CSCI 4530/6530 Advanced Computer Graphics

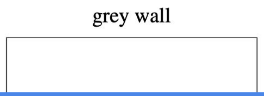
https://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S25/

Lecture 14: The Rendering Equation & Irradiance Caching & Photon Mapping

The Light of Mies van der Rohe

Henrik Wann Jensen
SIGGRAPH 2000

Worksheet: Progressive Radiosity



Perform 5 iterations of progressive refinement radiosity.

undistributed @ iter 1	(0 , 0 , 0)			

Final Project Brainstorming

- Each student should post two different project ideas on the forum. For each idea:
    - Describe the idea, motivation, & an … mple … nti… sult…
    - A significant/interesting technical imple…
- Have you already decided on one idea…
- Do you already have a partner? Who? (e… an idea and/or a partner everyone must post … s)
- Part 1: Due Friday 2/28 @ 11:59pm
- Teams of 2 strongly recommended (individuals & teams >2 require instructor permission)
- Projects from prior terms are on the website

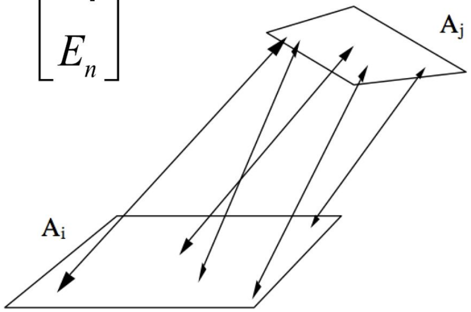
Part 2: Peer Feedback
Due Monday 3/10 @ 11:59pm
Reply to 3 of your classmates' ideas posts!

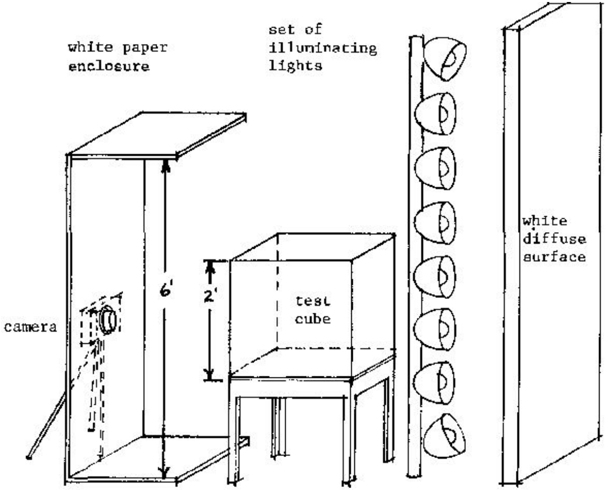
Last Time?

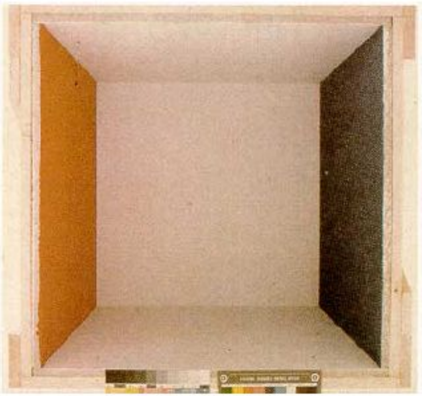
- Cornell Box
- Form Factors
- Radiosity Matrix
- Progressive Radiosity (Incremental Solver)

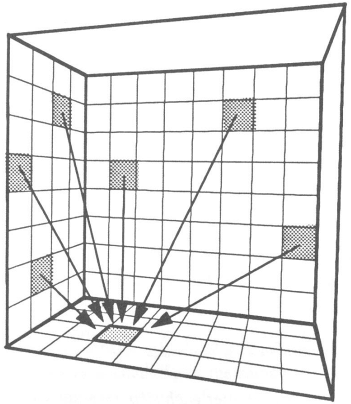
solve for B_i

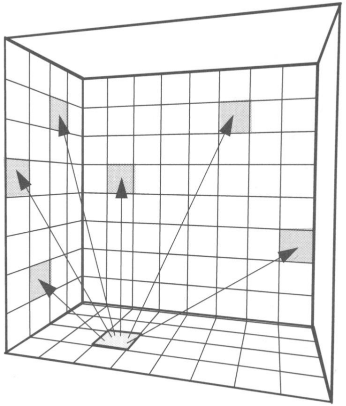
$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & & \\ \vdots & & \ddots & \\ -\rho_n F_{n1} & \cdots & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$









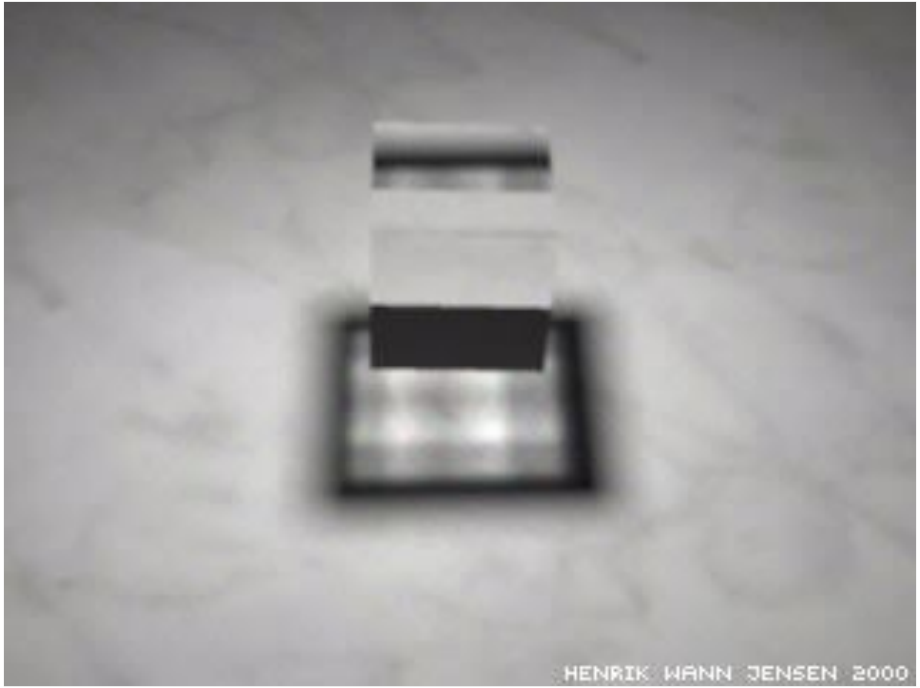


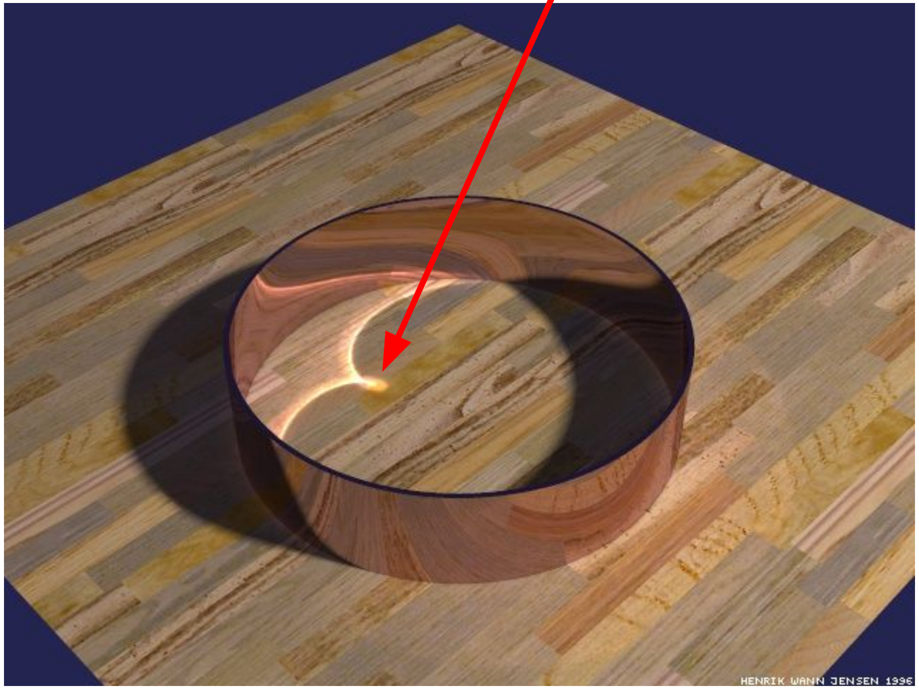
Today

- Worksheet: 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

Is this Traditional Ray Tracing?

We don't know that this spot is extra bright.
We don't know what directions to cast rays
from diffuse surface to find this mirror caustic.



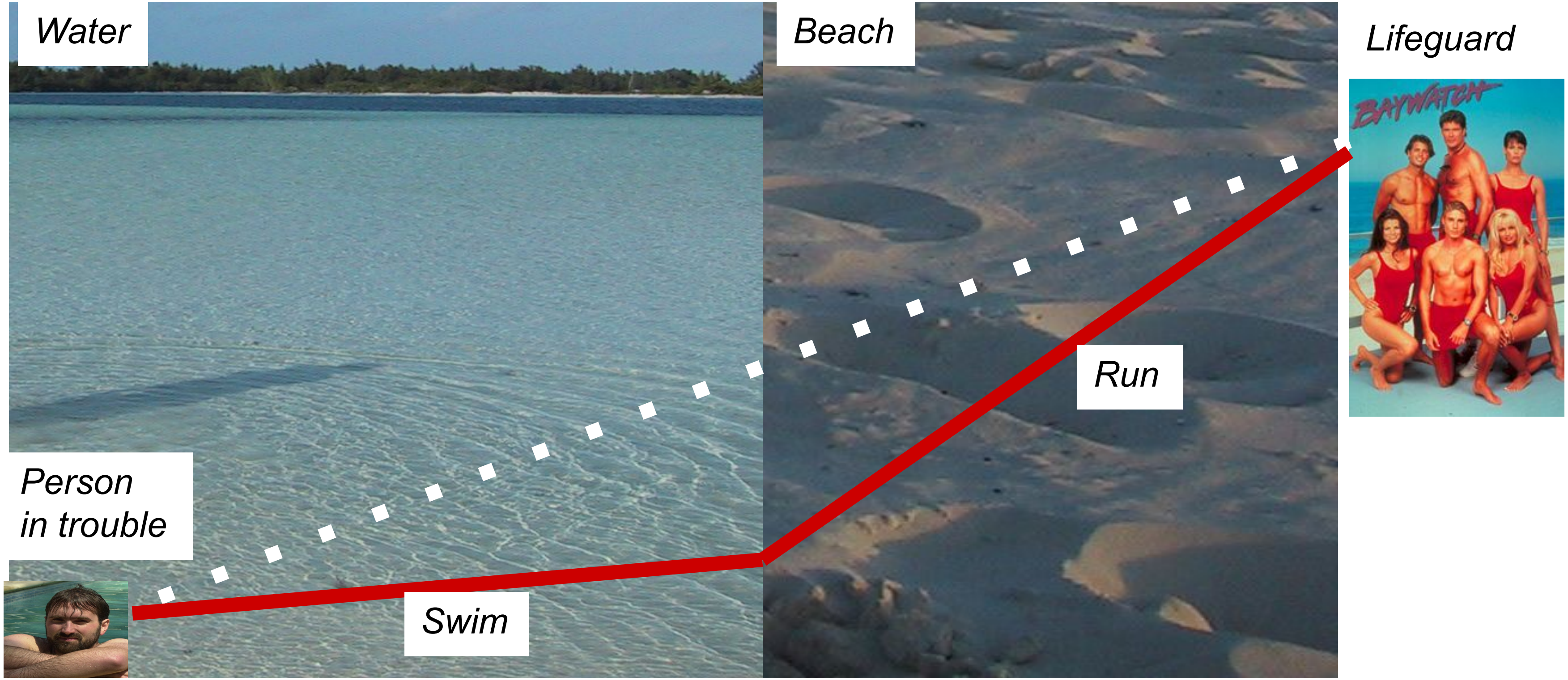


Images by Henrik Wann Jensen

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

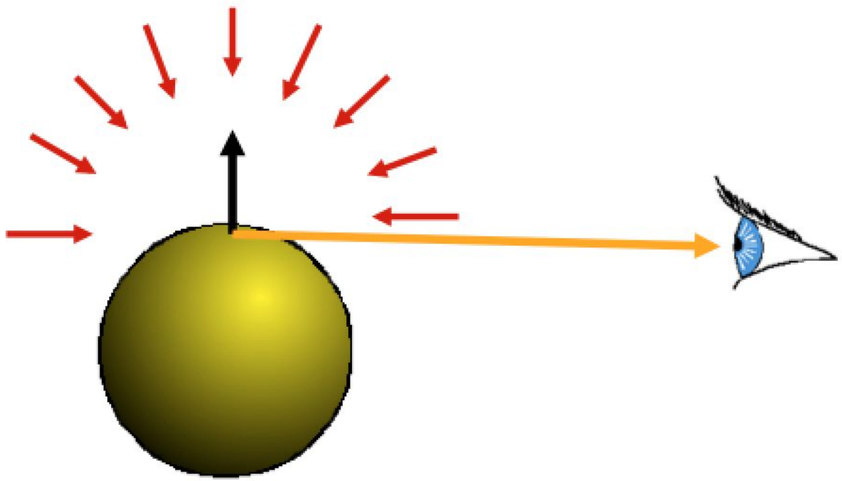
Refraction and the Lifeguard Problem

- Running is faster than swimming: What is the optimal rescue path?



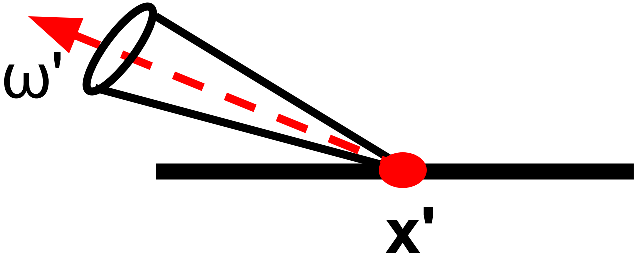
The Rendering Equation

- 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*

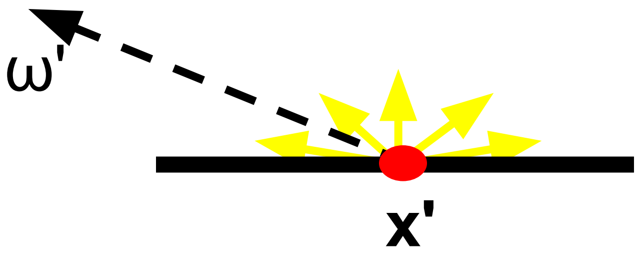
The Rendering Equation



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

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

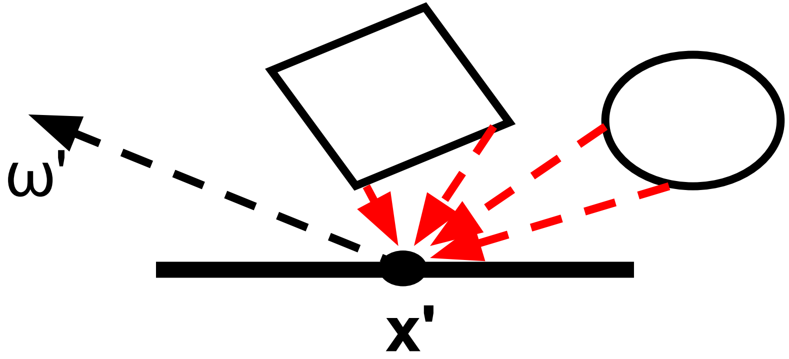
The Rendering Equation



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

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

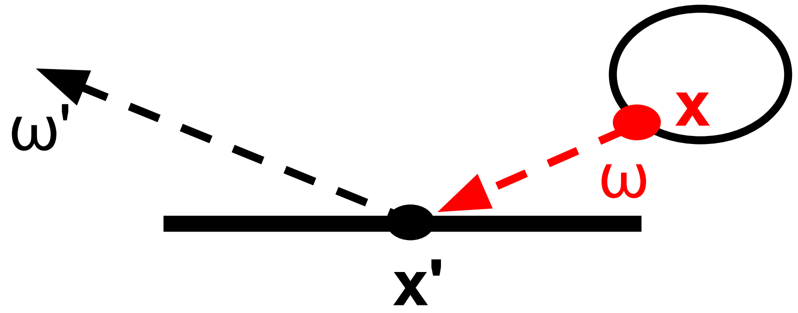
The Rendering Equation



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

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

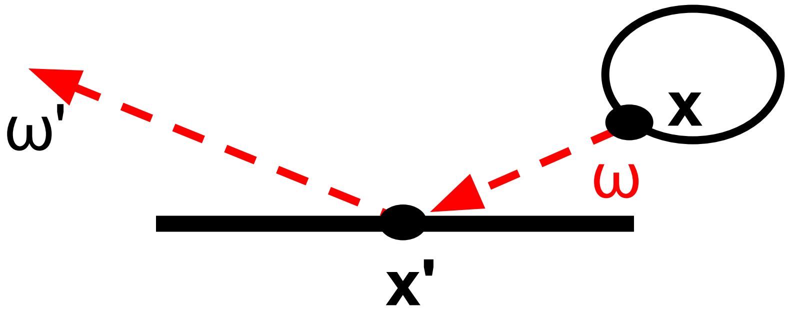
The Rendering Equation



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

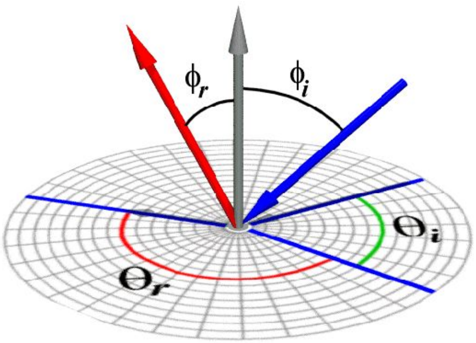
For each x, compute L(x, ω), the radiance at point x in the direction ω (from x to x')

The Rendering Equation

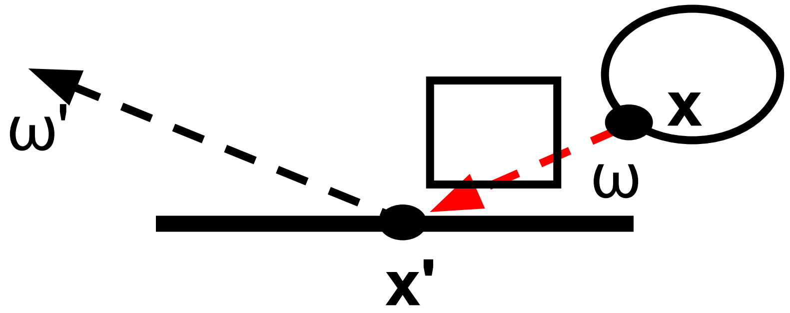


$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x') \, dA$$

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



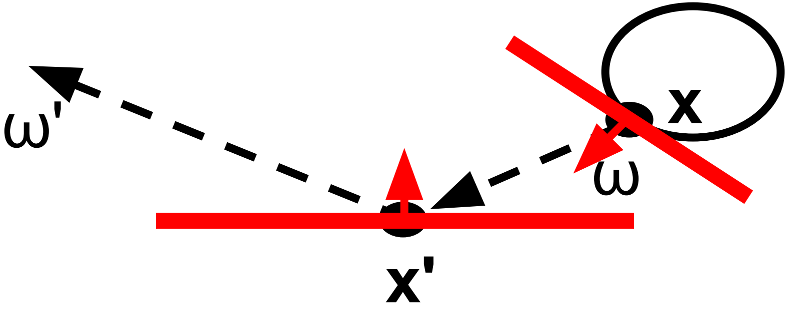
The Rendering Equation



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

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

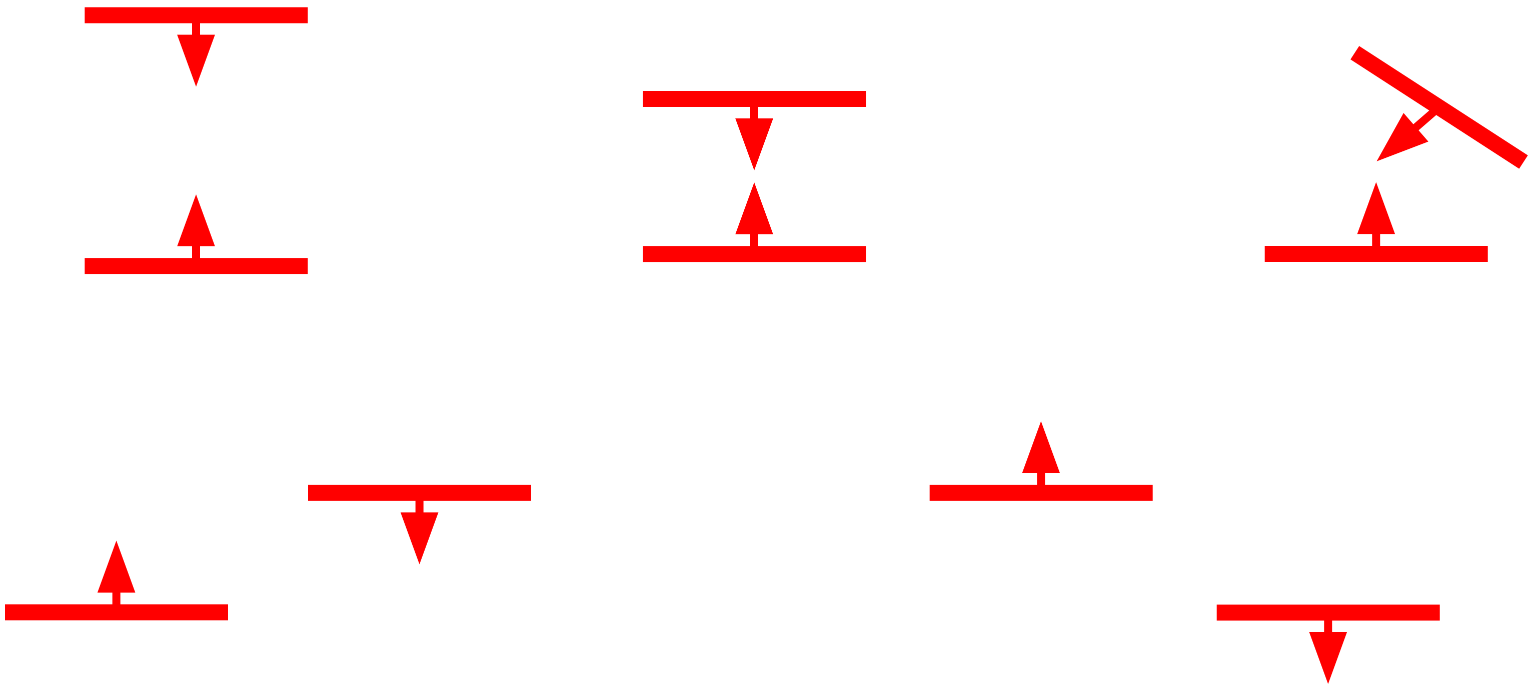


$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x')\, dA$$

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

Intuition about Geometry Term G(x,x')?

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



Rendering Equation → Radiosity

Different rendering techniques can be expressed as a subset/approximation of the full rendering equation.

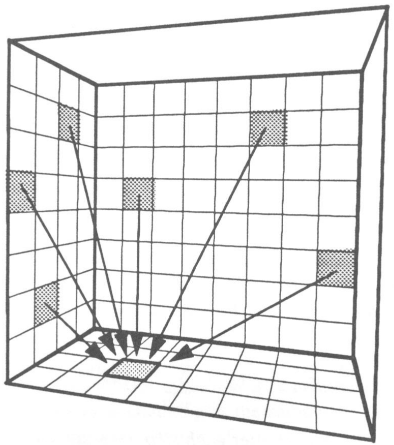
$$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega') L(x,\omega) G(x,x') V(x,x')\, dA$$

Radiosity assumption:
perfectly diffuse surfaces (not directional)

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

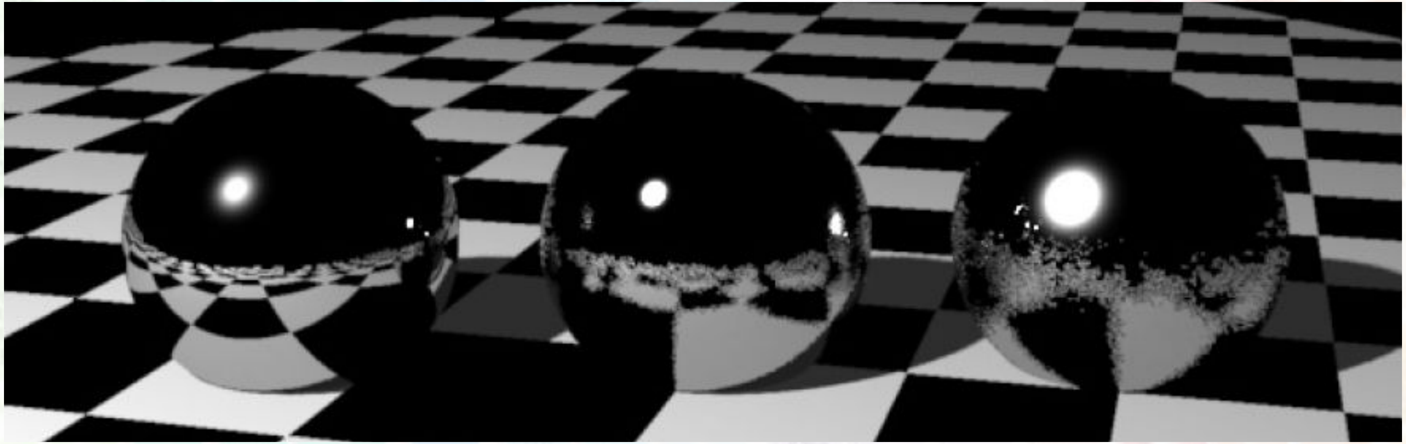
discretize

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

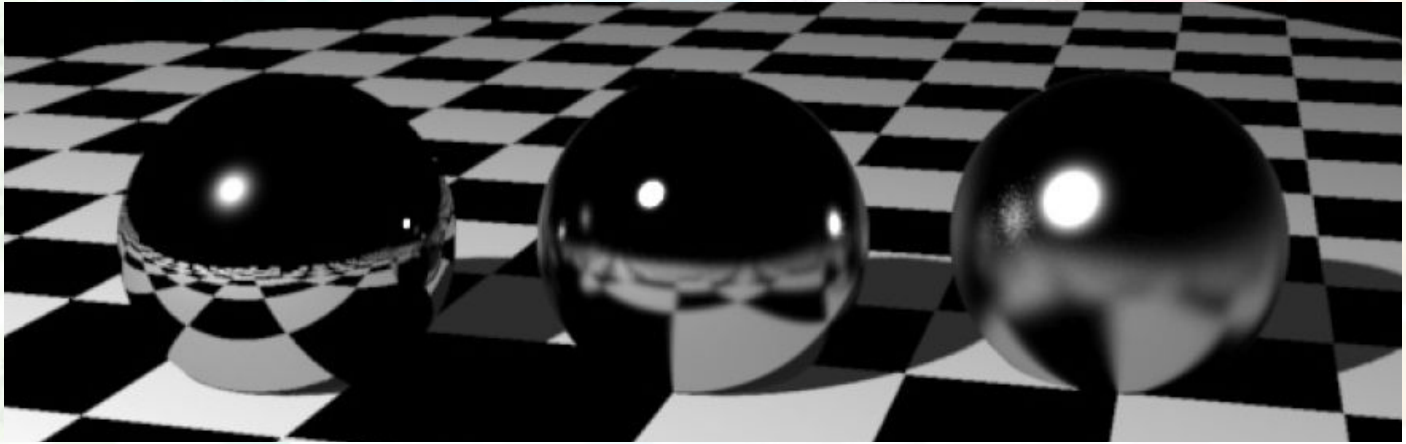


Questions?

1 glossy sample per pixel



256 glossy samples per pixel

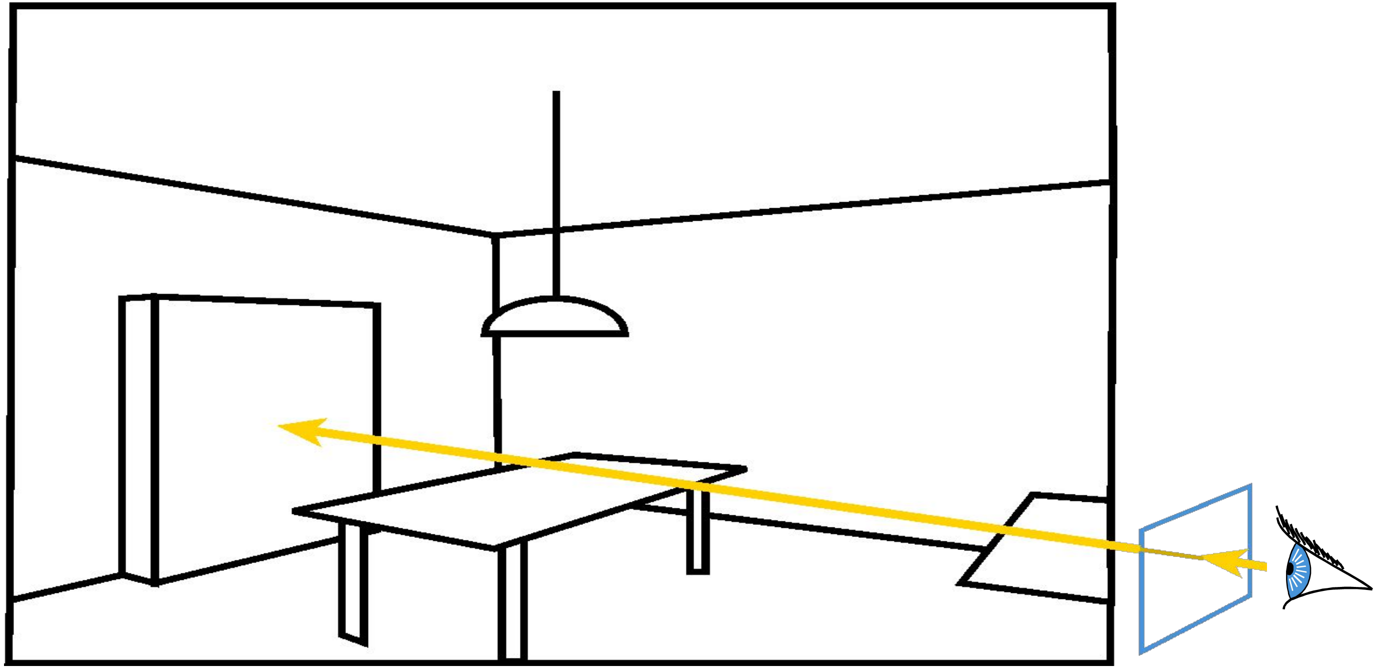


Today

- Worksheet: 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

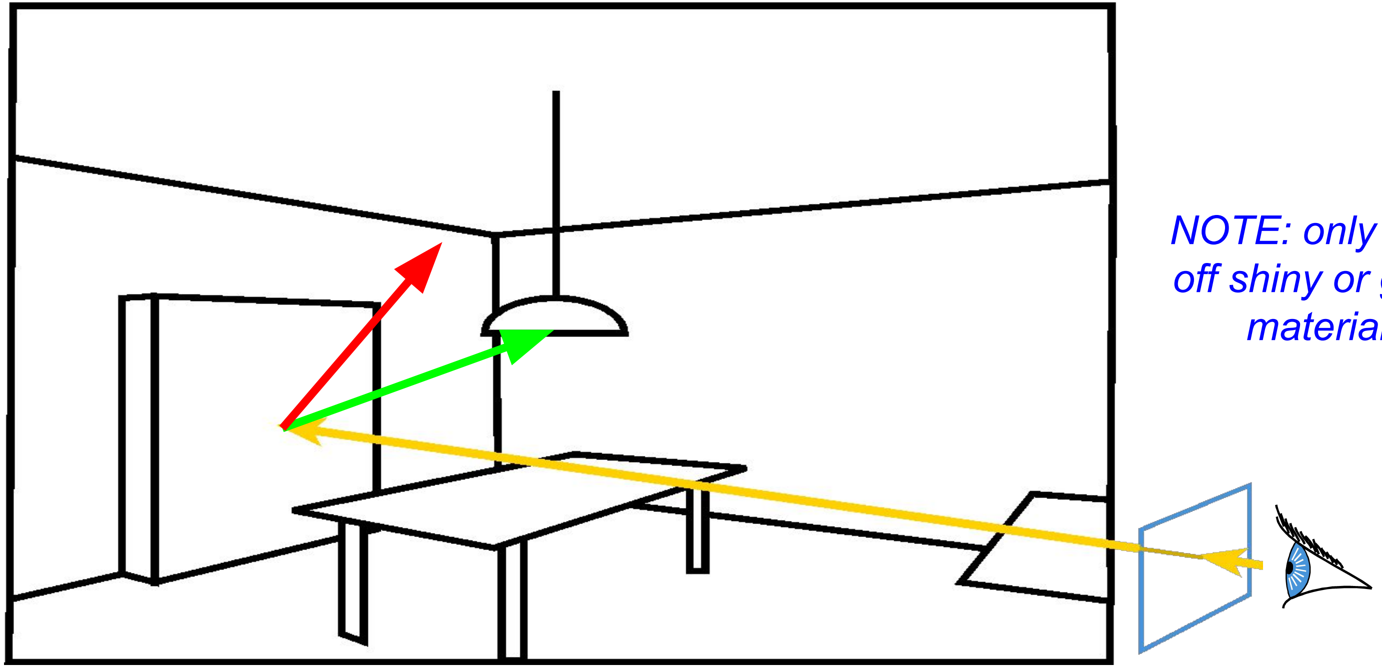
Ray Casting

- Cast a ray from the eye through each pixel



Ray Tracing

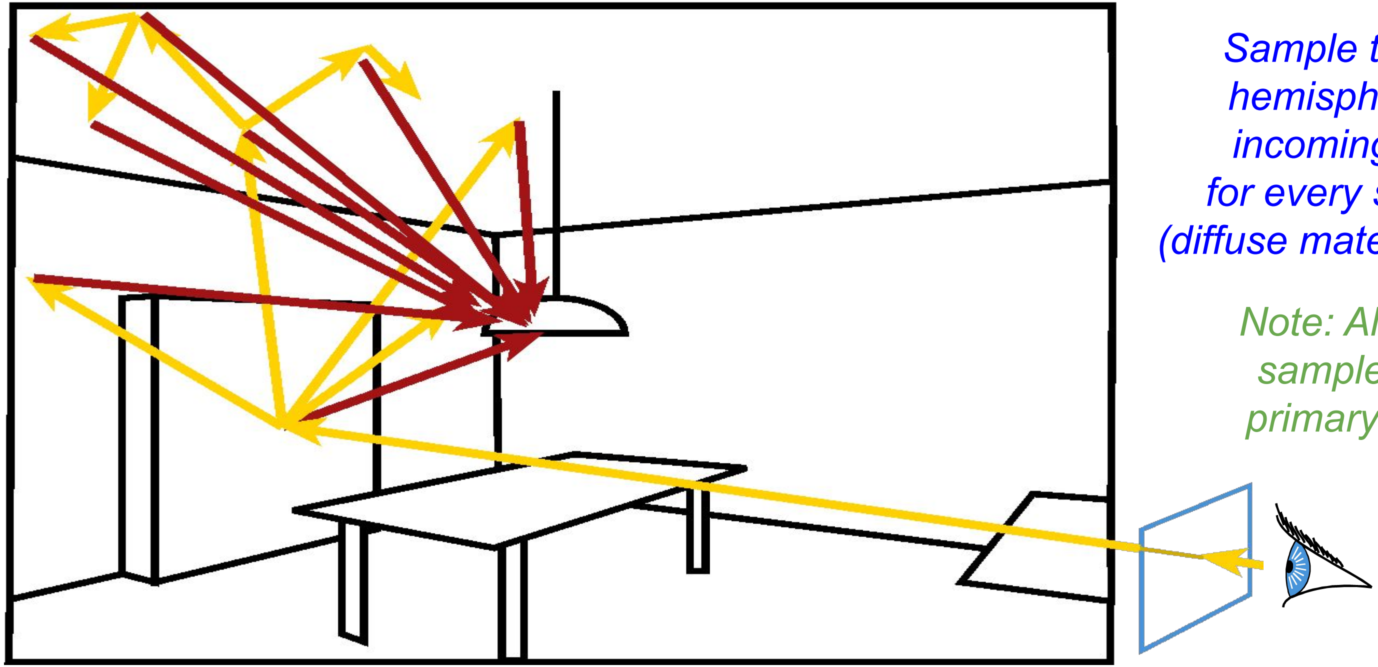
- Cast a ray from the eye through each pixel
- Trace secondary rays (shadow, reflection, refraction)



NOTE: only reflect off shiny or glossy materials!

Monte Carlo Ray Tracing

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

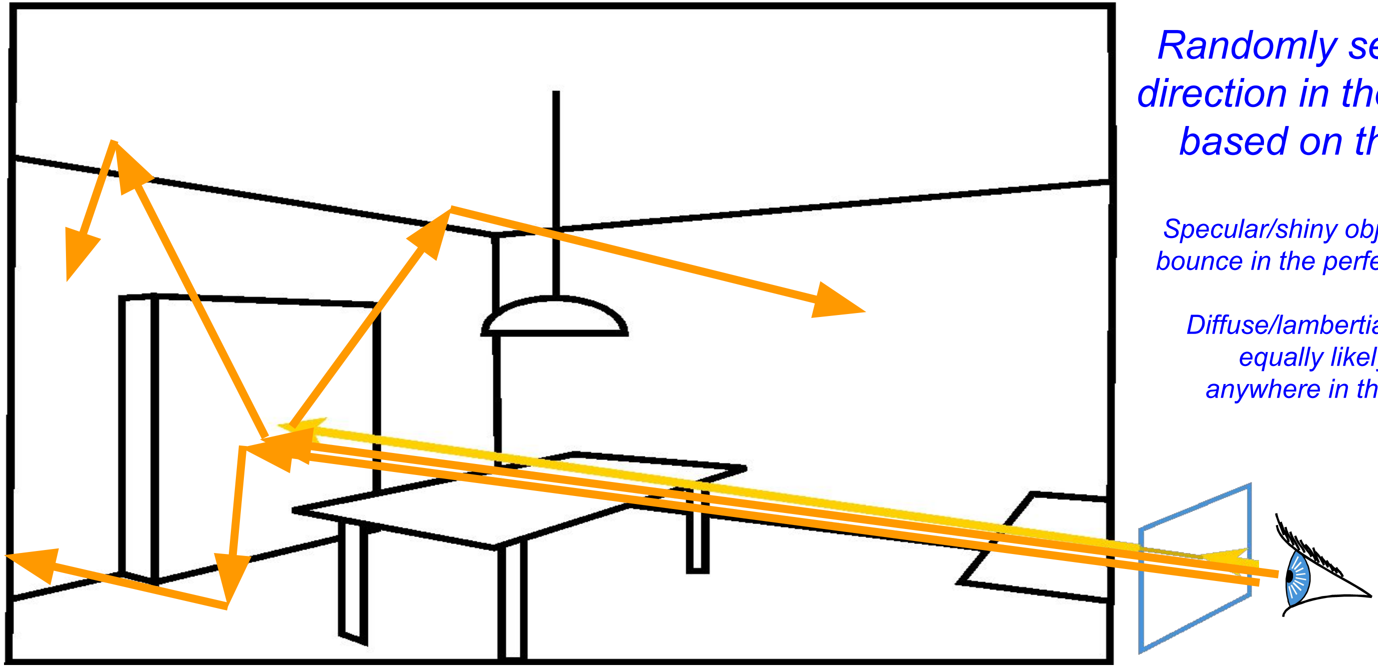


Sample the full hemisphere of incoming light for every surface (diffuse materials too!)

Note: Always sample the primary light

(Monte Carlo) Path Tracing

- Trace *only one* secondary ray per recursion
- But send many primary rays per pixel - NOTE: anti-aliasing for free!

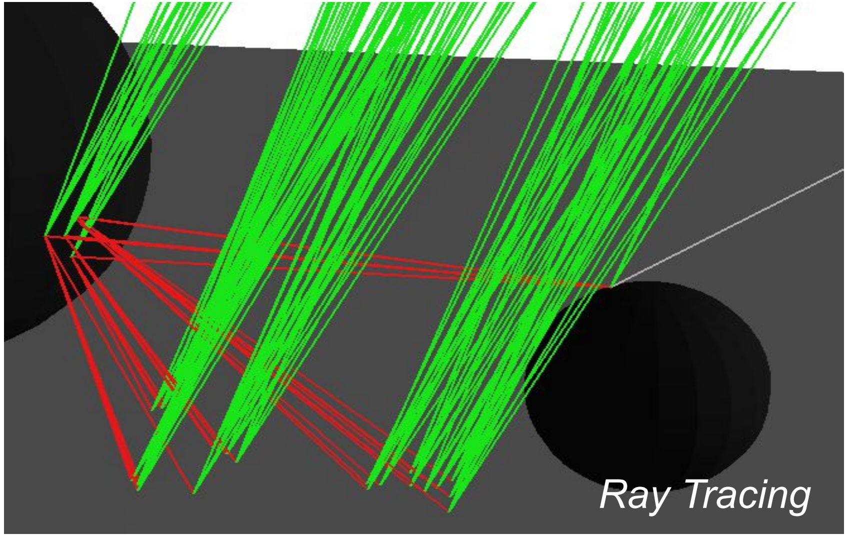


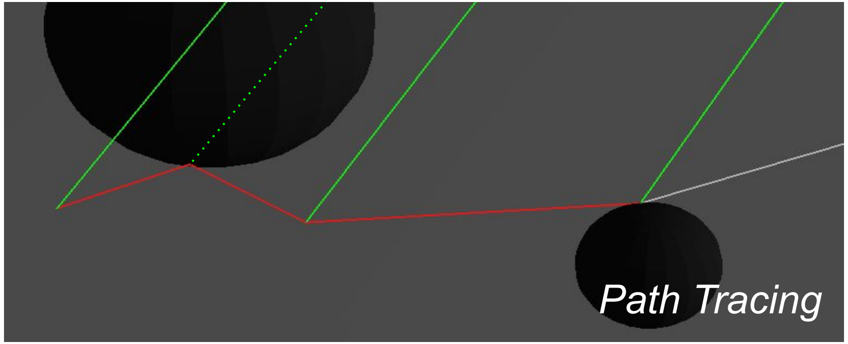
Randomly select one ray direction in the hemisphere based on the material.

Specular/shiny objects most likely to bounce in the perfect mirror direction.

Diffuse/lambertian/matte objects equally likely to bounce anywhere in the hemisphere.

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

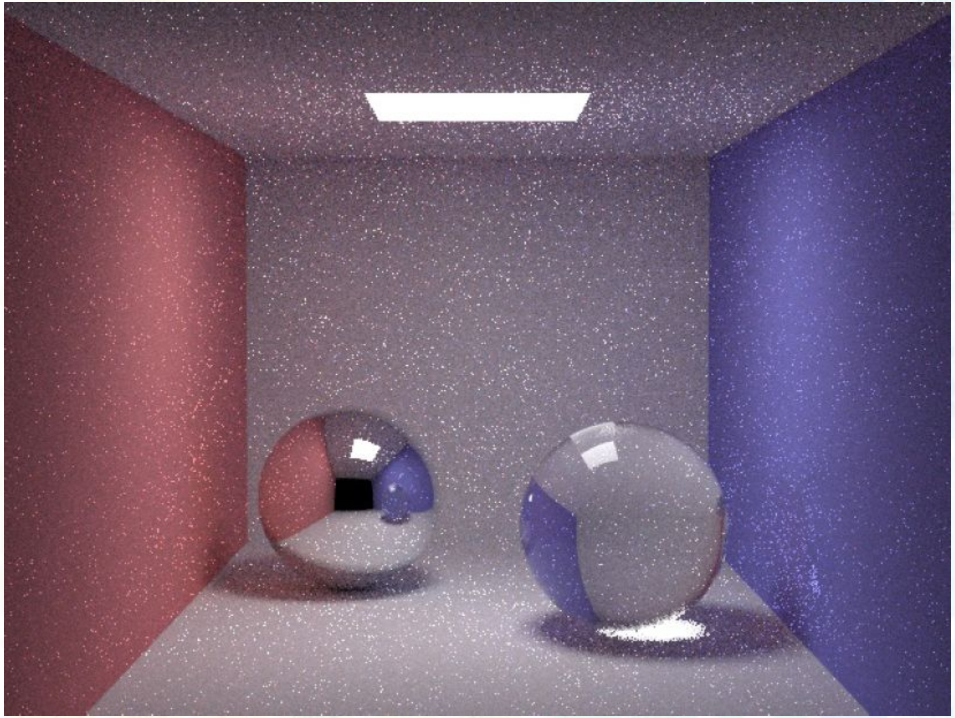
How many 3 bounce paths can we trace per pixel for the same cost?

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

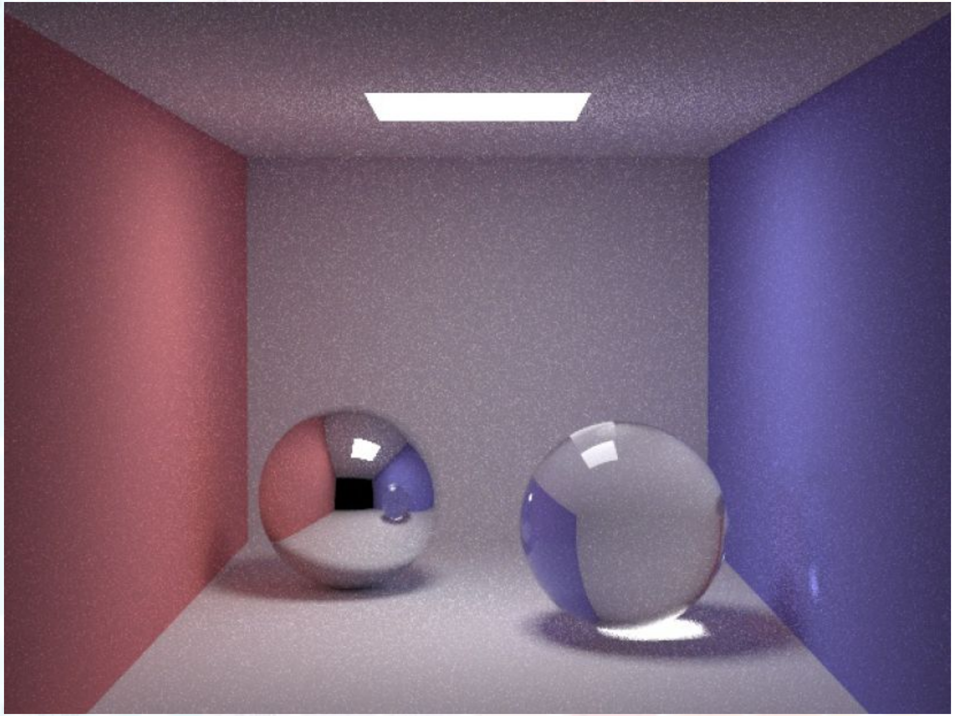
Which will probably have less error?

Questions?

10 paths/pixel



100 paths/pixel



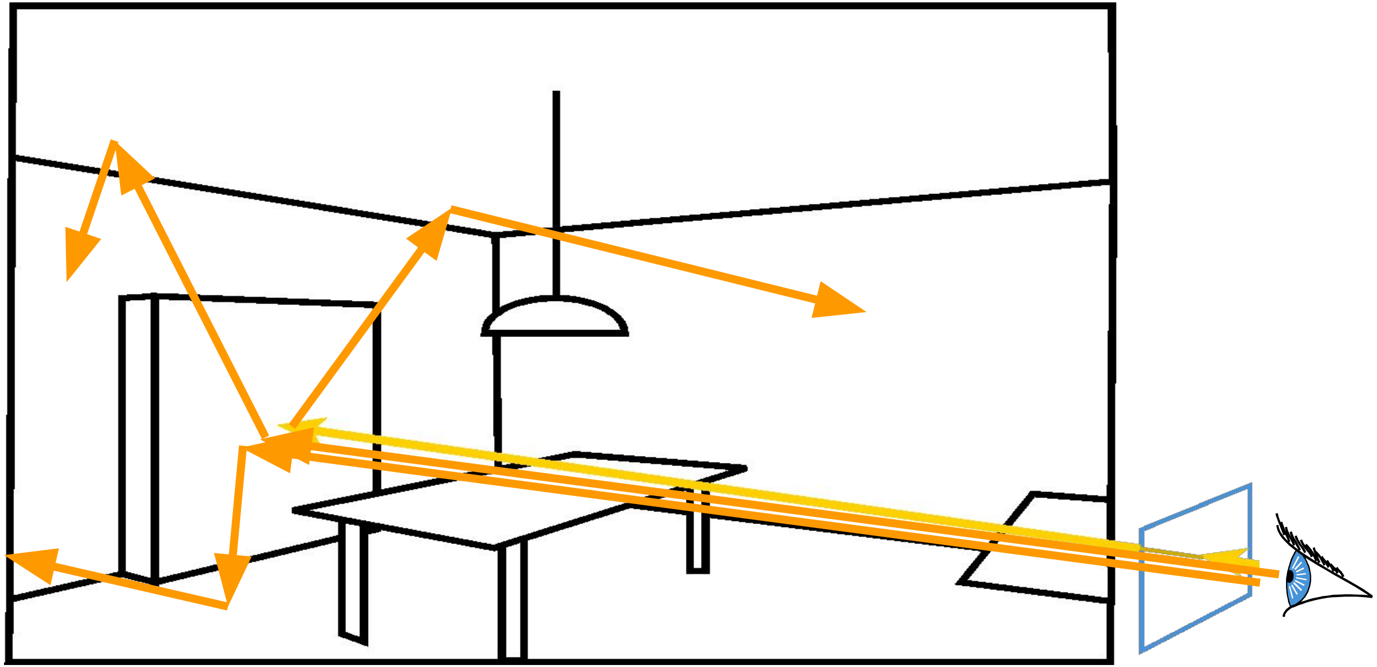
Images from Henrik Wann Jensen

Today

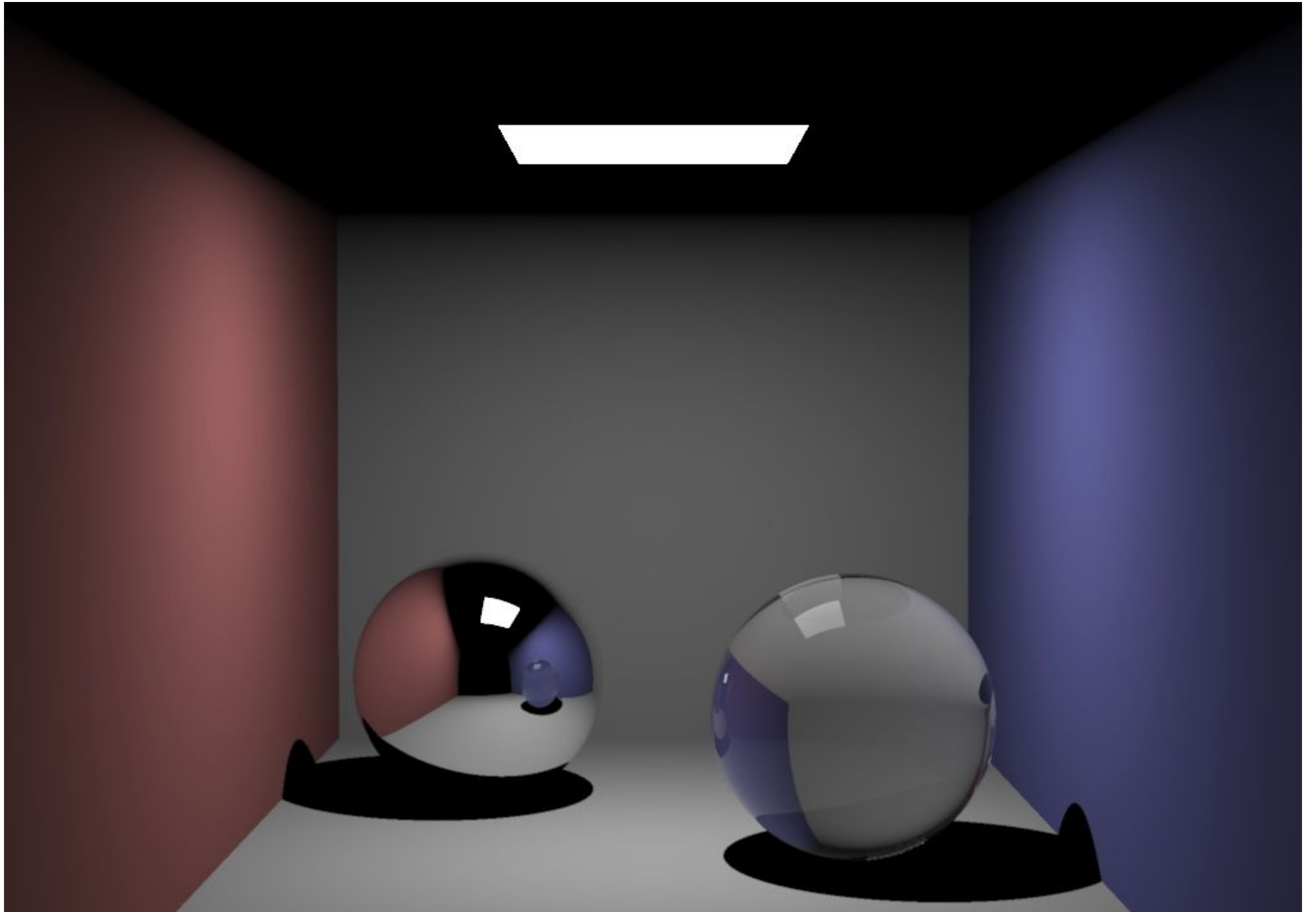
- Worksheet: Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

Problem: Path Tracing is Costly

- Requires tons of rays per pixel

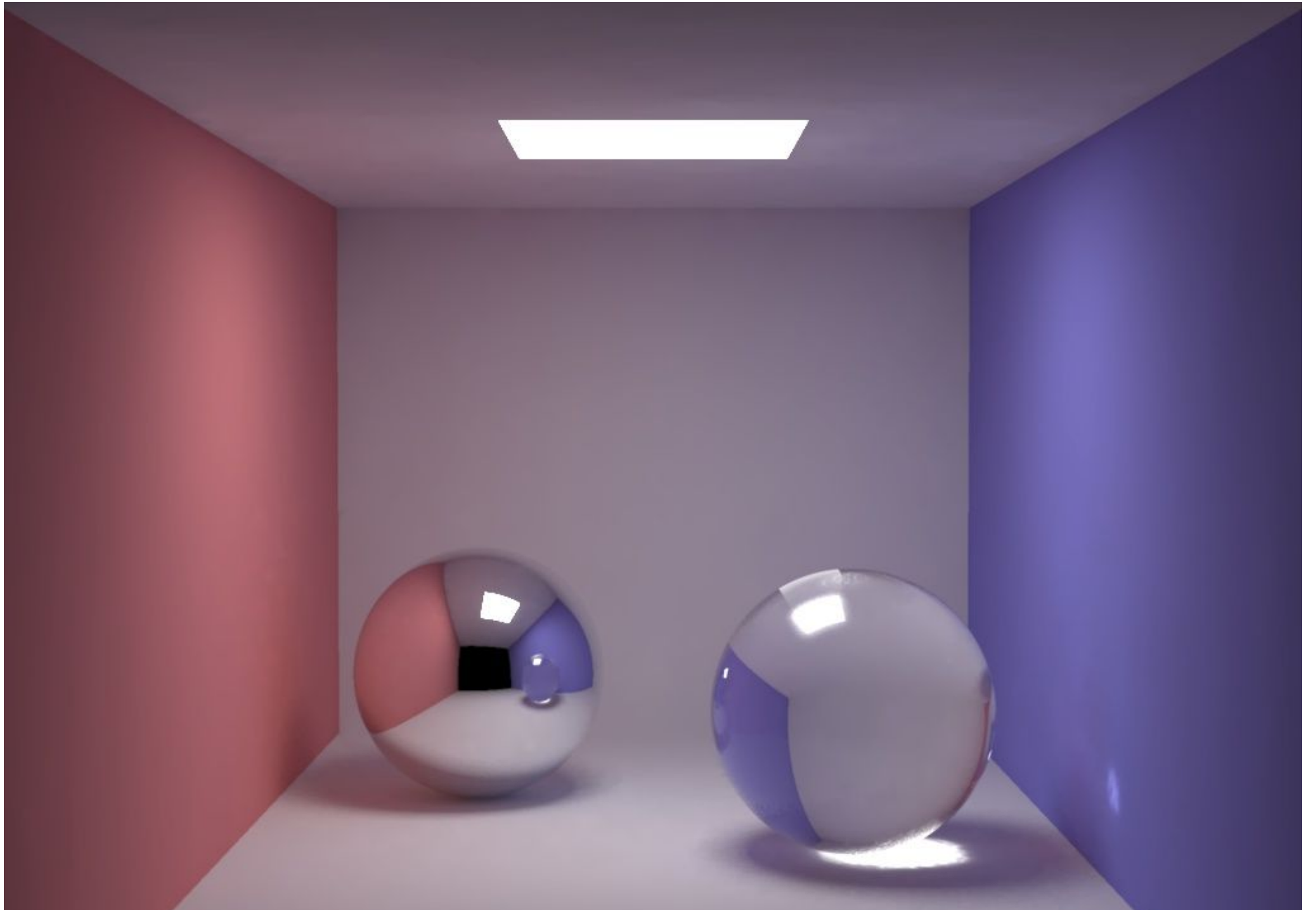


Direct Illumination



Henrik Wann Jensen

Global Illumination



Henrik Wann Jensen

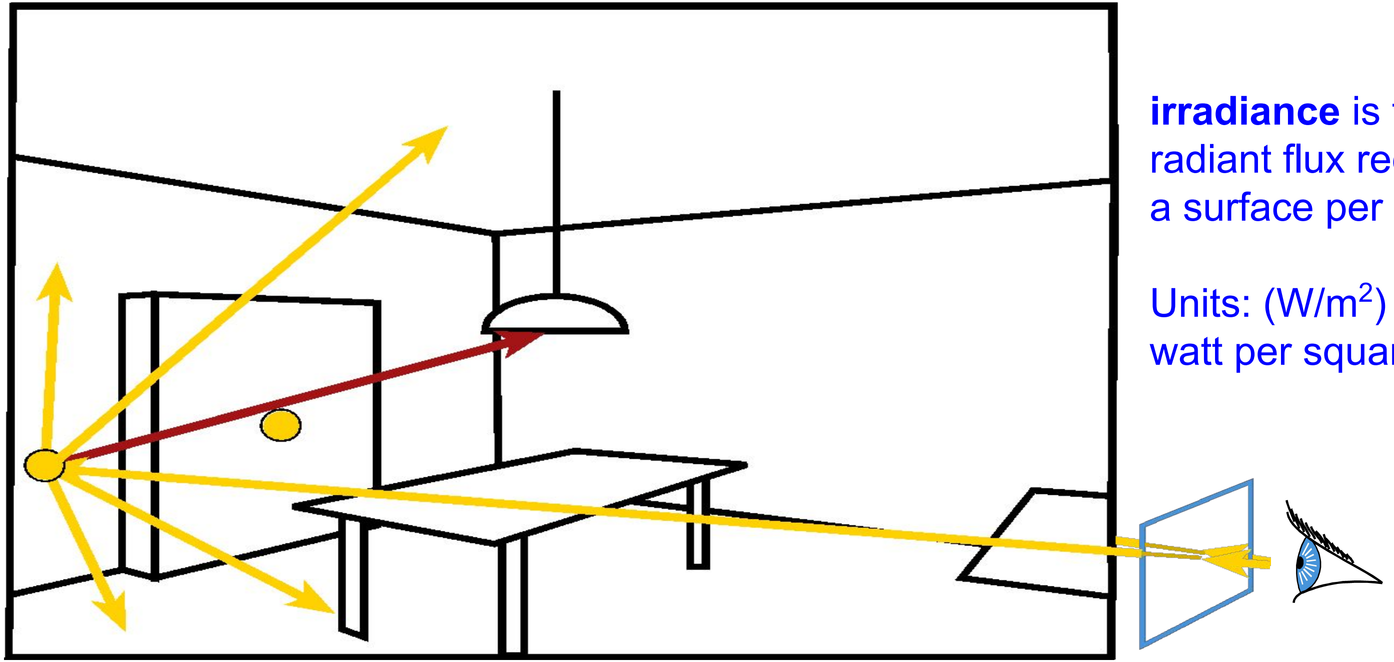
Global - Direct = Indirect Illumination

Notice that indirect Illumination is very smooth, except at discontinuities!

Henrik Wann Jensen

Irradiance Cache

- The indirect illumination is smooth
- So let's store and re-use the computed indirect illumination

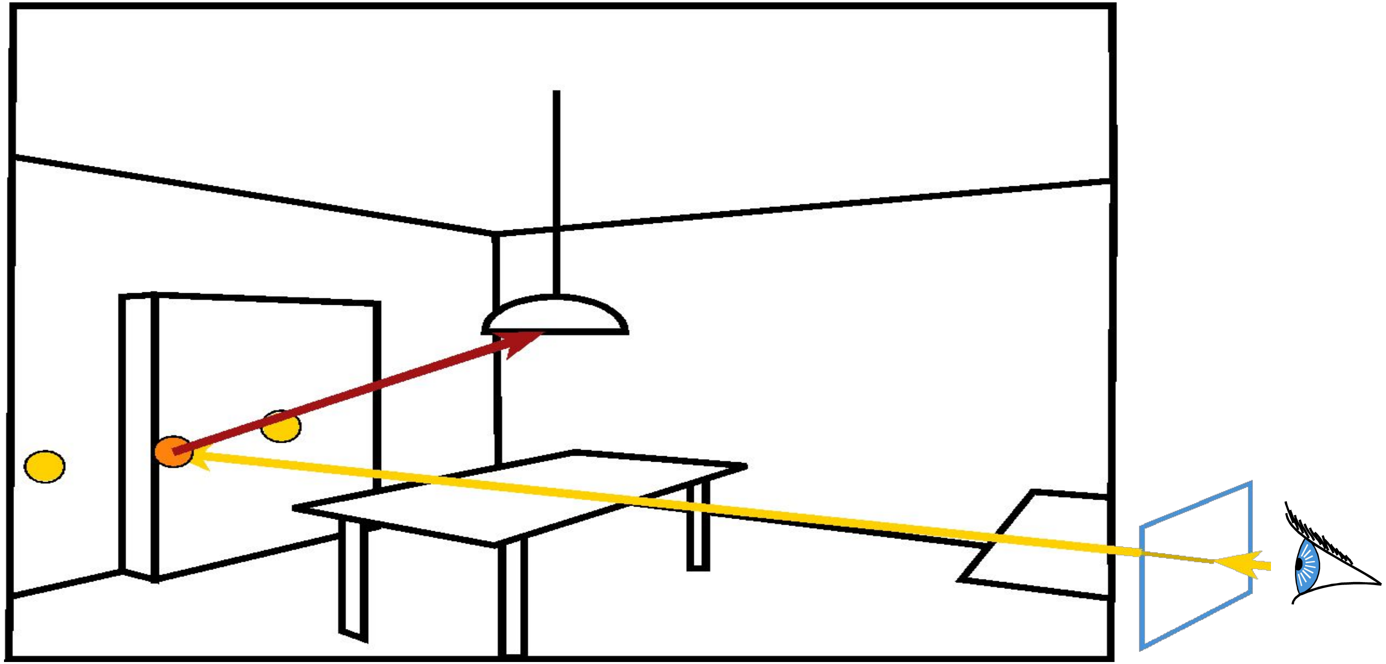


irradiance is the radiant flux received by a surface per unit area.

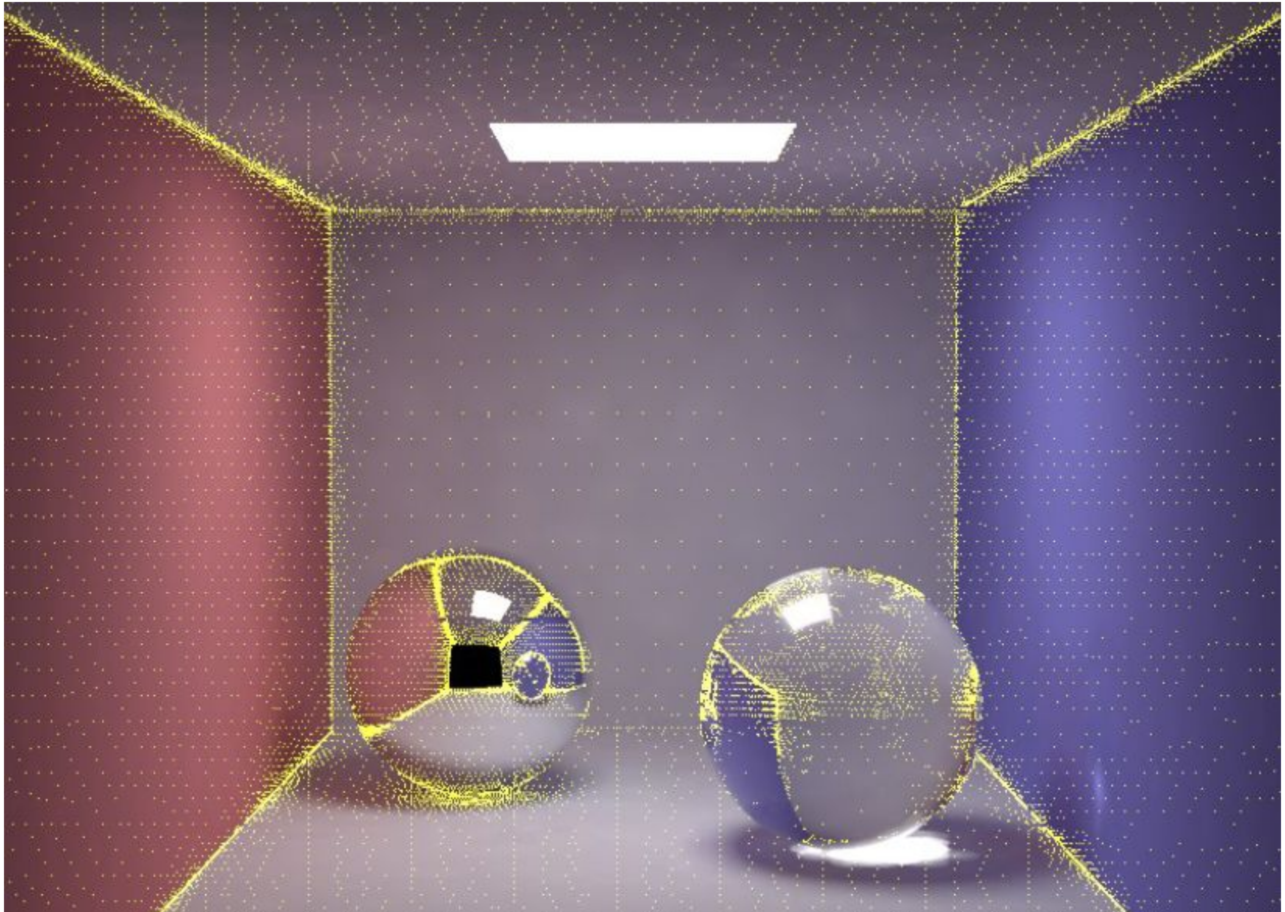
Units: (W/m^2)
watt per square meter

Irradiance Cache

- Interpolate nearby cached irradiance values, *when they are available*
- Always re-calculate direct lighting (e.g., shadows from primary light)



Irradiance Cache



Henrik Wann Jensen

Questions?

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

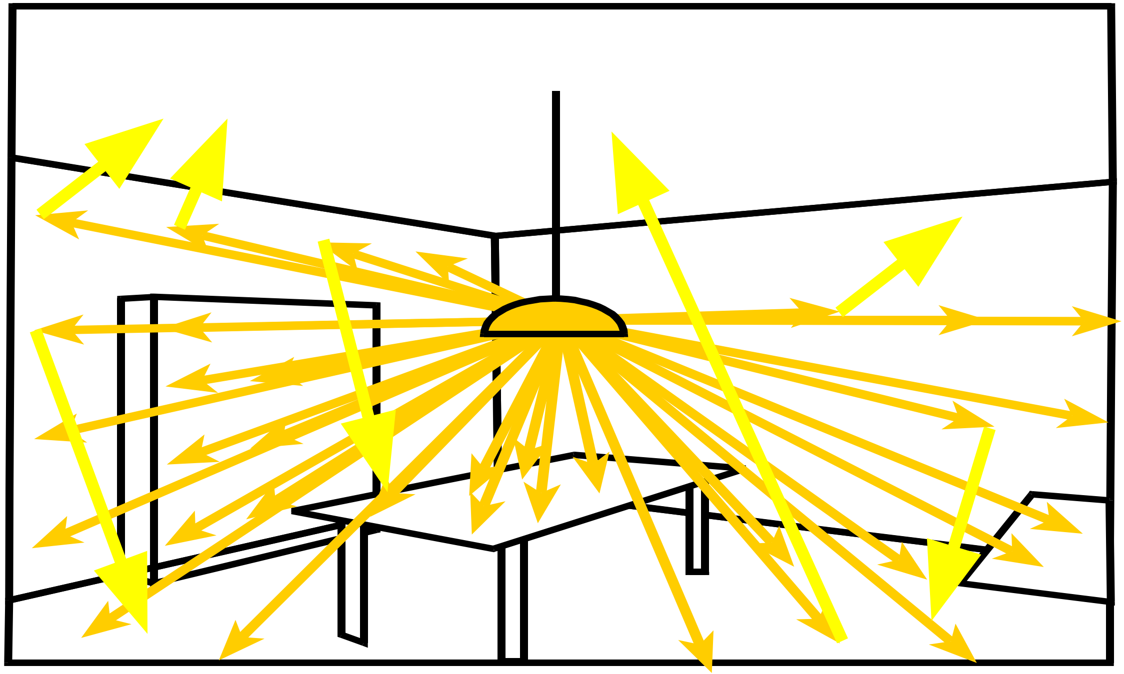
Henrik Wann Jensen

Today

- Worksheet: 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

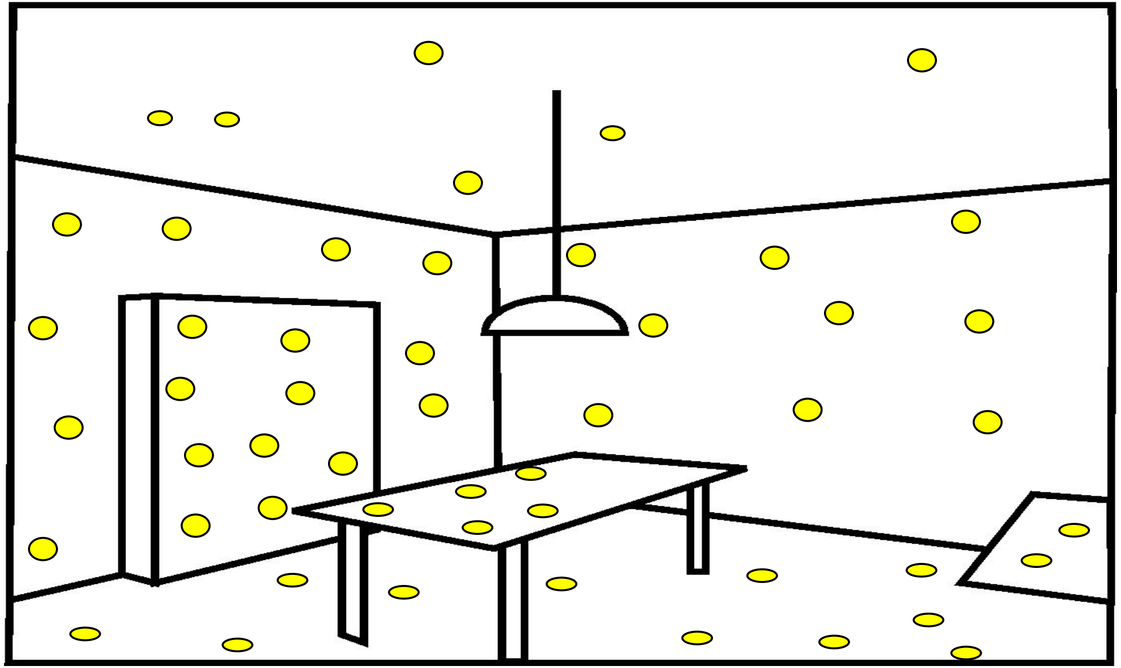
Photon Mapping

- Preprocess: Cast rays from light sources
  - *NOTE: This is independent of camera/eye viewpoint*



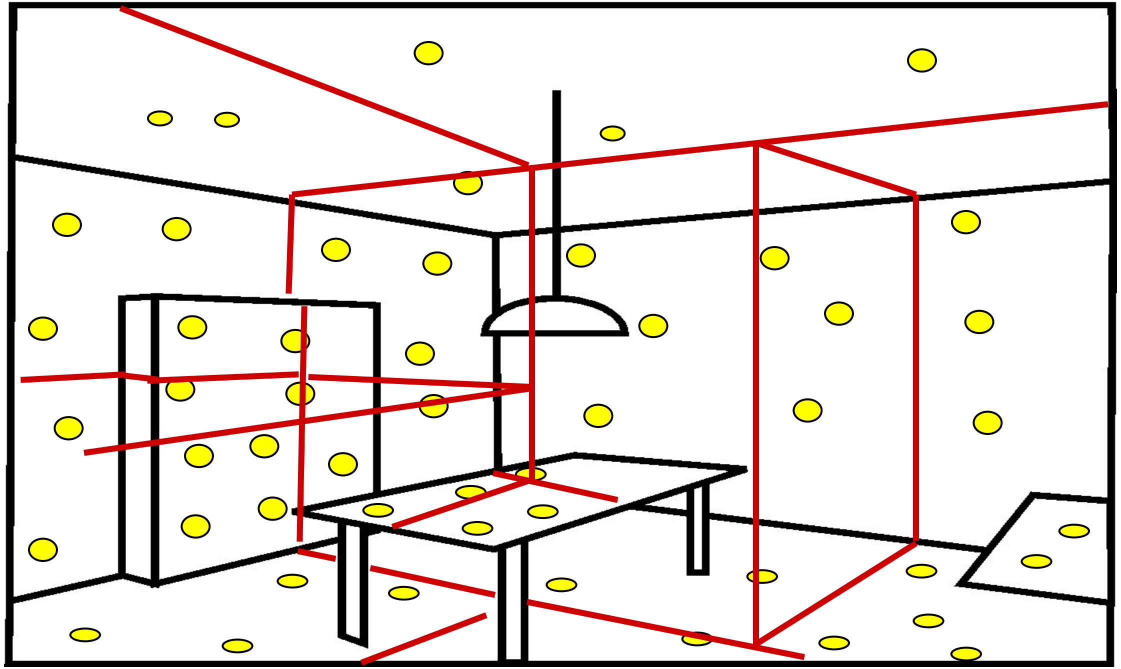
Photon Mapping

- Preprocess: Cast rays from light sources
- Store photons: position + light power + incoming direction



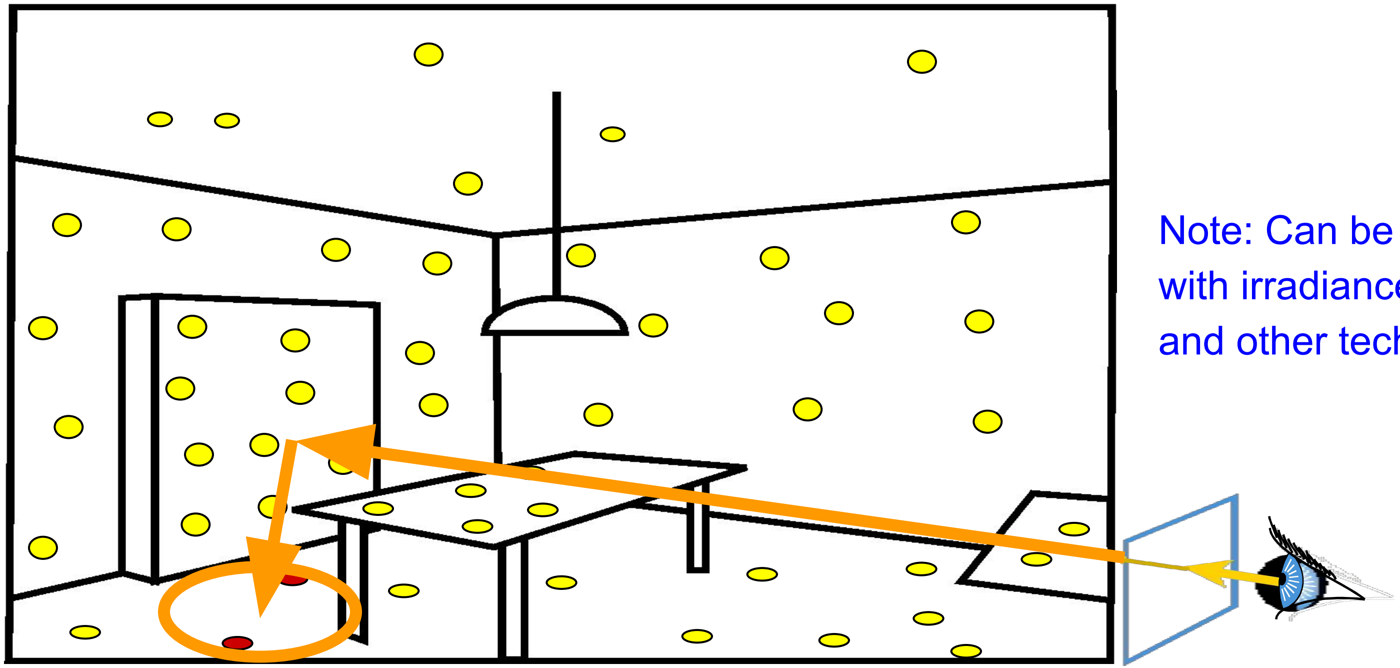
Using a kD-tree to storing the Photon Map

- Efficiently store photons for fast access
- Use hierarchical spatial structure (e.g., in a kd-tree)



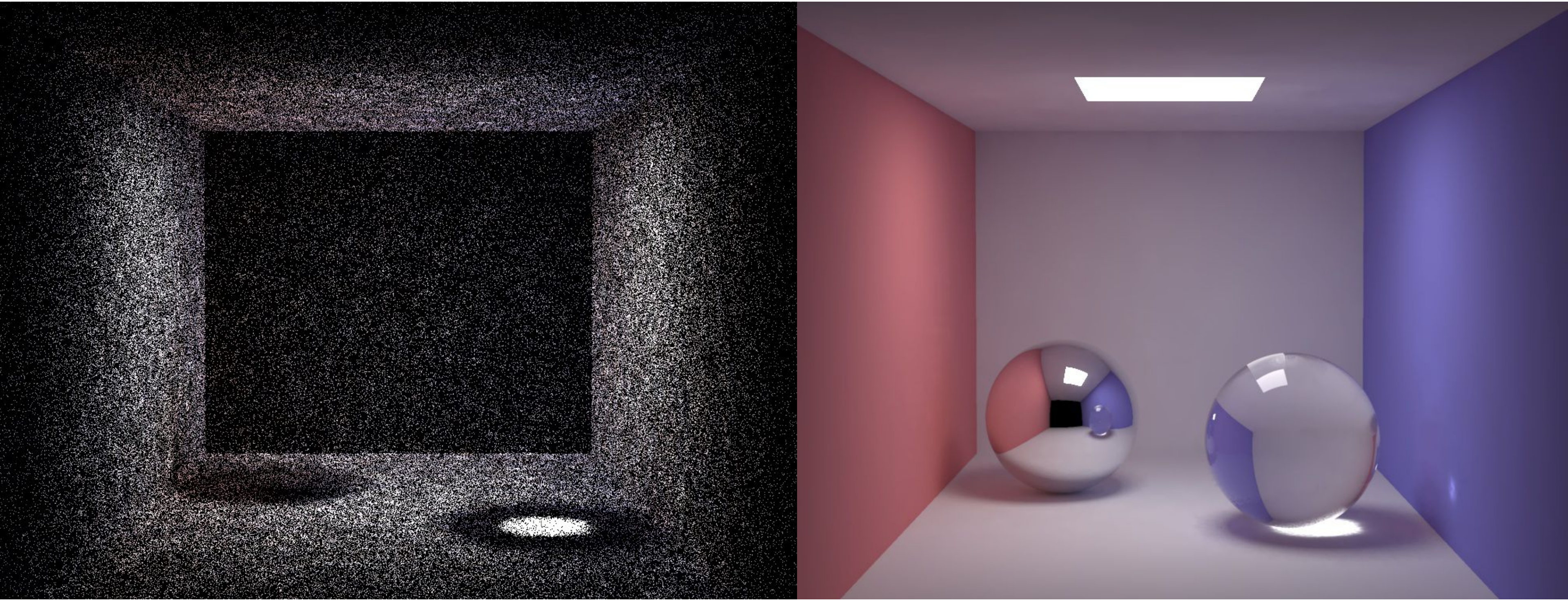
Rendering with Photon Map

- We still trace primary rays and bounce once based on the material
- For secondary rays: Estimate irradiance using *k* closest photons



Note: Can be combined with irradiance caching and other techniques

Photon Map Results



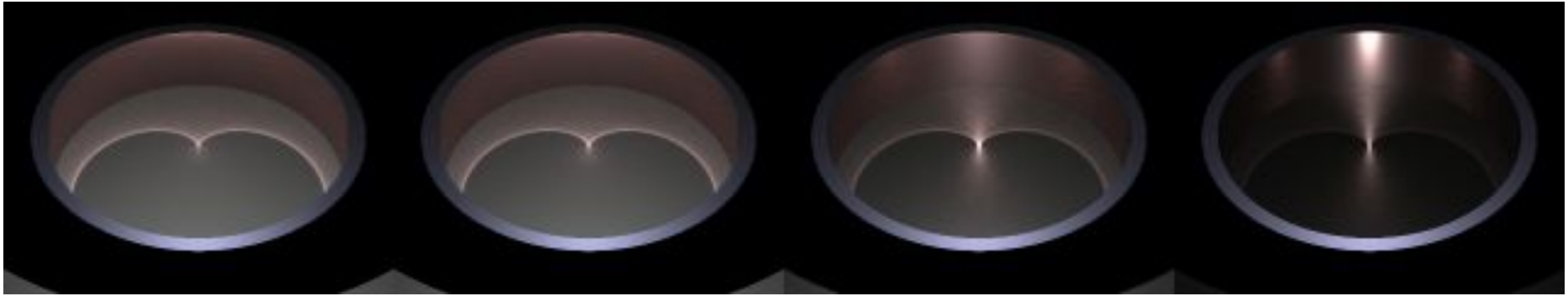
Henrik Wann Jensen

Today

- Worksheet: 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

Reading for Today

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



- Balancing the kd-tree makes a big difference
- Careful memory use, details about design for bit usage & total size
- Russian Roulette - random chance based on intensity to continue
    - Is it an appropriate & intuitive name?
- Various ‘magic’ constants, which (they say) are not difficult to set.
- Expecting paper about physics.. got paper about data structures!
- Well written, intuitive, easy to read, clear references to prior work
- Acknowledges limitations with the methods
- One column v. two column: specified by the publisher (journal/conference), e.g. this was in a book with pages < 8.5”x11”

Reading for Today

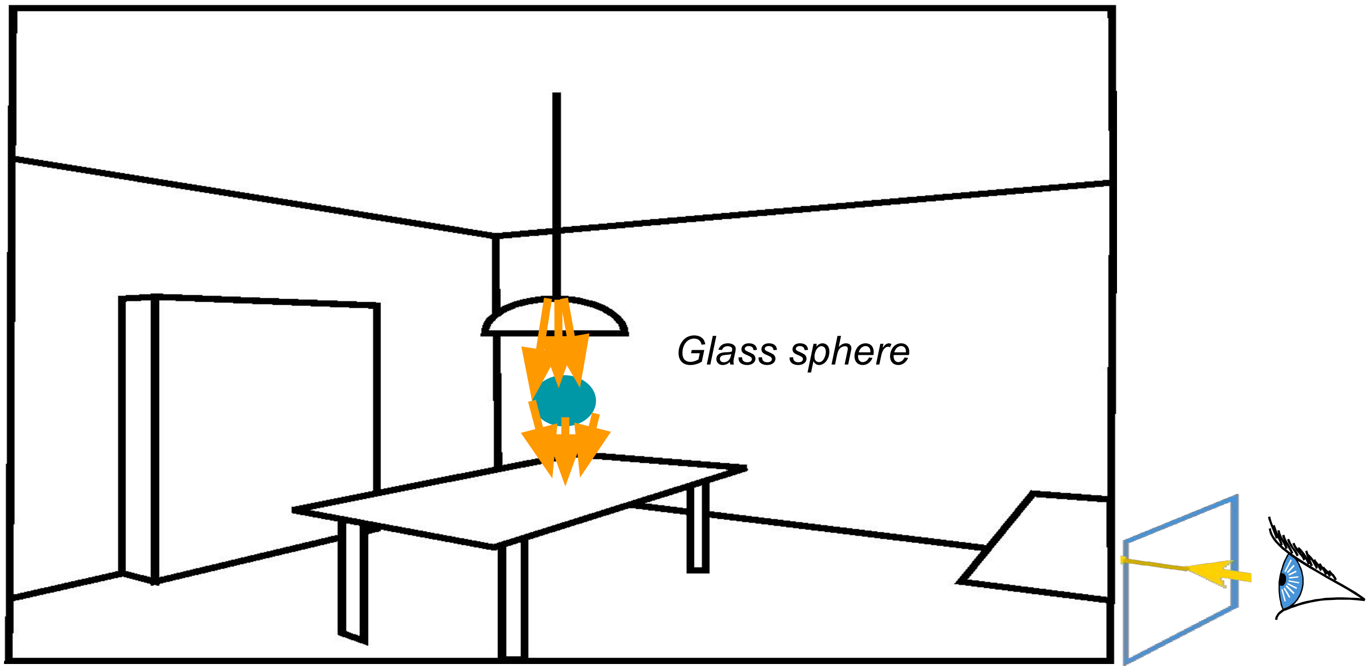
- “Global Illumination using Photon Maps”,
  Henrik Wann Jensen, *Rendering Techniques* 1996.

- Offloaded explanation of existing techniques to other papers (necessary to read other papers to understand this paper)
- Full details of hardware used & running times, direct comparison to other techniques/implementations
- Tradeoff between # of photons, memory, rendering time, & final quality
- Separate photon maps (with different resolutions) for different (matte v. specular) materials
- Are we finished? Have we correctly modeled/rendered all aspects of the physics of light – just limited by computational power?
- Take shortcuts (skip caustic photon map), when appropriate, for the scene & materials
- Important of random numbers. What random number generator is used? Are there artifacts if a bad number generator is used?

https://www.housepaintingtutorials.
com/painting-tone-on-tone-wall-stri

Photon Mapping - Caustics

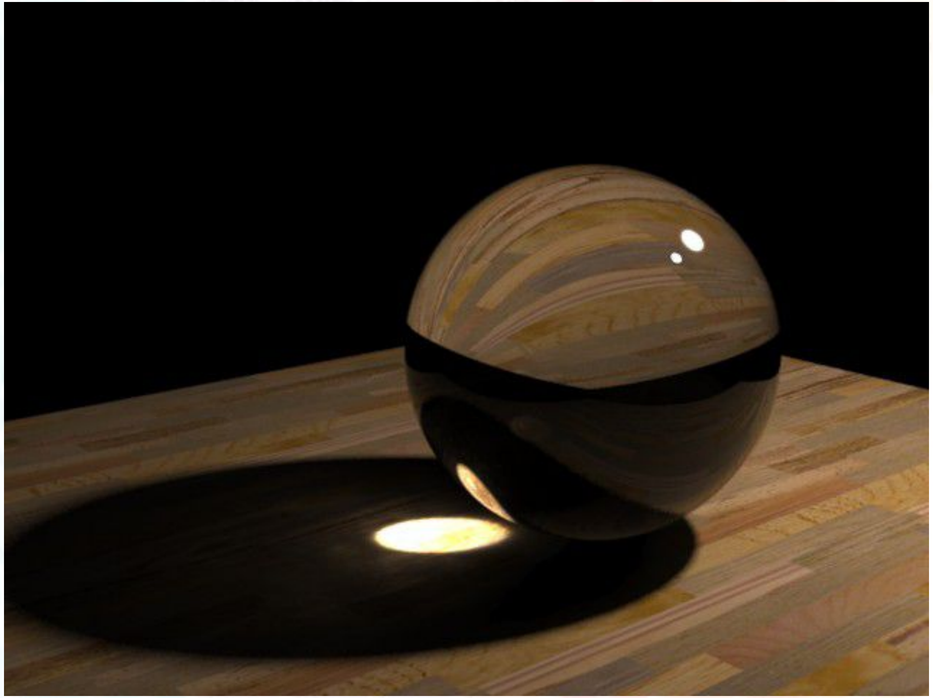
- Special photon map for specular reflection and refraction



Path Tracing & Photon Mapping Comparison

Path Tracing w/ 1000 paths/pixel

Photon mapping

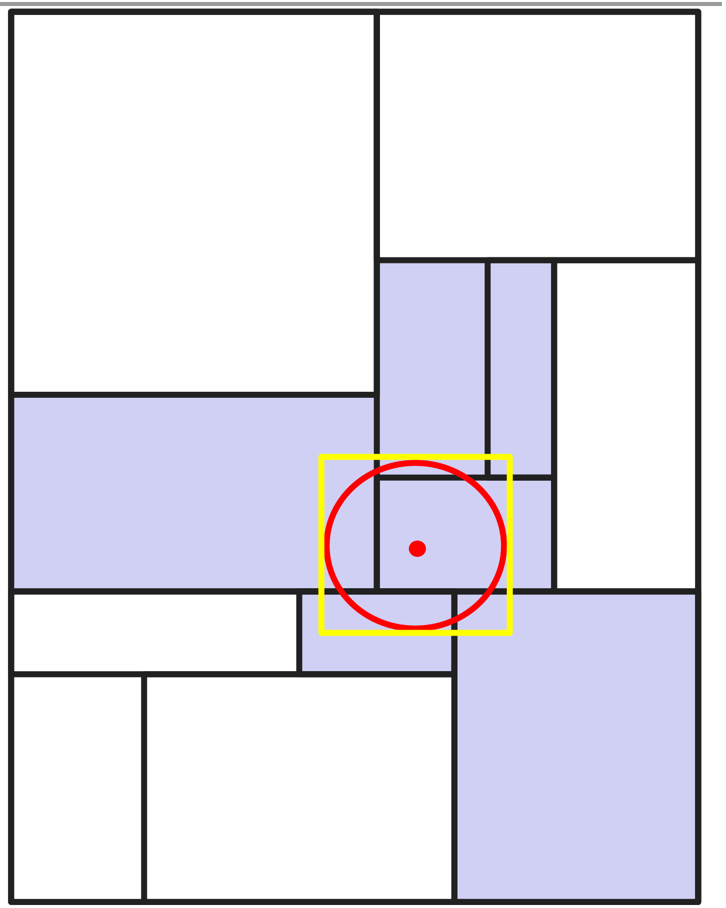


(similar rendering time)

Henrik Wann Jensen

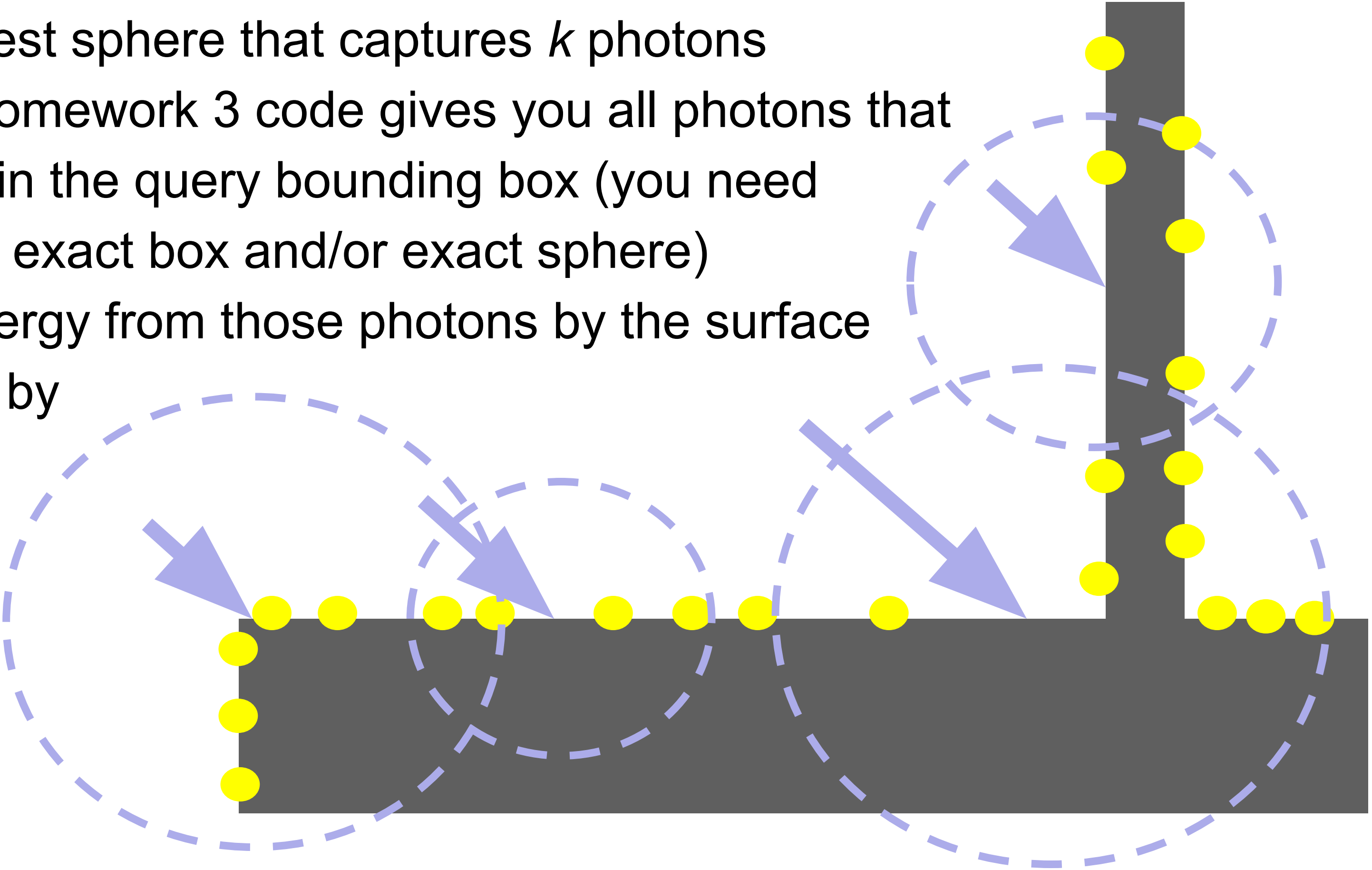
Homework 3: Photons in the *k*-D Tree

- You start with query point & radius (red)
- You input to the function KDTree::CollectPhotonsInBox 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: Homework 3 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 or convex corners?



Today

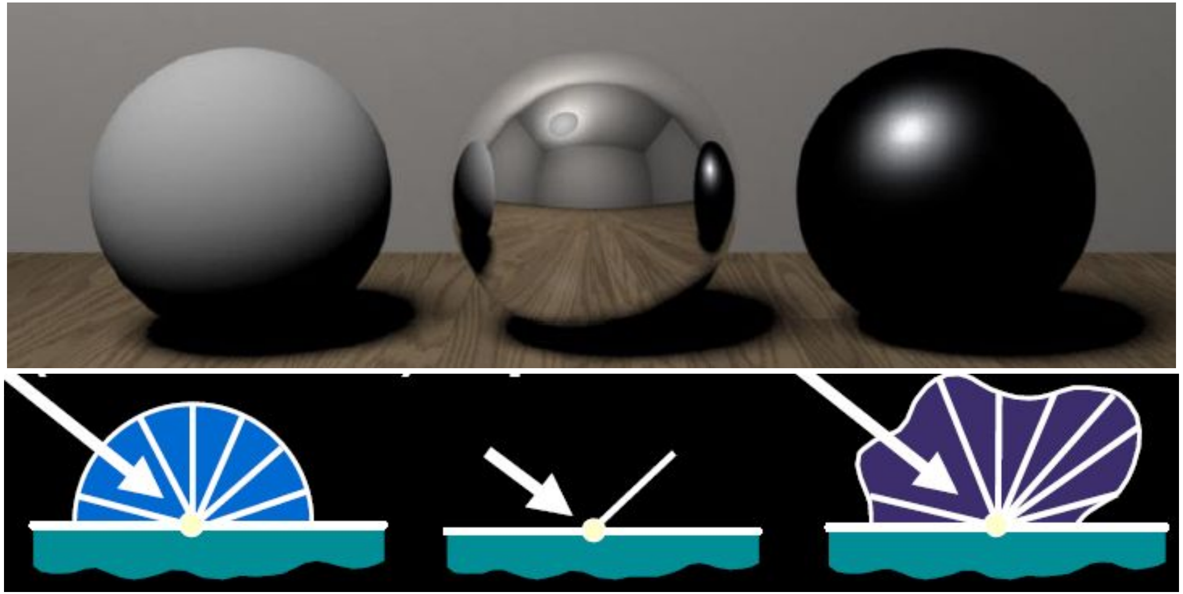
- Worksheet: Progressive Radiosity
- The Rendering Equation
- Ray Casting vs. Ray Tracing vs. Monte-Carlo Ray Tracing vs. Path Tracing
- Irradiance Caching
- Photon Mapping
- Papers for Today
- Ray Grammar
- Papers for Next Time

Ray Grammar

Classify local interaction:

- E = eye
- L = light
- S = perfect specular (reflection or refraction)
- G = glossy scattering
- D = diffuse scattering

D **S** **G**

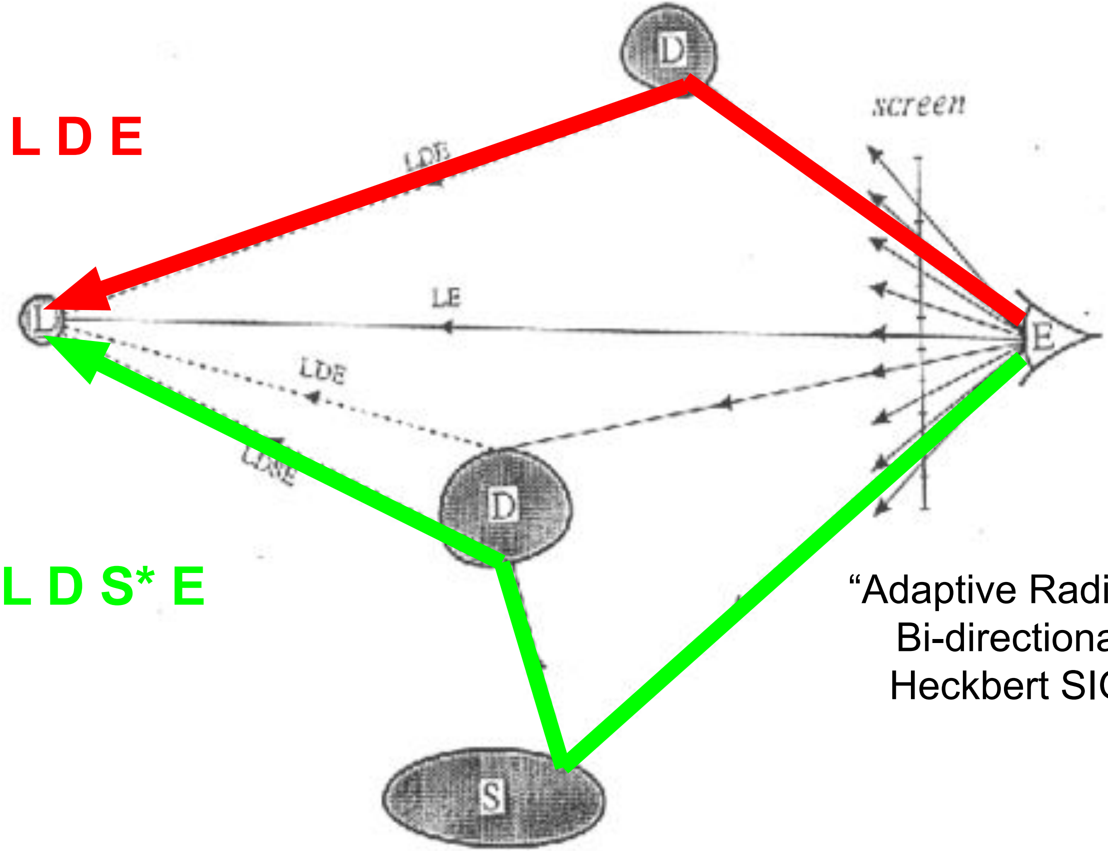


From Dutre et al.'s slides

Classic Ray Casting/Tracing

Ray casting: L D E

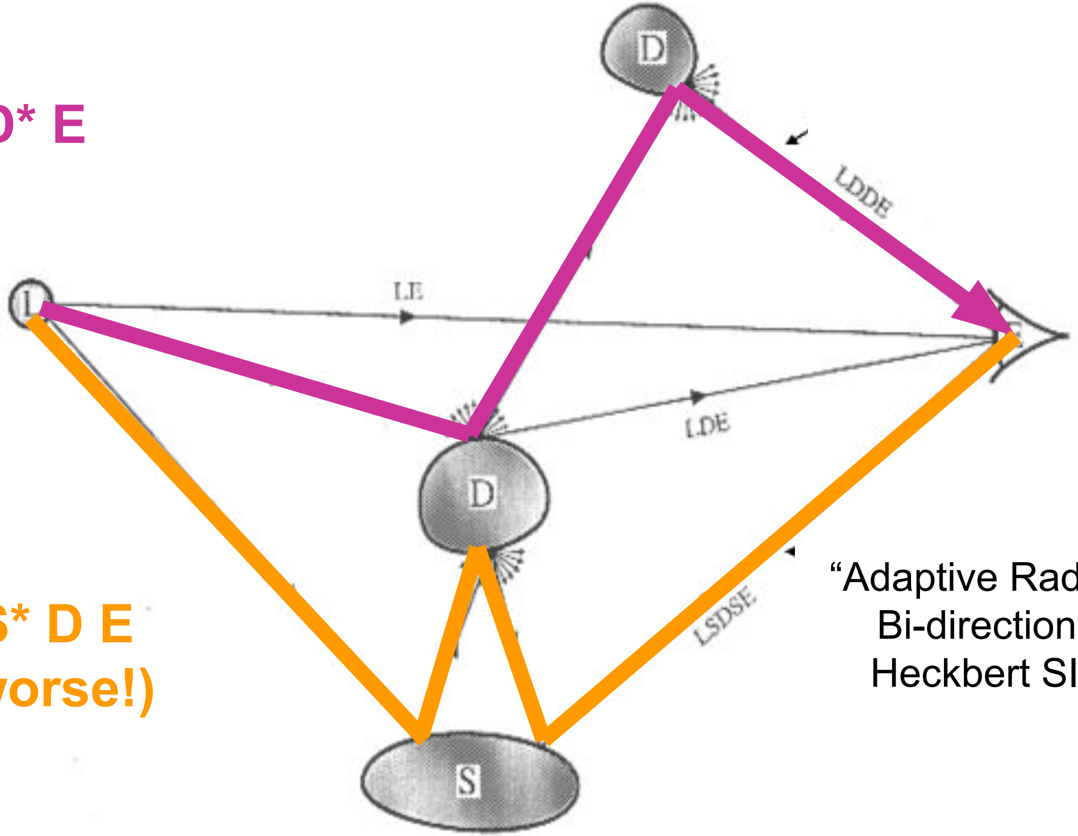
Ray tracing: L D S* E



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

Photon Tracing

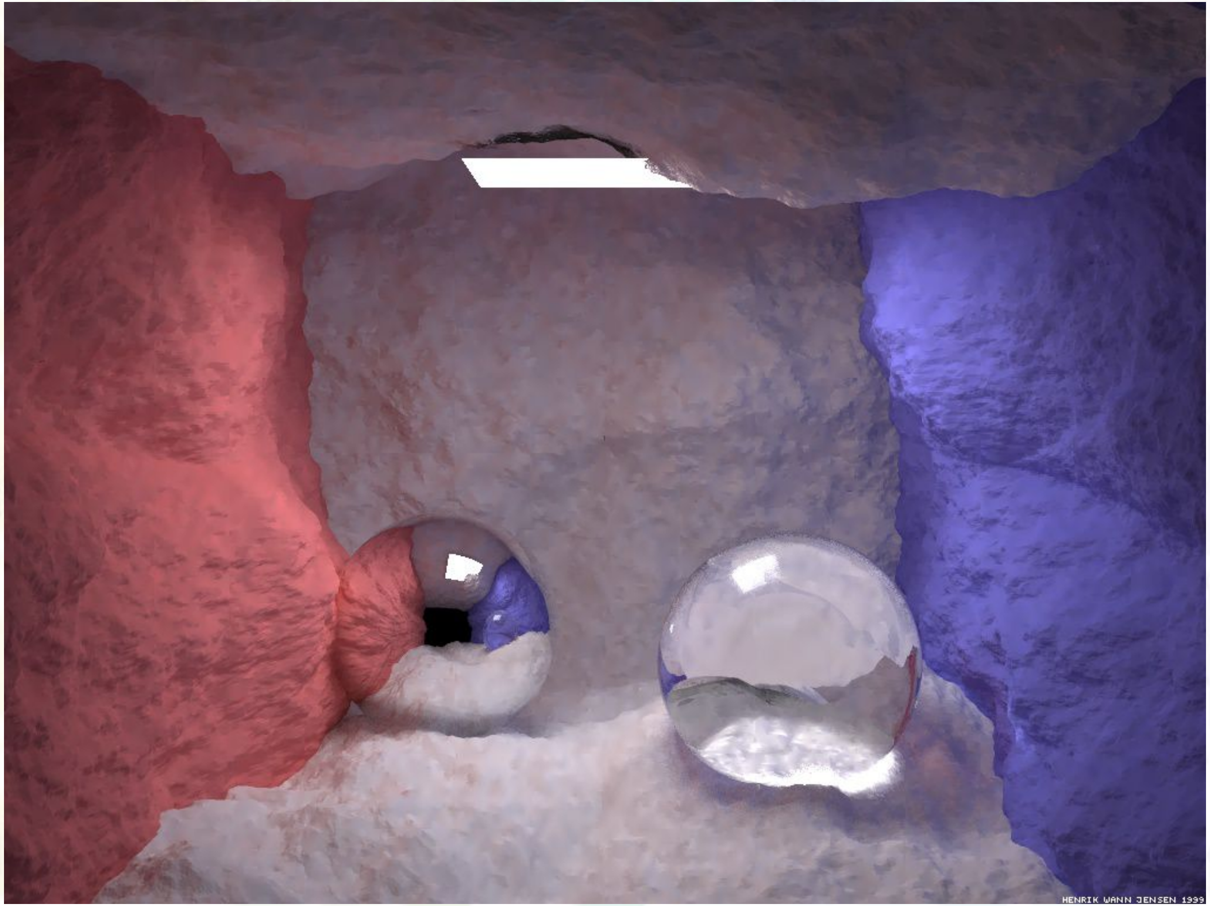
Radiosity: L D* E



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

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

Questions?



Henrik Wann Jensen

Today

- Worksheet: 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 Tuesday After Break: *(pick one)*

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

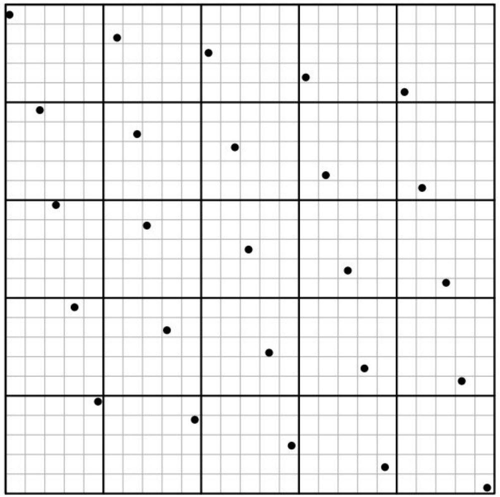


Figure 1: The canonical arrangement. Heavy lines show the boundaries of the 2D jitter cells. Light lines show the horizontal and vertical substrata of N-rooks sampling. Samples are jittered within the subcells.

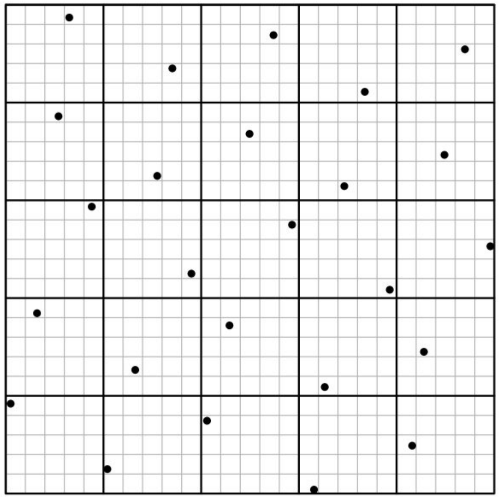


Figure 3: With correlated shuffling.

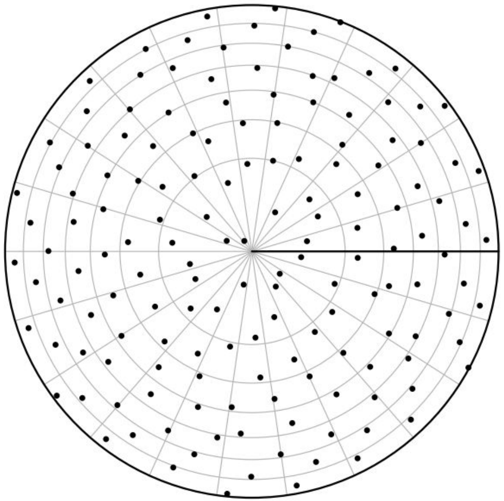


Figure 9: Polar warp with $m = 22, n = 7$.

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

Readings for Tuesday After Break: *(pick one)*

- "Implicit Visibility and Antiradiance for Interactive Global Illumination", Dachsbacher, Stamminger, Drettakis, and Durand Siggraph 2007

Readings for Tuesday After Break: *(pick one)*

- "Recognition of Surface Reflectance Properties from a Single Image under Unknown Real-World Illumination", Dror, Adelson, & Willsky, 2001.

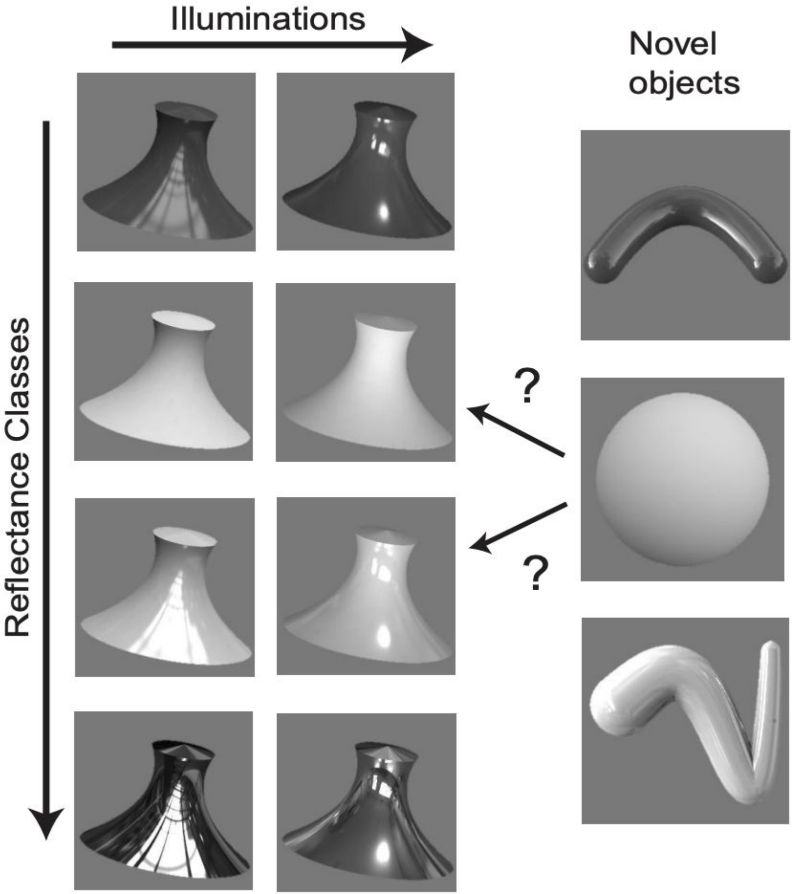


Figure 1. The task addressed by our classifier. Using images of several surface materials under various illuminations as a training set, we wish to classify novel objects under novel illumination according to their surface material.