Lecture 12: Photon Mapping

CS 6620, Spring 2009

Kavita Bala
Computer Science
Cornell University

Biased Methods

- MC problems
- Biased methods: store information (caching)
  - Better type of noise: blurring
  - Greg Ward's Radiance
  - Photon Mapping

  - Radiosity (even)
  - Assumption: common case is diffuse

Irradiance Caching

- Introduced by Greg Ward 1988
- Implemented in RADIANCE
  - Public-domain software
- Exploits smoothness of irradiance
  - Cache and interpolate irradiance estimates
  - Also has error, but the right kind of error

Basic Idea

- “Cache” directionally invariant irradiance when possible
  - Path reuse
- When we need irradiance at a new point
  - Compute irradiance estimate using cached samples
- Where do samples come from?
- How do we store them?
- How do we compute the irradiance estimate?
- When do we use this cache of irradiance values?

What information to store?

$$E = \int_{\substack{Solid\\Angle}} L(x \leftarrow \Psi) \cdot \cos\theta \cdot d\omega_\Psi$$

$$E = \int \int L(x \leftarrow \Psi) \cdot \cos\theta \cdot \sin\theta \cdot d\theta d\phi$$

$$E = \frac{\pi}{TP} \sum_t \sum_p L_i(\theta_t, \phi_p)$$

- Compute Li using Monte Carlo

Validity of Sample

- Assign a region where a sample can be reused
  - The radius
- Larger region where irradiance varies smoothly
- Smaller region when it varies a lot

Distance to surfaces

- Harmonic mean heuristic
- Range of sample is given by a radius Ri

$$\frac{1}{R_i} = \frac{1}{N}\sum \frac{1}{d_j}$$

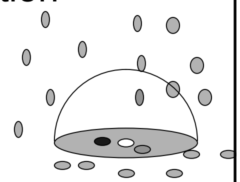
- Where dj is distance to closest surface
- Why not just average?

Storing Samples

- Question of precisely what to store:
  - Irradiance (Ei)
  - The location (xi)
  - The normal at that location (ni)
  - The distance that we went to find nearby surfaces (Ri)

How to compute the irradiance estimate?

- When new sample requested
  - Query octree for samples near location
  - Check ε at x, x_i is a nearby sample



$$\varepsilon_i(x,\vec{n}) = \frac{\| x_i - x \|}{R_i} + \sqrt{1 - \vec{n} \bullet \vec{n}_i}$$

  - Weight samples inversely proportional to ε_i

$$E(x,\vec{n}) = \frac{\sum_{i, w_i > 1/a} w_i(x,\vec{n}) E_i(x_i)}{\sum_{i, w_i > 1/a} w_i(x,\vec{n})}$$

  - Otherwise, compute new sample

Summary: Irradiance Caching

- Start with a basic path tracing algorithm, but …
  - Store irradiance in octree
  - If hit diffuse surface, build an estimate of irradiance at that point
  - Always compute direct lighting explicitly
  - If estimate is good, use it
  - If not, use path tracing to estimate the irradiance and store it

Regexps to classify paths

- **L** light source
- **D** diffuse reflection
- **G** glossy / semi-diffuse reflection
- **S** perfect specular reflection
- **E** eye

Regexps to classify paths

- “Looking” directly at a light source
  - **L E**
- Seeing a diffuse surface
  - **L D E**
- Seeing a diffuse surface reflected in a mirror
  - **L D S E**

Regexp operators

- Standard stuff:

  ***** zero or more

  + one or more

  ? zero or one

  | or

The regexp quiz!

- "All possible" paths
  - **L (D|G|S)* E**
- 'Plain' recursive raytracing?
  - **L D S* E**
  - …maybe **L (D|G) S* E**
- Directly rendering radiosity?
  - **L D* E**

© Kavita Bala, Computer Science, Cornell University

- Bidirectional path tracing is…?
  - **L (G|D|S)* (G|D|S)* E**
  - = **L (G|D|S)* E**

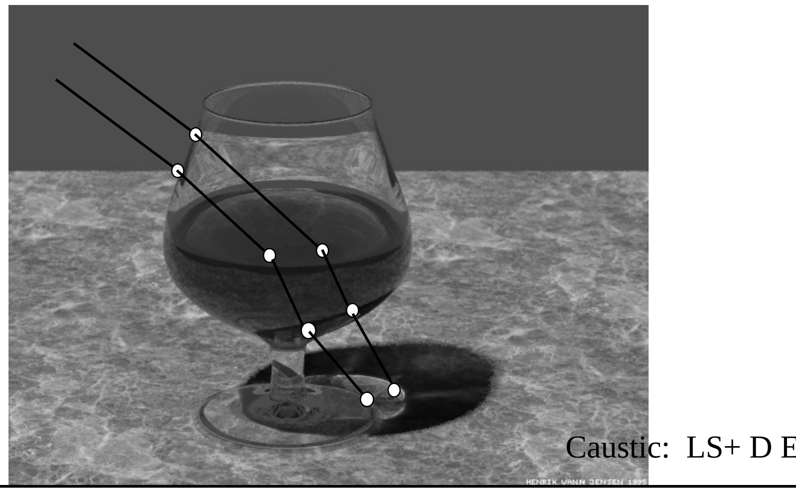
Light Tracing

- Some sub-paths are best sampled from the light
  - Where light is focused – *caustics*
  - Where most of the light's energy goes to a small region of the world

- Several variants

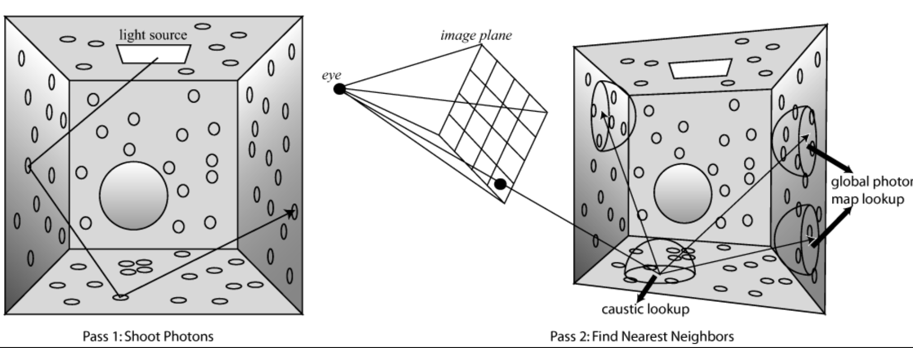
Photon Map

- Build on irradiance caching
- Use bidirectional ray tracing

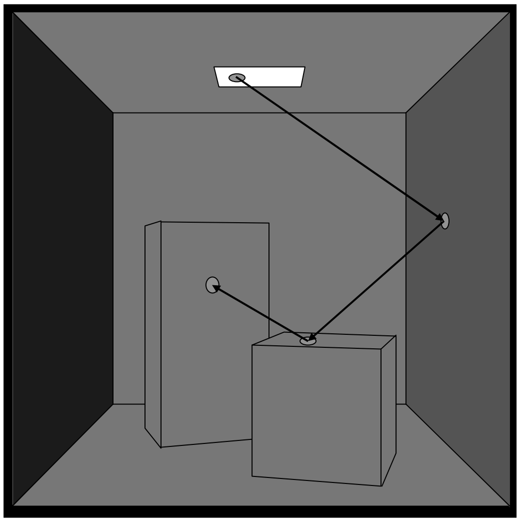


Photon Map

- 2 passes:
  - shoot “photons” (light-rays) and record any hit-points
  - shoot viewing rays, collect information from stored photons

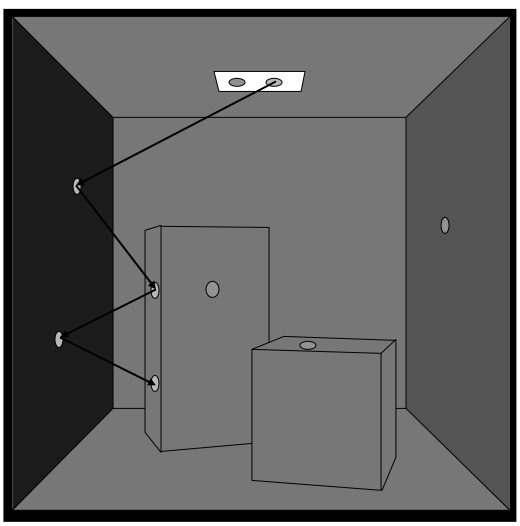


Pass 1: shoot photons



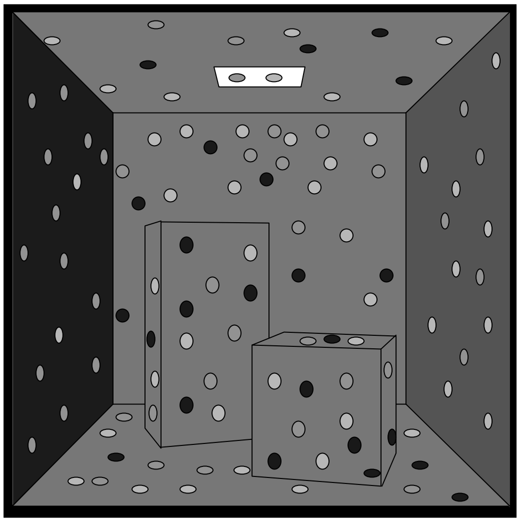
- Light path generated using MC techniques and Russian Roulette
- Store:
  - position
  - incoming direction
  - color
  - ...

Pass 1: shoot photons



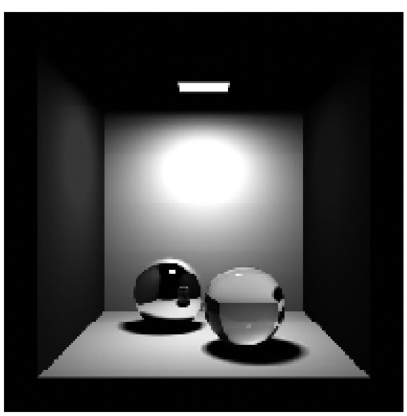
- Light path generated using MC techniques and Russian Roulette
- Store:
  - position
  - incoming direction
  - color
  - ...

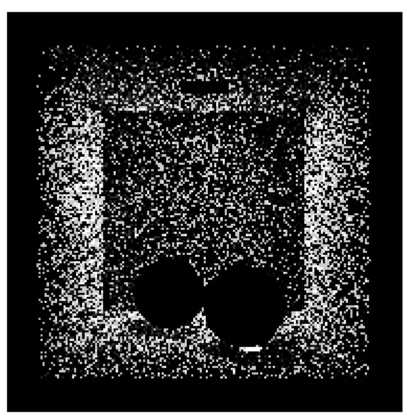
Pass 1: shoot photons



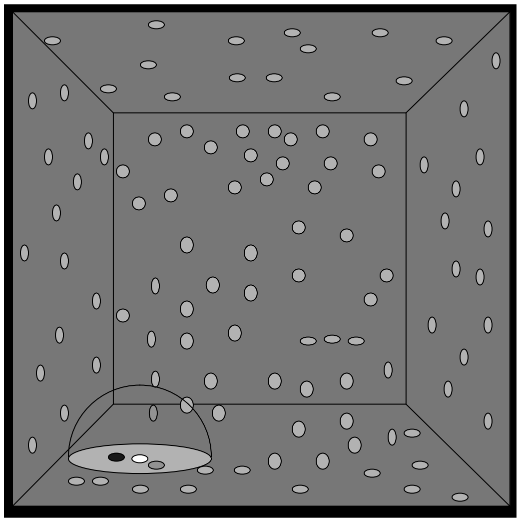
- Light path generated using MC techniques and Russian Roulette
- Store:
  - position
  - incoming direction
  - color
  - ...

Pass 1: shoot photons



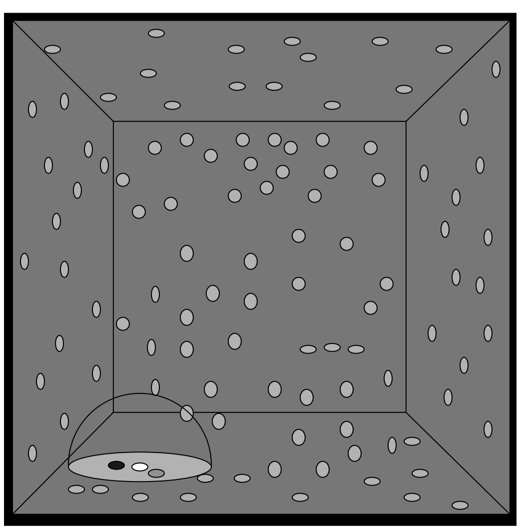


Pass 2: viewing ray (naive)



- Search for N closest photons
- Assume these photons hit the point we're interested in
- Compute estimate

Pass 2: viewing ray (better)



- Search for N closest photons (+check normal)
- Assume these photons hit the point we're interested in
- Compute estimate

Direct Visualization

Figure 4: Direct visualization of the global photon map in the Museum scene

Does this work?

- Using photon maps as is create noisy images
  - – Need EXTREMELY large amount of photons
  - – Filtering techniques can be used with different type of kernels
    - ▪ The filtered results often look too blurry !!!
- Only use photon maps for some paths

© Kavita Bala, Computer Science, Cornell University

Indirect Visualization

Figure 3: The Museum scene

Figure 4: Direct visualization of the global photon map in the Museum scene

Photon Map

- Break lighting into several components
  - – LDS*E and LS*E paths (direct + specular)
  - – LS+DS*E paths (caustic)
  - – L(S|D)*DDS*E paths (indirect)
- Use a different technique for each component
  - – Note that S*E part is common – always start by tracing rays from the eye through specular bounces to a diffuse surface (or a light)

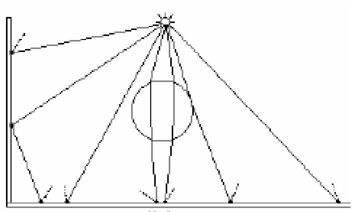
Two Photon Maps

- Create two maps
  - Global photon map
  - Caustic map

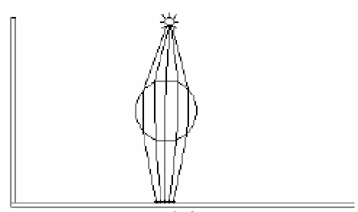
- Global map
  - Low resolution, viewed indirectly
- Caustic map
  - High resolution, viewed directly

2 Photon Maps

- Global Photon Map
  - Low resolution
    - Captures low frequency variation in shading
  - Will not be viewed directly
  - Not good enough for caustics and specular



Global photon map



Caustic photon map

2 Photon Maps

- Caustic Map
  - High resolution
  - But need a projection map
- Projection Map
  - Guide photons in directions that are important
- When does it not work?

Reflection Equation Decomposition

$$f_r(\underline{\omega}_i, \underline{x}, \underline{\omega}_o) = f_{r,spec}(\underline{\omega}_i, \underline{x}, \underline{\omega}_o) + f_{r,diffuse}(\underline{\omega}_i, \underline{x}, \underline{\omega}_o)$$

$$L(\underline{x}, \underline{\omega}_i) = L_{direct}(\underline{x}, \underline{\omega}_i) + L_{caustic}(\underline{x}, \underline{\omega}_i) + L_{indirect}(\underline{x}, \underline{\omega}_i)$$

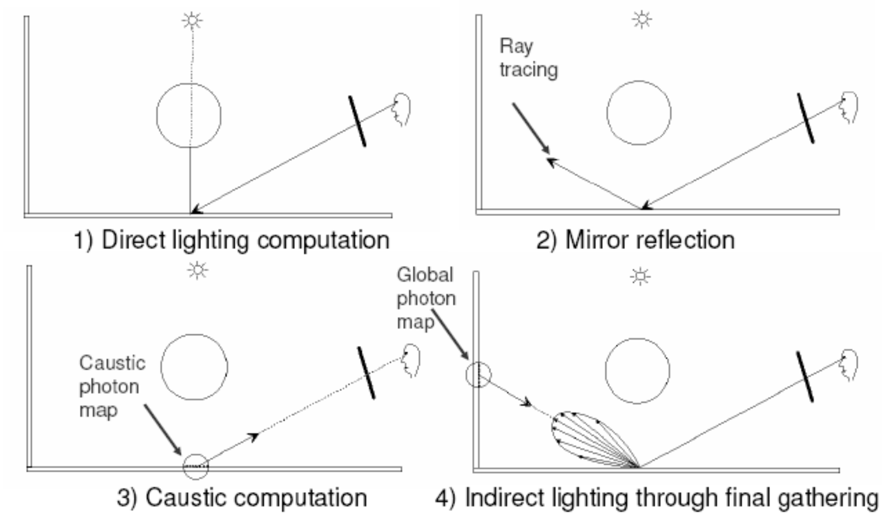
$$L\ (\underline{x}, \underline{\omega}_o) = \int_{\Omega_+} f_r(\underline{\omega}_i, \underline{x}, \underline{\omega}_o) L(\underline{x}, \underline{\omega}_i) \cos\theta_i \, d\omega_i =$$

1) $$\int_{\Omega_+} f_r(\underline{\omega}_i, \underline{x}, \underline{\omega}_o) L_{direct}(\underline{x}, \underline{\omega}_i) \cos\theta_i \, d\omega_i +$$

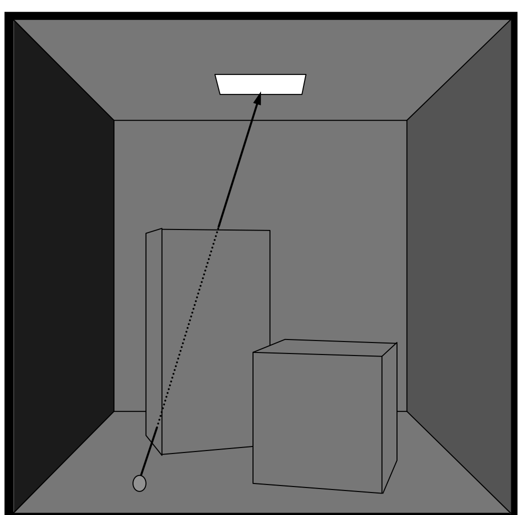
2) $$\int_{\Omega_+} f_{r,spec}(\underline{\omega}_i, \underline{x}, \underline{\omega}_o)(L_{caustic}(\underline{x}, \underline{\omega}_i) + L_{indirect}(\underline{x}, \underline{\omega}_i)) \cos\theta_i \, d\omega_i +$$

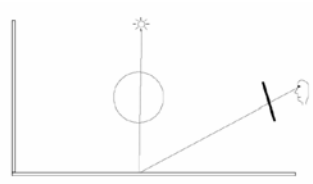
3) $$\int_{\Omega_+} f_{r,diffuse}(\underline{\omega}_i, \underline{x}, \underline{\omega}_o) L_{caustic}(\underline{x}, \underline{\omega}_i) \cos\theta_i \, d\omega_i +$$

4) $$\int_{\Omega_+} f_{r,diffuse}(\underline{\omega}_i, \underline{x}, \underline{\omega}_o) L_{indirect}(\underline{x}, \underline{\omega}_i) \cos\theta_i \, d\omega_i$$

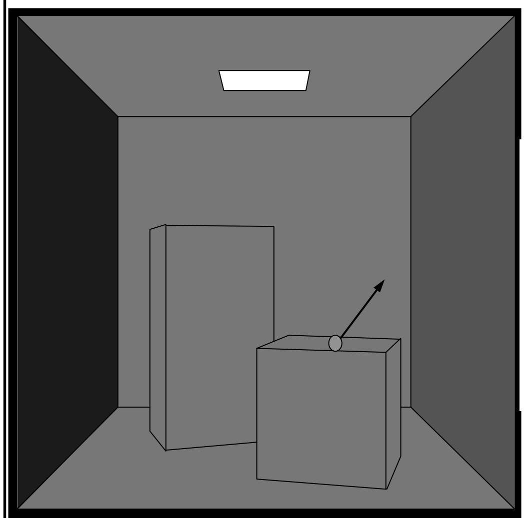


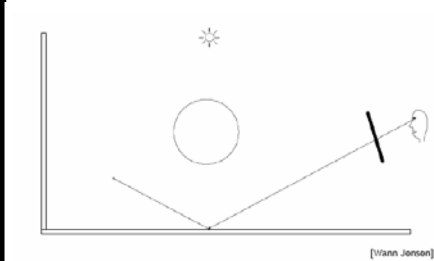
Pass 2: Direct Illumination LDS*E



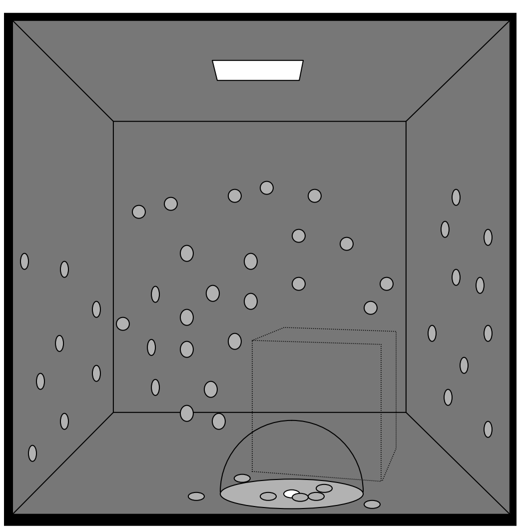


Pass 2: Specular reflections LS*E



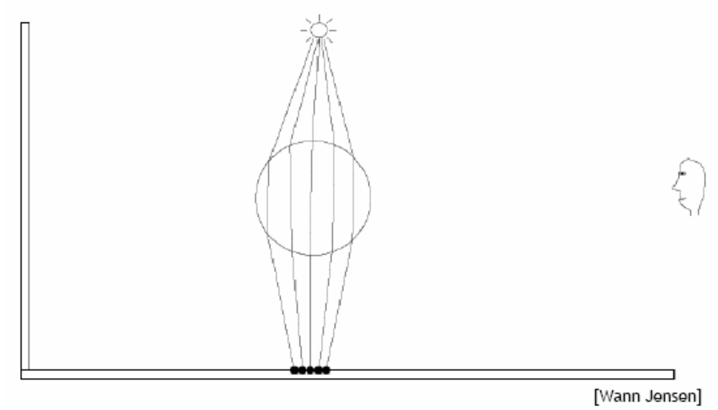


Pass 2: Caustics LS*DS*E



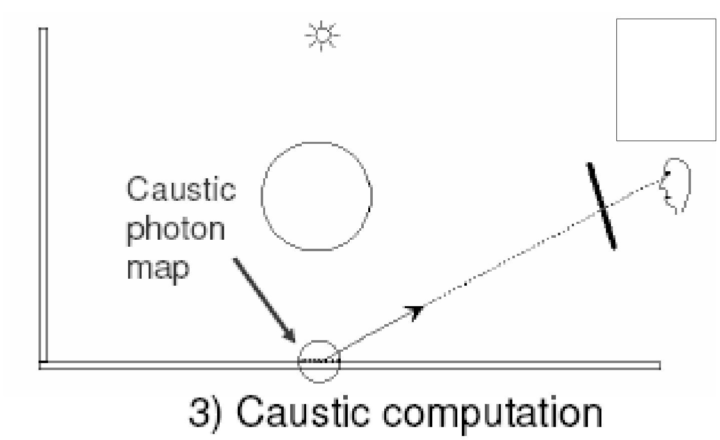
- Direct use of “caustic” maps
- The “caustic” map is similar to a photon map but treats LS*D path
- Density of photons in caustic map usually high enough to use as is

Pass 1: Caustics LS*DS*E



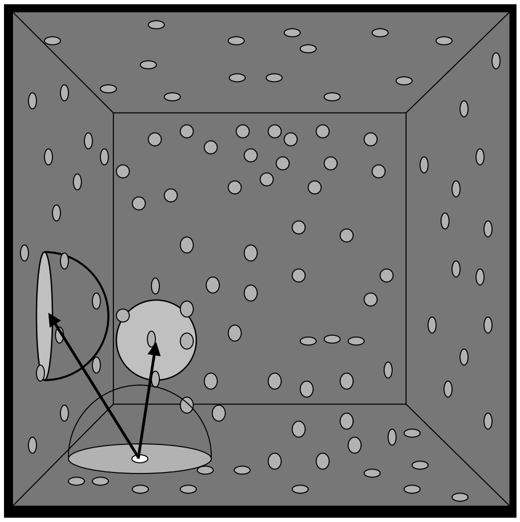
Need projection map to compute caustic map

Pass 2: Visualize caustic map directly



Pass 2:Indirect Diffuse

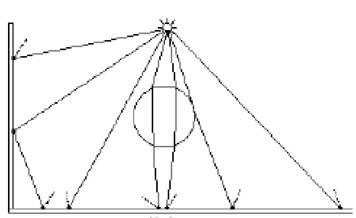
L(S|D)*DDS*E



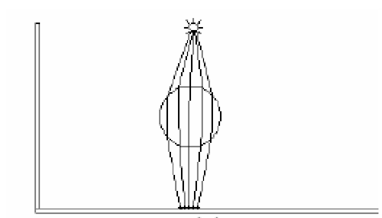
- Compute estimate by importance sampling of hemisphere
- For each of them
  – Search for N closest photons
  – Assume these photons hit the point

2 Photon Maps

- Global map: low resolution, viewed indirectly
- Caustic map: high resolution, viewed directly



Global photon map



Caustic photon map

- Need projection map to compute caustic map