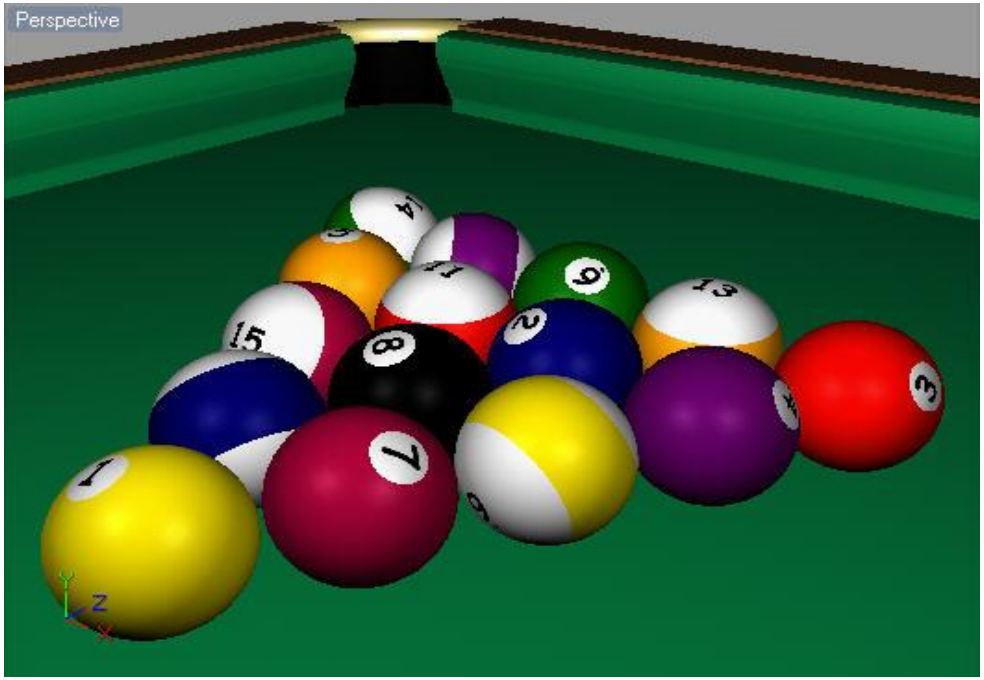
Global Illumination: Radiosity, Photon Tracing, Photon Mapping

Before We Go to Photon Mapping

Local vs. Global Illumination

- The techniques of rendering
  - Local illumination techniques
  - Global illumination techniques



Local Illumination Methods

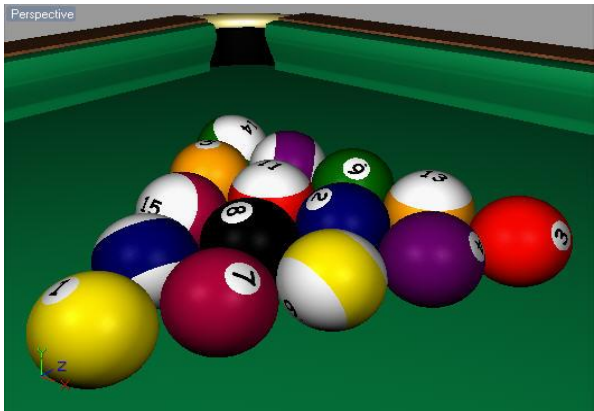
Considers light sources and surface properties only.

Phong Illumination, Phong shading, Gouraud Shading

Using techniques like Shadow maps, shadow volume, shadow texture for producing shadows

Very fast

Used for real-time applications such as 3D computer games



Rendering Equations

- The total light leaving a point =
  - Exitance from the point +
  - Incoming light from other sources reflected at the point

$$L(\mathbf{x},\theta_o,\phi_o) = L_e(\mathbf{x},\theta_o,\phi_o) + \int_{\Omega} \rho_{\mathbf{x}}(\theta_o,\phi_o,\theta,\phi) L_i(\mathbf{x},\theta,\phi)\cos\theta d\omega$$

Light leaving

Exitance

Sum

BRDF

Incoming light

Incoming light reflected at the point

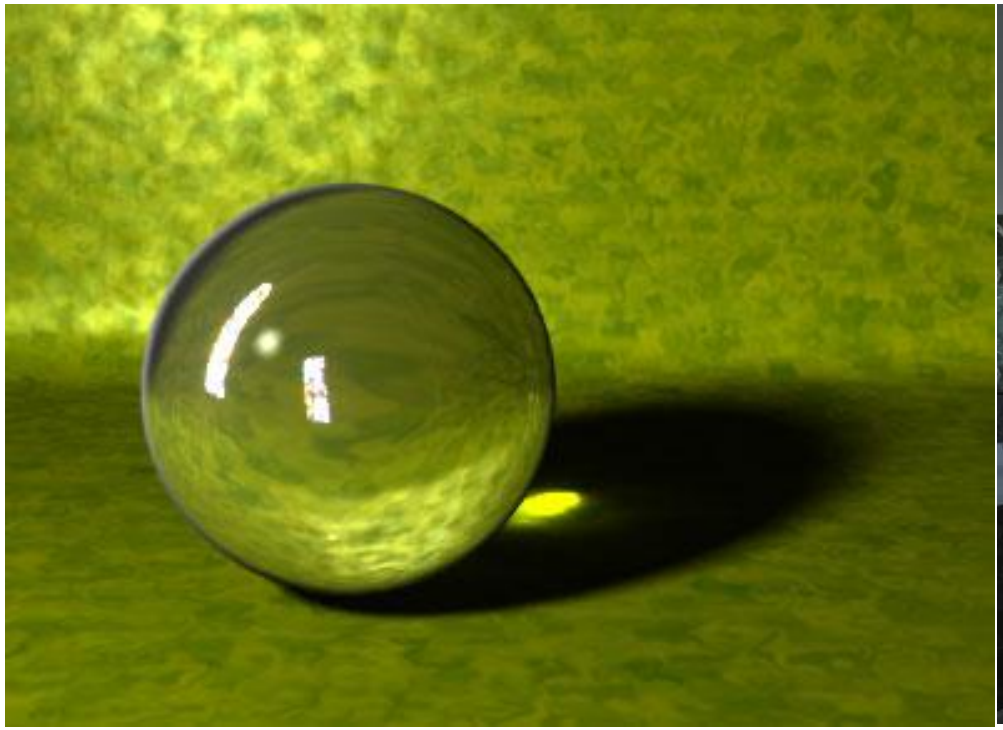
Photorealistic Lighting

- Solve the equation
  - To compute light leaving x, need to find how much light reaches it
  - To find how much light reaches x, need to compute light leaves every other point
  - Hard because BRDFs are high dimensional
  - But some light interaction in the scene is diffused
    - View angle independent

Global Illumination

- Traditional Ray-tracing only considers specular reflections
  - Does not account for (indirect) diffused reflections reaching eye
- Need to “trace” those rays also
  - Stochastic methods
  - Monte Carlo techniques to solve rendering equation
- Separate diffused and specular reflection
  - Radiosity

Global Illumination Techniques



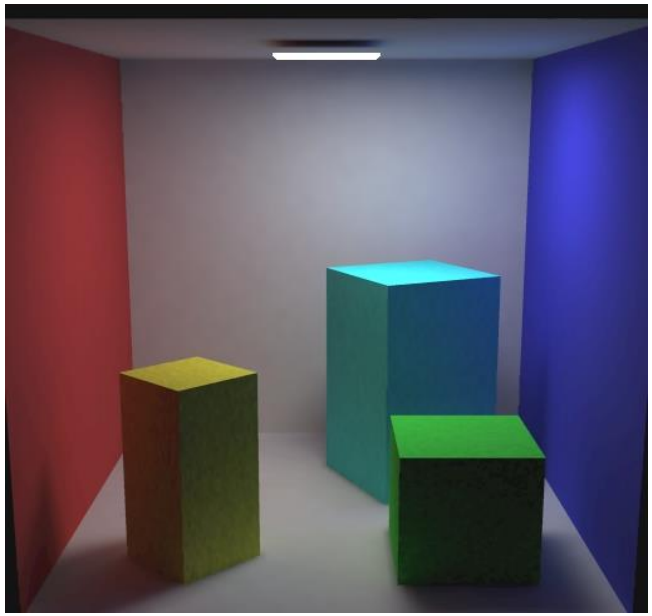
Global Illumination

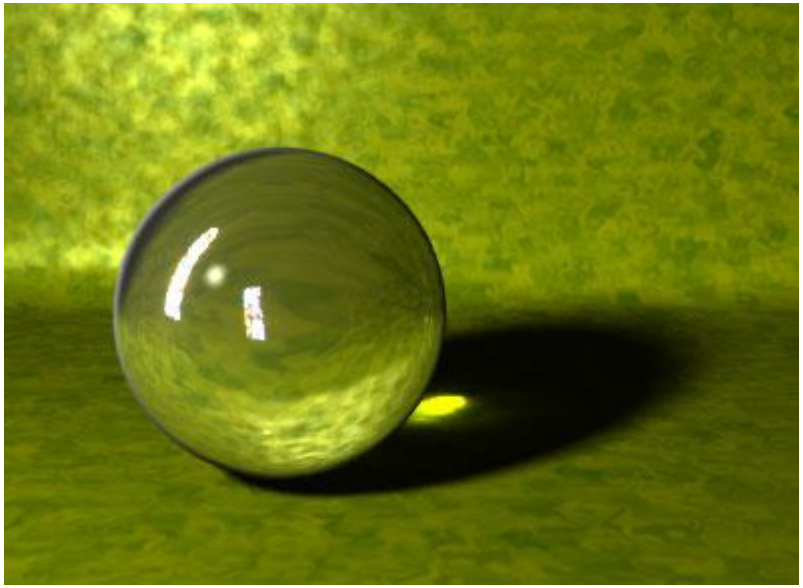
- Methods that simulate not only the direct illuminations but also the light and indirect illuminations
    - Monte-Carlo ray tracing
    - Radiosity, Photon Mapping
- Global illuminations can handle
  - Reflection (one object in another)
  - Refraction (Snell's Law)
  - Shadows
  - Caustics

  under the same frame work
- Requires more computation and is slow

Global Illumination Methods

- Radiosity (classic)
- Photon Mapping (relatively new)





Radiosity Assumptions

- All surfaces are perfectly diffuse
- Illumination is constant over a patch
- Subdivide triangles into small patches
  - Problems at sharp illumination boundaries, e.g., shadows
  - Less space/time efficient solutions
  - Discontinuity meshing
- Can be pre-computed
  - With view-dependent illumination computed and added later

The Radiosity Method

- View independent
- The rendering calculation does not have to be done even though the viewpoint is changed
- The basic method can only handle diffuse color

  → need to be combined with ray-tracing to handle specular light

(a)

Radiosity Example

- Extreme color bleeding here
- Textures are post illumination
- Notice meshing artifacts like the banding around the pictures on the wall

From Alan Watt, “3D Computer Graphics”

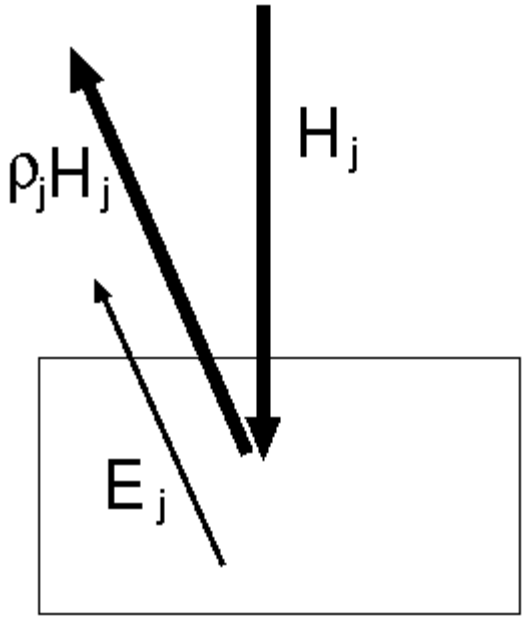
Radiosity Meshing

- Each patch is colored with its illumination
- The previous image was obtained by pushing color to vertices and then Gourand shading

From Alan Watt, “3D Computer Graphics”

The Radiosity Method

- At each surface in a model the amount of energy that is given off (Radiosity) is comprised of
  - the energy that the surface emits internally, plus
  - the amount of energy that is reflected off the surface



$$B_j = \rho_j H_j + E_j$$

B_j – Radiosity of surface j

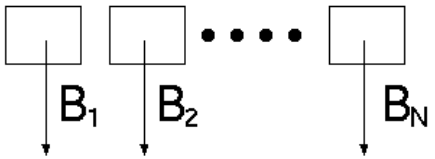
ρ_j – Reflectivity of surface j

H_j – Energy incident on surface j

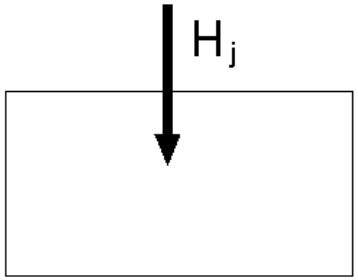
E_j – Energy emitted by surface j

The Radiosity Model

- The amount of incident light hitting the surface can be found by summing for all other surfaces the amount of energy that they contribute to this surface



$$H_j = \sum_{i=1}^{N} B_i F_{ij}, \quad j = 1..N$$



H_j – Energy incident on surface j

B_i – Radiosity of surface i

F_{ij} – Form factor ij

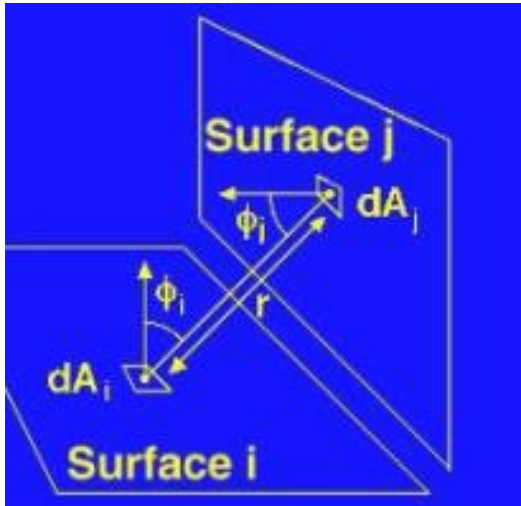
Form Factor (Fij)

- The fraction of energy that leaves surface I and lands on surface j
- Between differential areas, it is

$$F_{i,dA_j} = \frac{\cos\phi_i \cos\phi_j}{\pi r^2}$$

- The overall form factor between i and j is

$$F_{ij} = \sum_i \sum_j \frac{\cos\phi_i \cos\phi_j}{\pi r^2} dA_j$$



The Radiosity Matrix

- The radiosity equation now looks like this:

$$B_j = E_j + \rho_j \sum_{i=1}^{N} B_i F_{ij}, \quad j = 1..N \qquad (1)$$

- The derived radiosity equations form a set of N linear equations in N unknowns. This leads nicely to a matrix solution:

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_N F_{N1} & -\rho_N F_{N2} & \cdots & 1-\rho_N F_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}$$

Radiosity Steps

1 - Generate Model

2 - Compute Form Factors

3 - Solve Radiosity Matrix

4 – Render

Only if the geometry of the model is changed must the system start over from step 1.

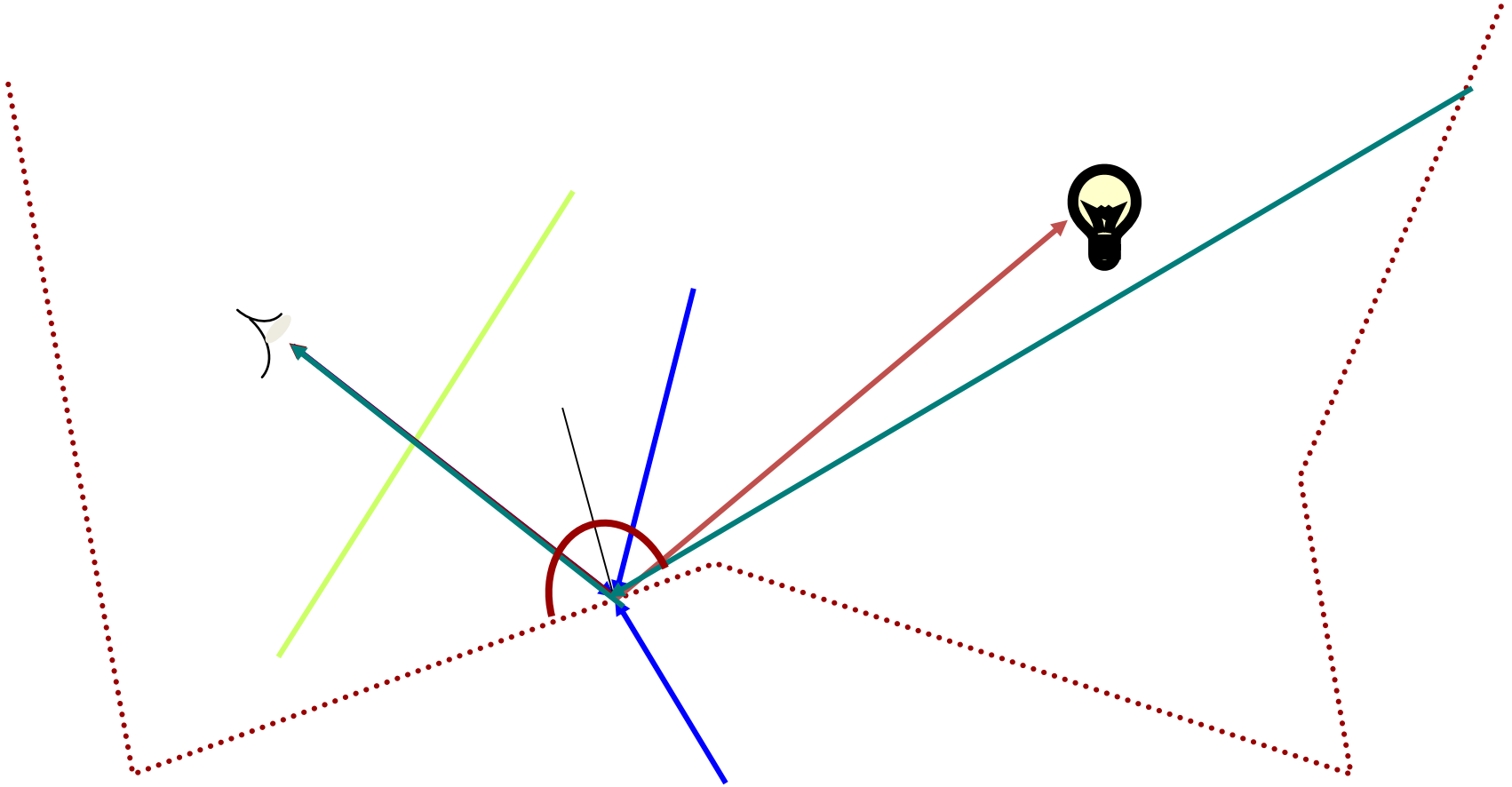
If the lighting or reflectance parameters of the scene are modified the system may start over from step 3.

If the view parameters are changed, the system must merely re-render the scene (step 4).

Radiosity Features

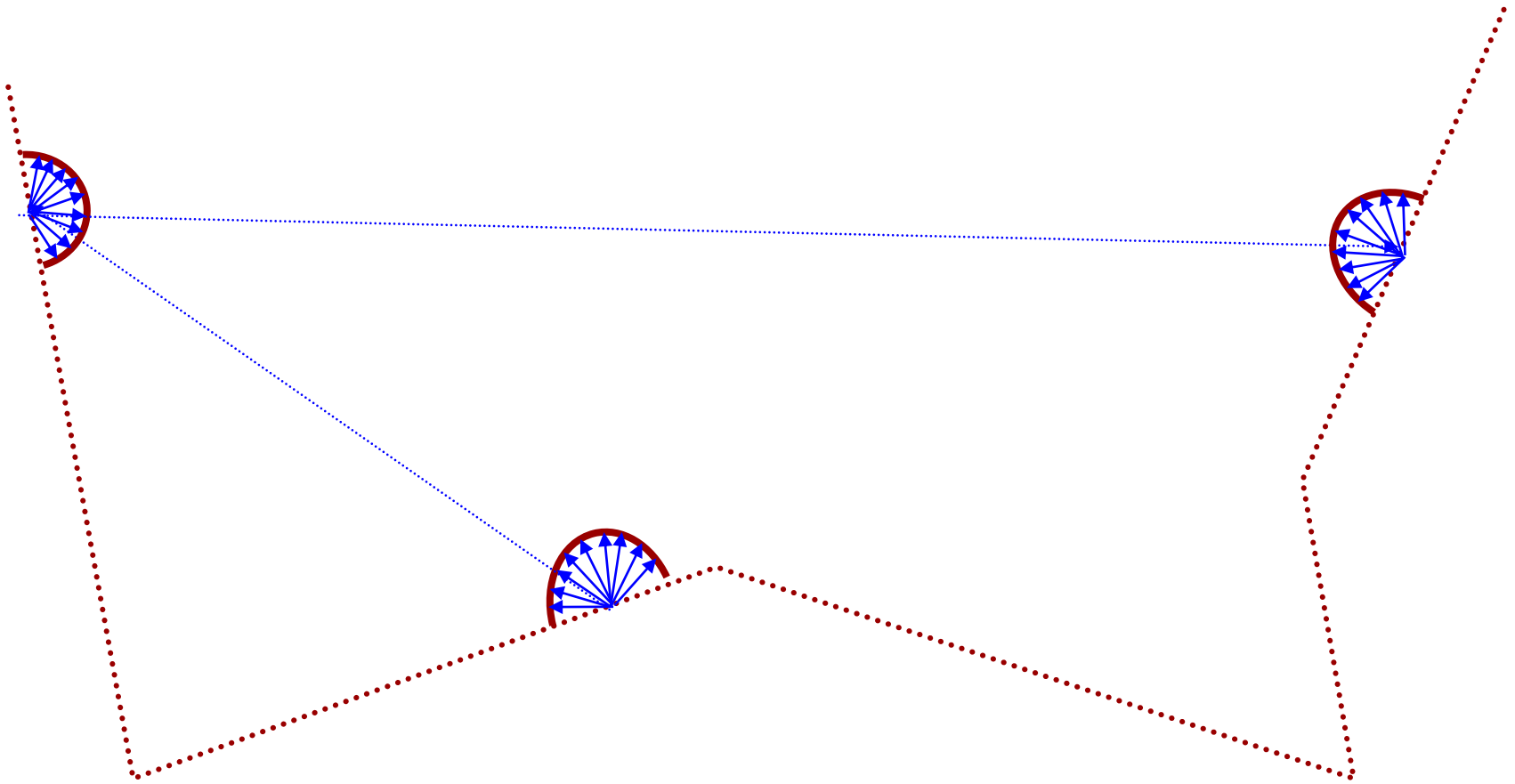
- The faces must be subdivided into small patches to reduce the artifacts
- The computational cost for calculating the form factors is expensive
  - Quadratic to the number of patches
- Solving for B_i is also very costly
- Cannot handle specular light

Tracing Rays



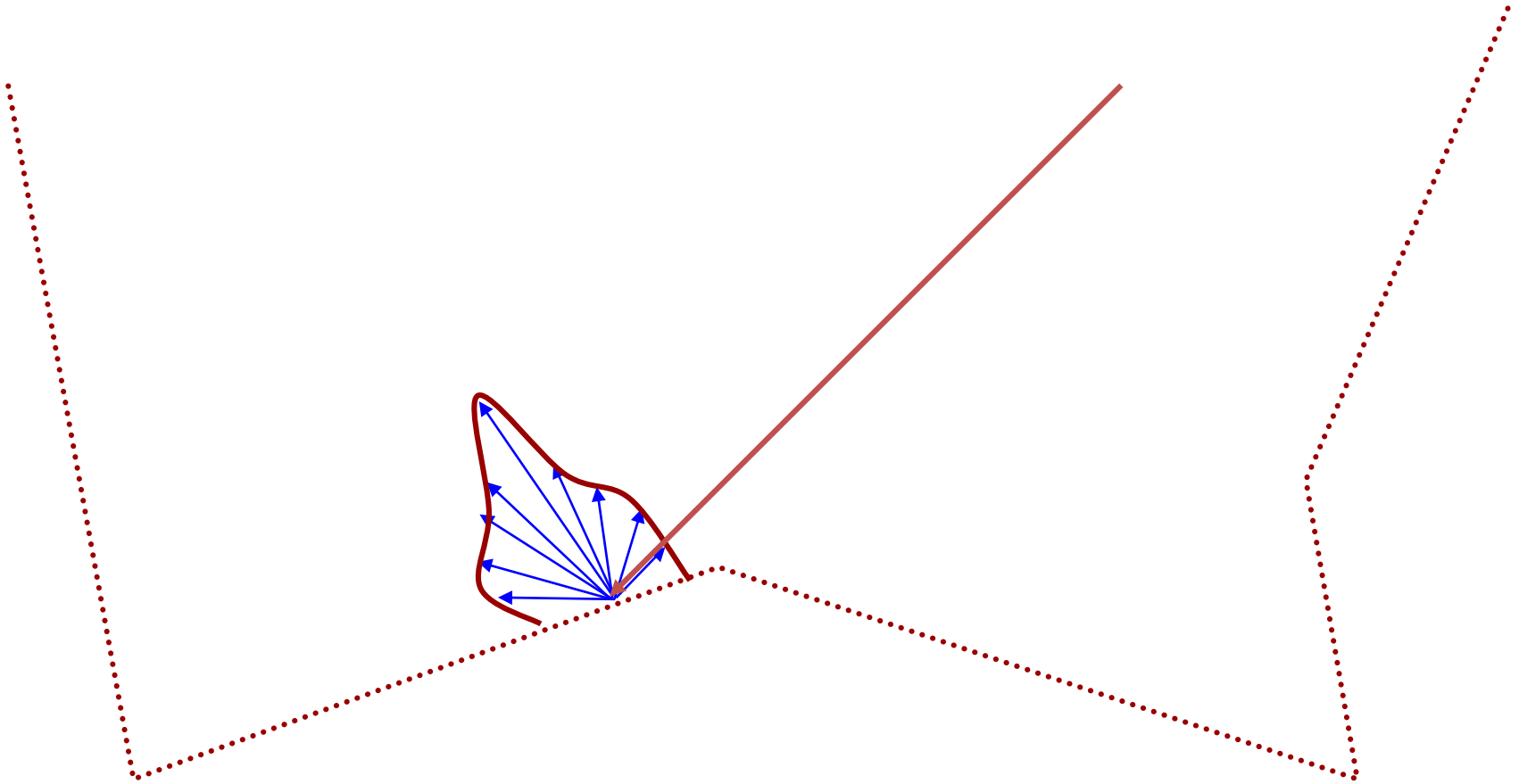
Trace many more rays

Tracing Rays



Not just in the ideal specular directions

Tracing Rays



Could distribute rays based on surface property

Global Illumination

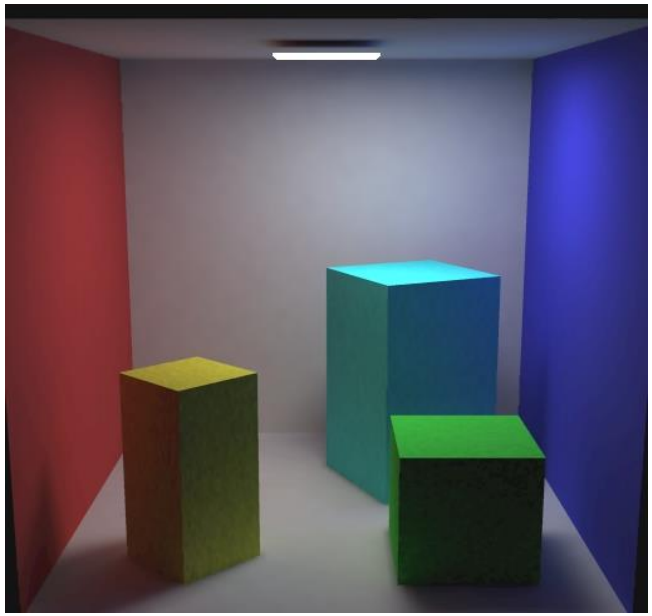
- What's wrong with factoring into Radiosity and Ray-tracing?
- Factor into direct illumination + indirect illumination
- Indirect illumination is the hard part

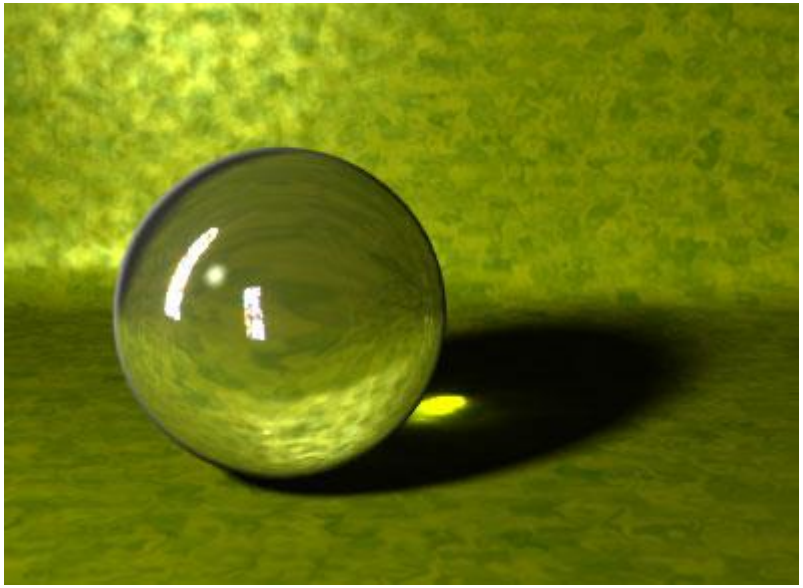
But:

- Indirect illumination has low variance
  - Can be sampled and interpolated (with care)

Global Illumination Methods

- Radiosity (classic)
- Photon Mapping (relatively new)





Photon Mapping

From http://graphics.ucsd.edu/~henrik/images/global.html

Photon Mapping Basics

- Enhancement to ray-tracing
- Can simulate caustics
  (focused light, like shimmering waves at the bottom of a swimming pool)
- Can simulate diffuse inter-reflections
  (e.g., the "bleeding" of colored light from a red wall onto a white floor, giving the floor a reddish tint)
- Can simulate clouds or smoke

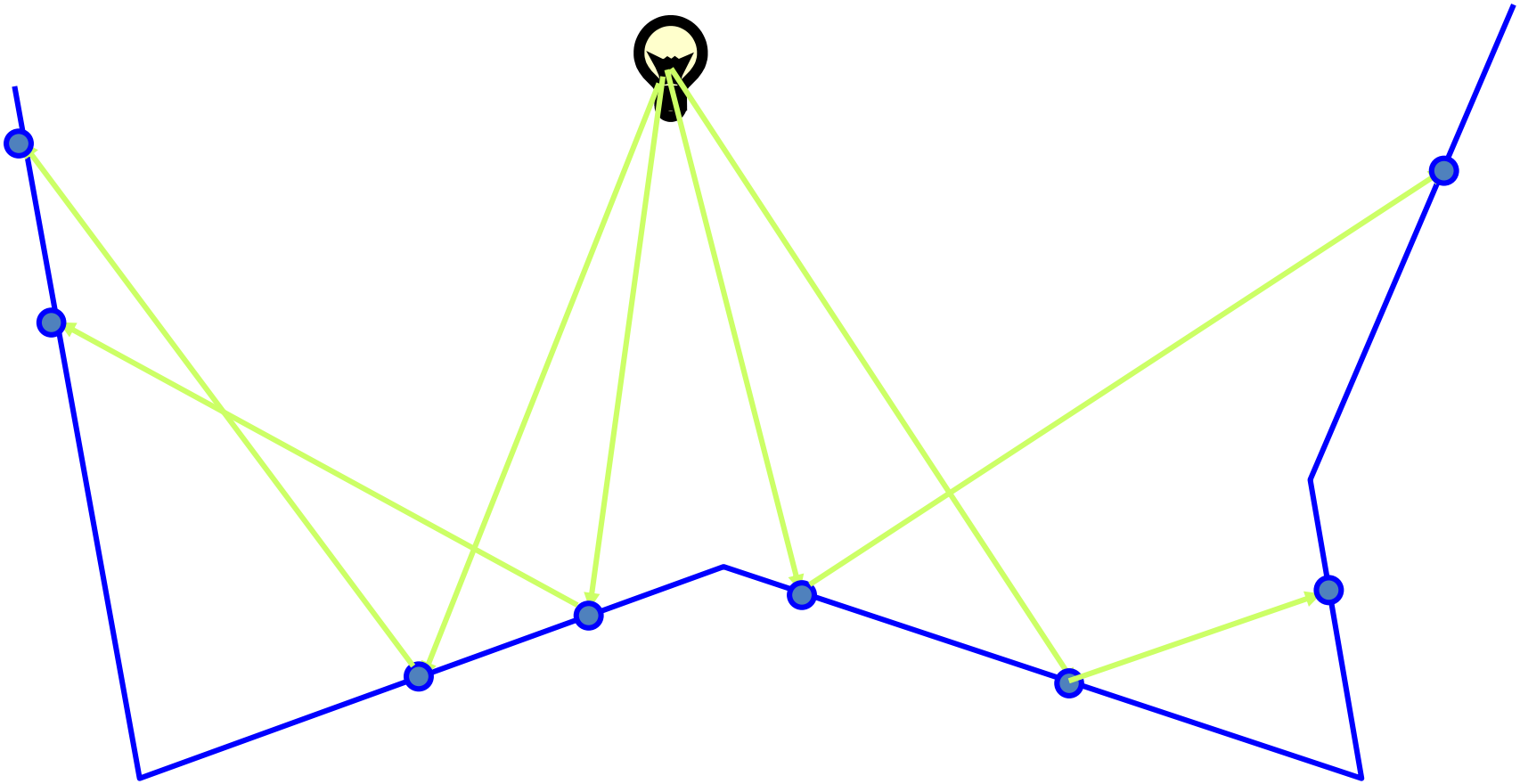
Photon Mapping

- A fast, global illumination algorithm based on Monte-Carlo method
- Casting photons from the light source, and saving the information of reflection when it hits a surface in the “photon map”, then render the results

Photon Mapping

- “Photons” are emitted (ray-traced) from light sources
- Photons either bounce or are absorbed
- Photons are stored in a *photon map,* with both position and incoming direction
- Photon map is decoupled from the geometry

Photon Mapping



[Jensen]

Two-pass Algorithm

- A two pass global illumination algorithm
  - First Pass - Photon Tracing
  - Second Pass – Rendering
- First Pass
  - Photons are generated and traced
  - They are associated with surface points and stored
  - Photon Map is created
  - View independent pass
- Second Pass
  - Photon map is a static and used to
    - Compute estimates of the incoming flux and the reflected radiance at any point in the scene
    - Look up indirect illumination from Photon map
    - Or, trace rays once and then look up illumination

Photon Tracing

- Indirect illumination on diffuse surfaces
- The process of emitting discrete photons from the light sources and tracing them through the scene
  - Store them at diffuse surfaces
- Brighter light => More Photons
  - Photons of uniform flux
  - Diffuse point light emits in uniformly distributed random directions
  - Directional light emits in its direction
  - Square light source emits from random positions on the square
    - with directions limited to a hemisphere

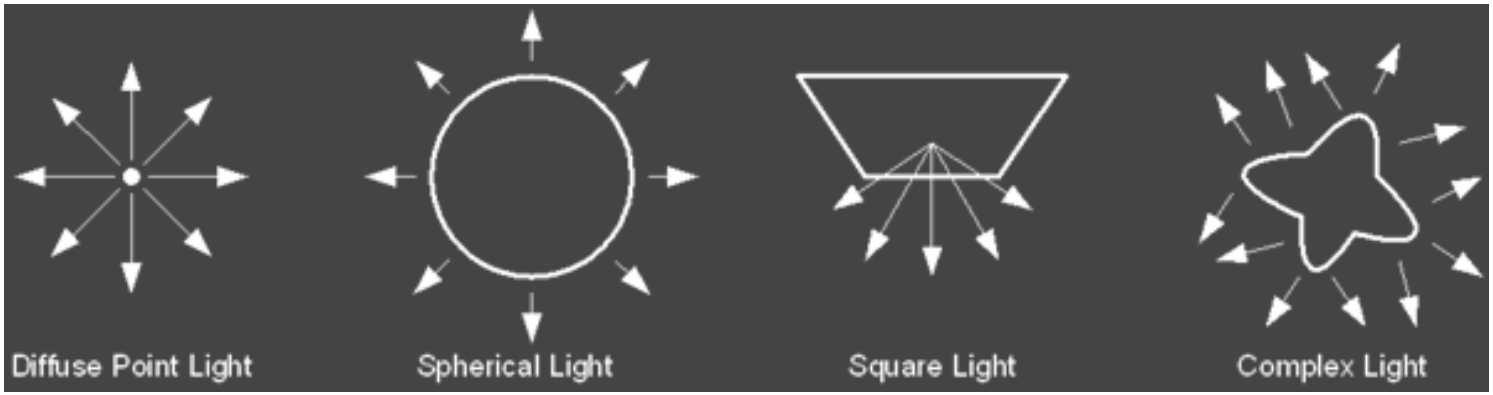
Photon Tracing

- The goal is to populate the photon maps that are used in the rendering pass to calculate the reflected radiance at surfaces

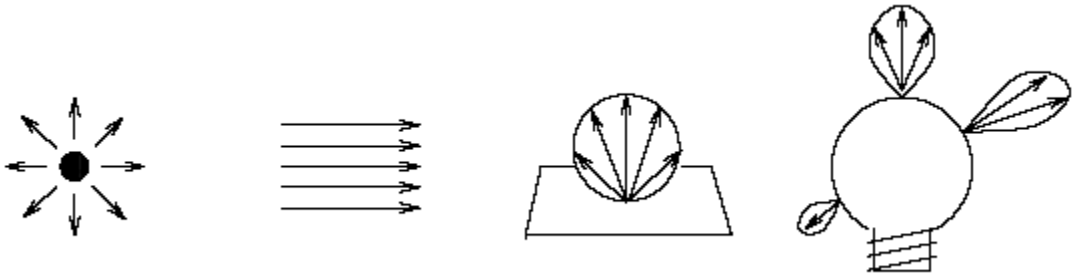
- Photon emission should follow light's emission properties
  - Controls origin and direction of each photon
  - Cosine distribution (more photon's emitted perpendicular to light)

Photon Emission

- A photon's life begins at the light source.
- For each light source in the scene we create a set of photons and divide the overall power of the light source among them.
- Brighter lights emit more photons



Photon Initialization

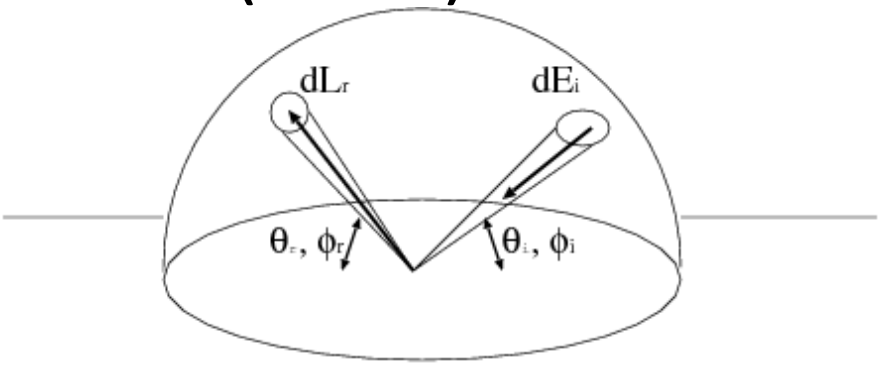


$$P_{photon} = P_{light} \; / \; n_{photon}$$

Bidirectional Reflectance Distribution Function (BRDF)

- The reflectance of an object can be represented by a function of the incident and reflected angles
- This function is called the Bidirectional Reflectance Distribution Function (BRDF)

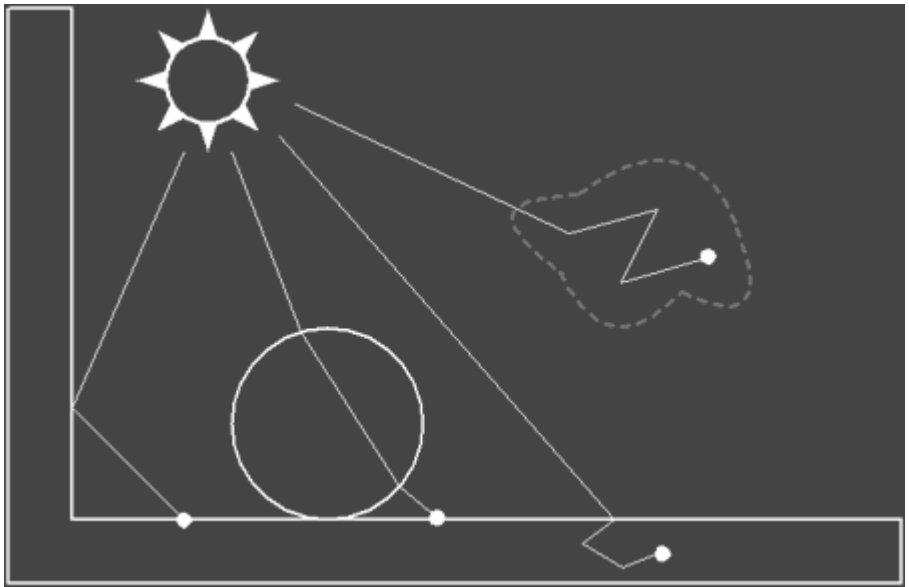
$$\rho(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{dL_r(\theta_r, \phi_r)}{dE_i(\theta_i, \phi_i)},$$



- where *E* is the incoming irradiance and *L* is the reflected radiance

Photon Scattering

- Emitted photons from light sources are scattered through a scene and are eventually absorbed or lost
- When a photon hits a surface we can decide how much of its energy is absorbed, reflected and refracted based on the surface's material properties



Photon Distribution

- Importance sampling
  - Multiple light sources
- Projection map
  - Project a bounding sphere for objects
  - Photons not wasted on background

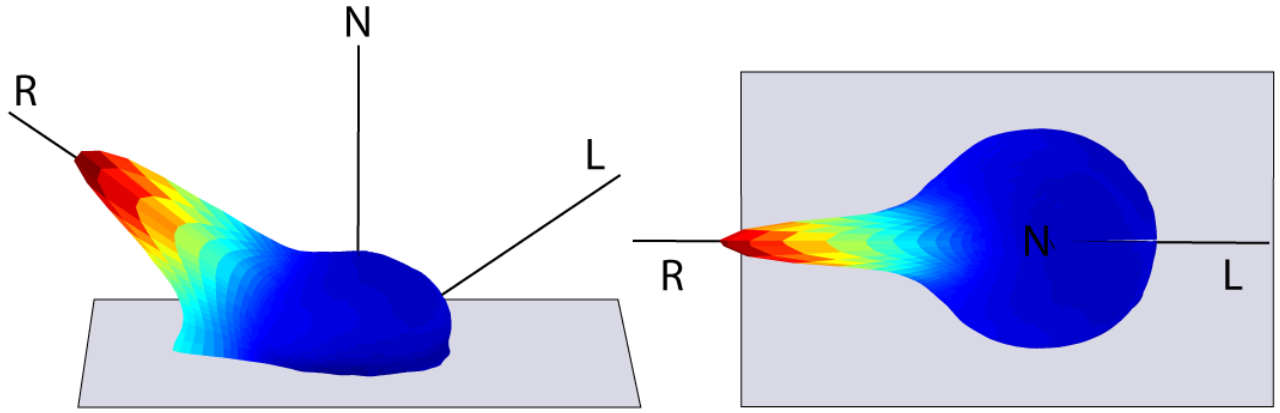
When the Photons Hits Surfaces

- Attenuate the power and reflect the photon
- Reflect with full power

- For arbitrary BRDFs
- Use Russian Roulette techniques
  - Decide whether the photon is reflected or not based on the probability

Arbitrary BRDF Reflection

- Can randomly calculate a direction and scale the power by the BRDF

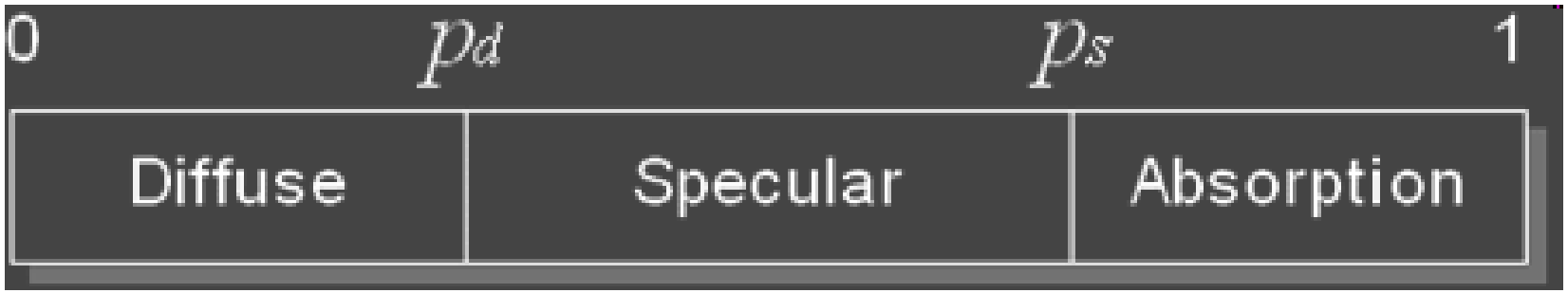


Photon Interactions

- On intersection, a photon can be
  - Reflected, transmitted, or absorbed
  - Probabilistic method based on the material properties
  - Russian roulette
    - Roll a dice and decide next step for the photon
    - Controls count not the power of the reflected/transmitted photon

Russian Roulette

- If the surface is diffusive+specular, a Monte Carlo technique called Russian Roulette is used to probabilistically decide whether photons are reflected, refracted, or absorbed.
- Produce a random number between 0 and 1
- Determine whether to transmit, absorb, or reflect in a specular or diffusive manner, according to the value



Monochromatic Photon Behavior

- Consider monochromatic simulation
- Reflective surface with
  - Diffuse reflection coefficient d
  - Specular reflection coefficient s (with d + s <= 1)
- Uniformly distributed random variable $X \in [0, 1]$
- Sample X. If
  - $X \in [0, d]$ => diffuse reflection
  - $X \in [d, s + d]$ => specular reflection
  - $X \in [s + d, 1]$ => absorption

“Chromatic” Photons

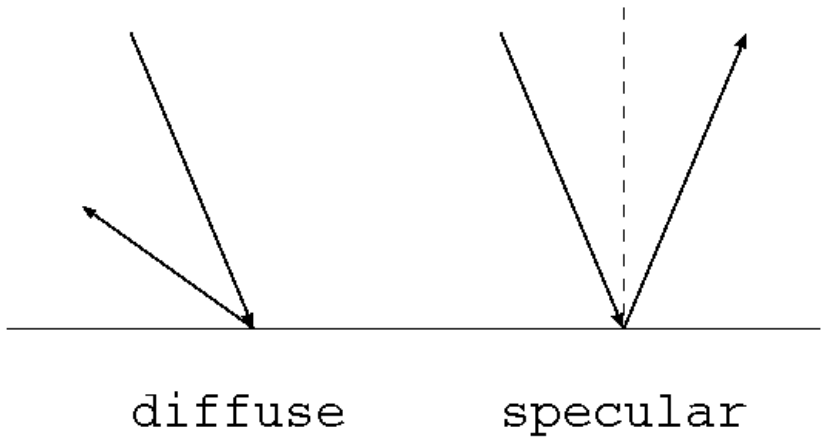
- Probability P_d for diffuse reflection =
  - $MAX(d_r P_r, d_g P_g, d_b P_b) / MAX(P_r, P_g, P_b)$
  - where (d_r, d_g, d_b) are the diffuse reflection coefficients
  - (P_r, P_g, P_b) are the powers of the incident photons
- Probability P_s for specular reflection is
  - $MAX(s_r P_r, s_g P_g, s_b P_b) / MAX(P_r, P_g, P_b)$
  - where (s_r, s_g, s_b) are the specular reflection coefficients
- Instead of MAX, probabilities can also be based on total power

Chromatic Photon Behavior

- Sample X. If
  - $X \in [0, P_d]$ => diffuse reflection
  - $X \in [P_d, P_s + P_d]$ => specular reflection
  - $X \in [P_s + P_d, 1]$ => absorption
- The power of the reflected photon must change
- If specular reflection was chosen:
  - $P_{refl,r} = P_{inc,r}\, s_r/P_s$
  - $P_{refl,g} = P_{inc,g}\, s_g/P_s$
  - $P_{refl,b} = P_{inc,b}\, s_b/P_s$
    - where P_{inc} is the power of the incident photon
- Similarly for diffused and absorbed photons

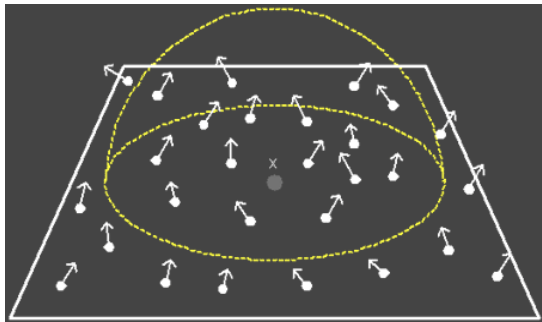
Diffuse and Specular Reflection

- If the photon is to make a diffuse reflection, randomly determine the direction
- If the photon is to make a specular reflection, reflect in the mirror direction



Photon Map

- When a photon makes a diffuse bounce, the ray intersection is stored in memory
  - 3D coordinate on the surface
  - Color intensity
  - Incident direction
- The data structure of all the photons is called Photon Map
- The photon data is not recorded for specular reflections

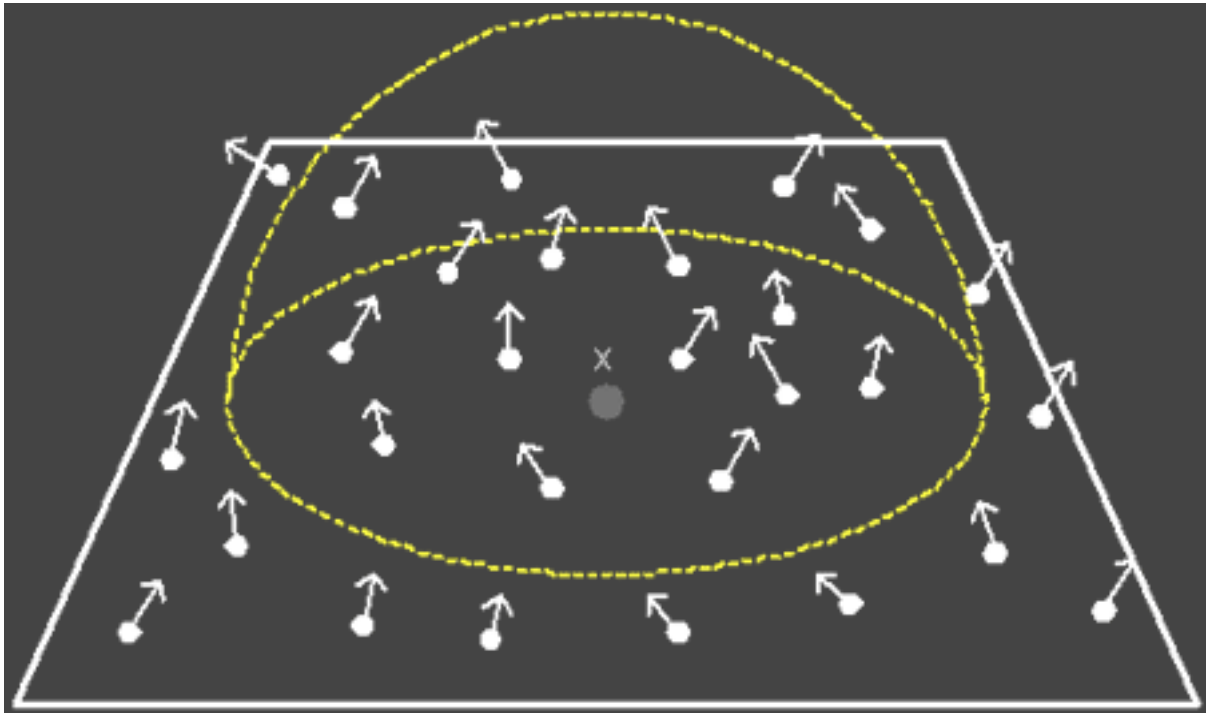


Photon Mapping

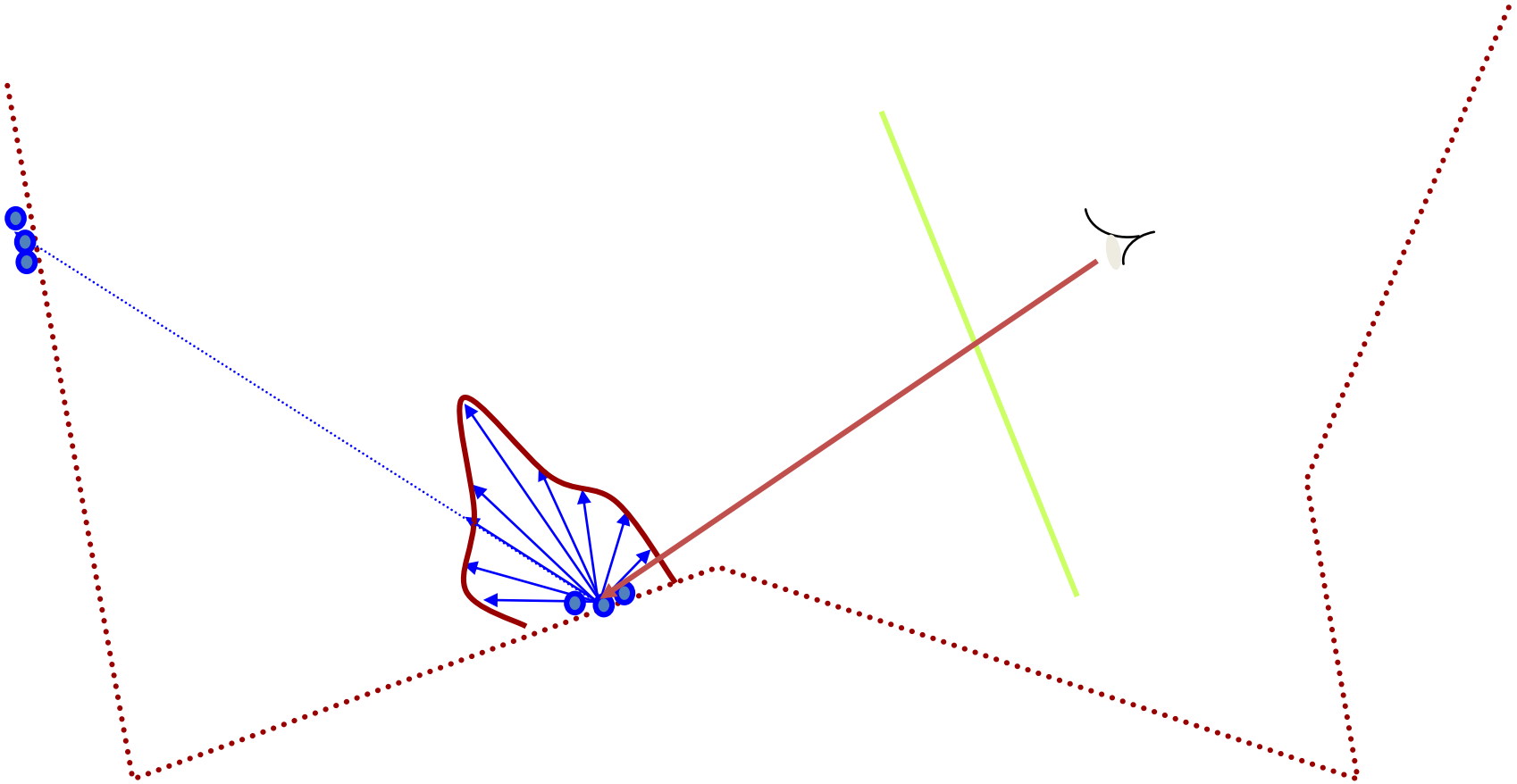
- The ray-tracing step uses the closest N photons to each ray intersection and estimates the outgoing radiance
- Specular can be done using “usual” ray-tracing to reduce the number of photons needed
- Numerous extensions to the idea to add more complex effects

Rendering – Second Pass

- Finally, a traditional ray-tracing procedure is performed by shooting rays from the camera
- At the location the ray hits the scene, a sphere is created and enlarged until it includes N photons



Rendering



Perform a single step of secondary illumination

Radiance Estimation

- Find a sphere around x containing n photons
- Use these n photons to estimate the radiance
- The radiance estimate can be written by the following equation

$$L_r(x,\vec{\omega}) = \sum_{p=1}^{N} f_r(x,\overrightarrow{\omega_p},\vec{\omega}) \frac{\Delta\Phi_p(x,\overrightarrow{\omega_p})}{\Delta A}$$

x: location the ray hits the scene

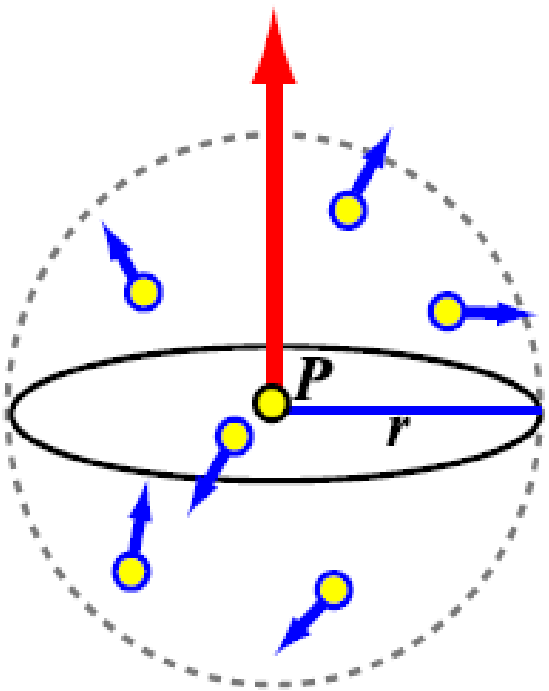
$\vec{\omega}$: direction towards the camera

$\overrightarrow{\omega_p}$: incident vector of photon p

f_r: BRDF

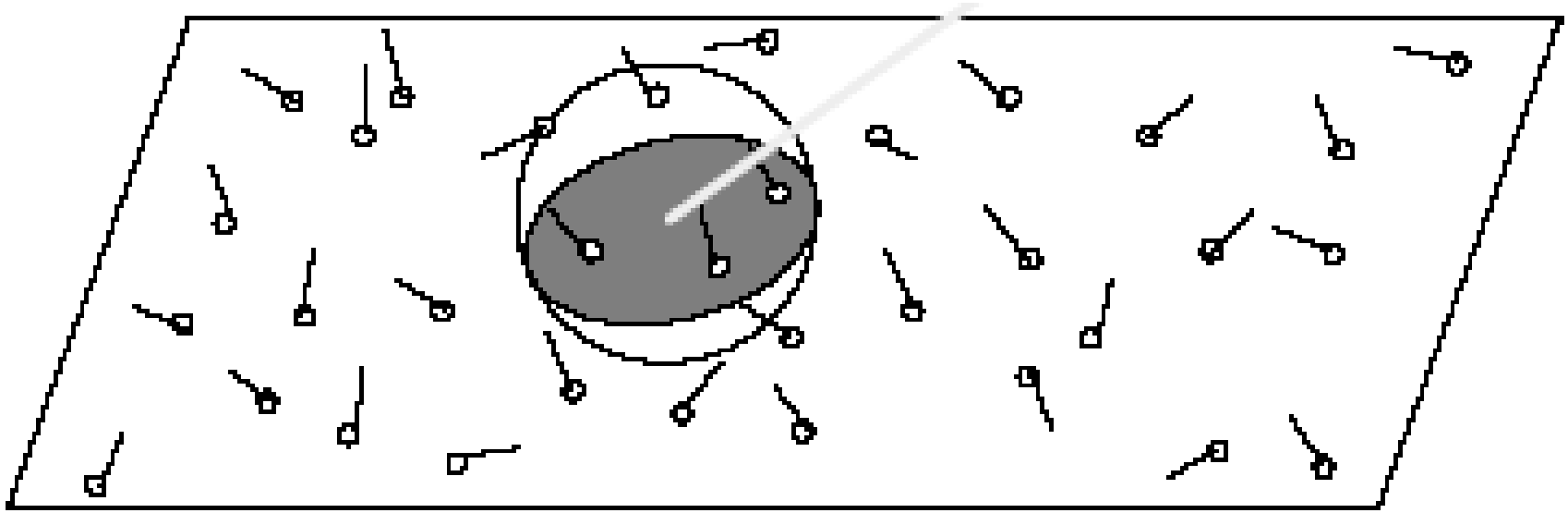
$\Delta\Phi_p$: power of photon p

ΔA: Area of the circle πr^2



Photon p has power $\Delta\Phi_p$

Radiance Estimate



$$L_r(\mathbf{x},\theta_o,\phi_o) = \frac{1}{\pi r^2}\sum_{p=1}^{n}\rho_{\mathbf{x}}(\theta_o,\phi_o,\theta,\phi)\Delta\Phi_p(x,\theta,\phi)$$

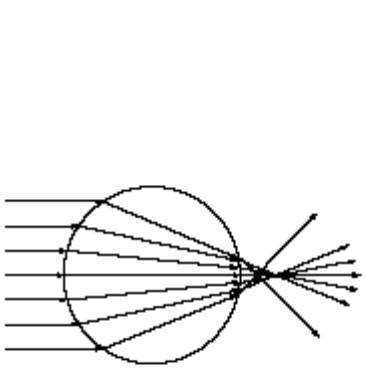
Estimate Filtering

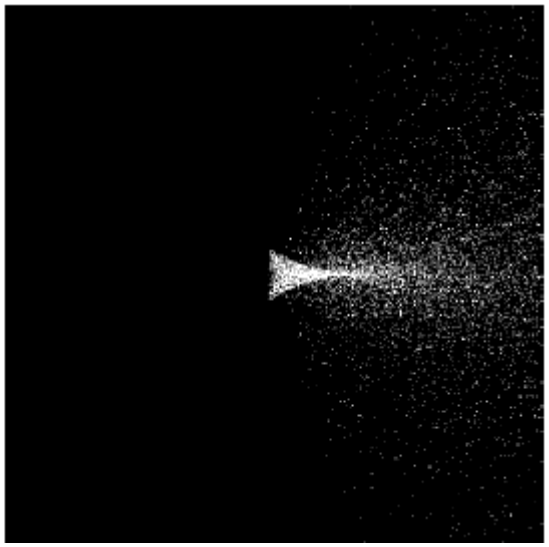
- Averaging samples creates blur
  - Good for radiosity, they are smooth
  - Not so good for caustics, they have sharp boundaries
- Use weighted averaging
- Use differential filters
  - Detect edges from the samples
  - “Zero” weights for samples outside edges

Participative Media

- Photons can be emitted from volumes (and surfaces and points)
- Probability of being *scattered* or *absorped* in the medium depends on
  - Density of the medium
  - Distance the photon travels through the medium
    - The denser the medium, the shorter the average distance before a photon interaction happens.
- Photons are stored at the positions where a scattering event happens
  - Exception photons directly from the light source
  - Direct illumination is evaluated using ray tracing
  - Not necessary, but this separation improves efficiency

Participative Medium





Glass sphere in fog illuminated by directional light

Images courtesy of Henrik Wann Jensen

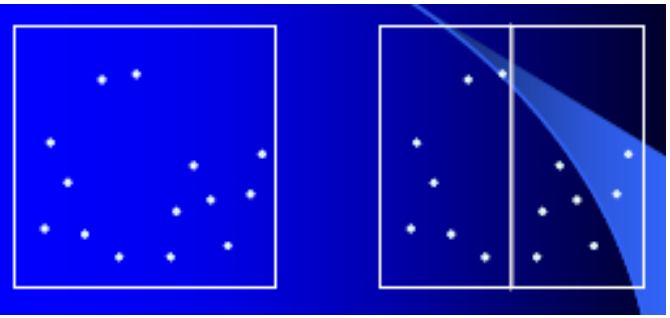
Photon Map

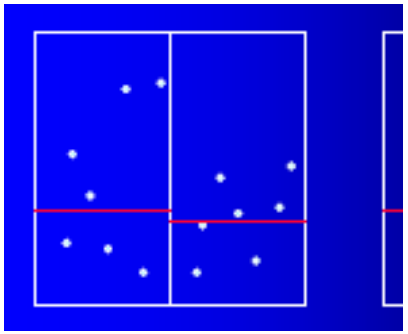
- Photons are only stored where they hit non-specular surfaces
- Global data structure: photon map stores
  - Position
  - Incoming photon power
  - Incident direction
  - A flag for help with sorting and look-up
- For participative media, one might construct a volume photon map

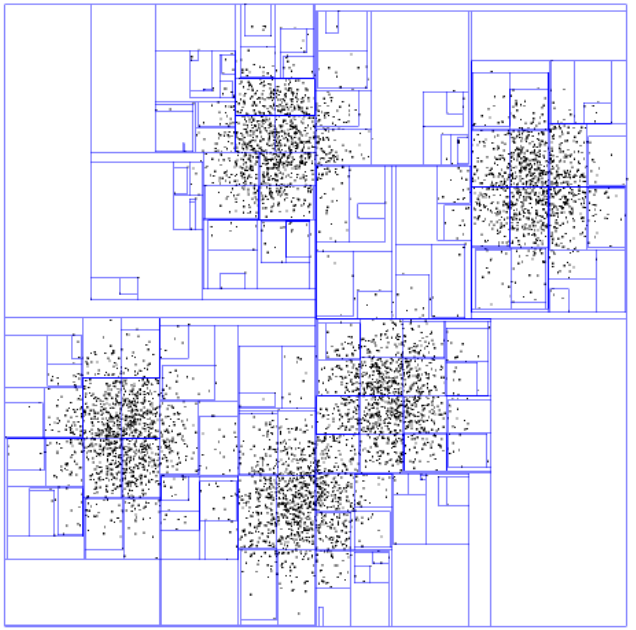
KD-Tree Implementation for Rendering Pass

- To find radiance at a point, seek the nearest photons
- The photon maps are classified and saved in a KD-tree
- KD-tree :
  - dividing the samples at the median
  - The median sample becomes the parent node, and the larger data set form a right child tree, the smaller data set f
  - Further subdivide the children trees

- Can efficiently find the neighbors when



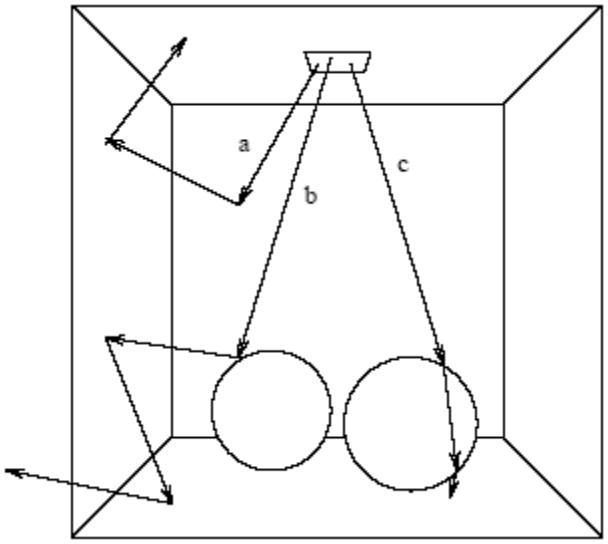




Three Photon Maps

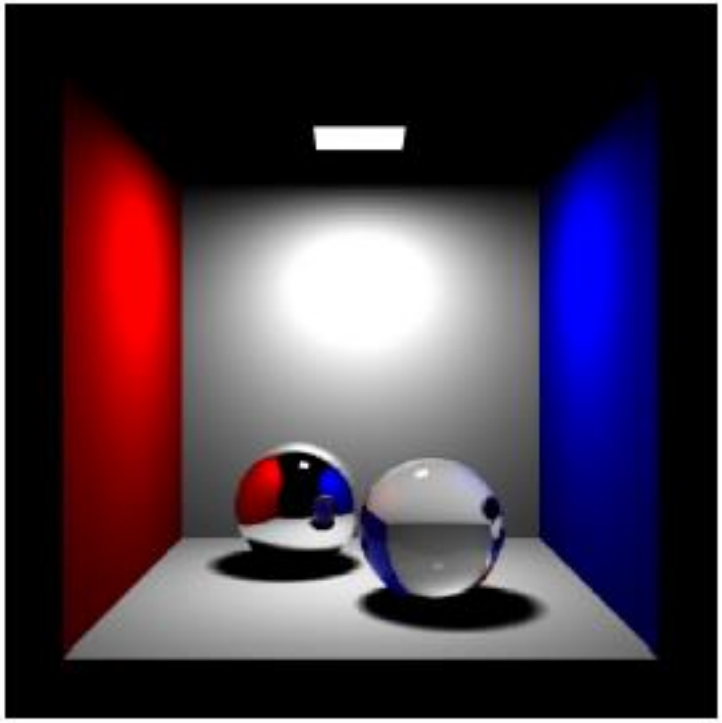
- Caustic photon map
  - Photons that have been through at least one specular reflection before hitting a diffuse surface: LS+D.
- Global photon map
  - Approximate representation of the global illumination solution
  - Scene for all diffuse surfaces
- Volume photon map
  - Indirect illumination of a participating medium

Photon Paths in Cornell Box

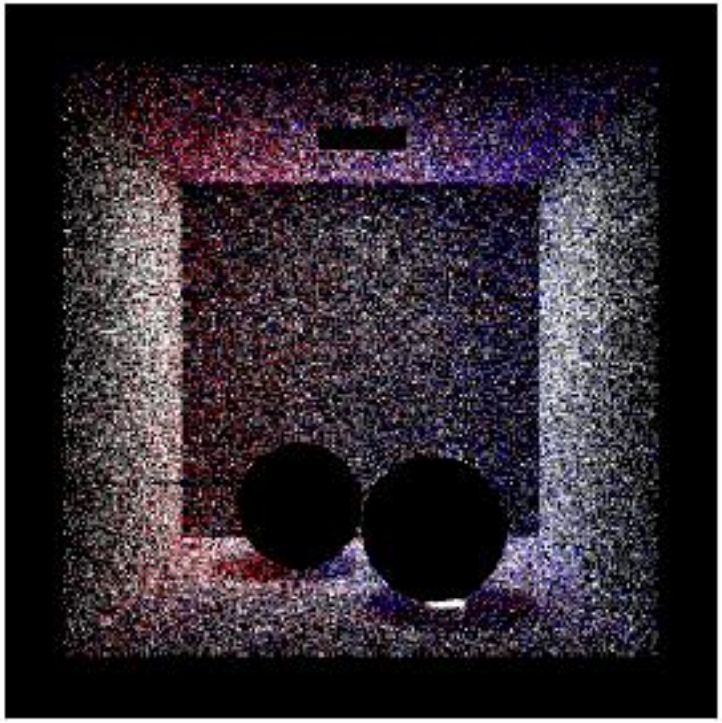


Chrome sphere (left) and Glass sphere (right)
(a) Two diffuse reflections followed by absorption
(b) Specular reflection followed by two diffuse reflections
(c) Two specular transmissions followed by absorption

Example



Ray-traced image (direct illumination, specular reflection and transmission)



Photons in the photon map.

Images courtesy of Henrik Wann Jensen

Photon Mapping - Pros

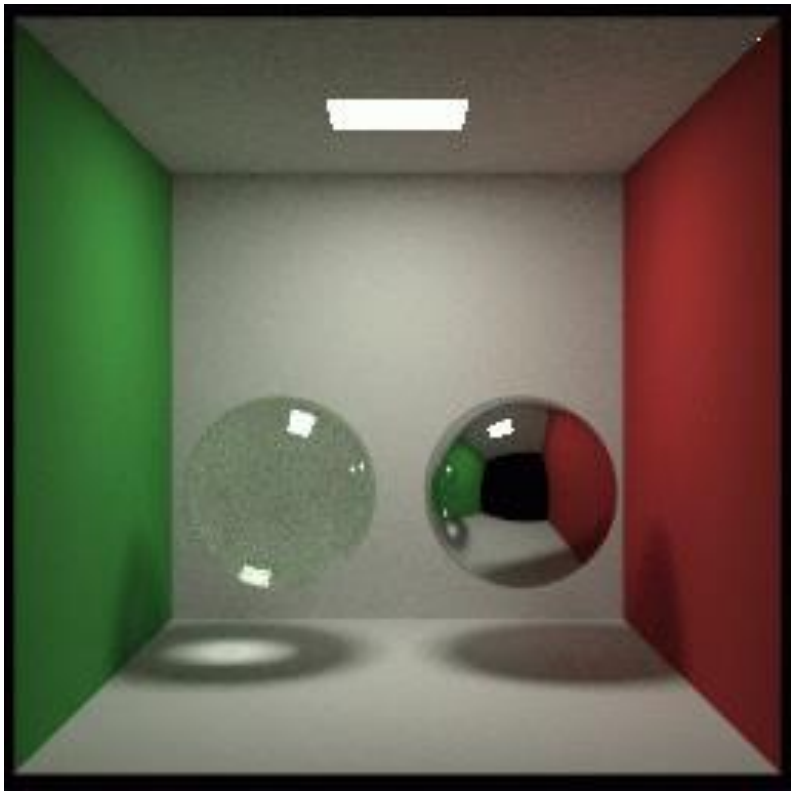
- Preprocessing step is view independent, so only needs to be re-done if the lighting or positions of objects change
- Inter-object interaction includes:
  - Shadows
  - Indirect lighting
  - Color bleeding
  - Highlights and reflections
  - Caustics – *current method of choice*
- Works for procedurally defined surfaces

Photon Mapping - Cons

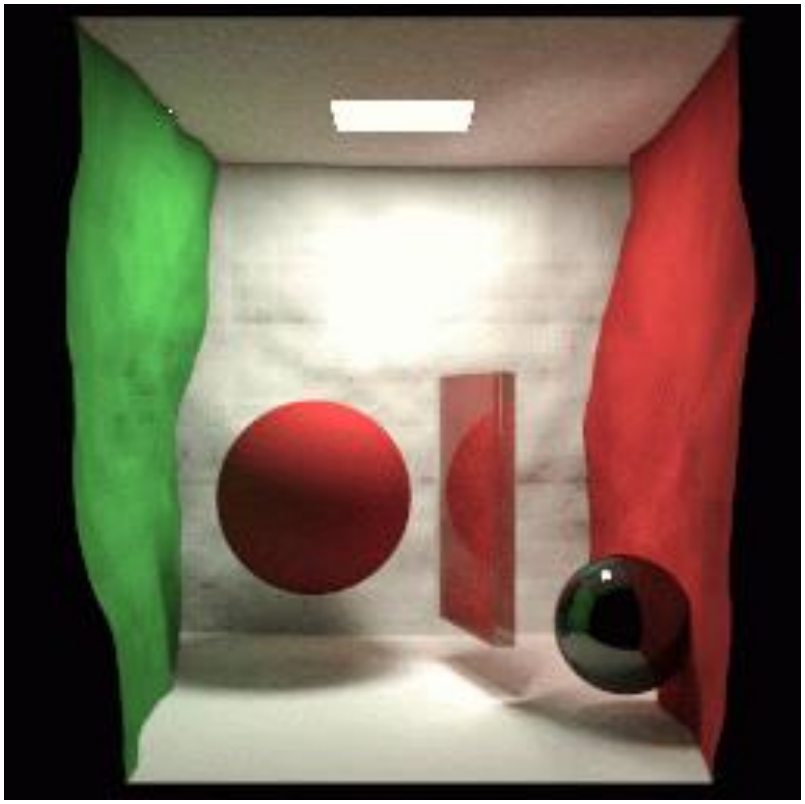
- Still time-consuming, although not as bad as comparable results from pure raytracing
- Photon map not easy to update if small changes are made to the scene

Precision

- The precision of the final results depends on
  - the number of photons emitted
  - the number of photons counted for calculating the radiance



Simple Cornell box scene with 343 samples
per pixel and 10 million photons.

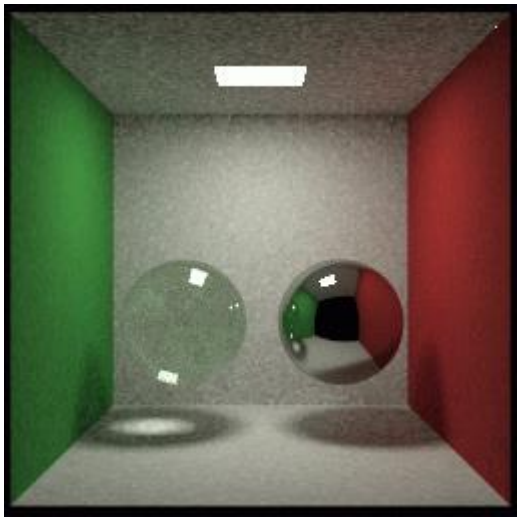


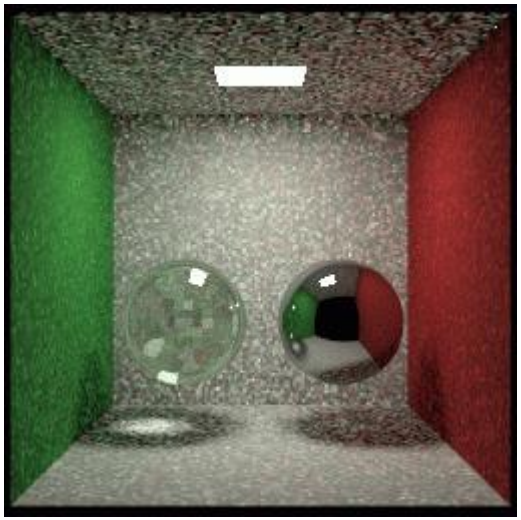
Cornell box on acid with 343 samples per
pixel and 10 million photons.

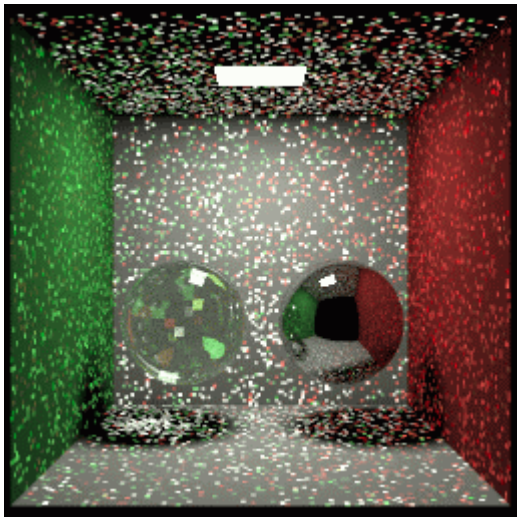
1,000,000 photons

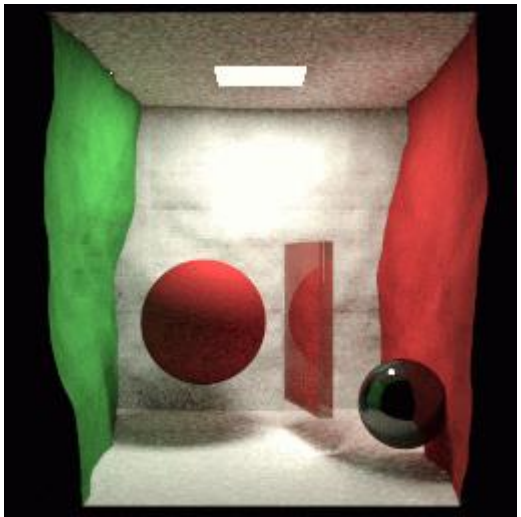
100,000 photons

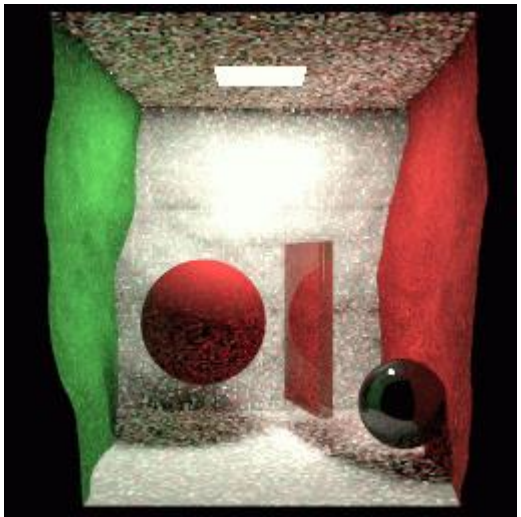
10,000 photons

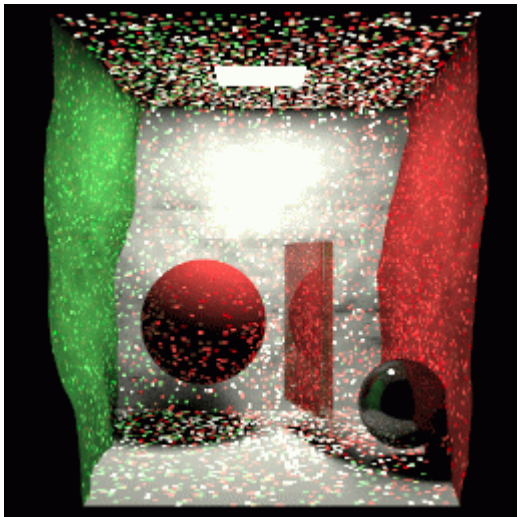






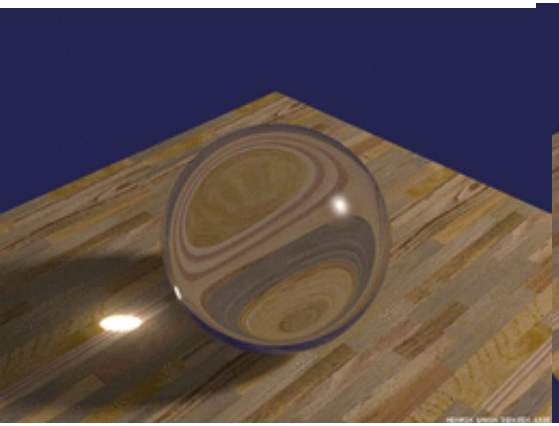
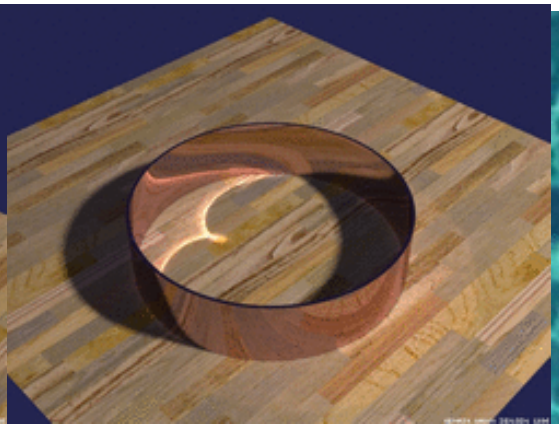






Photon Mapping Features

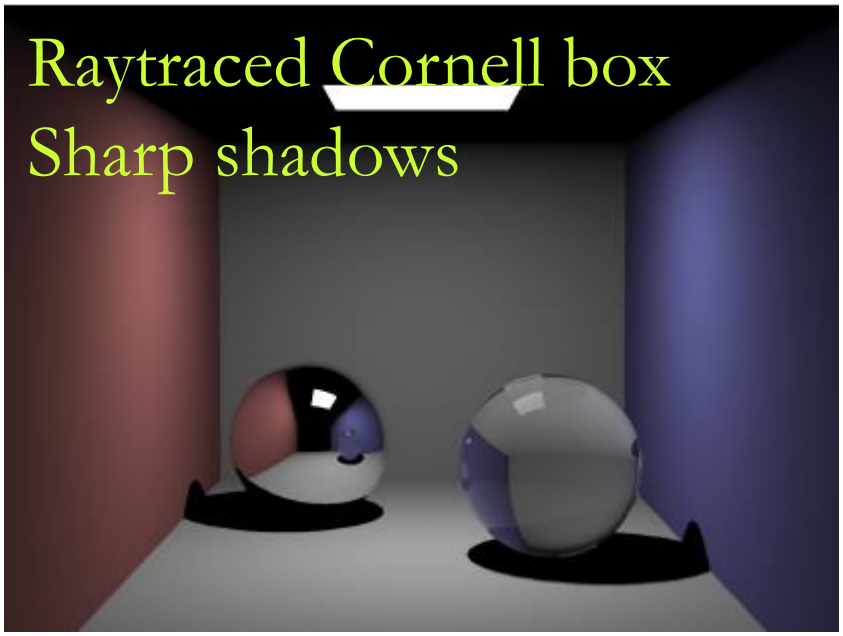
- Can render caustics
  - Ray tracing cannot render caustics
- Computationally efficient
  - Much more efficient than path-tracing


Photon Mapping is Efficient!

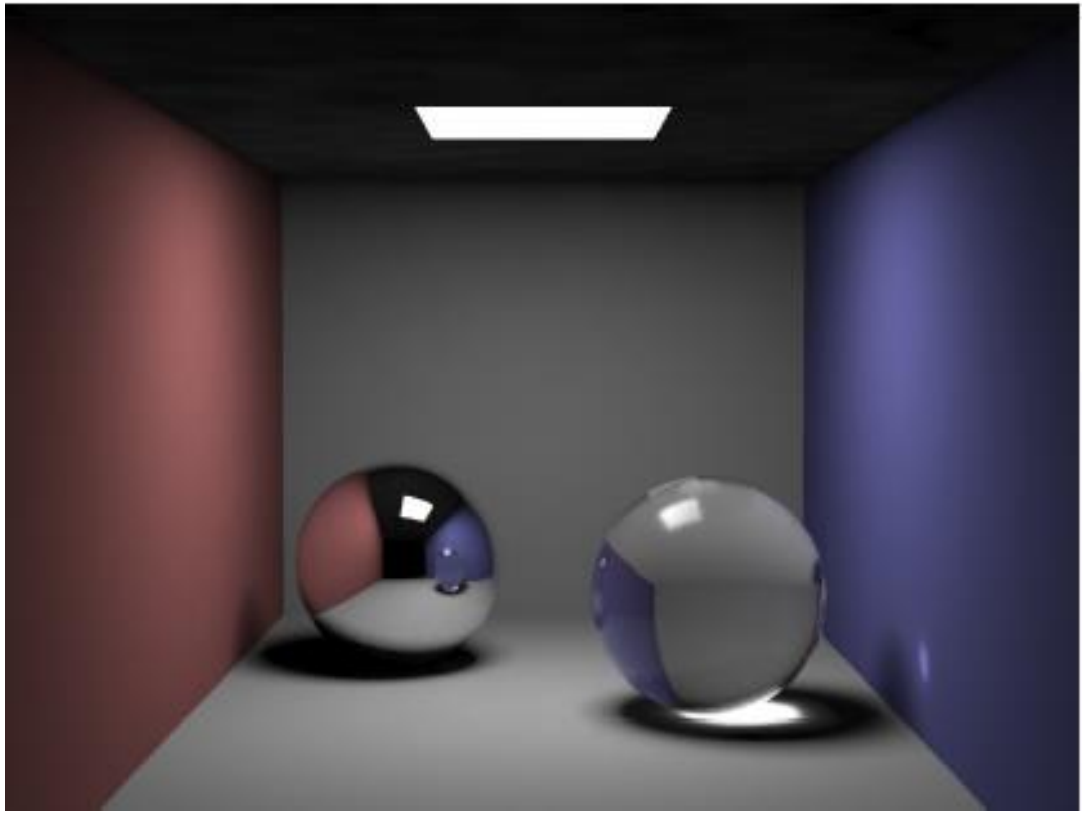
- It is a stochastic approach that estimates the radiance from a few number of samples
- Kernel density estimation: operating directly on the samples, instead of creating a histogram of samples associated with the geometry

Examples



Images courtesy of Henrik Wann Jensen

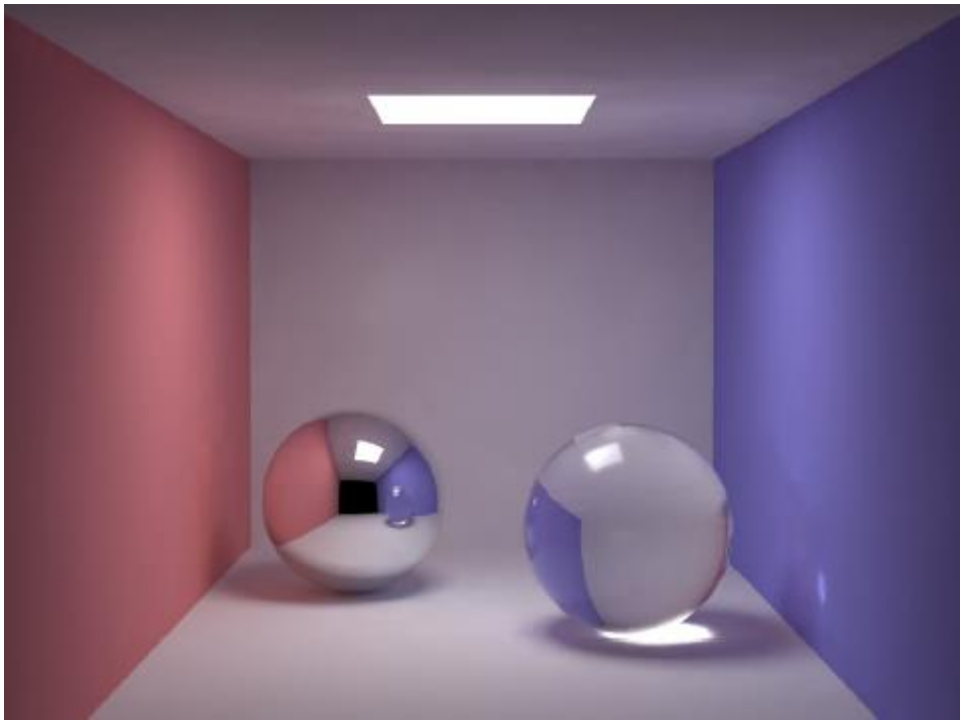
Caustic Photon Map



Photon map has 50000 photons and the estimate uses 60 photons

Images courtesy of Henrik Wann Jensen

Photon Map

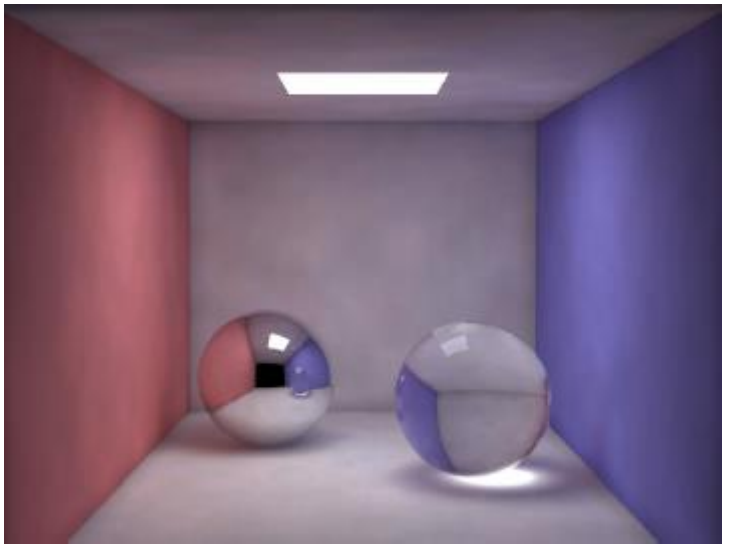


200000 photons in the global photon map. 100 photons used in the estimate.

Images courtesy of Henrik Wann Jensen

Photon Neighborhood

Sphere Estimate



Disk Estimate

Global photon map radiance estimates visualized directly using 500 photons

Images courtesy of Henrik Wann Jensen

Cornell Box with Water

Displacement-mapped water surface (20,000 tris). 500,000 photons in both the caustics and the global photon maps, and 100 photons in the radiance estimate.

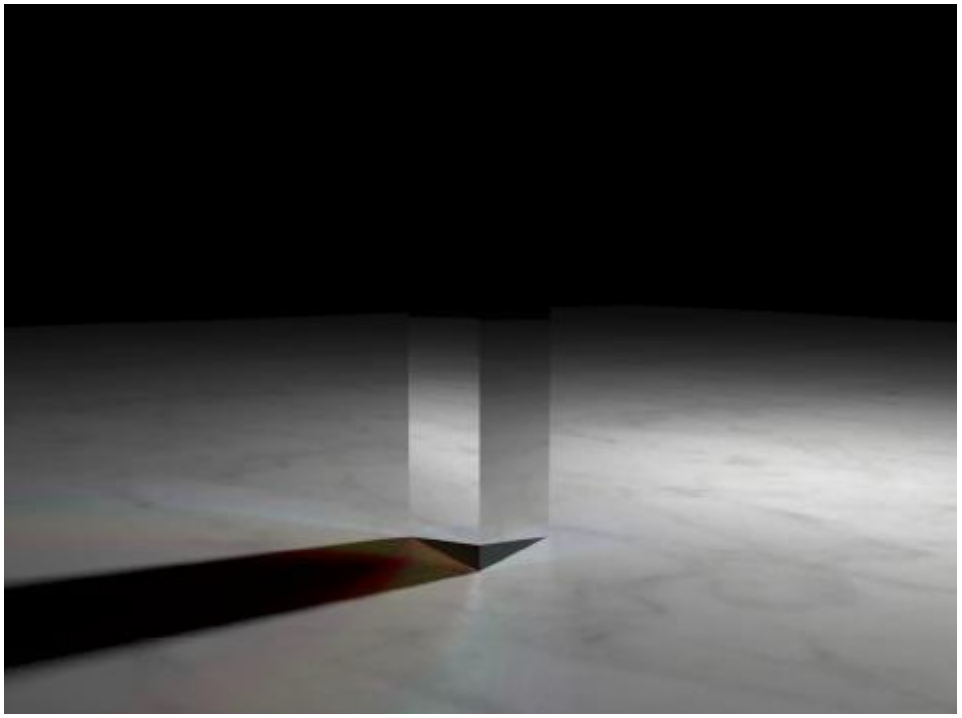
Images courtesy of Henrik Wann Jensen

Cognac Glass and Caustic

Glass made with 12000 triangles. 200,000 photons in the map. The radiance estimates with 40 photons.

Images courtesy of Henrik Wann Jensen

Caustic through Prism Dispersion



Only three separate wavelengths have been sampled. 500,000 photons in the caustics and 80 photons in the radiance estimate.

Images courtesy of Henrik Wann Jensen

Diana, the Huntress

Light behind translucent marble. (200,000 photons)

Summary

- Local and Global Illuminations
- Radiosity
- Photon Mapping
- Further readings:
  - Realistic Image Synthesis Using Photon Mapping by Henrik Wann Jensen
  - Global Illumination using Photon Maps (EGRW ‘96) Henrik Wann Jensen