PHOTON MAPPING

Image Rendering

ABSTRACT
Rendering photo-realistic images by using photon mapping.

Reema Bajwa
Syed Rizwan Gilani
CS 452 – Computer Graphics
Photon Mapping is a two-pass global illumination algorithm developed by Henrik Wann Jensen. In our class, we have studied and implemented as an assignment, the basic ray tracer. Photon Mapping, as we have learnt it, is one step ahead of ray tracing that not only takes into account the contribution from each and every photon that is emitted from a light source. When light rays hit certain objects present in a scene, there are certain paths that it can follow. These paths depend on the nature of the surface these rays hit. These surfaces can be classified as follow:

- Diffused Surface
- Specular Surface
- Refractive/Reflective Surface

In Photon Mapping we take into account the following paths:
We have attached a pipeline below to explain the steps involved in the process of Photon Mapping.

**Project Initiation:**

We started off with our project with the ray tracer base code in our hand. The first few days were spent on understanding the concept of Photon Mapping and how to integrate it with our ray tracer code. By the end of our research we decided to follow the approach explained in detail in the book named ‘Real Synthet’ by Jensen. The book explained every step of Photon Mapping in detail and gave us a direction to follow.

**Code Structure**

**First Pass**

**Phase 1: Photon Emission:**

As defined by general physics, a photon is an elementary particle - the quantum of light. Therefore, photons basically define the light source. In this first phase, photons are generated and shot into the scene. A light source with higher intensity will produce more photons, and the direction of each photon is randomly selected based on the type of light source (e.g. point light, directional light etc). Random direction is computed using Russian roulette.
**Phase2: Photon Scattering and Storing:**

When a photon hits an object, it can be reflected, transmitted or absorbed (photon scattering). This depends on the type of surface that a photon hits and Russian roulette.

If a photon hits a perfect mirror surface (for which reflectance = 1), it is either totally reflected or absorbed (terminated), depending on the value obtained by Russian roulette. Using the random number generated by Russian roulette, we check if that random number is less than material's reflectance constant. If the case is true, we reflect the photon using reflection equation (as in ray tracing). If the case is not true, photon is absorbed. Photon is not stored in the photon map.

If a photon hits a surface that has both, diffuse and a specular component, it is first stored in the global photon map (defined later), and then can either be reflected, transmitted or absorbed (depending on the random number generated by Russian roulette). If the random number is less than material's diffusion constant, it is reflected in random direction or transmitted (refracted) through the material (in case of a refractive material). If the random number is between material's diffusion constant and materials diffusion+specular constant, it is reflected again (specular reflection).

If a photon hits a surface that has diffusion + specular > the value of random number, it is simple absorbed (terminated). This entire process is summarized in figure 2.0.

Now that the photons are shot and stored (at the point of intersection), the first pass is complete.

*Note that the photons used for implementing caustic illumination are stored in a separate caustic photon map (defined later).*
Second Pass

Phase 3: Radiance Estimate and Rendering

The second pass renders the scene with the help of the photon map (Global and Caustic) built in the first pass. This part is integrated within our previous TraceRay function. Radiance is computed using a function of photon map called irradiance_estimate. The function takes in the intersection point, direction of the ray, lookout value and the total number of photons stored in the photon map as input arguments. Radiance is computed in the following way:

![Figure 1: Radiance Estimate](image)

Not every photon in \( S(P, r) \) would contribute to the radiance at \( P \). In fact, a photon with incident direction \( d \) can contribute only if \( d \cdot N > 0 \), because if \( d \cdot N \leq 0 \), its direction goes inside of the surface. If a photon does not contribute, it is ignored in this radiance estimate for point \( P \). From the illumination equation, the radiance contribution of a photon with incidence direction is:

\[
\text{intensity} \times (d \cdot N) \times \text{diffuse-factor}
\]

Let the sum of all radiance contribution be \( s \). The radiance estimate at \( P \) is \( s / (\pi r^2) \), where \( \pi r^2 \) is the area of a great circle of sphere \( S \). Therefore, the color at \( P \) is the sum of this radiance contribution and the radiance calculated from ray tracing. This method is theoretically sound; however, due to its required mathematics, we do not offer any proof. The sum of the radiance estimate and the radiance collected from ray tracing may be larger than one, and normalization is needed. Additionally, if the number of photons that can contribute to radiance estimate is too small, they are all ignored because the computed radiance estimate from very few photons may produce blurred images.

This global illumination radiance and caustic radiance values are now to be added to the final rendering equation.
Implementing Caustic Illumination:

Light refracted or reflected causes patterns called caustics, usually visible as concentrated patches of light on nearby surfaces. This is implemented using a separate emission function. In this case, photons from each light source are emitted in the direction of refractive objects. Equation of refraction from ray tracing is used to refract these photons within the object (on intersection with the object). As the refracted ray leaves the object, it is refracted again using the same equation and photons are then stored in caustic photon map when they hit the next diffused surface (e.g. from air to glass to air to wall/floor).

Photon Map

The photon hit information is stored in a photon map, which is usually a balanced kd-tree. Each node of the kd-tree stores the information of a photon hit, which include the coordinates of the hit point \((x, y, z)\) (used as the key for building the tree), color intensity \((r, g, b)\), incident direction of the photon, and other important information. Since the number of photons and their hit points may be very large, some form of compression may be needed. While the concept of kd-tree is simple, the implementation of a balanced kd-tree is a non-trivial task. After all photons are shot and the kd-tree is built, the first pass completes.
The Final rendering equation has the following components:

1. Direct Illumination – Ray Tracing
2. Specular – Ray Tracing
3. Global Illumination – Photon Mapping
4. Caustics – Photon Mapping

Final = Direct + Specular + Global + Caustics

Most Challenging Part

- Storing Photons in a photon Map that was a k-d tree and was balanced at every insertion caused a lot of memory errors that were extremely hard to debug and took a major proportion of our time
- Computing radiance values for global illumination and caustics
Intermediate Results – Simple Cornell Box

We have pasted below the results that we got on our way. These results are not only how they should have been and hence portray the stages that we had to overgo in order to achieve the desired results. Some of the results show over exposure of the photons, while some show in proper implementation of the photon map storing structure.

Simple ray tracing – Direct Illumination + Reflection + refraction + shadows
Caustics only – Radiance 1.2
Caustics (Radiance 1.2) with simple ray tracing
Caustics only – Radiance 2.0
Caustics only – Radiance 0.6
Global Illumination – Photons reflected as they hit the walls – Radiance 13.0
Global Illumination - Photons reflected as they hit the walls – Radiance 20.0
Global (Radiance 13) + Caustics (Radiance 0.6)
Global Illumination (Radiance 13.0 and No of photons doubled) + caustics
(Radiance 1.4)
Experimental global illumination result
Global + Specular + Direct with increased reflectivity. Color absorption at each intersection.
Final Render – Global + Direct + Specular + Caustic
Ray Tracing VS Photon Mapping
References:

- Realistic Image Synthesis using Photon Mapping – Henrik Wann Jensen
- Photon Mapping made Easy – Tin Tin Yu, John Lowther and Ching-Kuang Shene
- Lecture Slides – Dr Murtaza Taj