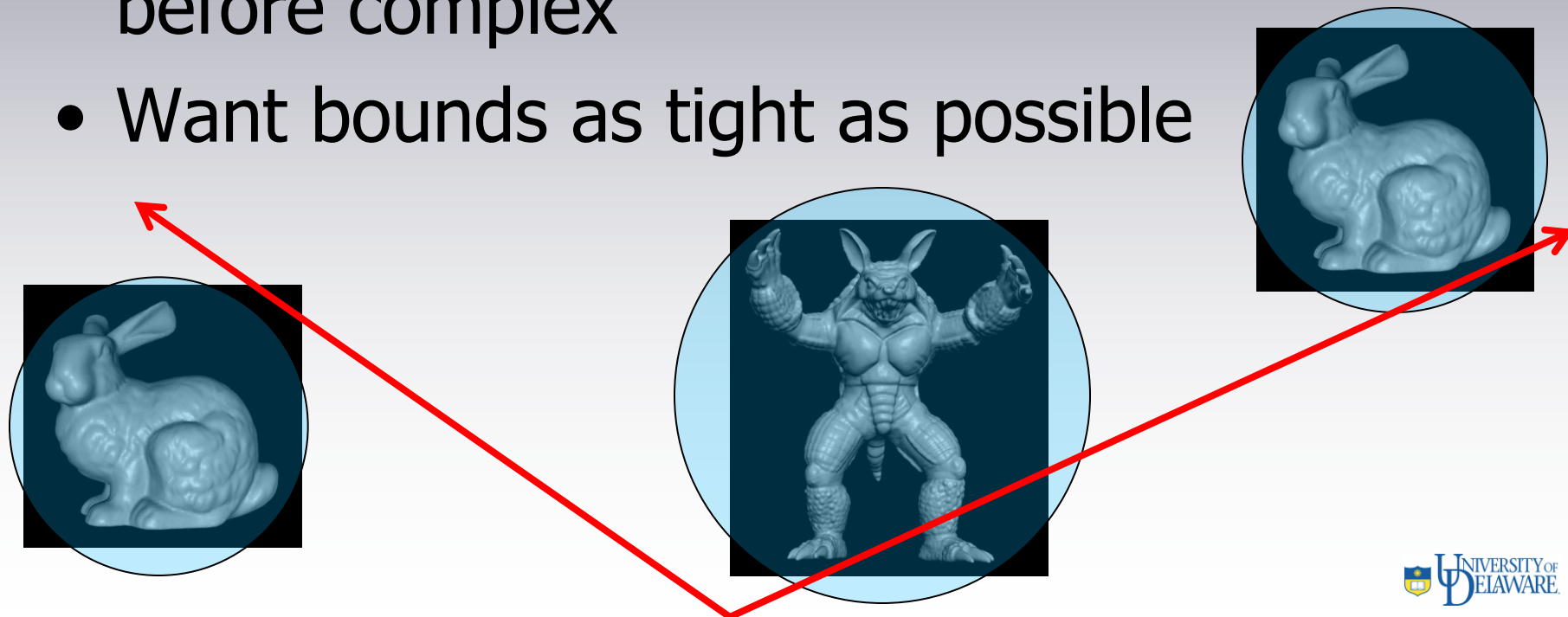
- Depth of field

- Motion blur

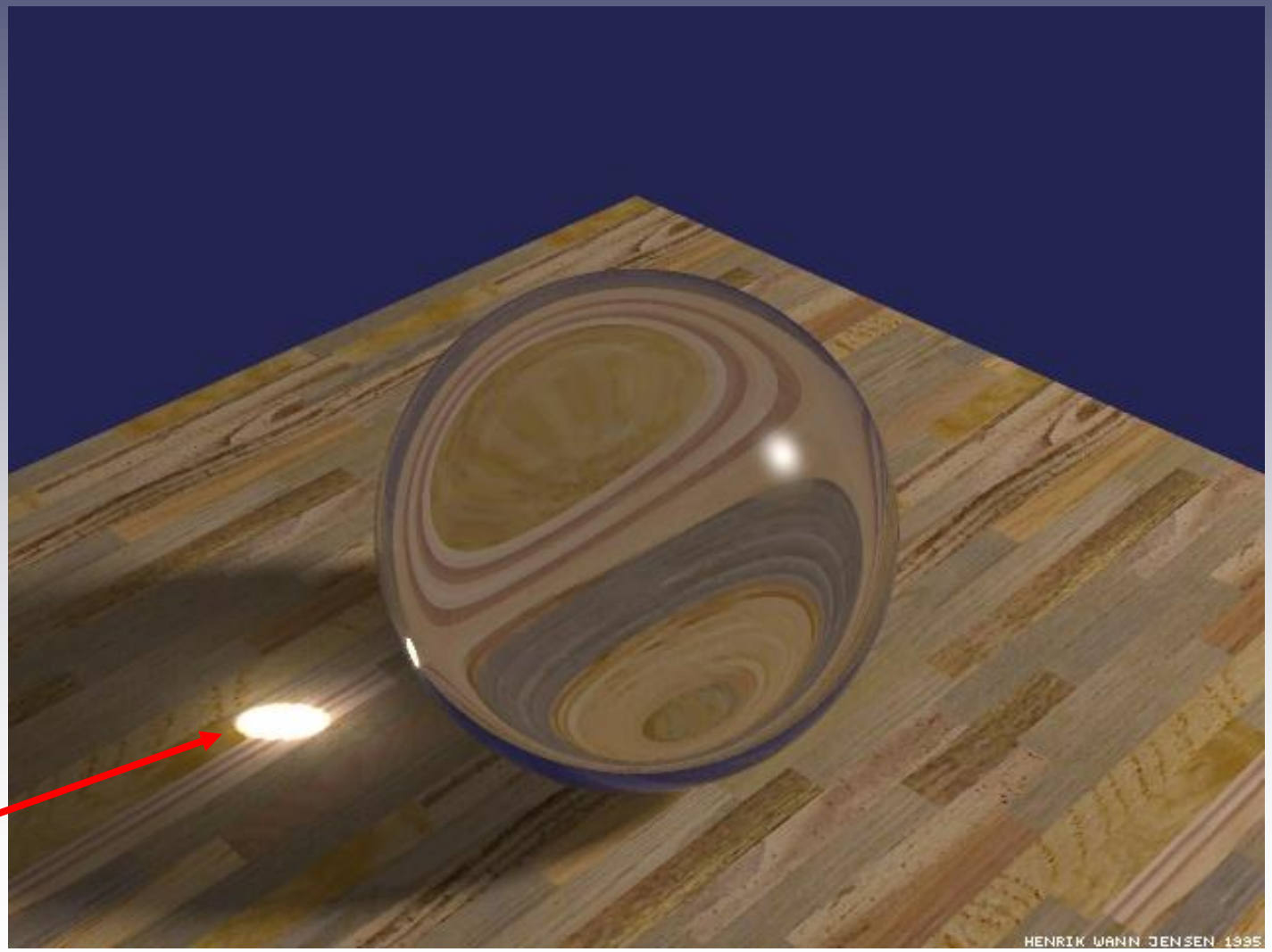


Bounding Volumes

- Idea: enclose complex objects (i.e., .obj models) in simpler ones (i.e., spheres, boxes) and test simple intersection before complex
- Want bounds as tight as possible



Can Ray Tracing Do This?

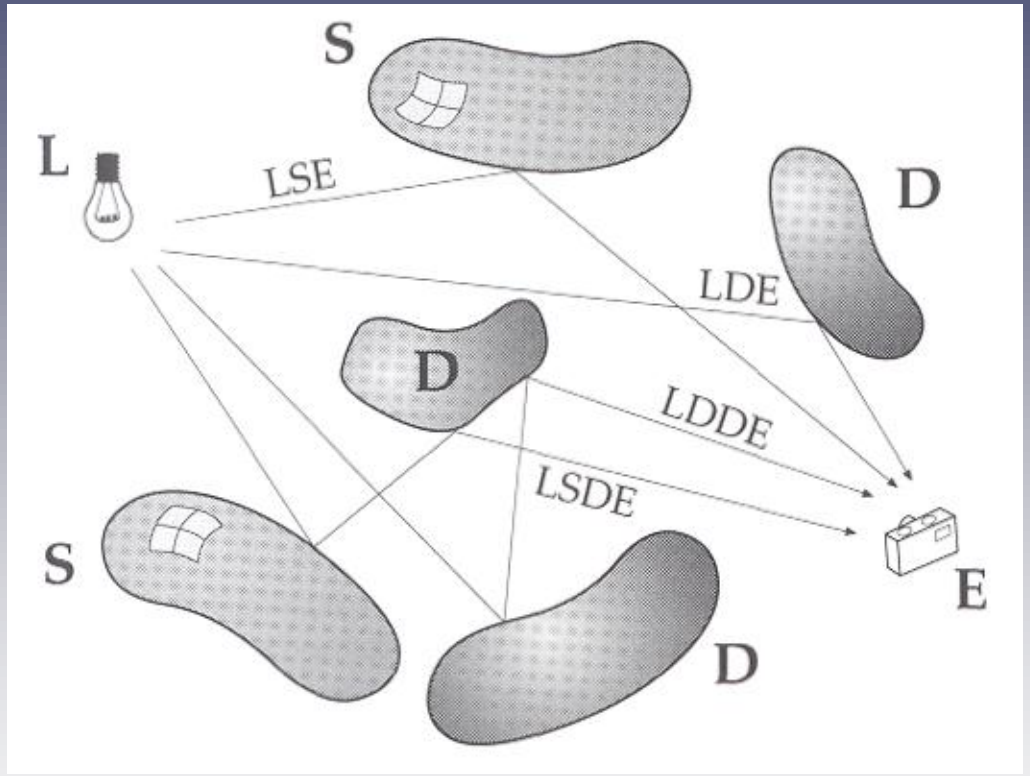


caustic

courtesy of H. Wann Jensen

Light Paths

- Consider the path that a light ray might take through a scene between the light source **L** and the eye **E**
- It may interact with multiple diffuse (**D**) and specular (**S**) objects along the way

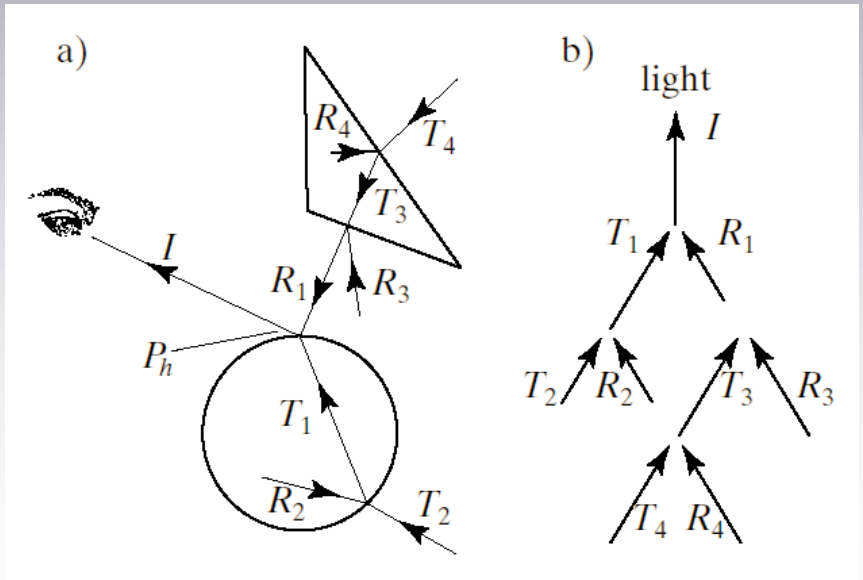


from Sillion & Puech

- We can describe this series of interactions with the regular expression **L** (**D** | **S**)* **E**
  - (If a surface is a mix of **D** and **S**, the combination is additive so it is still OK to treat in this manner)

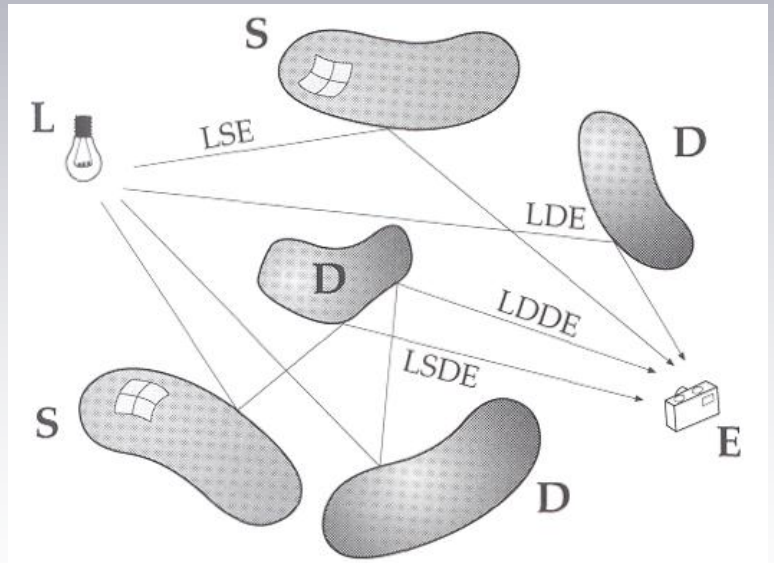
Light Paths: Examples

- Direct visualization of the light: **LE**
- Local illumination: **LDE**, **LSE**
- Ray tracing: **LS*E**, **LDS*E**



from Hill

Ray tracing light paths

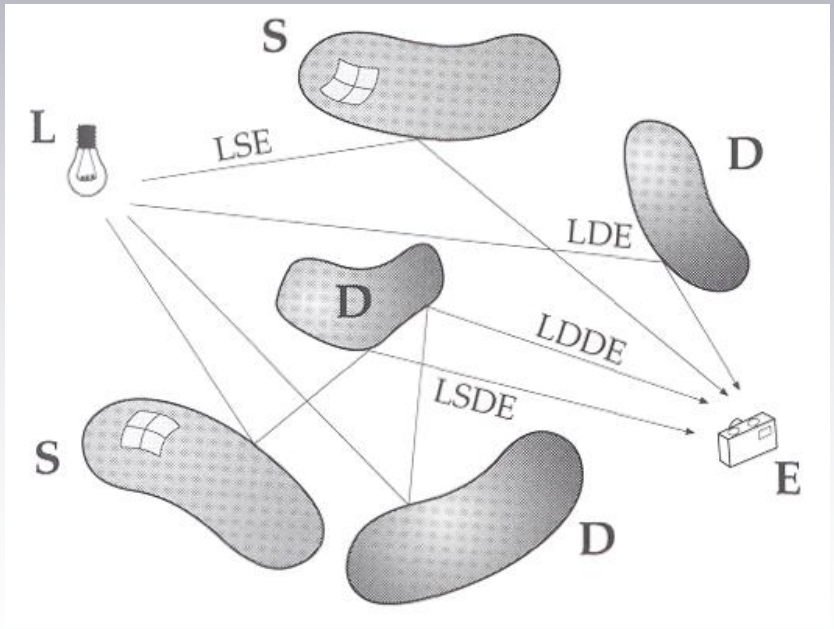


from Sillion & Puech

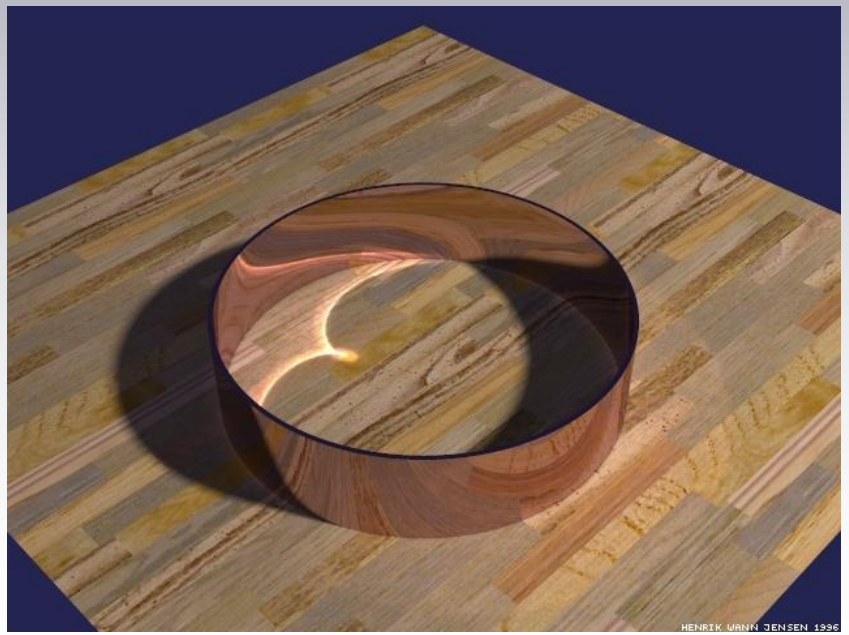
General light paths

Caustics

- Definition: (Concentrated) specular reflection/refraction onto a diffuse surface
  - In simplest form, follow an **LSDE** path
- Standard ray tracing **cannot** handle caustics—only paths described by **LDS*E**



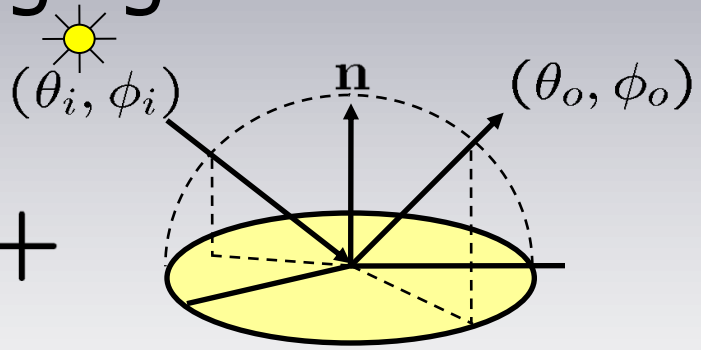
from Sillion & Puech



courtesy of H. Wann Jensen

More about caustics

- What is the problem with **LS⁺DE** paths for ray tracing?
- Review: Radiance for a viewing direction given all incoming light:



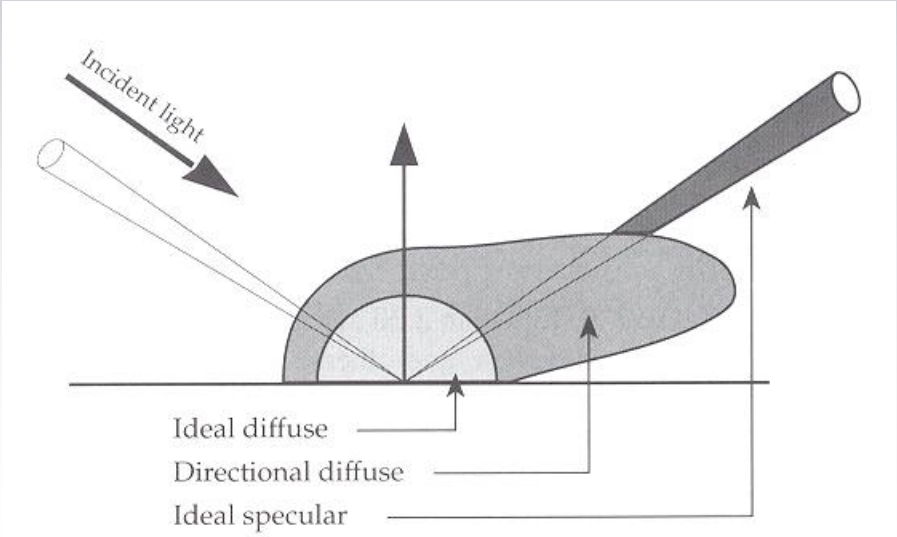
emitted light

$$L_o(\mathbf{x}, \theta_o, \phi_o) = L_e(\mathbf{x}, \theta_o, \phi_o) + \int_\Omega f(\theta_o, \phi_o, \theta_i, \phi_i) L_i(\mathbf{x}, \theta_i, \phi_i) \cos\theta_i d\omega$$

reflected light

Review: BRDFs

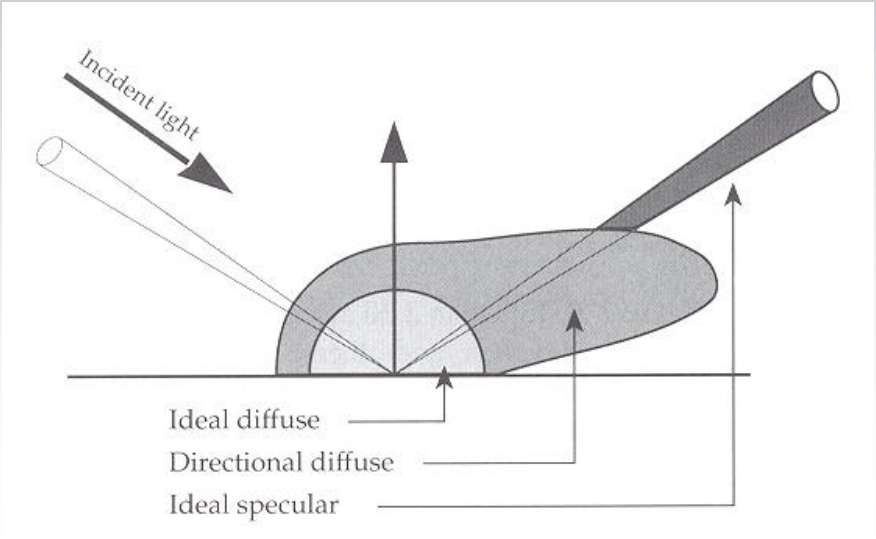
- **Bidirectional Reflectance Distribution Function** (BRDF): Ratio of outgoing radiance in one direction to incident irradiance from another
- Can view BRDF as **probability** that incoming photon will leave in a particular direction (given its incoming direction)



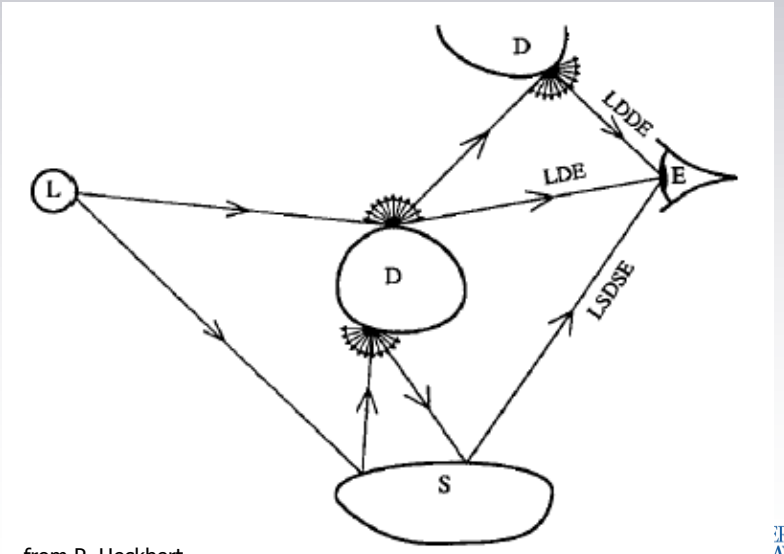
from Sillion & Puech

The Problem with Diffuse Surfaces

- For specular surfaces, we "know" where the photon will go (= "came from", if going backwards), whereas for diffuse surfaces there's much more uncertainty
  - If we're tracing a ray from the eye and we hit a diffuse surface, this uncertainty means that the source of the photon could be anywhere in the hemisphere
  - Conventional ray tracing just looks for lights at this point, but for **LS$^+$DE** paths we need to look for other specular surfaces
    - How to find them?



from Sillion & Puech



from P. Heckbert

HW #3 (due next Thursday, Apr. 26)

- Basic requirements
  - Complete shade_ray_**diffuse**()
  - Complete shade_ray_local(), which adds **specular and shadow** effects
  - Complete **reflection** component of shade_ray_recursive()
  - Add **sphere intersection testing** in intersect_ray_sphere()
  - Scene complexity and creativity
- Grad student requirements
  - Add support for **refraction** in shade_ray_recursive()
  - Add some version of adaptive supersampling, glossy reflection, ambient occlusion, or another advanced **distributed-ray** technique
  - Implement **bounding spheres** around objects to speed intersection calculations