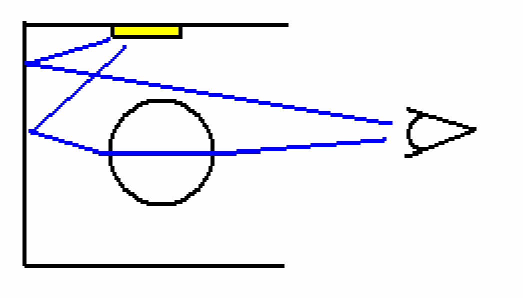
CS667	Photon Mapping	27 April 2006

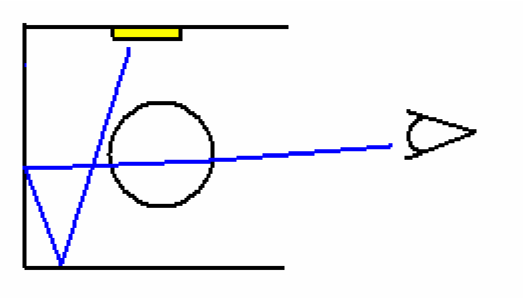
Victor Kwok **Lecturer: Steve Marschner**

Review of previous rendering methods:



a) Ray Tracing

Handles LDS*E light paths



b) Path Tracing

Handles LD*S*E light paths
However, it performs poorly for LS+DS*E paths, i.e. "caustic illumination paths".

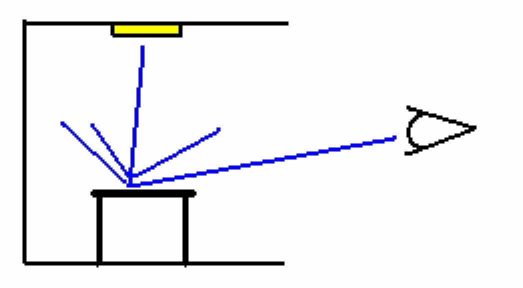
There are 2 classes of methods that are able to handle many of these difficult light paths:

Bidirectional – Trace rays from the eye AND from the light source. One way is good at finding caustics, while the other is good at dealing with transmission paths.

2 pass – Instead of tracing rays from light and eye together, we run sets of paths from the light, and then run sets of paths from the eye. We will focus on this method.

Background information for the 2 pass method

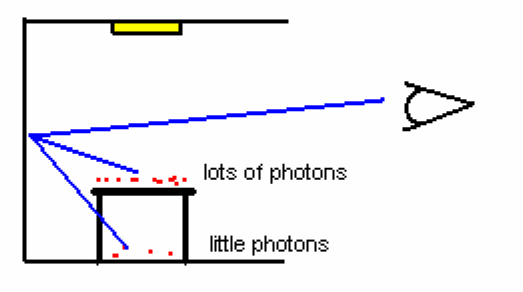
The original 2 pass method was used by radiosity algorithms:



1. Compute radiosity in scene
2. Ray trace, computing illumination from results of 1. This is known as the "final gather" method.

Handles LD*S*E paths.
We get sharp shadows.

2 pass method in photon mapping – Instead of radiosity, we store a large set of particles.



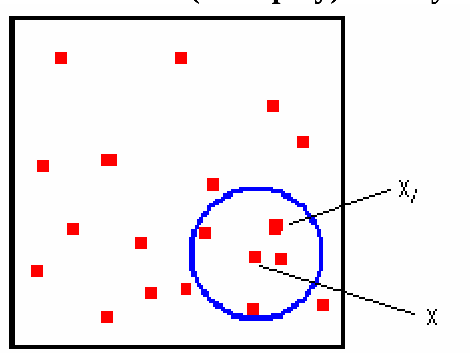
The counting of photons at a surface is called Density Estimation. Density Estimation is used in many probability and statistics problems. It consists of constructing an estimate based on data gathered from a probability density function that is by itself hard to solve or unsolvable.
In other words, we produce a coarse estimate $\hat{p}$ (x) of the distribution p(x):

Distribution p(x) $\rightarrow$ samples x_i } density estimation $\rightarrow$ $\hat{p}$ (x)

An example of this is a histogram.

There are two ways of doing Density Estimation: Parametric, and non-parametric. We will focus on the non-parametric method, known as:

Kernal Based (local poly) density estimation



$\hat{p}(x) = \sum_{i=1}^{n}[h(|x - x_i|)]$, where h is the weighting function, $\int h(x) = 1$

However, instead of using a fixed kernel size and counting number of points within it, we keep the number of points fixed and vary the kernel size, e.g. find kernel size such that we get 6 neighbors. In other words, we keep increasing the size of the kernel until the condition is satisfied.

In photon mapping, we apply this Kernal based method when counting the number of photons in a given area to determine how much irradiance is at that point. Each photon is modeled as a point in the density estimation method.

(Photos from lecture slideshow)
We use photon map for secondary rays only, in order to get rid of "blotchy" artifacts. In other words, we apply direct light first before using photon mapping.
We can see that photon mapping creates more noise at the corners of an enclosed scene.
(end photos)

Summary:

1st pass (from light)

Algorithm for photon tracing (similar to path tracing):

1. Generate rays from light sources in the scene. We find the total power in the scene and distribute it equally to the photons.
2. Find ray intersection.
3. Deposit photon.
4. Do a Russian roulette algorithm to determine whether to continue or not.
5. Generate a new direction, and recurse.

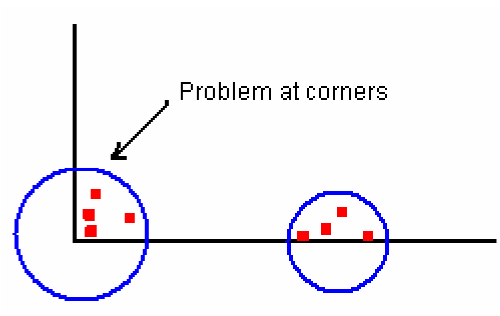
Photons are just a structure of 3 points storing its xyz location in the scene.
After the 1st pass, we now have photons scattered around the scene proportionally to the irradiance.

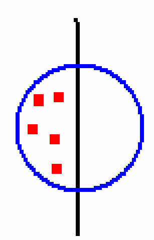
2nd pass (from eye)

Lookup algorithm:

1. Locate n nearest photons by using a kernel of radius r_{max}.
2. Compute the "size" of the region. Area = $\pi * (r_{max})^2$

Problems:





Here, it is darker than it should be, because there are no photons behind the other side of the wall.