The Rendering Equation & Irradiance Caching & Photon Mapping

The Light of Mies van der Rohe

Henrik Wann Jensen, SIGGRAPH 2000

The Light of Mies van der Rohe

Today

- Worksheet on Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

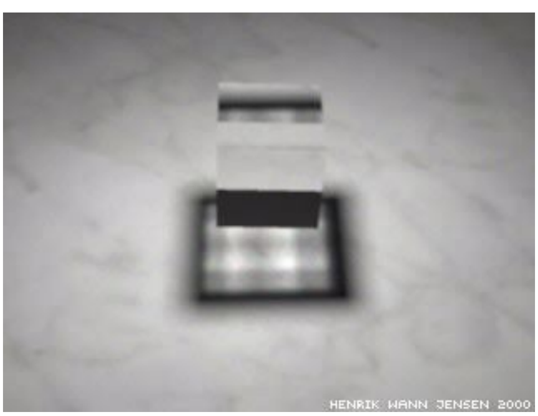
Pop Worksheet!

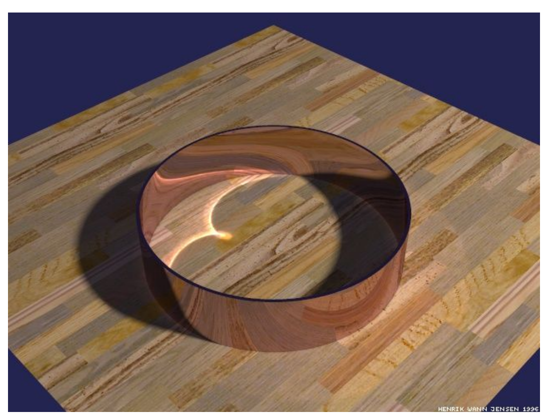
grey wall

Perform 5 iterations of progressive refinement radiosity

radiance @ iter 1	(10 , 10 , 10)			
undistributed @ iter 1	(0 , 0 , 0)			

Is this Traditional Ray Tracing?



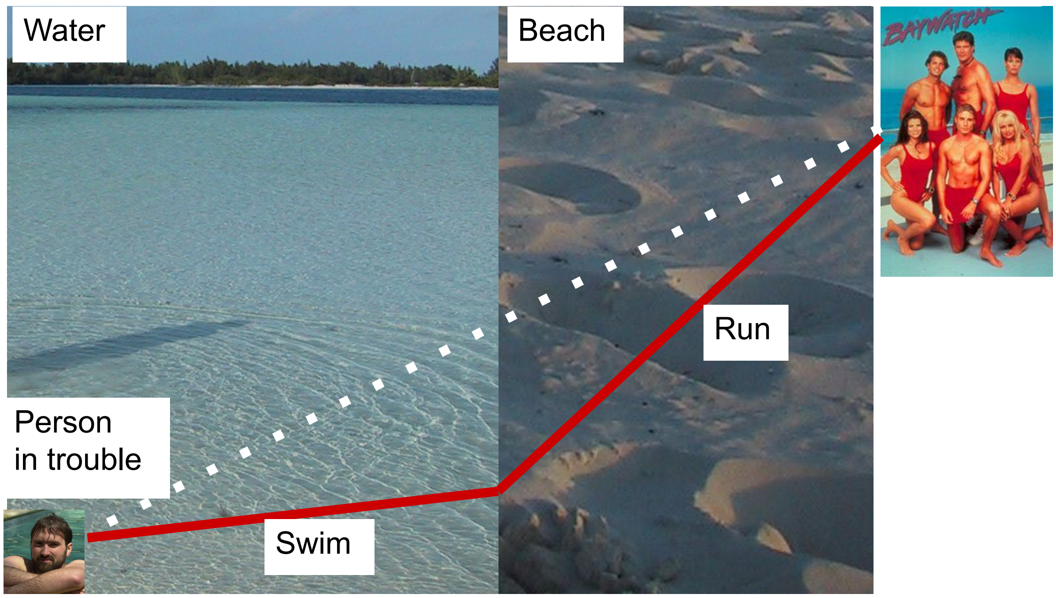


Images by Henrik Wann Jensen

No. Refraction and complex reflections for illumination are not handled properly in traditional (backward) ray tracing.

Refraction and the Lifeguard Problem

- Running is faster than swimming

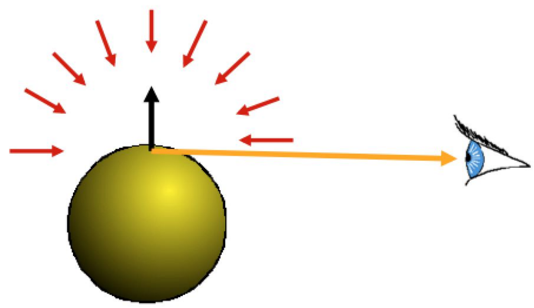


Today

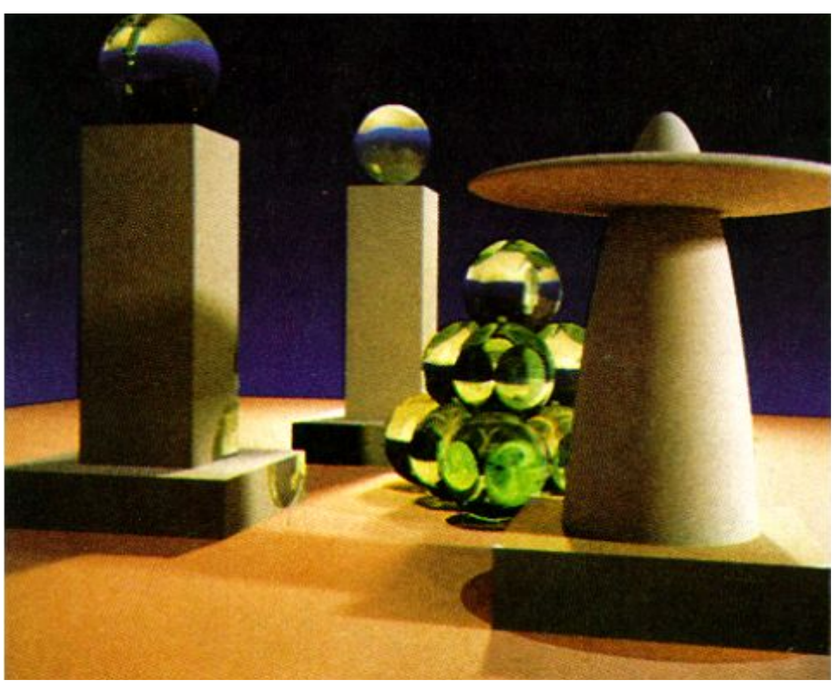
- Worksheet on Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

The Rendering Equation

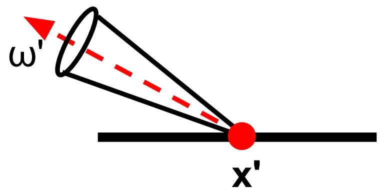
- Clean mathematical framework for light-transport simulation
- At each point, outgoing light in one direction is the integral of incoming light in all directions multiplied by reflectance property



"The Rendering Equation", Kajiya, SIGGRAPH 1986



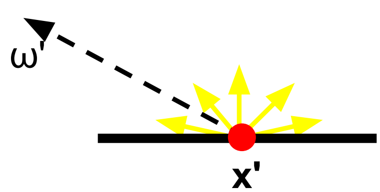
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

L (x',ω') is the radiance from a point on a surface in a given direction ω'

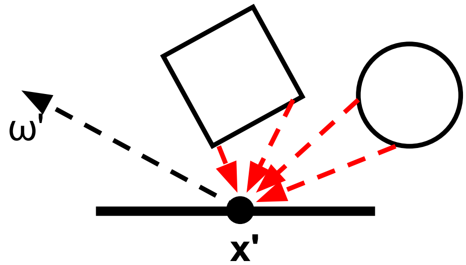
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

E(x',ω') is the emitted radiance from a point: E is non-zero only if x' is emissive (a light *source*)

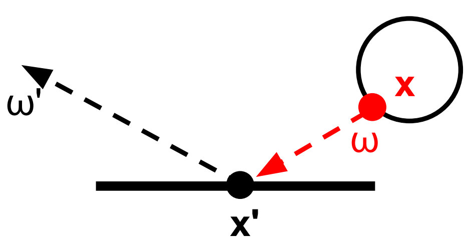
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x')\,dA$$

Sum the contribution from all of the other surfaces in the scene

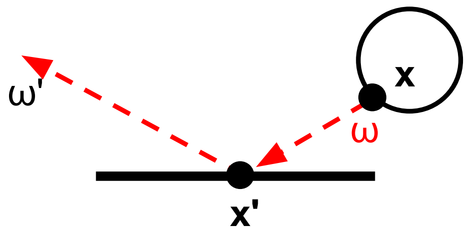
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x')\,dA$$

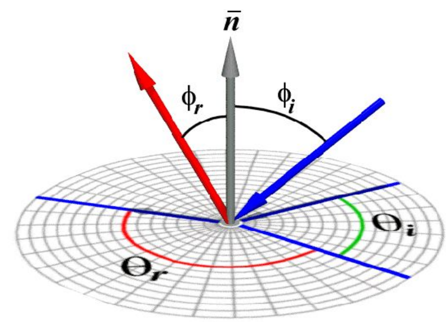
For each x, compute $L(x, \omega)$, the radiance at point x in the direction ω (from x to x')

The Rendering Equation

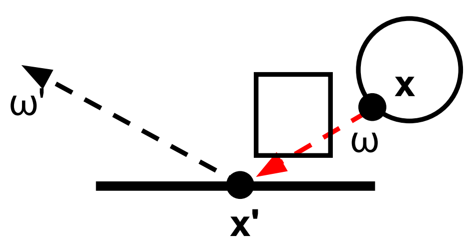


$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

scale the contribution by $\rho_{x'}(\omega,\omega')$, the reflectivity (BRDF) of the surface at x'



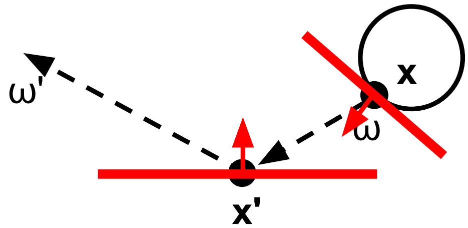
The Rendering Equation



$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

For each x, compute V(x,x'),
the visibility between x and x':
1 when the surfaces are unobstructed
along the direction ω, 0 otherwise

The Rendering Equation

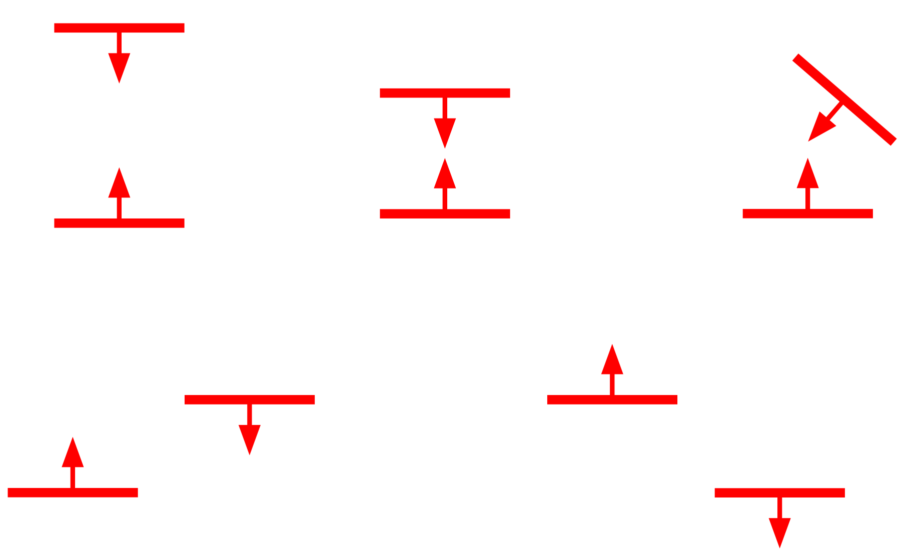


$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

For each x, compute G(x, x'), which describes the on the geometric relationship between the two surfaces at x and x’

Intuition about G(x,x')?

- Which arrangement of two surfaces will yield the greatest transfer of light energy? Why?



Rendering Equation → Radiosity

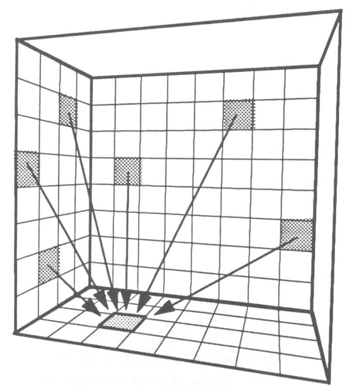
$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

Radiosity assumption:
perfectly diffuse surfaces (not directional)

$$B_{x'} = E_{x'} + \rho_{x'} \int B_x \, G(x,x') V(x,x')$$

discretize

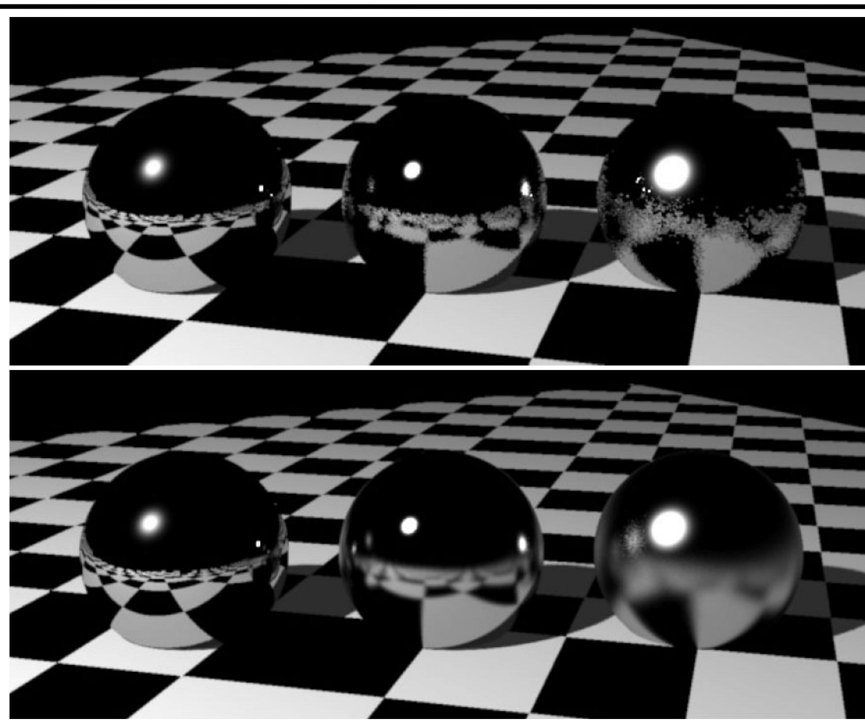
$$B_i = E_i + \rho_i \sum_{j=1}^{n} F_{ij} B_j$$



Questions?

1 glossy sample per pixel

256 glossy samples per pixel

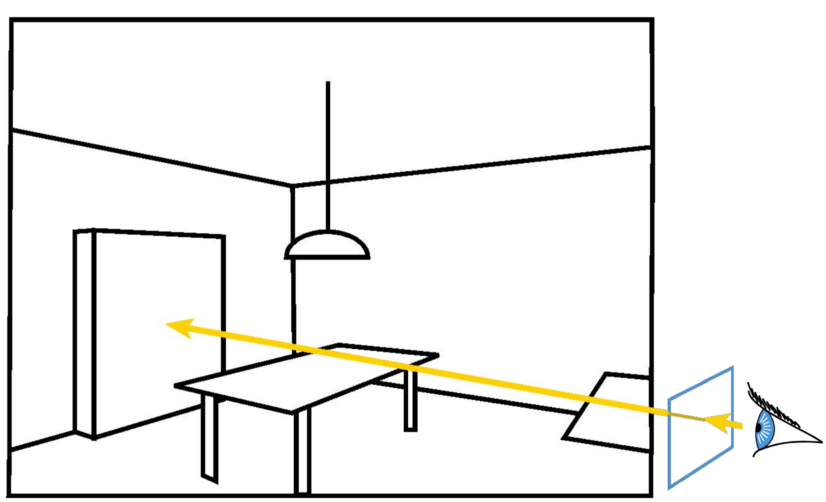


Today

- Worksheet on Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

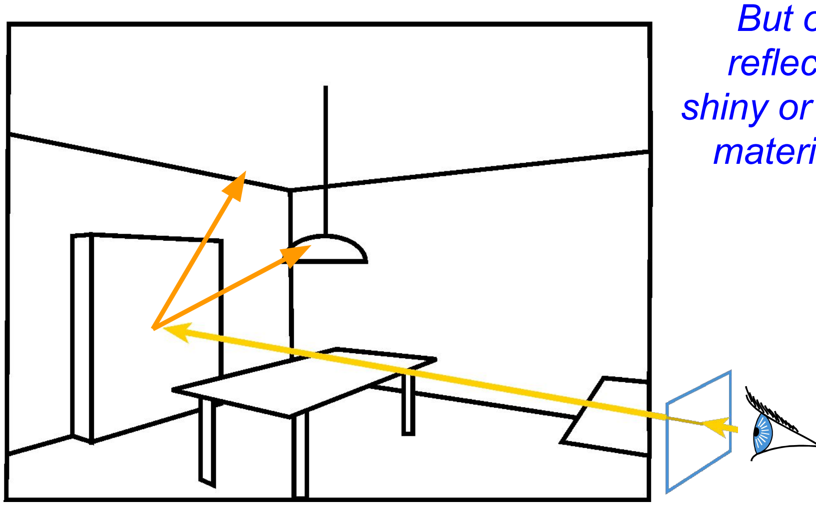
Ray Casting

- Cast a ray from the eye through each pixel



Ray Tracing

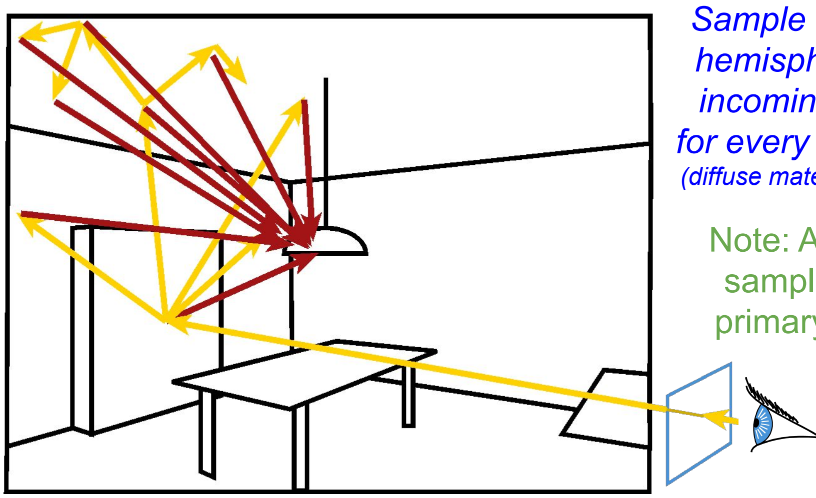
- Cast a ray from the eye through each pixel
- Trace secondary rays (light, reflection, refraction)



But only reflect off shiny or glossy materials...

Monte Carlo Ray Tracing

- Cast a ray from the eye through each pixel
- Cast random rays to accumulate radiance contribution
  - Recurse to solve the Rendering Equation

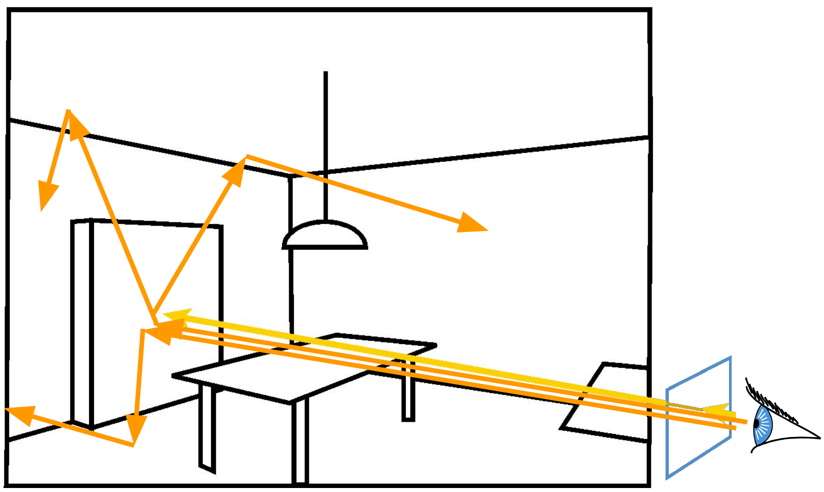


Sample the full hemisphere of incoming light for every surface
(diffuse materials too!)

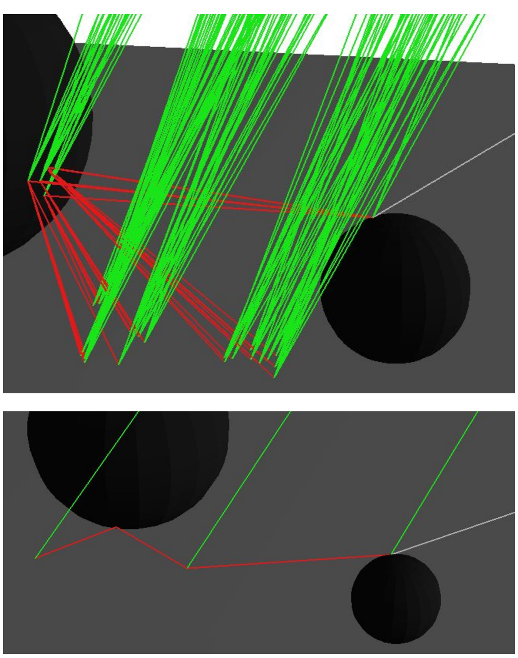
Note: Always sample the primary light

(Monte Carlo) Path Tracing

- Trace only one secondary ray per recursion
- But send many primary rays per pixel
  (performs antialiasing as well)



Ray Tracing vs. Path Tracing



2 bounces
5 glossy samples
5 shadow samples

How many rays cast per pixel?

1 main ray + 5 shadow rays +
5 glossy rays + 5x5 shadow rays +
5*5 glossy rays + 5x5x5 shadow rays
= 186 rays

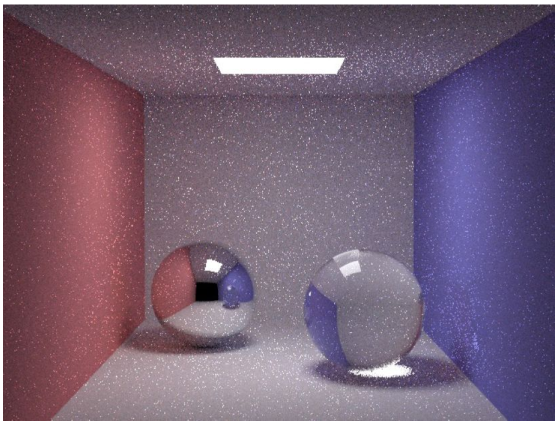
How many 3 bounce paths can we trace per pixel for the same cost?

186 rays / 8 ray casts per path
= ~23 paths

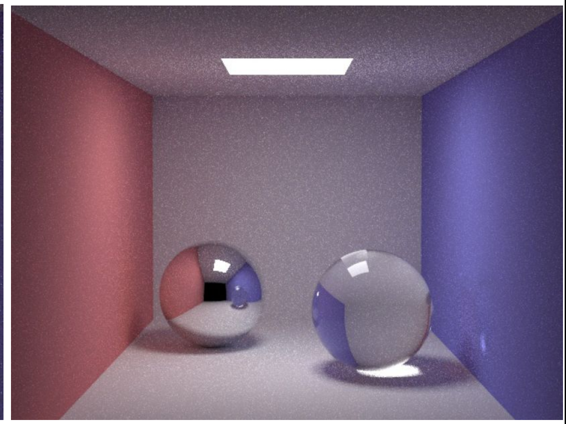
Which will probably have less error?

Questions?

10 paths/pixel



100 paths/pixel



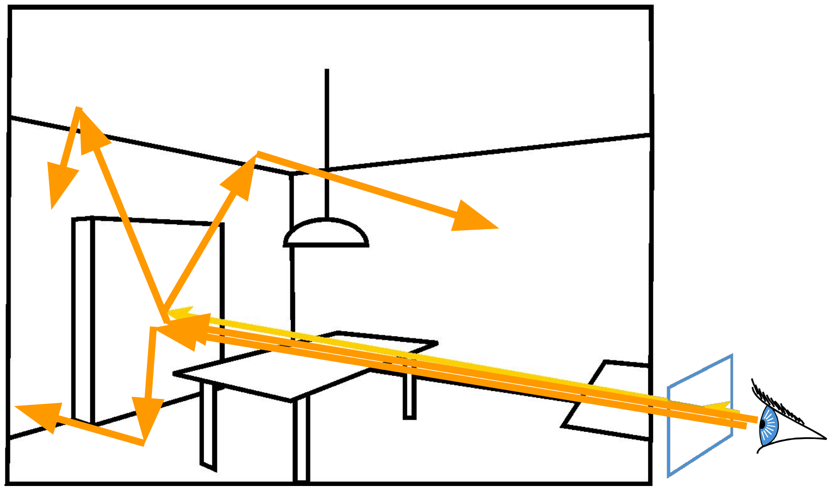
Images from Henrik Wann Jensen

Today

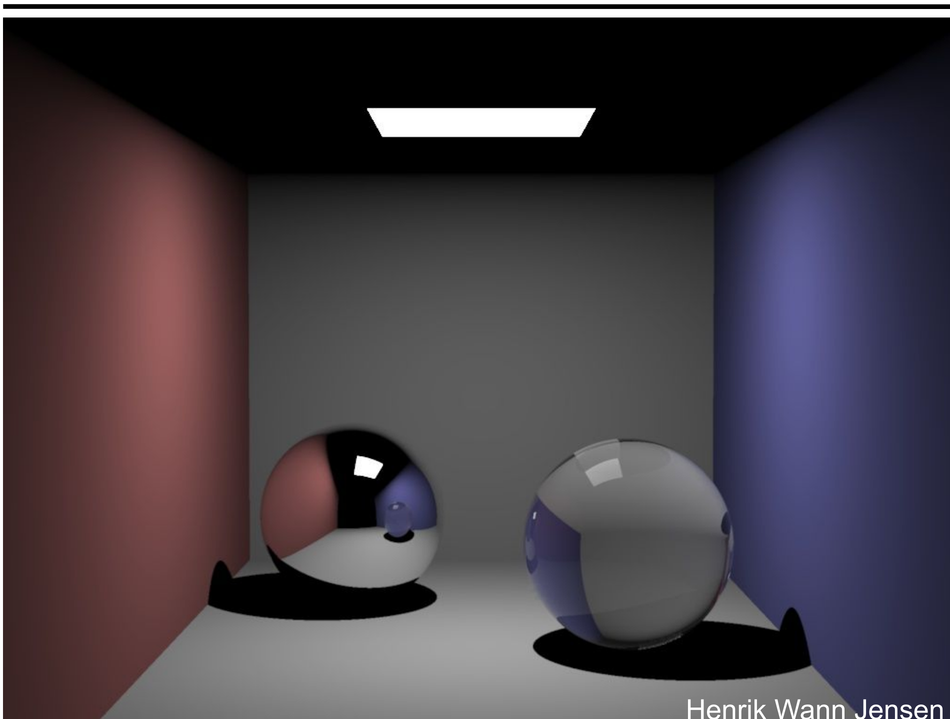
- Worksheet on Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

Path Tracing is costly

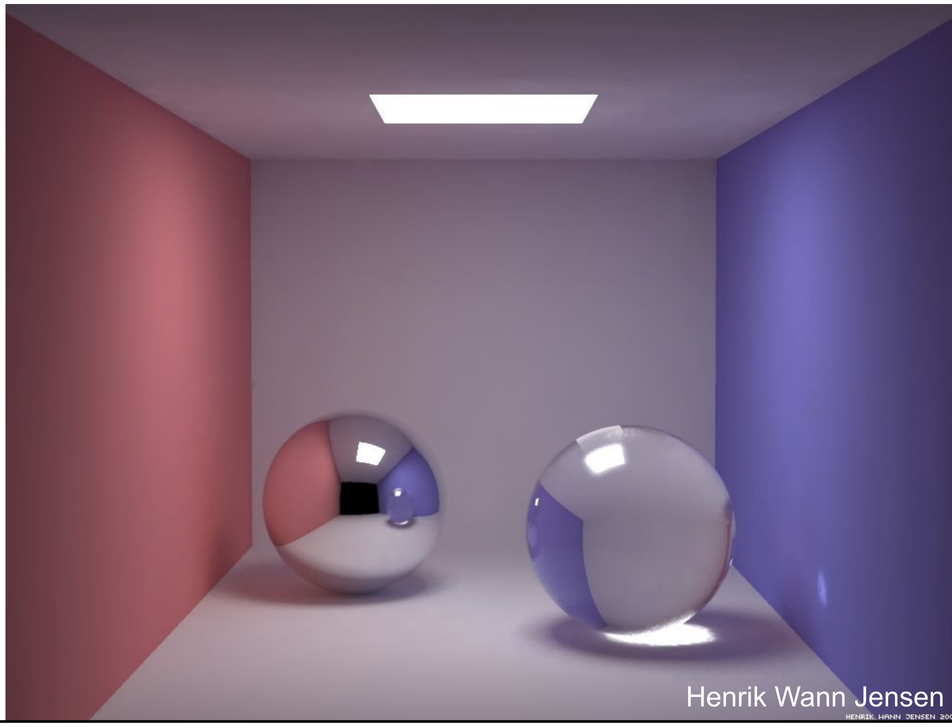
- Needs tons of rays per pixel



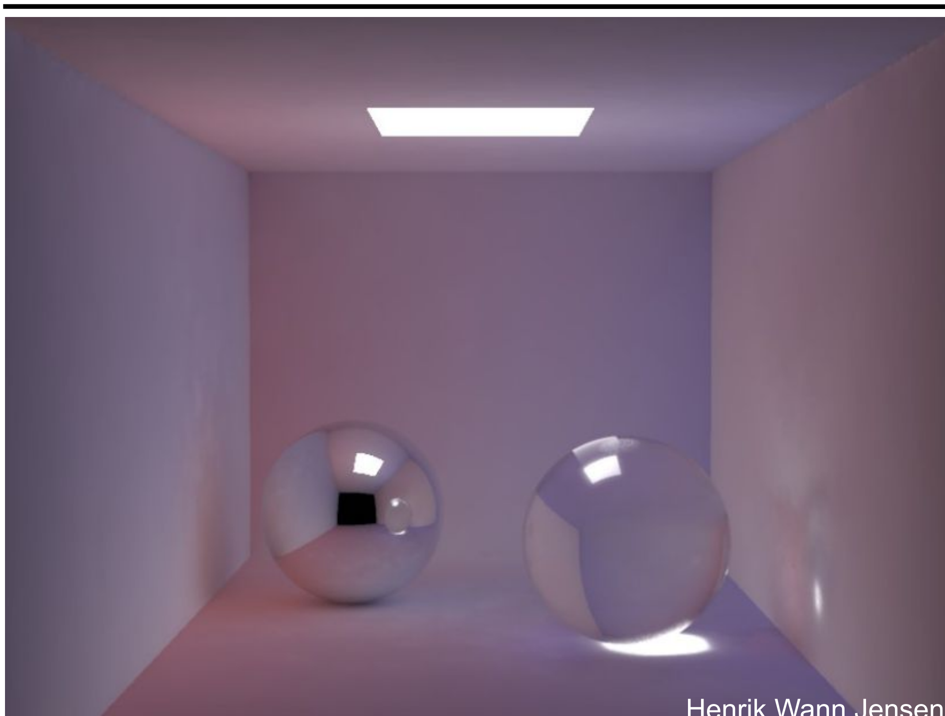
Direct Illumination



Global Illumination

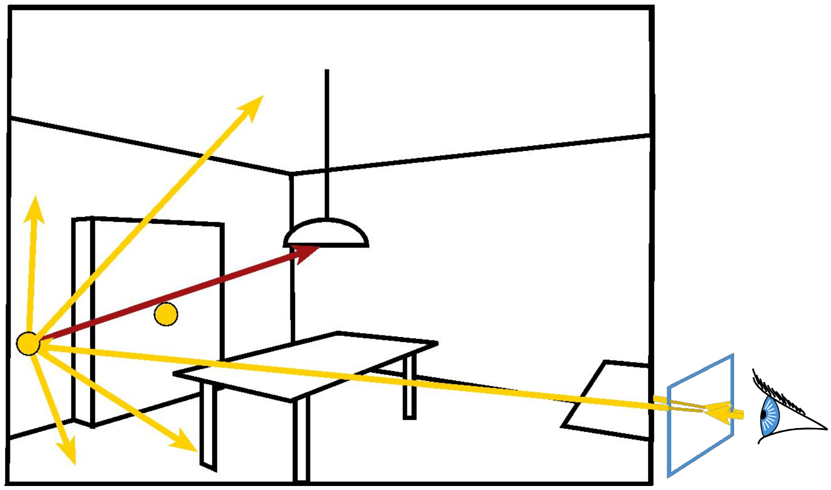


Indirect Illumination: smooth



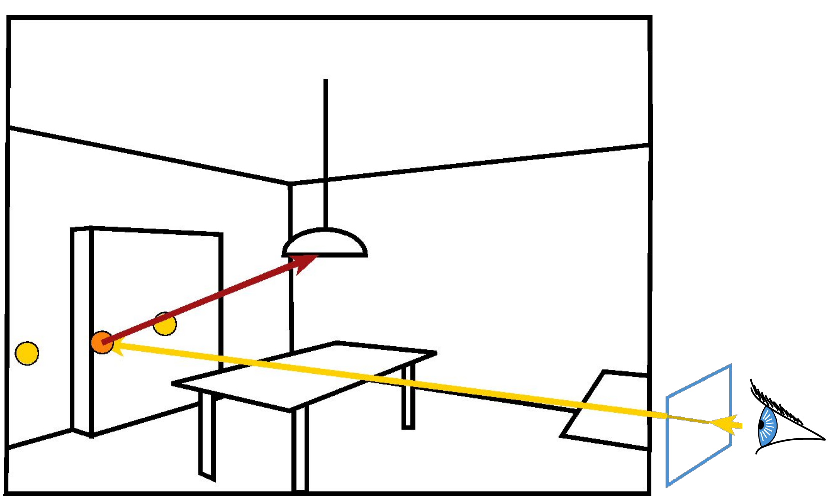
Irradiance Cache

- The indirect illumination is smooth
- Store the indirect illumination

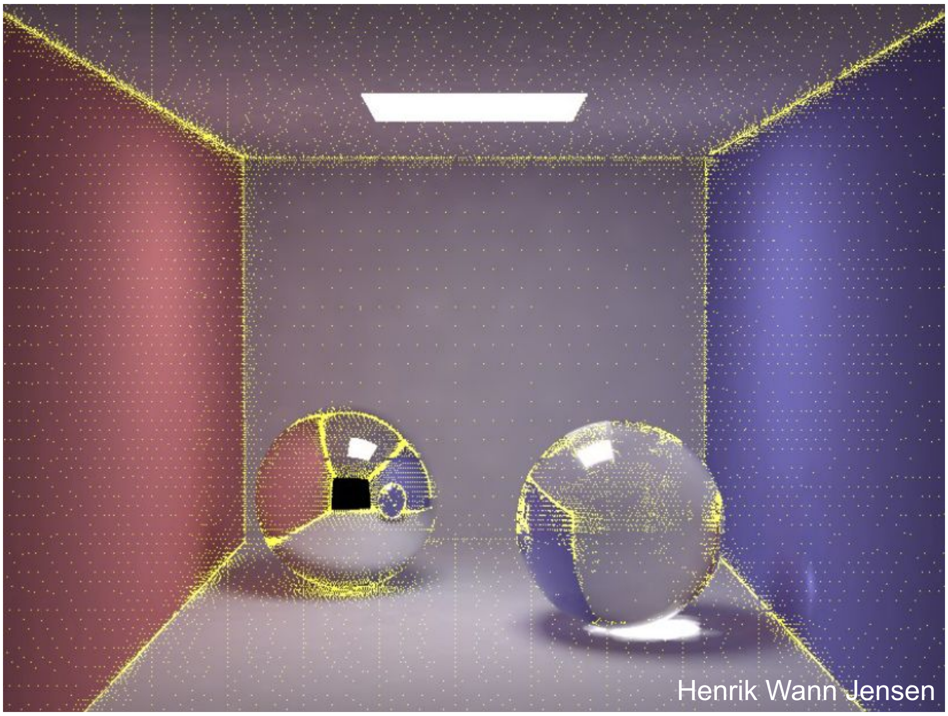


Irradiance Cache

- Interpolate nearby cached values
- But do full calculation for direct lighting

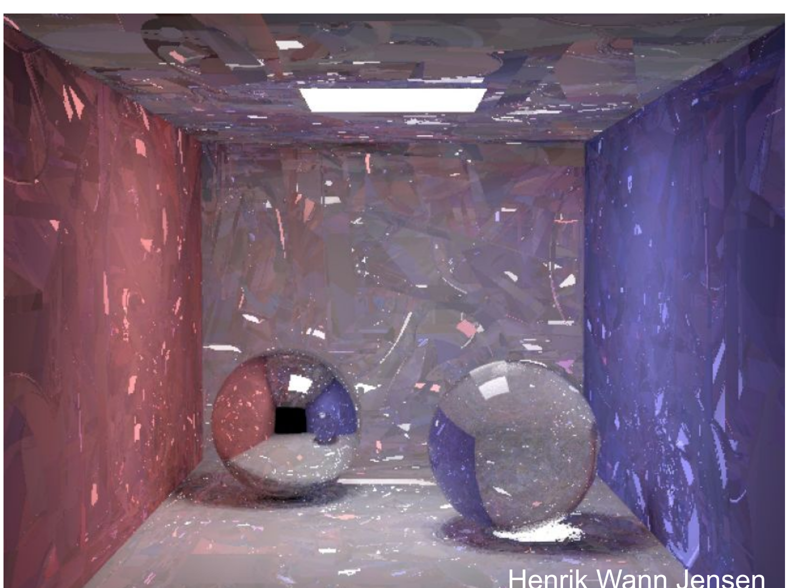


Irradiance Cache



Questions?

- Why do we need “good” random numbers?
  - With a fixed random sequence, we see the structure in the error

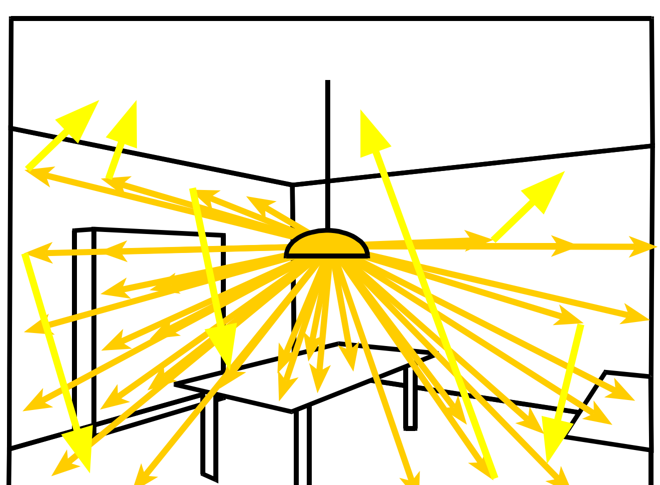


Today

- Worksheet on Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs.
  Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

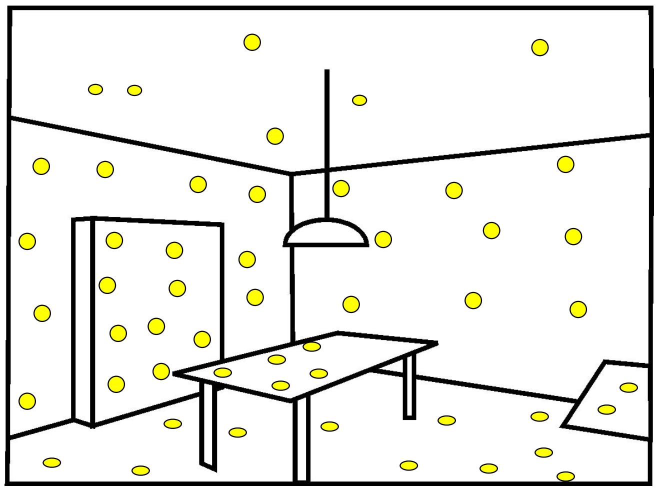
Photon Mapping

- Preprocess: cast rays from light sources
  - independent of viewpoint



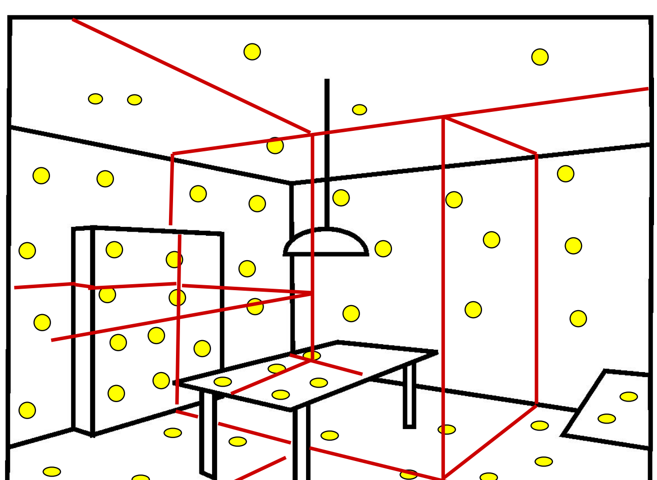
Photon Mapping

- Store photons
  - position + light power + incoming direction



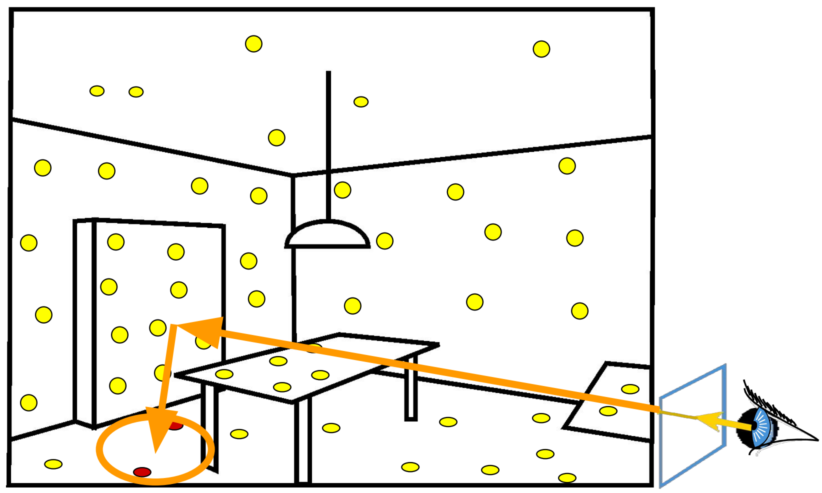
Storing the Photon Map

- Efficiently store photons for fast access
- Use hierarchical spatial structure (kd-tree)

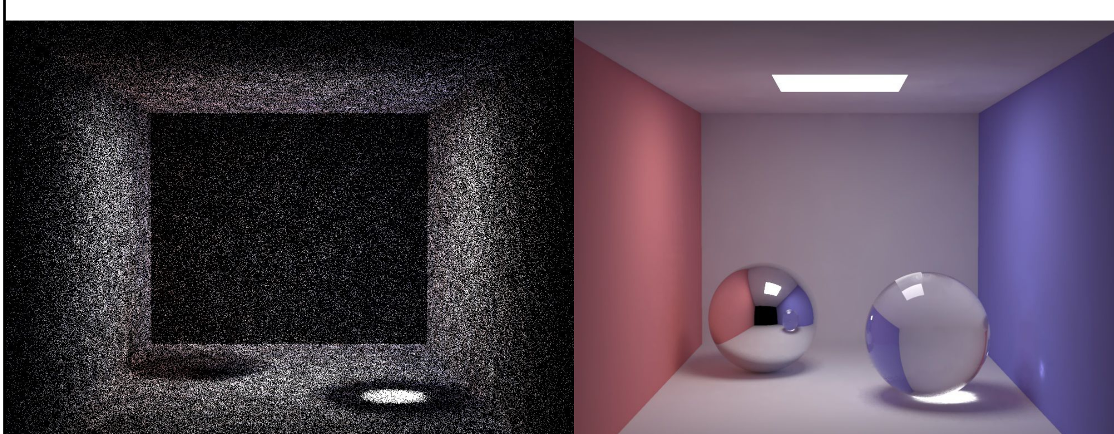


Rendering with Photon Map

- Cast primary rays
- For secondary rays: reconstruct irradiance using k closest photons
- Combine with irradiance caching and other techniques

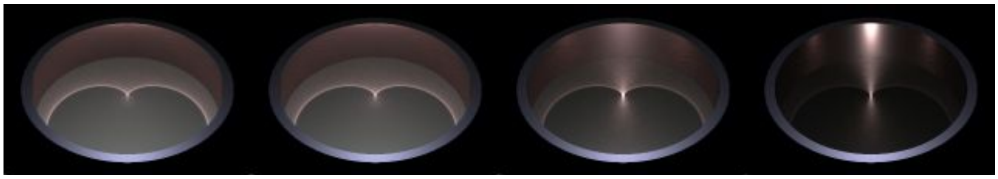


Photon Map Results

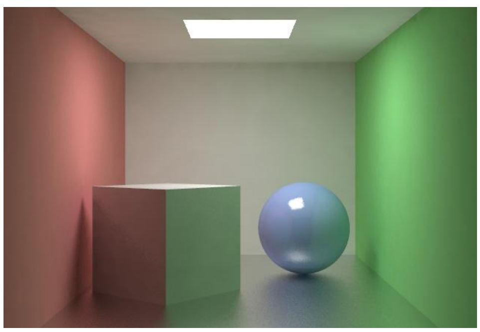


Readings for Today:

- "Rendering Caustics on Non-Lambertian Surfaces", Henrik Wann Jensen, *Graphics Interface* 1996.

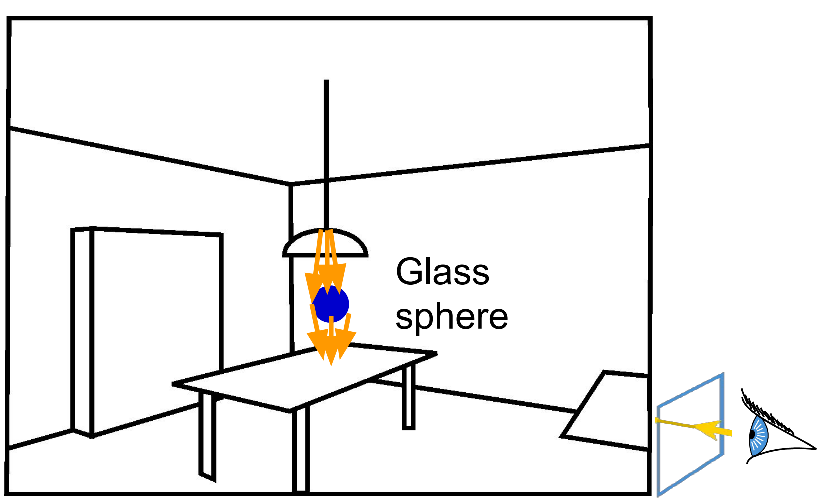


- "Global Illumination using Photon Maps", Henrik Wann Jensen, *Rendering Techniques* 1996.



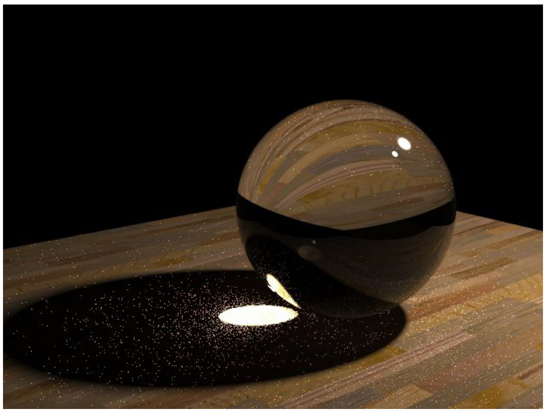
Photon Mapping - Caustics

- Special photon map for specular reflection and refraction

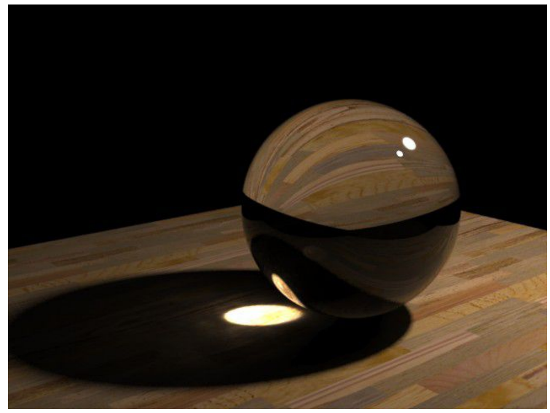


Comparison

Path Tracing
1000 paths/pixel

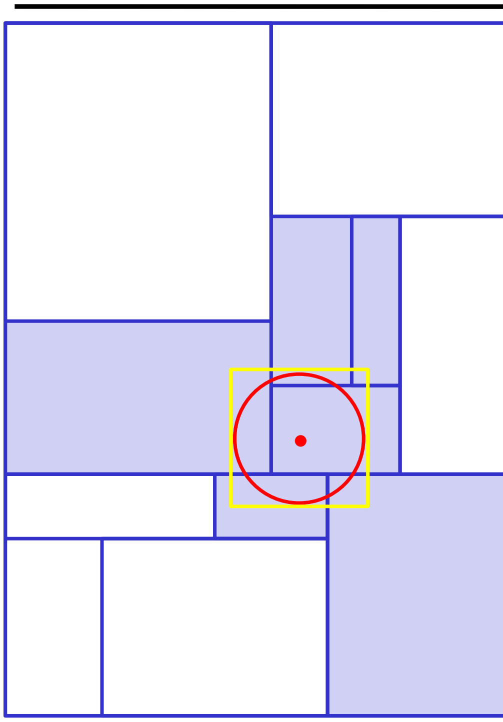


Photon mapping



(similar rendering time)

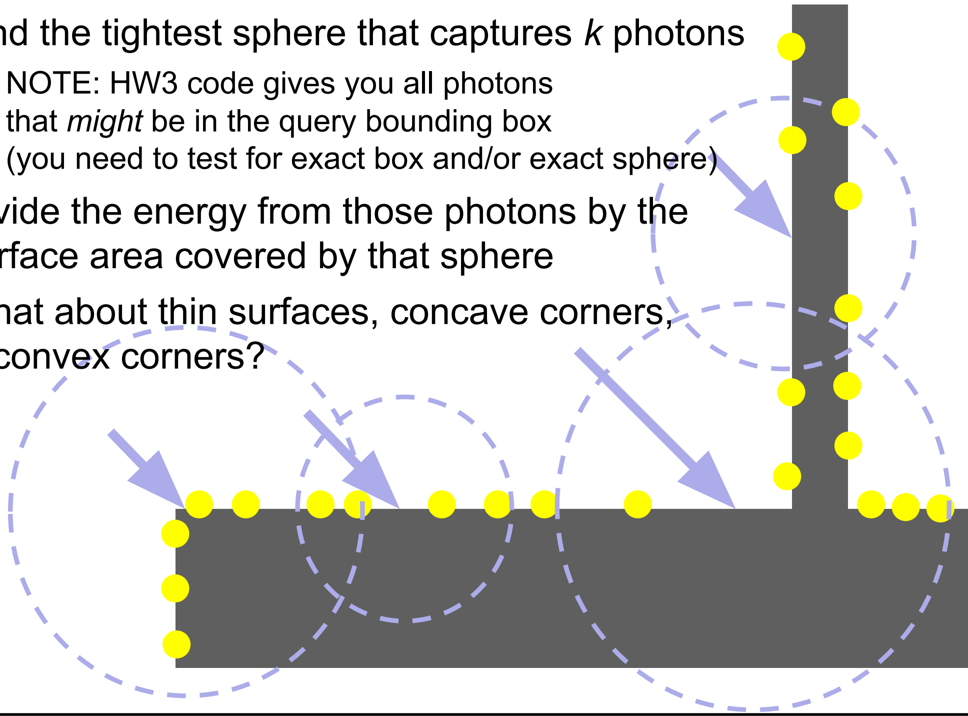
HW3: Photons in the k-d tree



- You start with query point & radius (red)
- You give the KDTree::CollectPhotonsInBox function a bounding box (yellow)
- The algorithm finds all k-d tree cells that overlap with bounding box (blue)
- The function returns all photons in those cells
- *You need to discard all photons not in your original query radius*

Closest Photon Details

- Find the tightest sphere that captures *k* photons
  - NOTE: HW3 code gives you all photons that *might* be in the query bounding box (you need to test for exact box and/or exact sphere)
- Divide the energy from those photons by the surface area covered by that sphere
- What about thin surfaces, concave corners, & convex corners?



Today

- Worksheet on Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

Ray Grammar

- Classify local interaction:

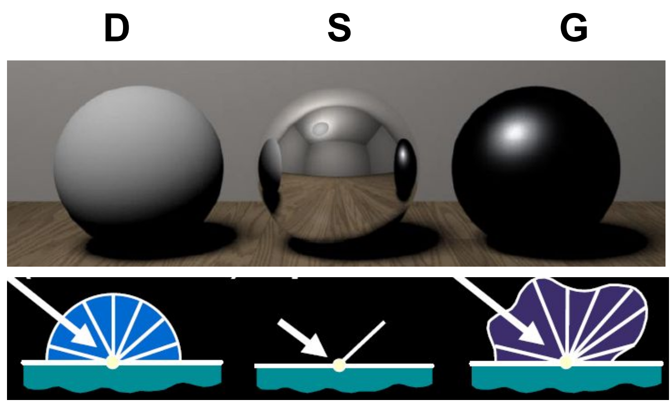
  E = eye

  L = light

  S = perfect specular reflection or refraction

  G = glossy scattering

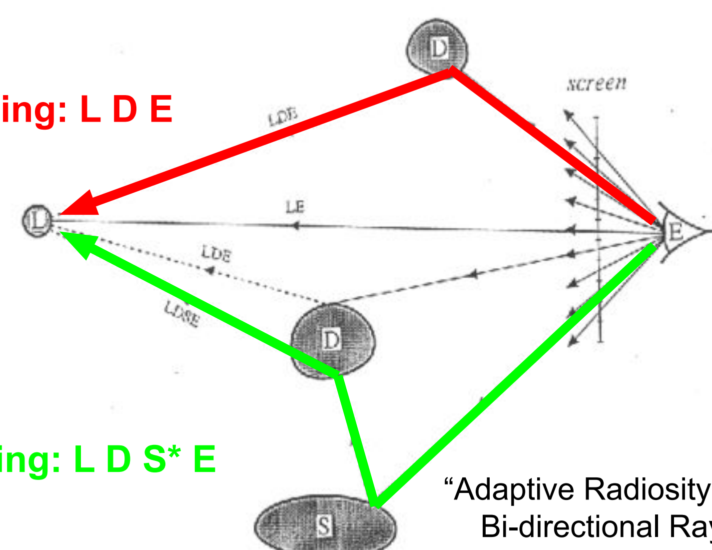
  D = diffuse scattering



From Dutre et al.'s slides

Classic Ray Casting/Tracing

Ray casting: L D E

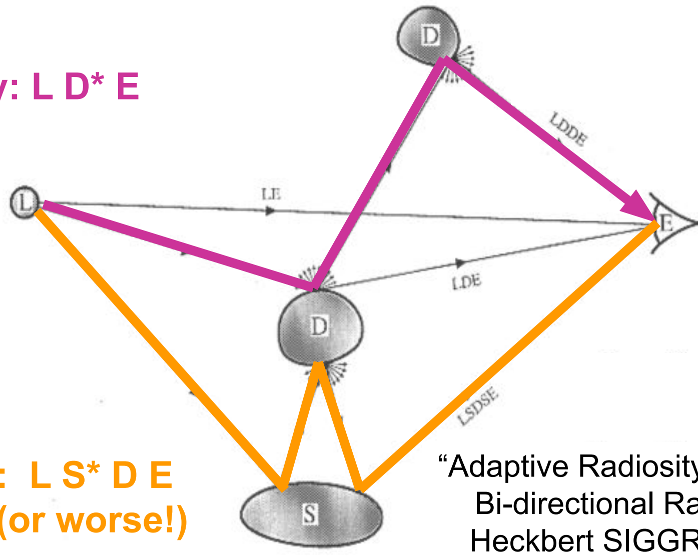


Ray tracing: L D S* E

"Adaptive Radiosity Textures for Bi-directional Ray Tracing"
Heckbert SIGGRAPH 1990

Photon Tracing

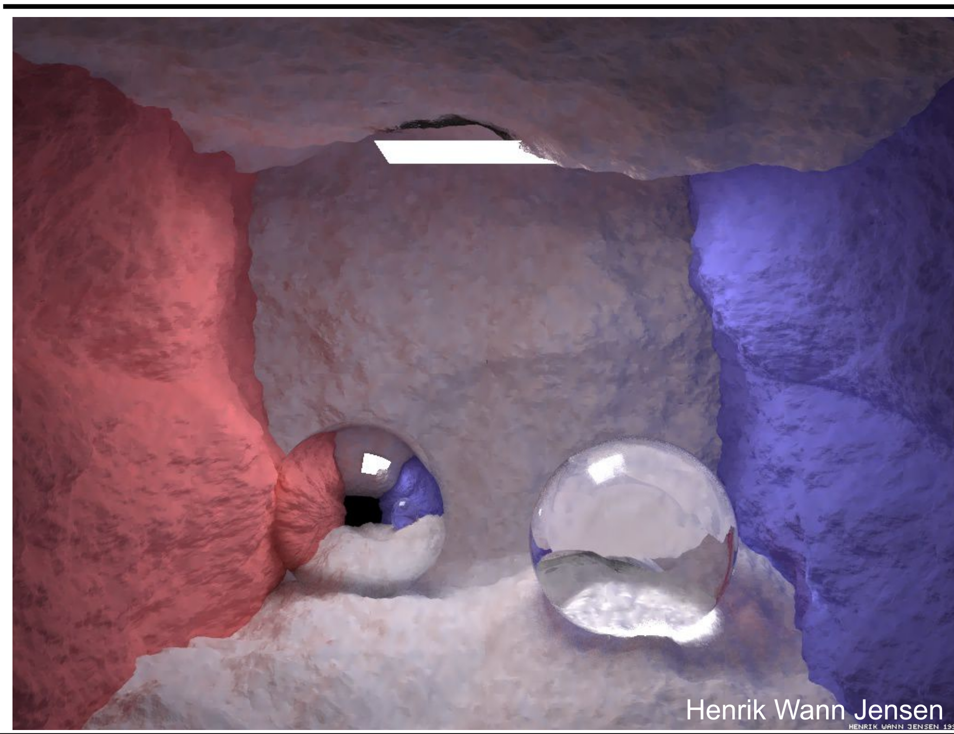
Radiosity: L D* E



Caustics: L S* D E
(or worse!)

"Adaptive Radiosity Textures for Bi-directional Ray Tracing"
Heckbert SIGGRAPH 1990

Questions?



Today

- Worksheet on Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

Readings for Next Time: *(pick one)*

"Correlated Multi-Jittered Sampling",
Andrew Kensler, Pixar Technical Memo, 2013

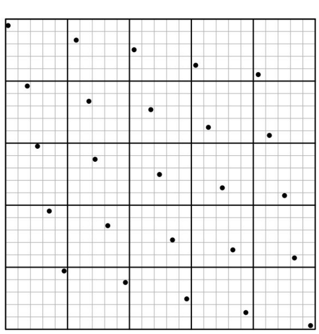


Figure 1: The canonical arrangement. Heavy lines show the boundaries of the 2D jitter cells. Light lines show the horizontal and vertical substrata of N-rooks sampling. Samples are jittered within the subcells.

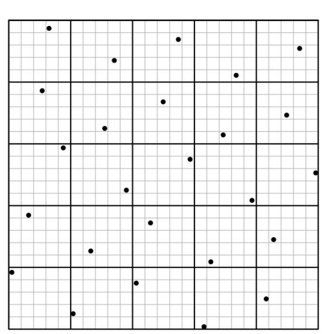


Figure 3: With correlated shuffling.

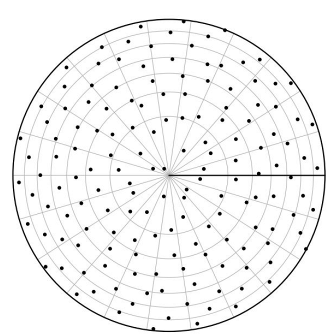


Figure 9: Polar warp with $m = 22, n = 7$.

[9]G. J. Ward and P. S. Heckbert. Irradiance gradients. In *Third Eurographics Rendering Workshop*, pages 85–98, May 1992.

Readings for Next Time: *(pick one)*

"Implicit Visibility and Antiradiance for Interactive Global Illumination"

Dachsbacher, Stamminger, Drettakis, and Durand
Siggraph 2007