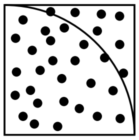
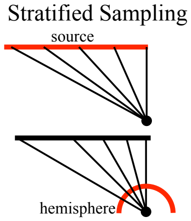
Irradiance Caching & Photon Mapping

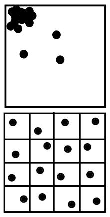
Last Time?

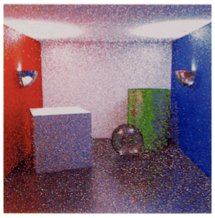
$L(x',\omega') = E(x',\omega') + \int \rho_{x'}(\omega,\omega')L(x,\omega)G(x,x')V(x,x')\,dA$

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







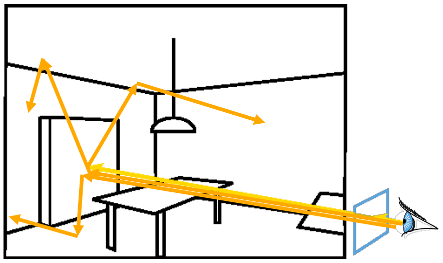


Today

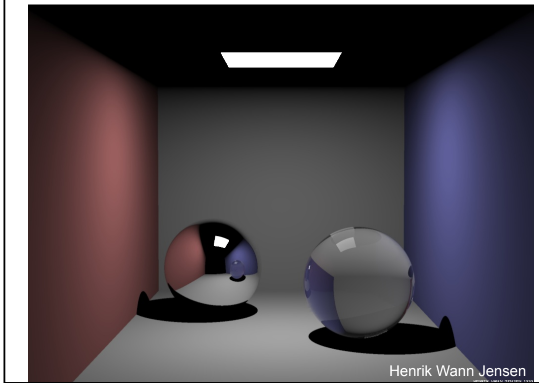
- Irradiance Caching
- Photon Mapping
- Acceleration Data Structures
- Ray Grammar

Path Tracing is costly

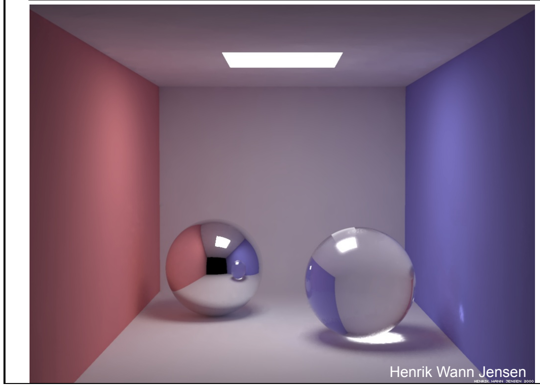
- Needs tons of rays per pixel



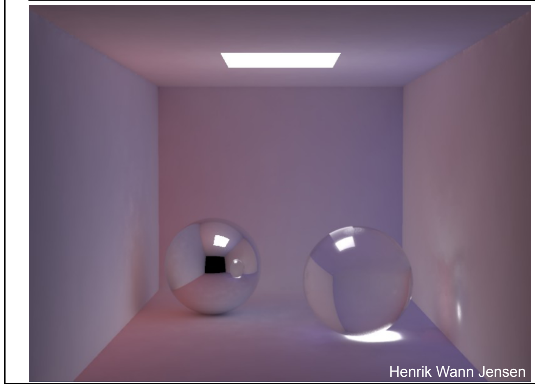
Direct Illumination



Global Illumination

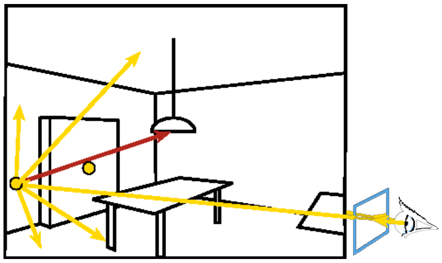


Indirect Illumination: smooth



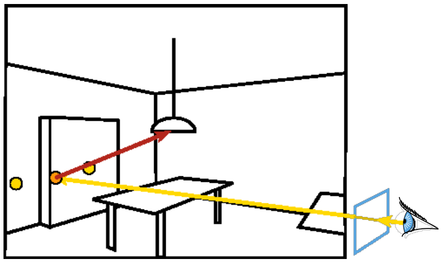
Irradiance Cache

- The indirect illumination is smooth
- Store the indirect illumination

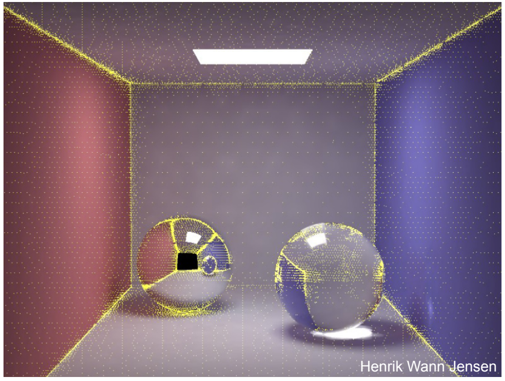


Irradiance Cache

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

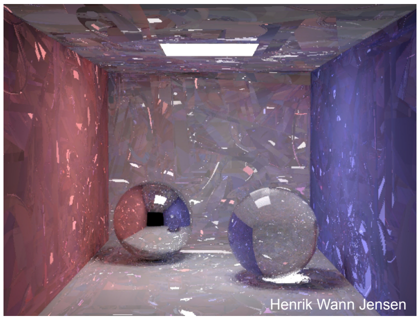


Irradiance Cache



Questions?

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

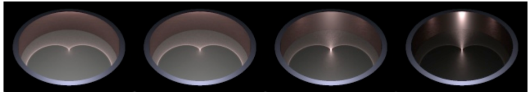


Today

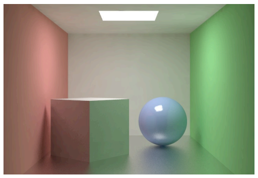
- Irradiance Caching
- Photon Mapping
- Acceleration Data Structures
- Ray Grammar

Readings for Today *(pick one)*

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

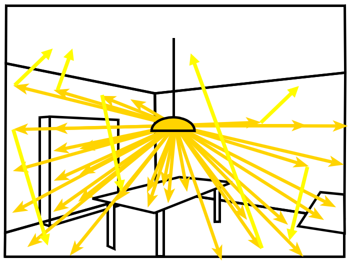


- "Global Illumination using Photon Maps", Henrik Wann Jensen, *Rendering Techniques* 1996.



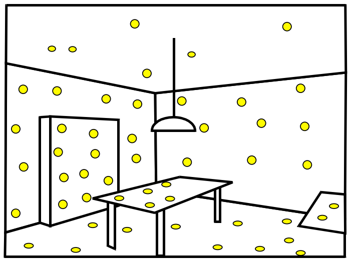
Photon Mapping

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



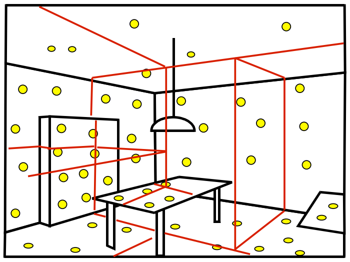
Photon Mapping

- Store photons
  - position + light power + incoming direction



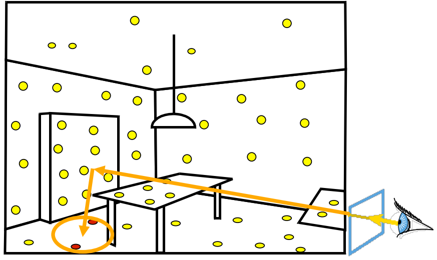
Photon Map

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

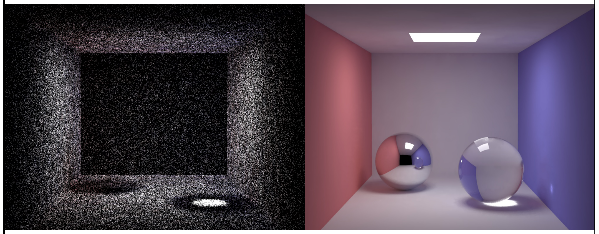


Rendering with Photon Map

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

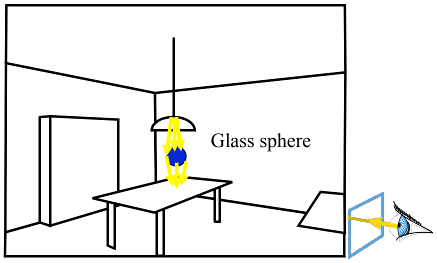


Photon Map Results



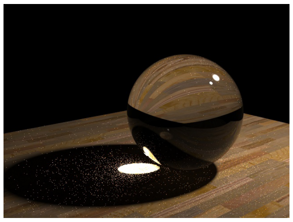
Photon Mapping - Caustics

- Special photon map for specular reflection and refraction

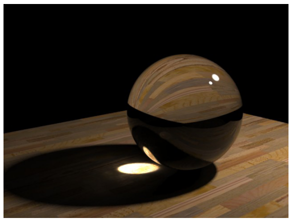


Comparison

Path Tracing
1000 paths/pixel

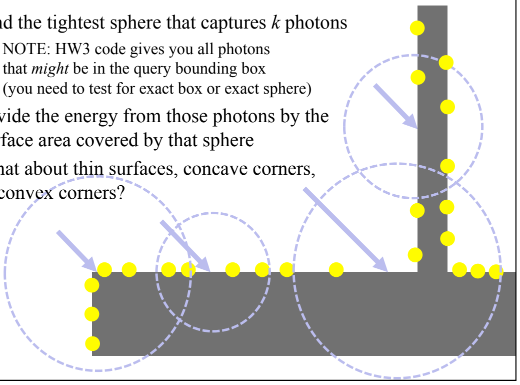


Photon mapping



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 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?

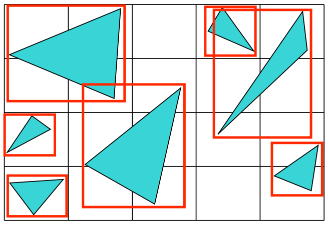


Today

- Irradiance Caching
- Photon Mapping
- Acceleration Data Structures
- Ray Grammar

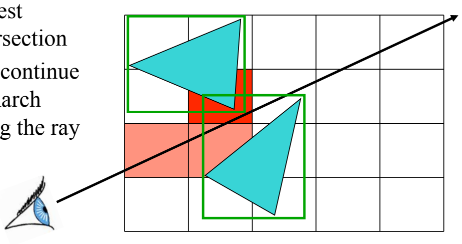
Regular Grid

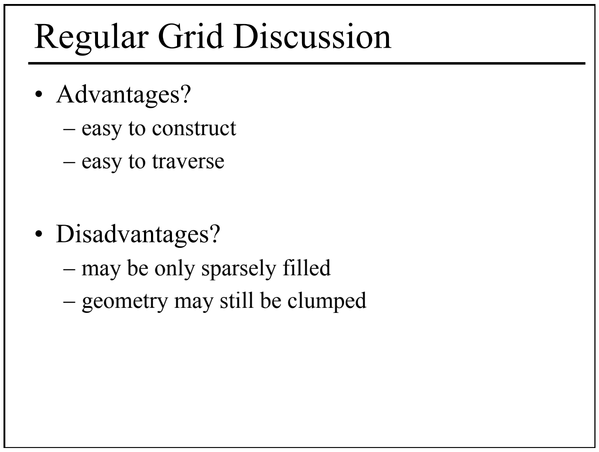
- Primitives that overlap multiple cells?
- Insert into multiple cells (use pointers)



For Each Cell Along a Ray

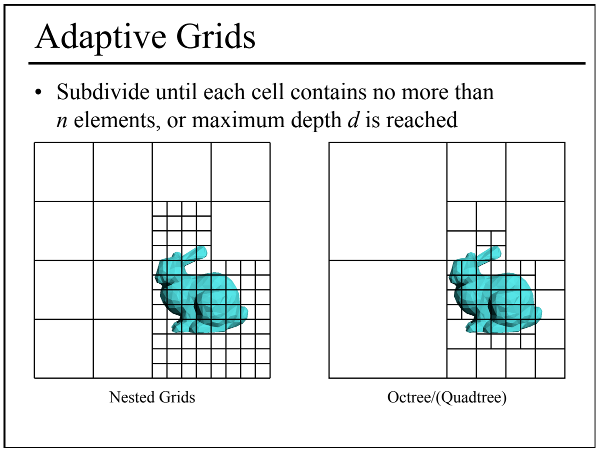
- Does the cell contain an intersection?
- Yes: return closest intersection
- No: continue to march along the ray





Regular Grid Discussion

- Advantages?
  - – easy to construct
  - – easy to traverse
- Disadvantages?
  - – may be only sparsely filled
  - – geometry may still be clumped

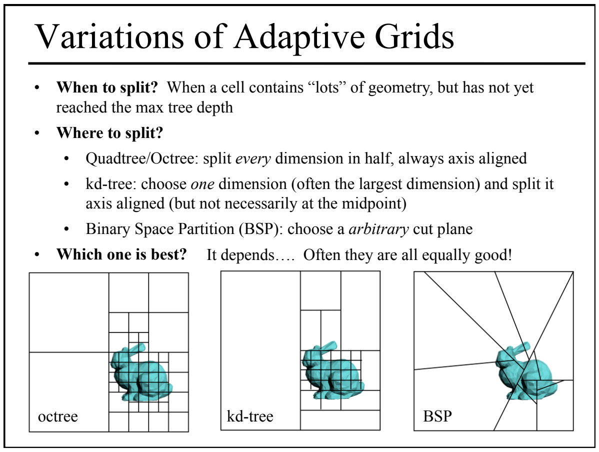


Adaptive Grids

- Subdivide until each cell contains no more than *n* elements, or maximum depth *d* is reached

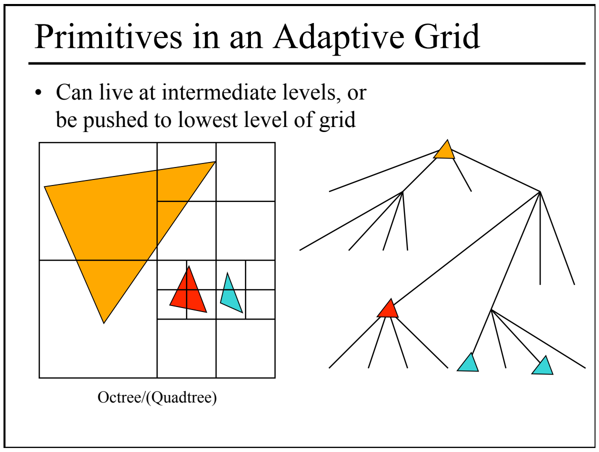
Nested Grids

Octree/(Quadtree)



Variations of Adaptive Grids

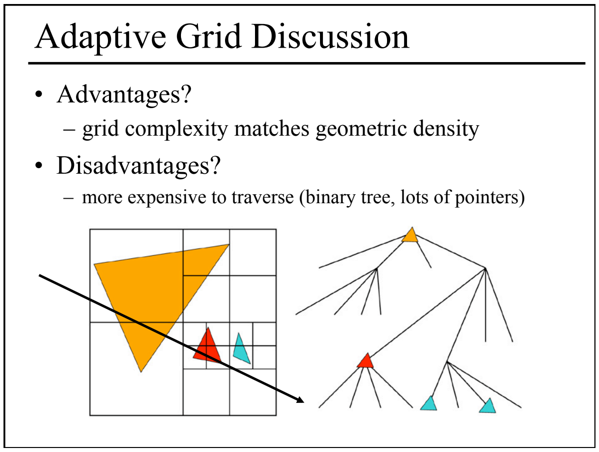
- **When to split?** When a cell contains "lots" of geometry, but has not yet reached the max tree depth
- **Where to split?**
  - Quadtree/Octree: split *every* dimension in half, always axis aligned
  - kd-tree: choose *one* dimension (often the largest dimension) and split it axis aligned (but not necessarily at the midpoint)
  - Binary Space Partition (BSP): choose a *arbitrary* cut plane
- **Which one is best?** It depends…. Often they are all equally good!



Primitives in an Adaptive Grid

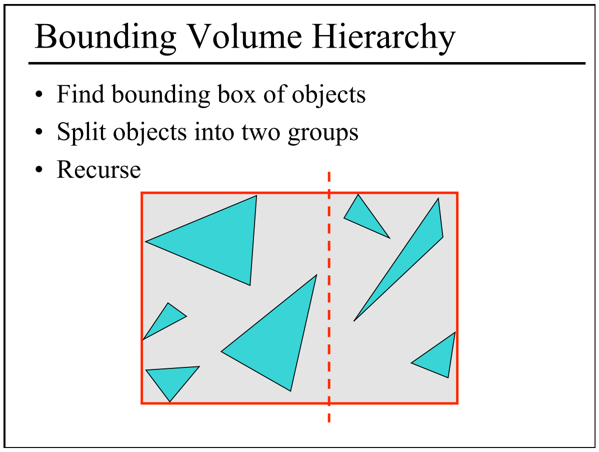
- Can live at intermediate levels, or be pushed to lowest level of grid

Octree/(Quadtree)



Adaptive Grid Discussion

- Advantages?
  - – grid complexity matches geometric density
- Disadvantages?
  - – more expensive to traverse (binary tree, lots of pointers)

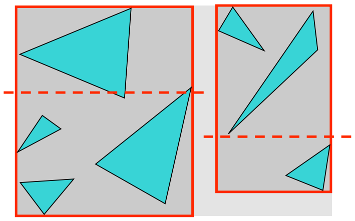


Bounding Volume Hierarchy

- Find bounding box of objects
- Split objects into two groups
- Recurse

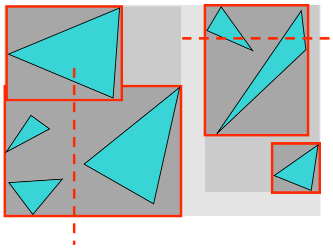
Bounding Volume Hierarchy

- Find bounding box of objects
- Split objects into two groups
- Recurse



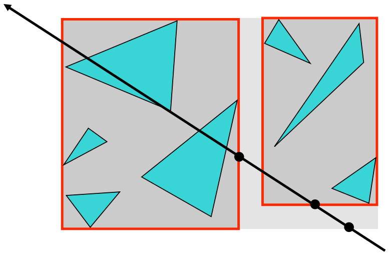
Where to split objects?

- At midpoint *OR*
- Sort, and put half of the objects on each side *OR*
- Use modeling hierarchy



Intersection with BVH

- Check sub-volume with closer intersection first

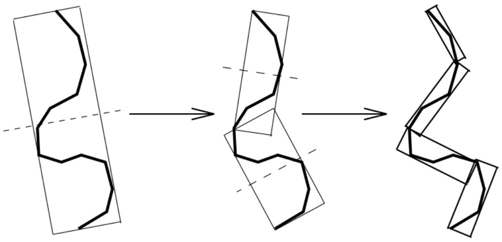


Bounding Volume Hierarchy Discussion

- Advantages
  - easy to construct
  - easy to traverse
  - binary
- Disadvantages
  - may be difficult to choose a good split for a node
  - poor split may result in minimal spatial pruning

Oriented Bounding Box (OBB)

- Generalization of the (axis-aligned) BVH



OBB-Tree: A Hierarchical Structure for Rapid Interference Detection, Gottschalk, Lin, & Manocha, SIGGRAPH 1996.

Today

- Irradiance Caching
- Photon Mapping
- Acceleration Data Structures
- Ray Grammar

Ray Grammar

- Classify local interaction:

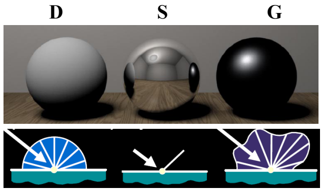
  E = eye

  L = light

  S = perfect specular reflection or refraction

  G = glossy scattering

  D = diffuse scattering

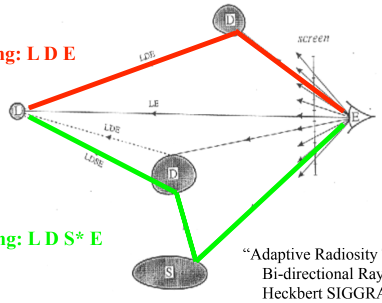


From Dutre et al.'s slides

Classic Ray Casting/Tracing

Ray casting: L D E

Ray tracing: L D S* E

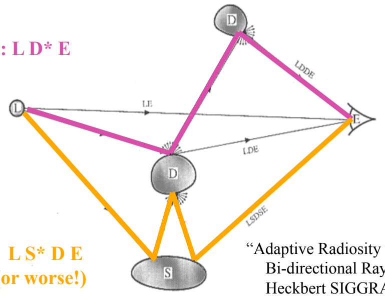


"Adaptive Radiosity Textures for Bi-directional Ray Tracing" Heckbert SIGGRAPH 1990

Photon Tracing

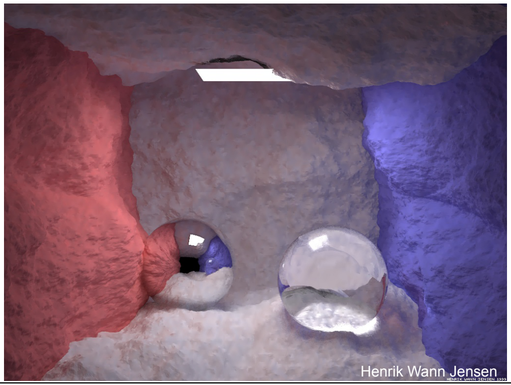
Radiosity: L D* E

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



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

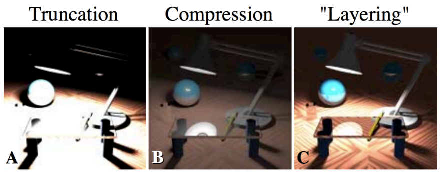
Questions?



Henrik Wann Jensen

Readings for Friday 3/25 *(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