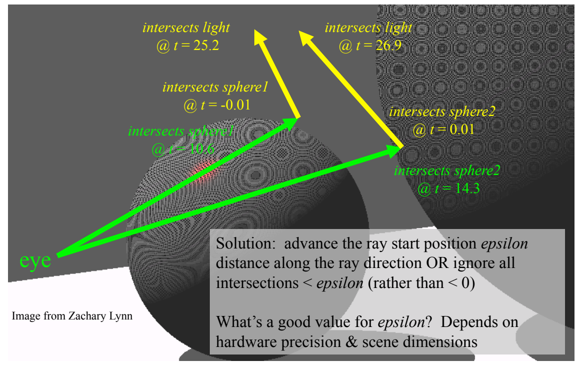
Irradiance Caching & Photon Mapping

Raytracing & Epsilon



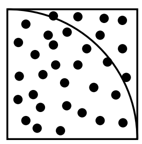
Solution: advance the ray start position *epsilon* distance along the ray direction OR ignore all intersections < *epsilon* (rather than < 0)

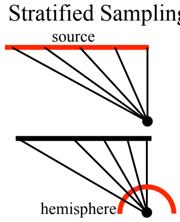
What's a good value for *epsilon*? Depends on hardware precision & scene dimensions

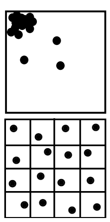
Last Time?

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

- Rendering Equation
- Monte-Carlo Integration
- Monte Carlo Rendering
- Forward Ray Tracing
- Stratified Sampling







Leftover from Last Time & Today

- *Sampling*
  - *Stratified Sampling*
  - *Importance Sampling*
- *Monte-Carlo Ray Tracing vs. Path Tracing*
- Irradiance Caching
- Photon Mapping
- Ray Grammar

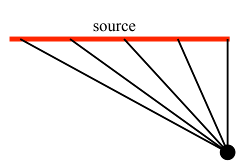
Domains of Integration

- Pixel, lens (Euclidean 2D domain)
- Time (1D)
- Hemisphere
  - Work needed to ensure *uniform* probability

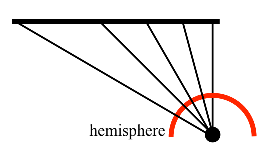
Example: Light Source

- We can integrate over surface *or* over angle
- But we must be careful to get probabilities and integration measure right!

Sampling the source uniformly

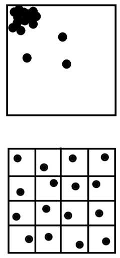


Sampling the hemisphere uniformly



Stratified Sampling

- With uniform sampling, we can get unlucky
  - E.g. all samples in a corner
- To prevent it, subdivide domain Ω into non-overlapping regions Ω_i
  - Each region is called a stratum
- Take one random samples per Ω_i



Stratified Sampling Example

$f(x) = e^{\sin(3x^2)}$

N	I
1	2.75039
10	1.9893
100	1.79139
1000	1.75146
10000	1.77313
100000	1.77862

Unstratified
$O(1/\sqrt{N})$

$f(x) = e^{\sin(3x^2)}$

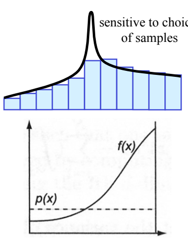
N	I
1	2.70457
10	1.72858
100	1.77925
1000	1.77606
10000	1.77610
100000	1.77610

Stratified
$O(1/N)$

Slide from Henrik Wann Jensen

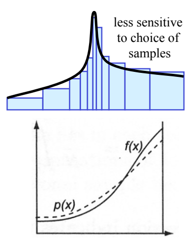
Sampling

uniform sampling
(or uniform random)



all samples
weighted equally

dense sampling where
function has greater magnitude

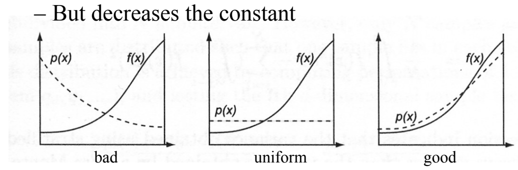


weights (width) for dense
samples are reduced

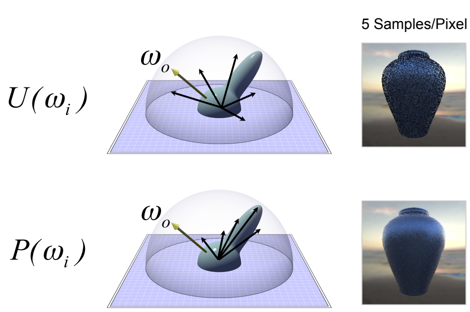
Importance Sampling

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

- Choose *p* wisely to reduce variance
  - Want to use a *p* that resembles *f*
  - Does not change convergence rate (still sqrt)
  - But decreases the constant

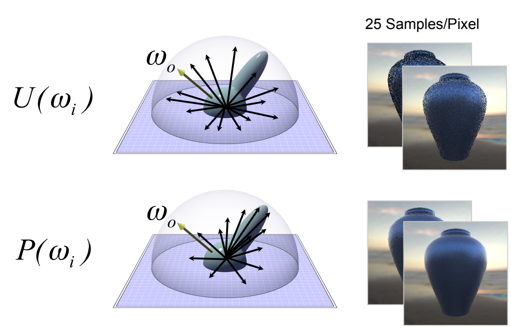


Uniform vs. Importance Sampling



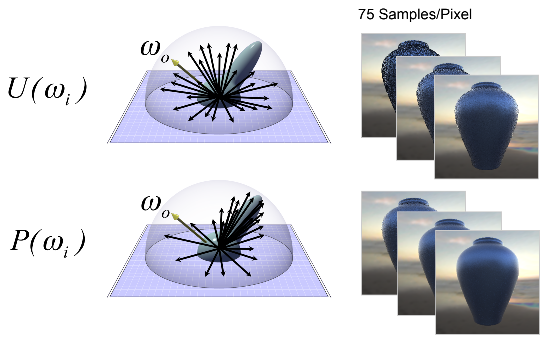
Slide from Jason Lawrence

Uniform vs. Importance Sampling



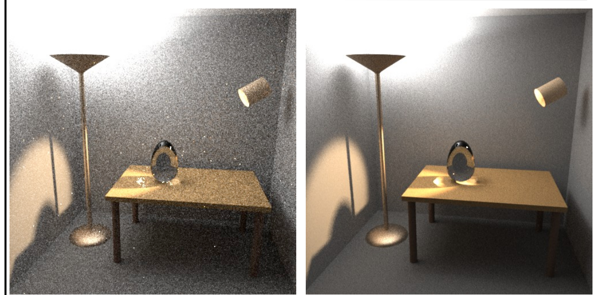
Slide from Jason Lawrence

Uniform vs. Importance Sampling



Slide from Jason Lawrence

Questions?



Naïve sampling strategy

Optimal sampling strategy

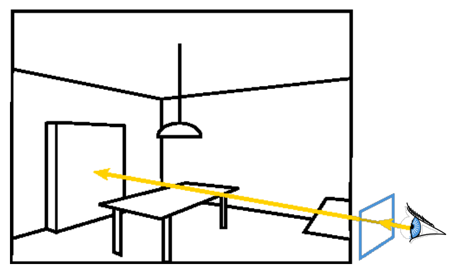
Veach & Guibas "Optimally Combining Sampling Techniques for Monte Carlo Rendering" SIGGRAPH 95

Leftover from Last Time & Today

- *Sampling*
- *Monte-Carlo Ray Tracing vs. Path Tracing*
- Irradiance Caching
- Photon Mapping
- Ray Grammar

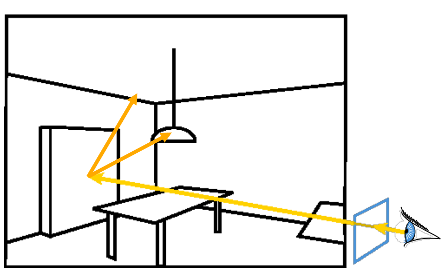
Ray Casting

- Cast a ray from the eye through each pixel



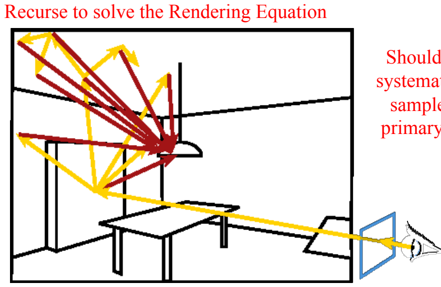
Ray Tracing

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



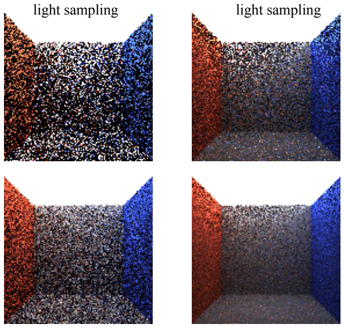
Monte-Carlo Ray Tracing

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



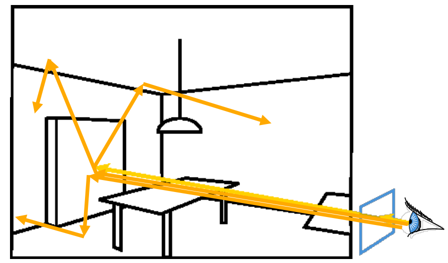
Should also systematically sample the primary light

Importance of Sampling the Light

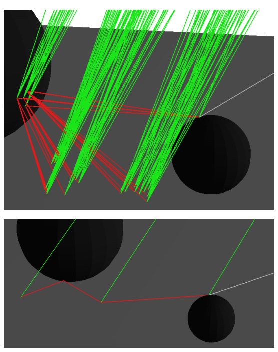


Monte Carlo Path Tracing

- 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?

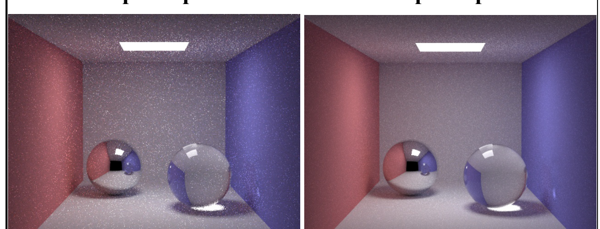
1main ray + 5 shadow rays +
5 glossy rays + 5x5 shadow rays +
5*5 glossy rays + 5x5x5 shadow rays
= 186 rays

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?



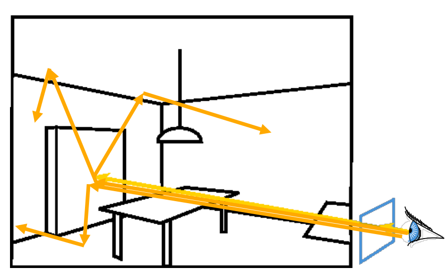
Images from Henrik Wann Jensen

Today

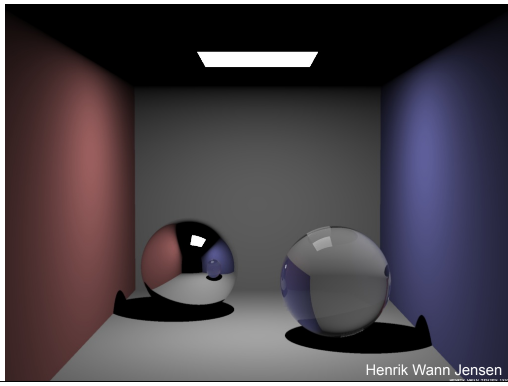
- *Sampling*
- *Monte-Carlo Ray Tracing vs. Path Tracing*
- Irradiance Caching
- Photon Mapping
- Ray Grammar

Path Tracing is costly

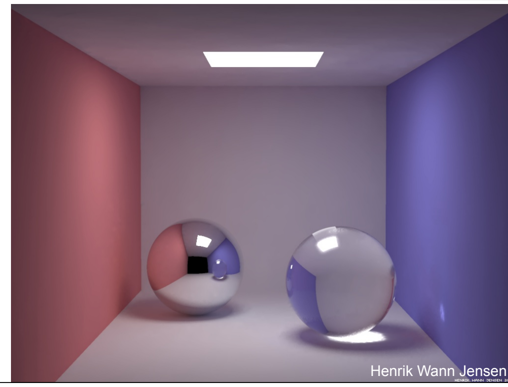
- Needs tons of rays per pixel



Direct Illumination



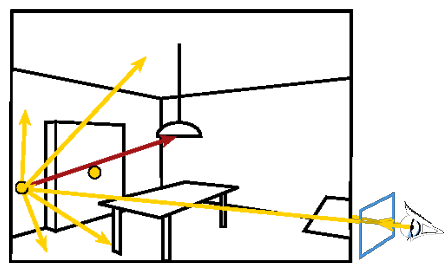
Global Illumination



Indirect Illumination: smooth

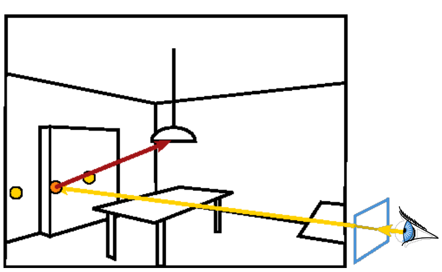
Irradiance Cache

- The indirect illumination is smooth
- Store the indirect illumination

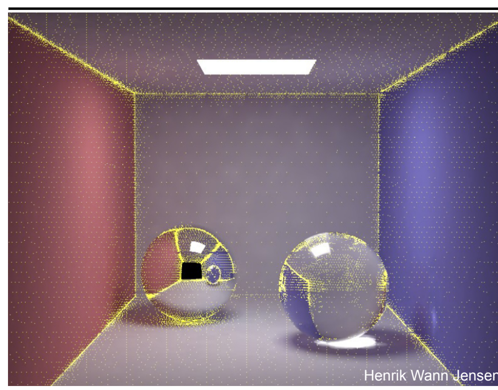


Irradiance Cache

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

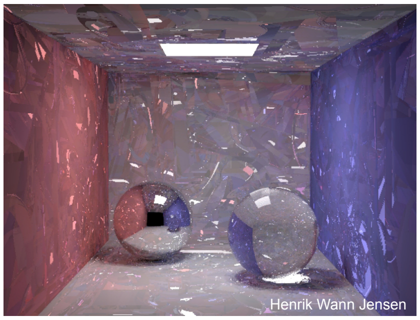


Irradiance Cache



Questions?

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

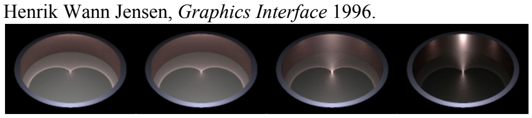


Today

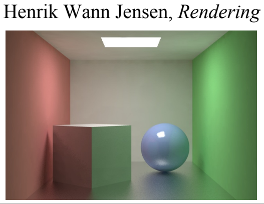
- *Sampling*
- *Monte-Carlo Ray Tracing vs. Path Tracing*
- Irradiance Caching
- Photon Mapping
- Ray Grammar

Readings for Today

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

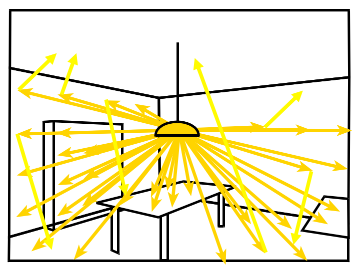


- "Global Illumination using Photon Maps", Henrik Wann Jensen, *Rendering Techniques* 1996.



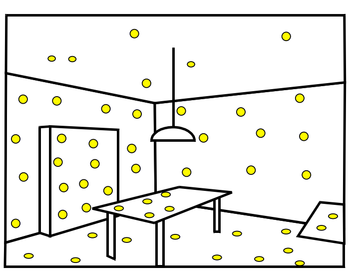
Photon Mapping

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



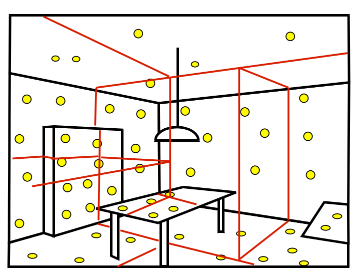
Photon Mapping

- Store photons
  - – position + light power + incoming direction



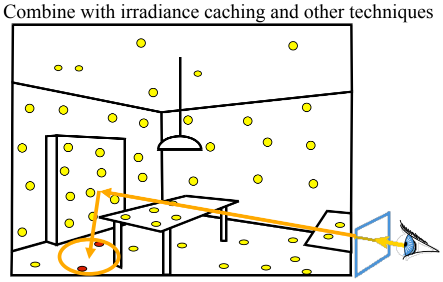
Photon Map

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

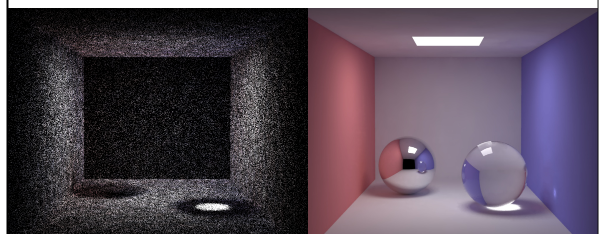


Rendering with Photon Map

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

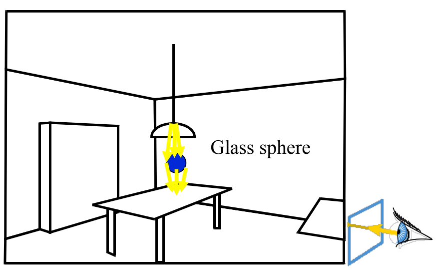


Photon Map Results



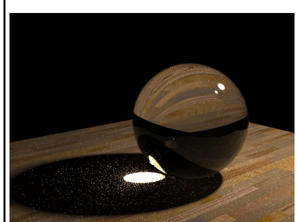
Photon Mapping - Caustics

- Special photon map for specular reflection and refraction

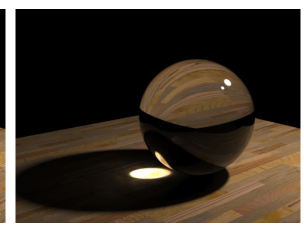


Comparison

Path Tracing
1000 paths/pixel

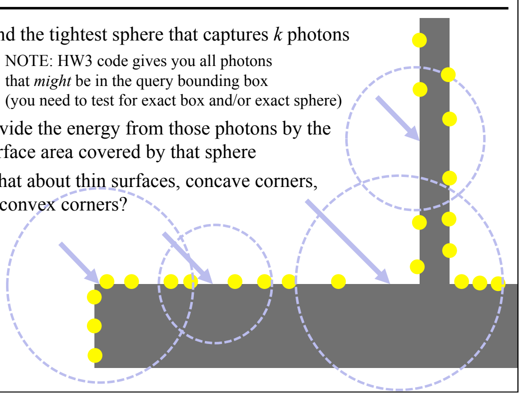


Photon mapping



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?

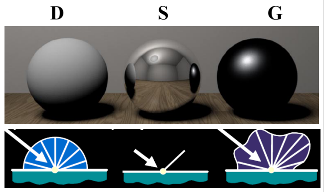


Today

- *Sampling*
- *Monte-Carlo Ray Tracing vs. Path Tracing*
- Irradiance Caching
- Photon Mapping
- Ray Grammar

Ray Grammar

- Classify local interaction:
  - E = eye
  - L = light
  - S = perfect specular reflection or refraction
  - G = glossy scattering
  - D = diffuse scattering

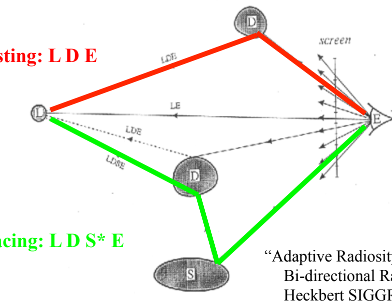


From Dutre et al.'s slides

Classic Ray Casting/Tracing

Ray casting: L D E

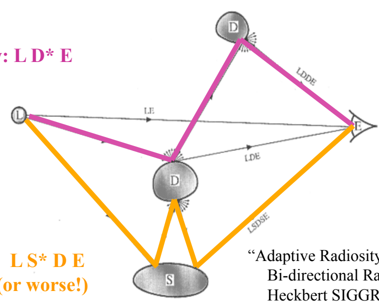
Ray tracing: L D S* E



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

Photon Tracing

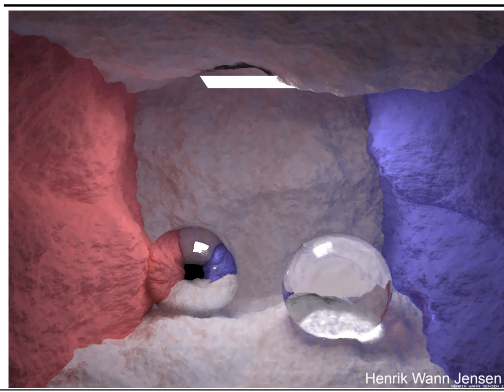
Radiosity: L D* E



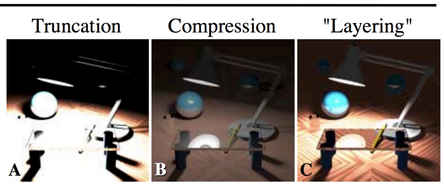
Caustics: L S* D E (or worse!)

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

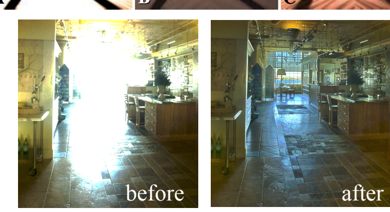
Questions?



Readings for Friday: *(pick one)*



"Two Methods for the Display of High Contrast Images", Tumblin, Hodgins, & Guenter, ACM Transactions on Graphics 1999



"Fast Bilateral Filtering for the Display of High-Dynamic Range Images", Durand & Dorsey, SIGGRAPH 2002