Photon Mapping Assignment

15-864 Advanced Computer Graphics, Carnegie Mellon University
Instructor: Doug L. James
TA: Christopher Twigg

Introduction

In this assignment you will implement (portions of) a photon mapping renderer. For simplicity, we will only consider scenes with a single area light source, and *assume surfaces are diffuse, or purely specular* (e.g., mirror or glass). To generate images for testing and grading, a test scene will be provided for you on the class website; this will be a very simple consisting of the Cornell box, an area light source, and specular spheres. Although a brief explanation of what needs to be done is given below, further implementation details can be found in [Jensen 2001; Jensen 1996], as well as other ray tracing [Shirley 2000], Monte Carlo [Jensen 2003], and global illumination texts [Dutré et al. 2003].

Getting Started: Familiarize yourself with the ray tracer

We are providing you with a ray tracer implementation that will load simple scenes, as well as perform ray casting, ray tracing, and optimized ray-scene intersection tests. This should make your life much easier, however you will still have to implement Monte Carlo sampling, Russian roulette, photon map construction, and put all the photon mapping pieces together. As discussed below, a good first step for this assignment is to use the ray tracer software to **implement direct illumination** on the diffuse surfaces using distributed ray tracing [Cook et al. 1984].

IMPORTANT NOTE: The ray tracer is for exclusive CMU-only academic use. Please do not distribute this software.

Photon Mapping Pass #1: Tracing Photons

In this step, you will implement the first pass of the photon mapping algorithm wherein you **construct the global and caustic photon maps** (see following figure) used by the second pass.

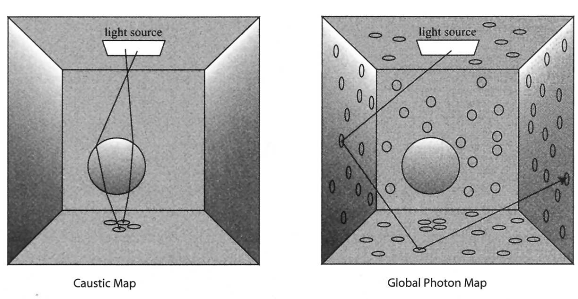

Figure Caustic map and global photon map. The caustic map captures photons which traverse the paths $LS{+}D$, while the global photon map represents all paths. *From [Dutre et al., 2003].*

Build the Global Photon Map (60 points): The global photon map represents the $L(S|D)^*D$ paths that are traced from the light source and terminate on diffuse surfaces. Use Monte Carlo sampling to emit photons of equal intensity from the diffuse area light source. Use the ray tracer's functionality to propagate photons (reflect, transmit and absorb) throughout the scene. To maintain photons of similar intensity, use *Russian roulette* [Arvo and Kirk 1990] to determine if photons are absorbed (diffuse), transmitted (transparent), or reflected at surfaces. Use Schlick's approximation to Fresnel's specular reflection coefficient to determine the probability of transmission and reflection at specular interfaces, e.g., glass. Store the photons in the photon map using Jensen's kd-tree data structure implementation (provided on the web page). Once these photons are stored, the data structure can compute the filtered irradiance estimates you need later.

Build the Caustic Photon Map (20 points): The high-resolution caustic photon map represents the LS^+D paths, and it is therefore only necessary to emit photons toward specular objects when computing the caustic photon map. For general objects, you can use the ray tracer's provided bounding box for the specular object(s) to avoid tracing photons through the entire scene since most will miss specular targets. However, the simplest thing to do is rejection testing on emitted photons using the hemispherical projection of the object's bounding sphere [Jensen and Christensen 1995], which for a scene composed of spheres is trivial. Any photons that still miss specular targets (as determined by the ray tracer) are simply discarded. Be sure to scale the power of the photons appropriately.

Visualize the Photon Maps (20 points): To debug your photon maps, generate (and hand-in) renderings of both global and caustic photon maps, for two cases involving 100 and 10000 photons. Although you can just draw the photon locations as points, you can get the photon map implementation to evaluate the surface irradiance–for Lambertian surfaces, this is proportional to radiosity. You can then use the ray tracer's ray casting ability to generate these images.

Once you can render the global and caustic photon maps directly, you can see differences associated with different photon counts, as well as changes in the number of photons used in the radiance density estimation. Notice the typically blotchy appearance of the global photon map, which explains why it is only used to compute indirect illumination in the photon mapping algorithm.

Photon Mapping Pass #2: Computing Images (100 points)

After the first pass (photon tracing) is completed, the second pass uses the photon maps to render the final images. Images can be rendered using path tracing as follows (from [Dutré et al. 2003]). Rays are traced through each pixel to find the closest visible surface. The radiance for a visible point is split into four components: direct illumination, specular (or glossy) illumination, illumination due to caustics, and the remaining indirect illumination. Each of these four components is computed as follows:

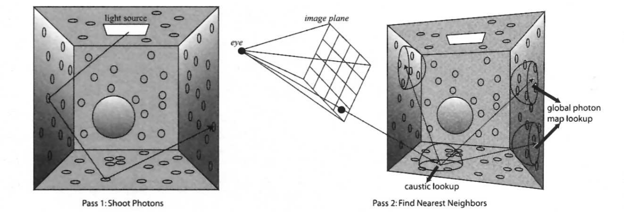

Figure Two passes of photon mapping in a Cornell box with a glass sphere. In Pass 1, photons are traced and deposited on nonspecular surfaces. In Pass 2, global illumination is indirectly computed using the global photon map (as shown). For each indirect ray, the N closest photons in the global photon map are found. Caustics are also found by doing a similar look-up in the caustic map at the visible point. Direct illumination, specular, and glossy reflections (not shown) are computed using ray tracing. *From [Dutre et al., 2003].*

1. **Direct illumination** for visible surfaces is computed using Monte Carlo sampling of the area light sources (as described in class).

2. **Specular reflections and transmissions** are path traced, with the radiance on any terminating diffuse surface determined using the other three components (direct, caustics, and indirect).

3. **Caustics** are computed using the caustic photon map. Since caustics occur only in a few parts of the scene, they are computed at a higher resolution to permit direct high-quality display.

4. **Indirect illumination** is computed by Monte Carlo sampling of the hemisphere (during path tracing). Use the global photon map implementation to compute irradiance on these intersected surfaces. This extra level of indirection decreases visual artifacts arising from the otherwise blotchy global photon map.

Generate at least one rendering of the test scene using full global illumination. Grades will be evenly distributed among the four illumination components.

Tips and Tricks

There are lot of steps before the photon mapping algorithm is complete, so here are some suggestions:

- **Make it easy for us to give you partial grades.** Many will finish this assignment, but some of you may have trouble with one or two steps. Therefore, implement and test the algorithm in steps, e.g., do each illumination component separately. Also, if you do not complete everything, please generate renderings for what you do have.

- **Code direct illumination first** to get familiar with the ray tracing software, and path tracing. You'll be able to render diffuse scenes (without interreflections) and generate images right away.

- **Implement the algorithm using global photon maps first, leaving caustics until later.** Once the rest of the renderer is nearly complete, adding caustics should be relatively easy.

- **The global photon map is a quick indirect illumination approximation.** Indirect illumination involves Monte Carlo sampling on the hemisphere to lookup photon map irradiances. However, you can get a quick-and-dirty indirect illumination up and running fast, by just using the global photon map estimate (as you did in *Visualize the Photon Maps*).

Rendering Contest!

Figure 1: **Futurama**, by Lutz Essers *(First place, Nov-Dec 2003, Internet RayTracing Competition (IRTC))*

Rendering contests, such as the IRTC (`http://www.irtc.org/stills`), can produce some pretty amazing images. So, to support some healthy competition (and higher education), everyone is encouraged to submit a still image to our own rendering contest. I'm offering a prize to *two people* with the best rendered image–as determined by a class vote. The world isn't made of spheres in boxes, so make any scene you wish–let your imagination run wild! The prize is a copy of the Photon Mapping book [Jensen 2001], or something comparable (at the winner's discretion). You can find many free OBJ meshes at `www.turbosquid.com.`

References

ARVO, J., AND KIRK, D. B. 1990. Particle Transport and Image Synthesis. 63–66.

COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed Ray Tracing. In *Computer Graphics (Proceedings of SIGGRAPH 84)*, vol. 18, 137–145.

DUTRÉ, P., BEKAERT, P., AND BALA, K. 2003. *Advanced Global Illumination*. A.K. Peters Ltd.

JENSEN, H. W., AND CHRISTENSEN, N. J. 1995. Photon maps in bidirectional Monte Carlo ray tracing of complex objects. *Computers & Graphics 19*, 2 (Mar.), 215–224.

JENSEN, H. W. 1996. Global Illumination using Photon Maps. In *Eurographics Rendering Workshop 1996*, 21–30.

JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. A.K. Peters Ltd.

JENSEN, H. W., 2003. Monte Carlo Ray Tracing, Siggraph 2003 Course 44, July.

SHIRLEY, P. 2000. *Realistic Ray Tracing*. A.K. Peters Ltd.