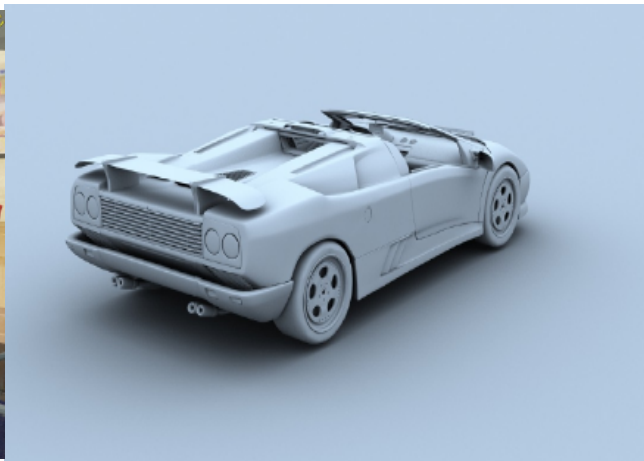




## **Global Illumination**

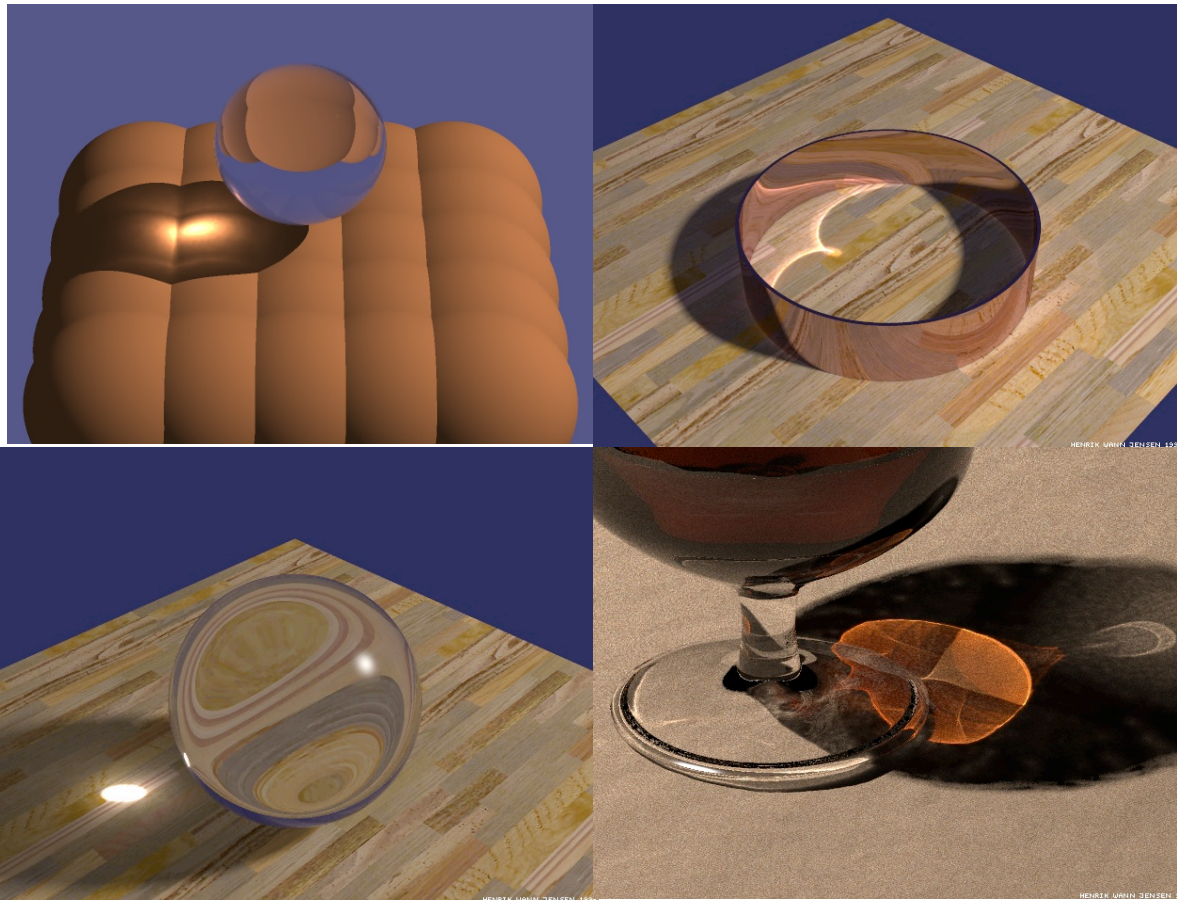
## Problems with Ray Tracing

- **World is diffuse!**



## Problems with Ray Tracing

➤ **Caustics and Volume Light**



## Distributed Ray Tracing

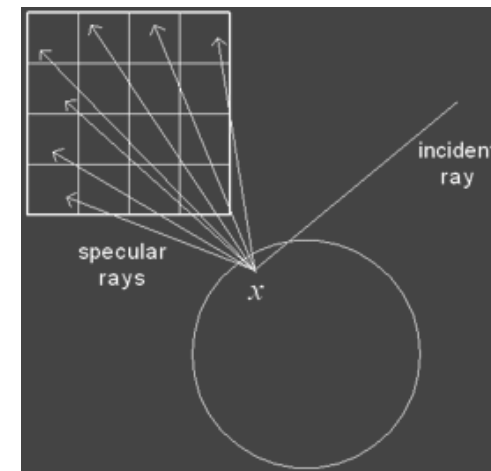
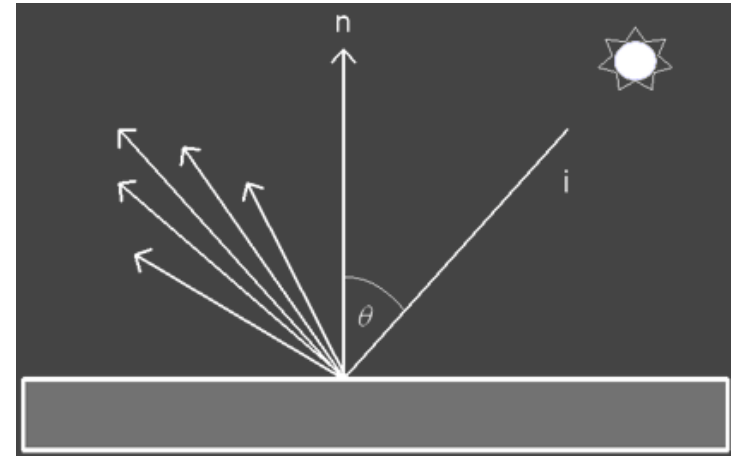
---

- **Distributed ray tracing is a ray tracing method based on randomly distributed oversampling**
- **Stochastic Oversampling**
  - Pixel for antialiasing
  - Light source for soft shadows
  - Reflection function for soft (glossy) reflections
  - Time for motion blur
  - Lens for depth of field

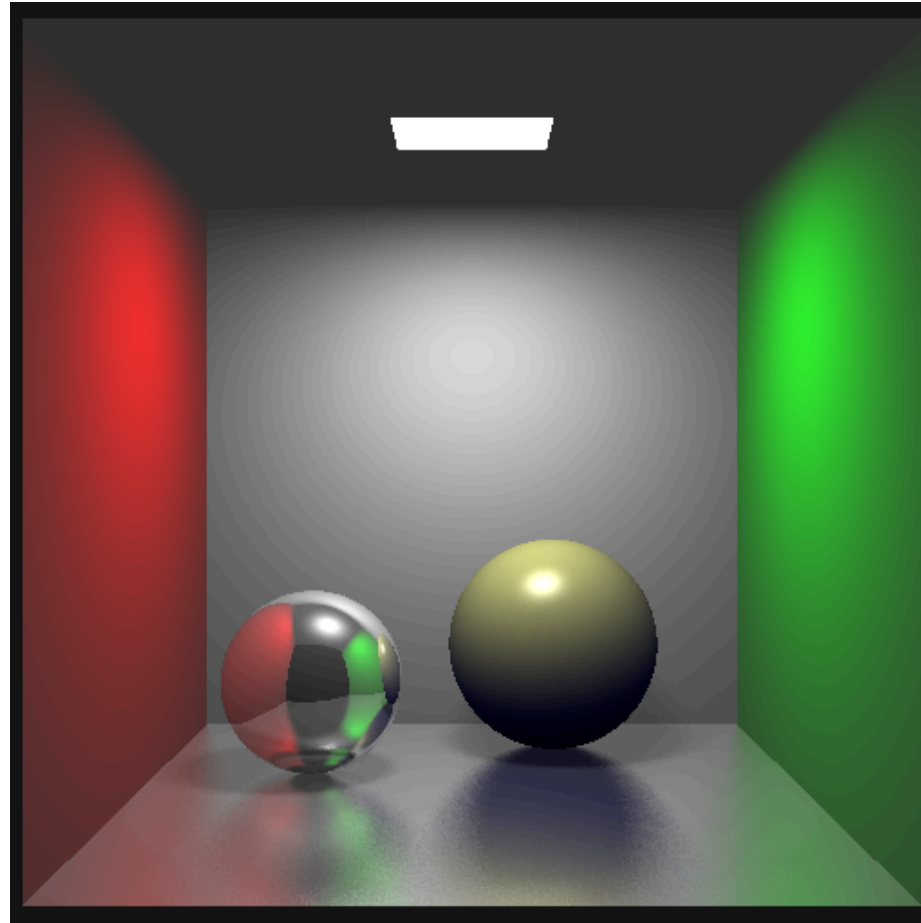


## Distributed Ray tracing

- **Distributing a set of reflection rays by randomly perturbing the ideal specular reflection ray.**
- **The spread of the distribution determines the glossiness where a wider distribution spread models a rougher surface.**



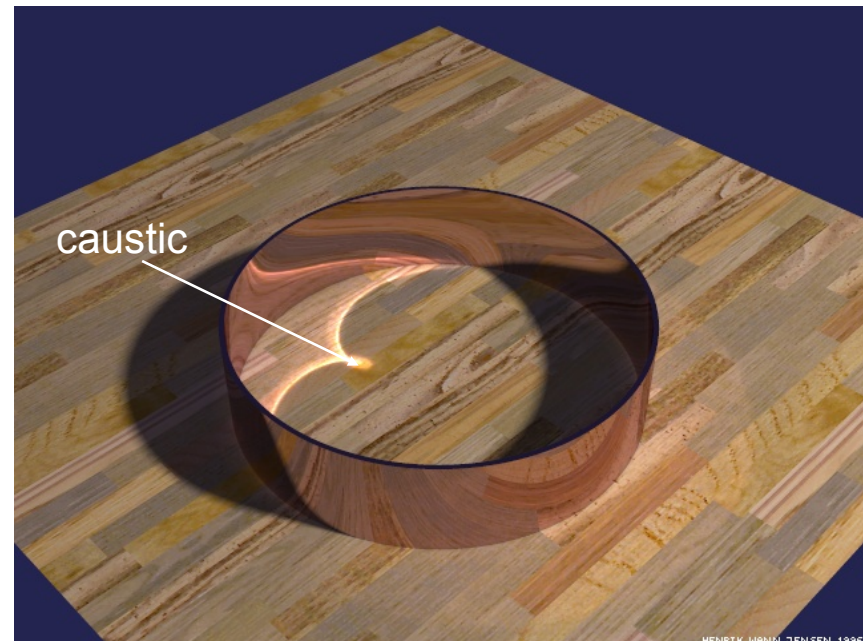
## Distributed Ray tracing



taken from [http://www.cs.wpi.edu/~emmanuel/courses/cs563/write\\_ups/zackw/realistic\\_raytracing.html](http://www.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/realistic_raytracing.html)

## Bidirectional Ray tracing

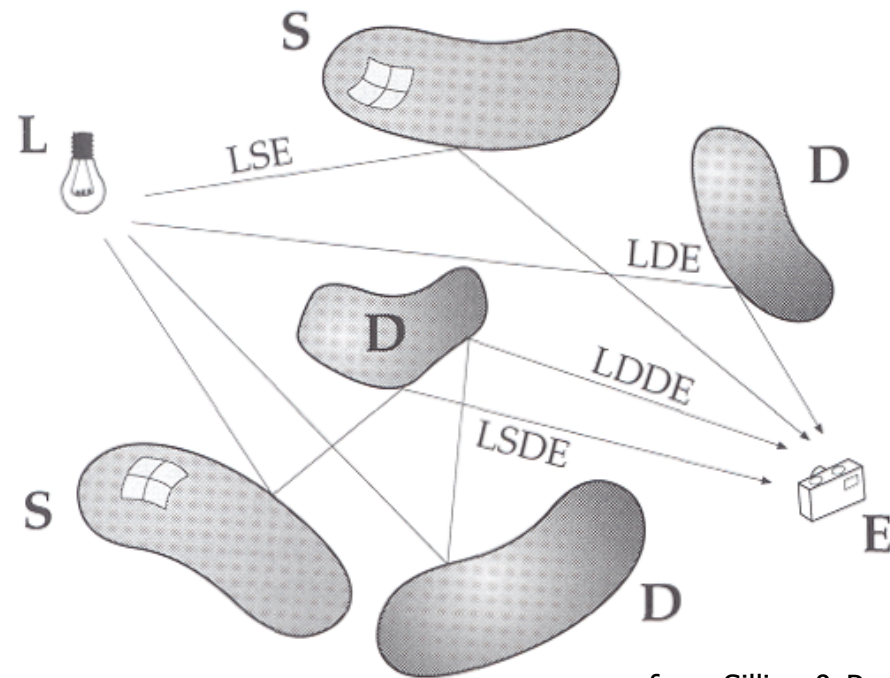
- **Caustic -  
(Concentrated)  
specular reflection  
/refraction onto a  
diffuse surface**
- **Standard ray  
tracing cannot  
handle caustics**



## Light Paths

➤ Interactions of the light ray can be expressed using regular expressions

- L is the light source
- E is the eye/camera
- D is a diffuse surface
- S is a specular surface

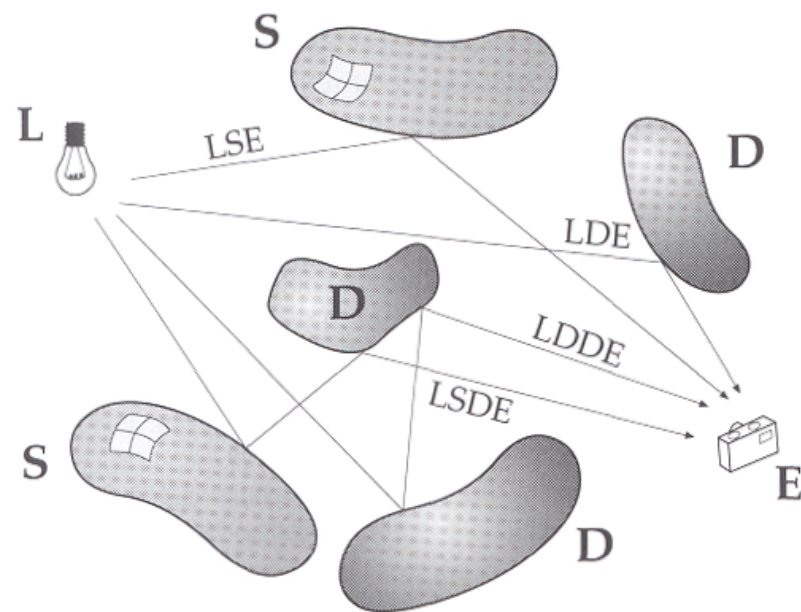


from Sillion & Puech



## Light Paths

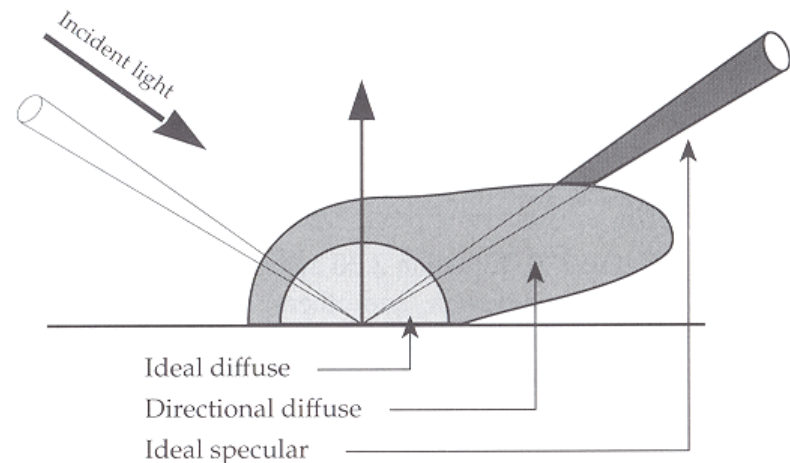
- Direct visualization of the light: LE
- Local illumination: LDE, LSE
- Ray tracing:  $LS^*E$ ,  $LDS^*E$
- Caustics:  $LS^+DE$



from Sillion & Puech

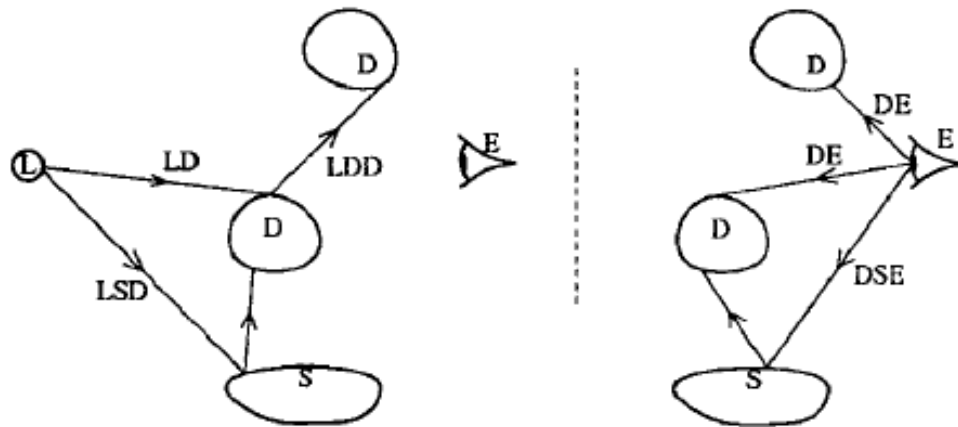
## Diffuse Surfaces

- **Uncertainty in the direction that a photon will take for diffuse surfaces**
- **For specular surfaces, the BRDF (probability that incoming photon will leave in a particular direction) has a thin profile**
  - We can predict the direction of the outgoing photon
- **For an ideal diffuse surfaces, the BRDF would be spherical**
  - The photon can travel along any of the direction with equal probability



from Sillion & Puech

## Bidirectional Ray tracing



from P. Heckbert

- **Idea: Trace forward light rays into scene as well as backward eye rays**
  - At diffuse surfaces, light rays additively “deposit” photons in **radiosity textures**, or “rexes”, where they are picked up by eye rays
- **The rays of the forward and backward pass “meet in the middle” to exchange information.**

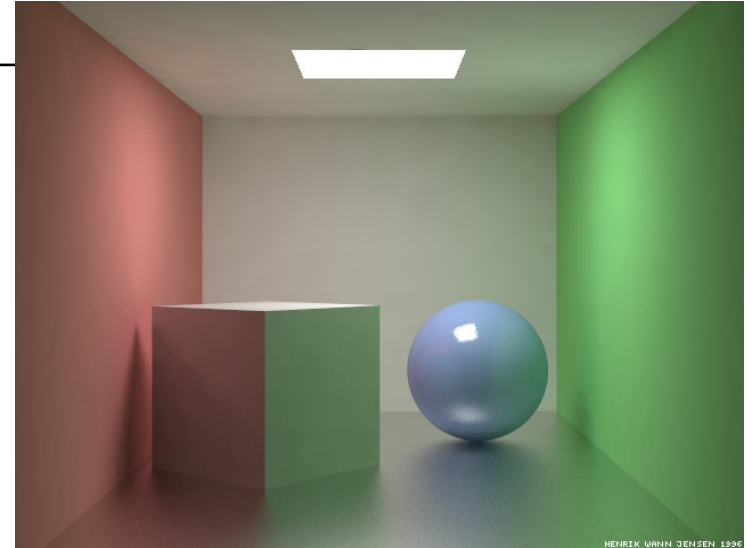
## Photon Mapping

---

- **Simulates the transport of individual photons**
- **Photons emitted from light sources**
- **Photons bounce off of specular surfaces**
- **Photons deposited on diffuse surfaces**
  - Held in a 3-D spatial data structure
  - Surfaces need not be parameterized
- **Photons collected by path tracing from eye**
  
- Jensen EGRW 95, 96

## Why Map Photons?

- High variance in Monte-Carlo renderings results in noise
- Collection of deposited photons into a “photon map” (a 3-D spatial data structure) provides a flux density estimate
- Flux samples filtered easier than path samples, resulting in error at lower frequencies
- Error is a result of bias, which decreases as the number of samples increase
- And, oh yeah, it’s a lot faster

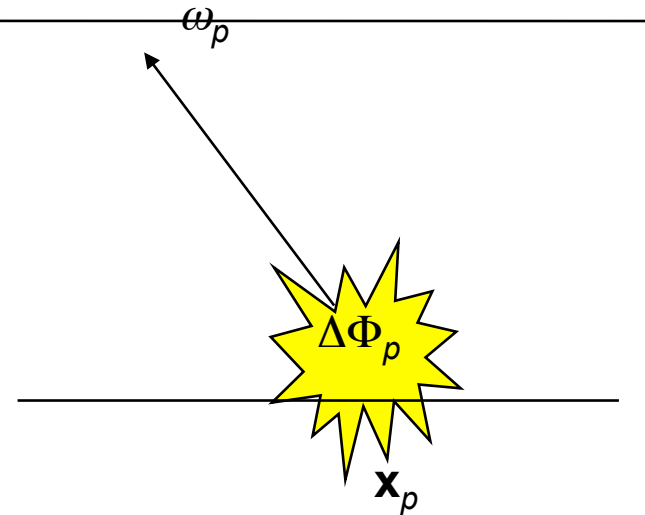


The scene above contains glossy surfaces, and was rendered in 50 minutes using photon mapping. The same scene took 6 hours for render with Radiance, a rendering system that used radiosity for diffuse reflection and path tracing for glossy reflection.



## What is a Photon?

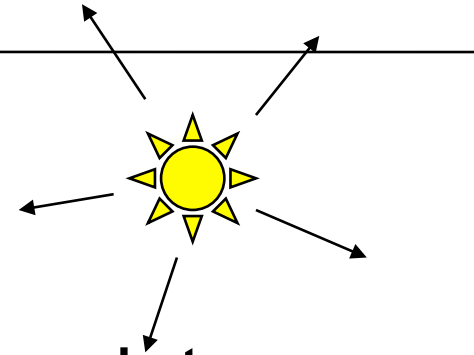
- **A photon  $p$  is a particle of light that carries flux  $DF$   $p(x_p, w_p)$** 
  - Power:  $DF_p$  – magnitude (in Watts) and color of the flux it carries, stored as an RGB triple
  - Position:  $x_p$  – location of the photon
  - Direction:  $w_p$  – the incident direction  $w_i$  used to compute irradiance
- **Photons vs. rays**
  - Photons propagate flux
  - Rays gather radiance



## Sources

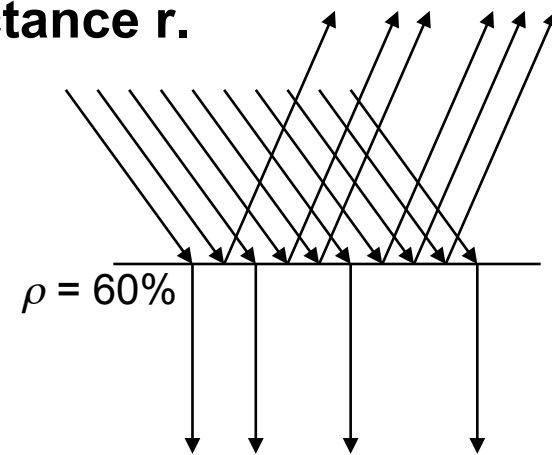
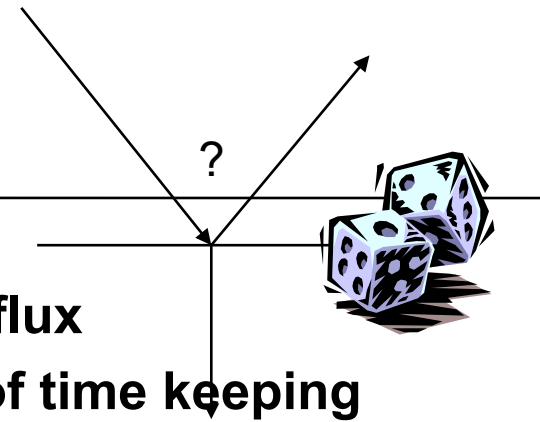
---

- **Point source**
  - Photons emitted uniformly in all directions
- **Power of source (W) distributed evenly among photons**
- **Flux of each photon equal to source power divided by total # of photons**
- **For example, a 60W light bulb would send out a total of 100K photons, each carrying a flux DF of 0.6 mW**
- **Photons sent out once per simulation, not continuously as in radiosity**



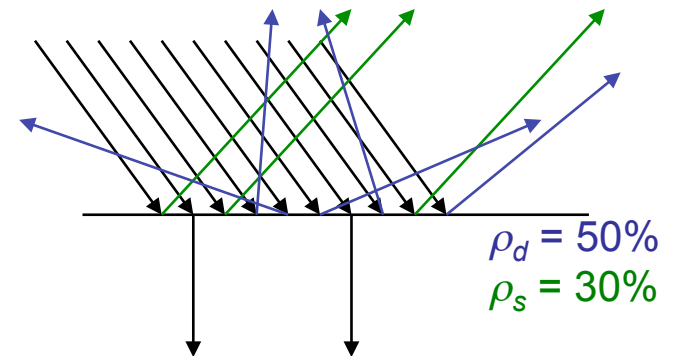
## Russian Roulette

- Reflected flux only a fraction of incident flux
- After several reflections, spending a lot of time keeping track of very little flux
- Instead, completely absorb some photons and completely reflect others at full power
- Spend time tracing fewer full power photons
- Probability of reflectance is the reflectance  $r$ .
- Probability of absorption is  $1 - r$ .



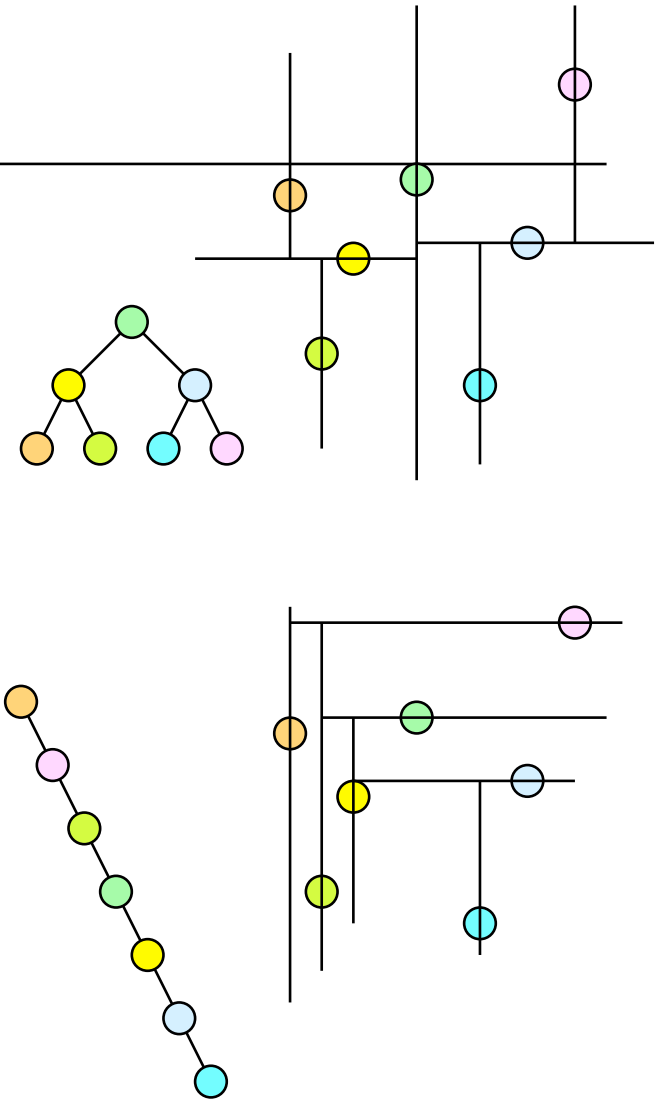
## Mixed Surfaces

- **Surfaces have specular and diffuse components**
  - $\rho_d$  – diffuse reflectance
  - $\rho_s$  – specular reflectance
  - $\rho_d + \rho_s < 1$  (conservation of energy)
- **Let  $z$  be a uniform random value from 0 to 1**
- **If  $z < \rho_d$  then reflect diffuse**
- **Else if  $z < \rho_d + \rho_s$  then reflect specular**
- **Otherwise absorb**



## Storing Photons

- **Uses a kd-tree – a sequence of axis-aligned partitions**
  - 2-D partitions are lines
  - 3-D partitions are planes
- **Axis of partitions alternates wrt depth of the tree**
- **Average access time is  $O(\log n)$**
- **Worst case  $O(n)$  when tree is severely lopsided**
- **Need to maintain a balanced tree, which can be done in  $O(n \log n)$**
- **Can find  $k$  nearest neighbors in  $O(k + \log n)$  time using a heap**





## Reflected Radiance

- Recall the reflected radiance equation

$$L_r(\mathbf{x}, \omega_r) = \int_{\Omega} f_r(\omega_i, \omega_r) L_i(\mathbf{x}, \omega_i) (N \cdot \omega_i) d\omega_i$$

- Convert incident radiance into incident flux

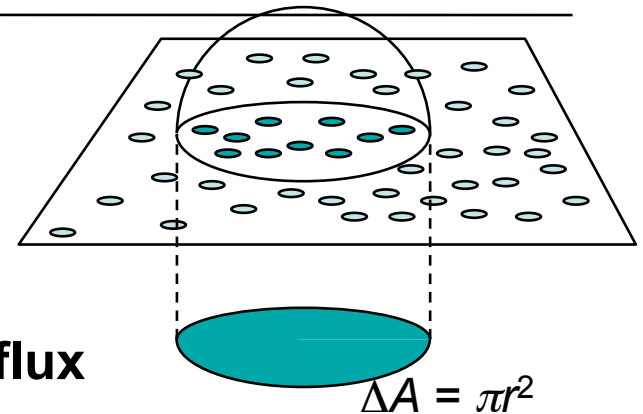
$$L_i(\mathbf{x}, \omega_i) = \frac{d^2\Phi_i(\mathbf{x}, \omega_i)}{(N \cdot \omega_i) d\omega_i dA_i}$$

- Reflected radiance in terms of incident flux

$$L_r(\mathbf{x}, \omega_r) = \int_{\Omega} f_r(\omega_i, \omega_r) \frac{d^2\Phi_i(\mathbf{x}, \omega_i)}{dA_i}$$

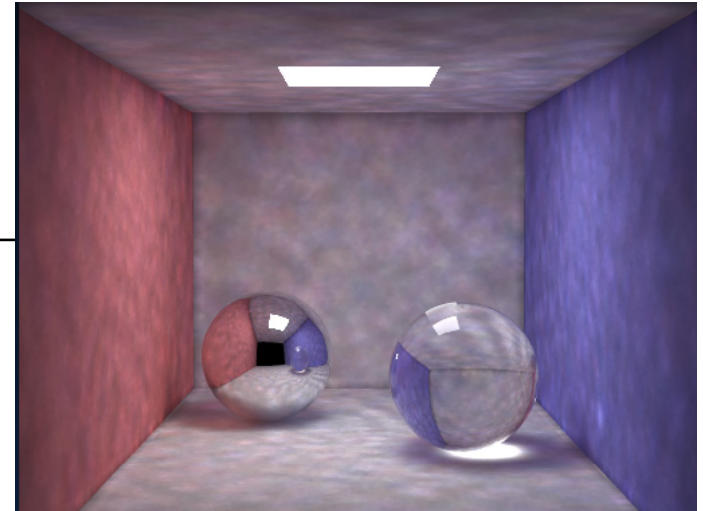
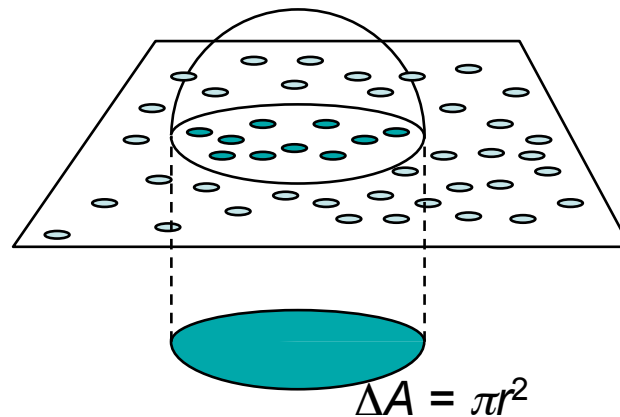
- Numerically

$$L_r(\mathbf{x}, \omega_r) \approx \frac{1}{\Delta A} \sum_{p=1}^n f_r(\omega_p, \omega_r) \Delta\Phi_p(\mathbf{x}, \omega_p)$$

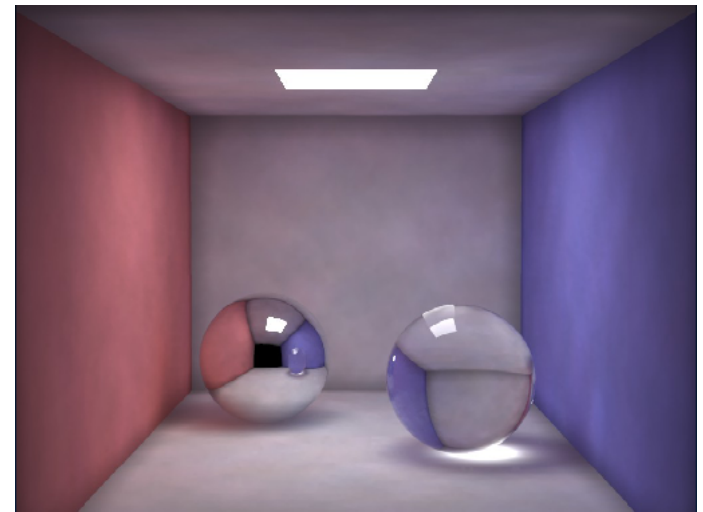


## How Many Photons?

- How big is the disk radius  $r$ ?
- Large enough that the disk surrounds the  $n$  nearest photons.
- The number of photons used for a radiance estimate  $n$  is usually between 50 and 500.



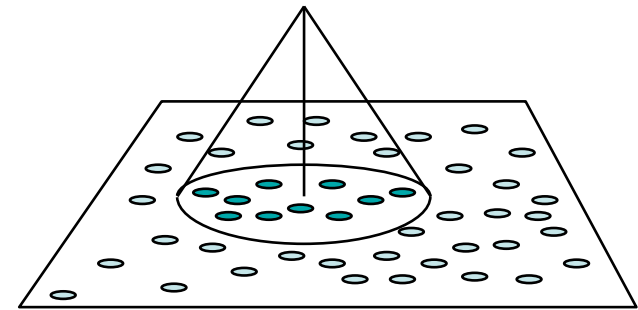
Radiance estimate using 50 photons



Radiance estimate using 500 photons

## Filtering

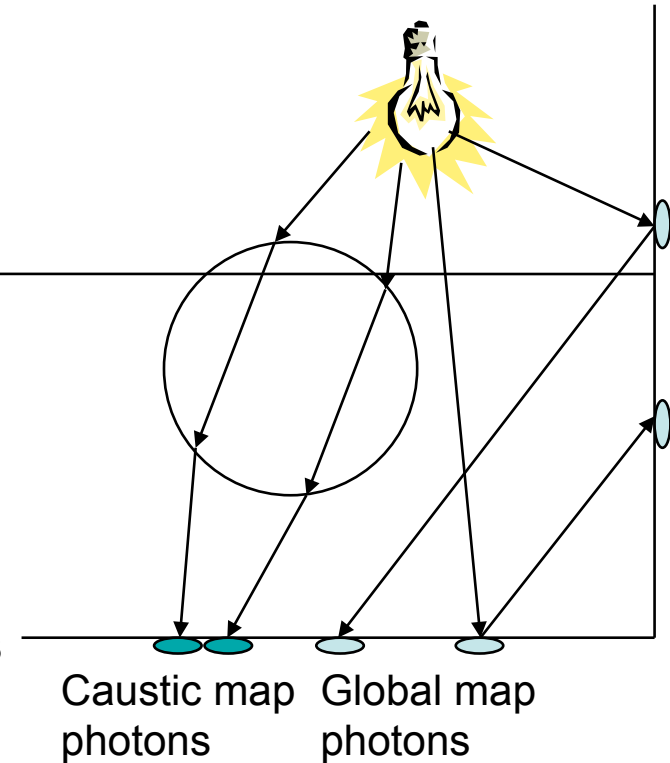
- Too few photons cause blurry results
- Simple averaging produces a box filtering of photons
- Photons nearer to the sample should be weighted more heavily
- Results in a cone filtering of photons



$$L_r(\mathbf{x}, \omega_r) \approx \frac{1}{\left(1 - \frac{2}{3k}\right)\pi r^2} \sum_{p=1}^n \left(1 - \frac{\|\mathbf{x} - \mathbf{x}_p\|}{kr}\right) f_r(\omega_p, \omega_r) \Delta\Phi_p(\mathbf{x}_p, \omega_p)$$

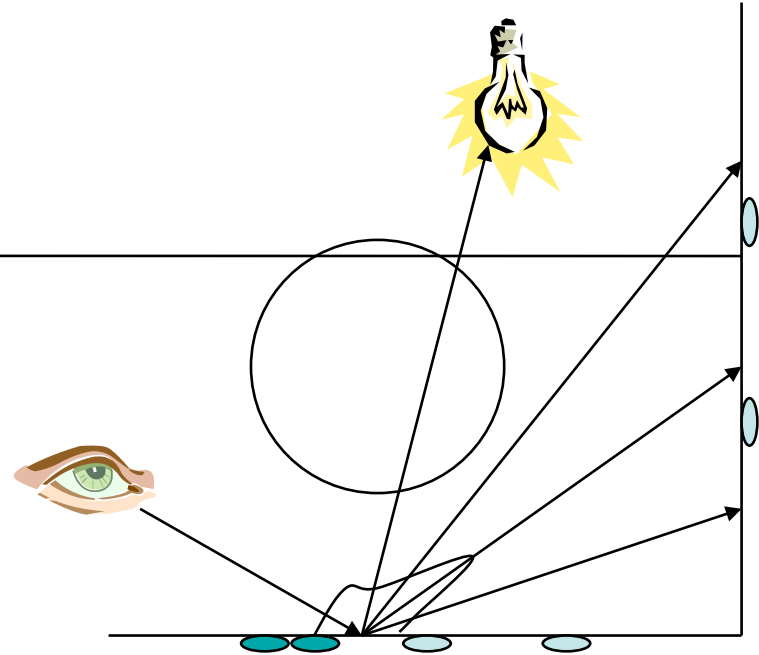
## Multiple Photon Maps

- **Global  $L(S|D)*D$  photon map**
  - Photon sticks to diffuse surface and bounces to next surface (if it survives Russian roulette)
  - Photons don't stick to specular surfaces
- **Caustic  $LSS*D$  photon map**
  - High resolution
  - Light source usually emits photons only in directions that hit the thing creating the caustic



## Rendering

- **Rendered by glossy-surface distributed ray tracing**
- **When ray hits first diffuse surface...**
  - Compute reflected radiance of caustic map photons
  - Ignore global map photons
  - Importance sample BRDF for as usual
  - Use global photon map to importance sample incident radiance function  $L_i$
  - Evaluate reflectance integral by casting rays and accumulating radiances from global photon map



First diffuse intersection. Return radiance of caustic map photons here, but ignore global map photons

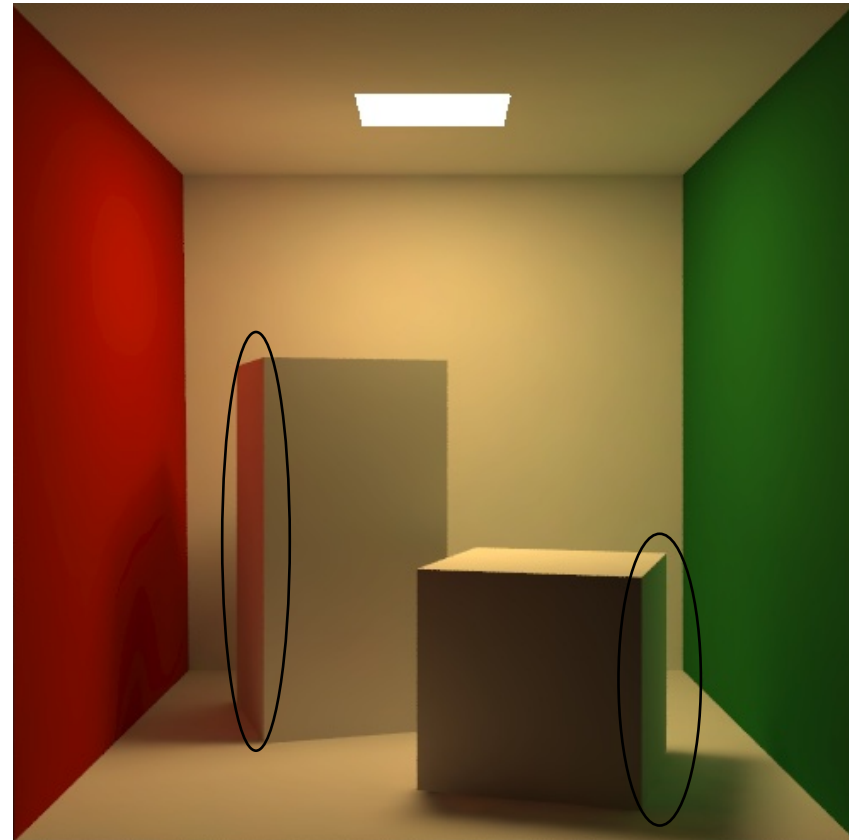
Use global map photons to return radiance when evaluating  $L_i$  at first diffuse intersection.





## Radiosity

- Handling cases such as  $LD^*E$
- “Color Bleeding”



courtesy of Cornell

## Radiosity

---

- **Based on heat transfer theory**
- **Assume diffuse reflections on all surfaces**
- **Solve for radiant exitance, or radiosity, (power per area) instead of radiance**
- **No directional variation, view independent**