An Analysis of Path Tracing and Photon Mapping in an Attempt to Simulate Full Global Illumination

Claurissa Tuttle
August 2003

Abstract

Realistic image synthesis is a major area of interest in computer graphics. One of the major challenges of rendering images realistically is achieving a full global illumination. The goal is to be able to define a scene with objects and lights and render that scene to make an image that looks real. Once the objects are defined, the challenge is to get the lighting right. Lighting an image is a complicated process because light is reflected, refracted, and absorbed by objects. Light is basically everywhere. In order to determine how much light that a specific point on an object receives, it is necessary to know the material properties of an object. The material properties of an object determine how light interacts with that object. For example, glass will reflect and refract light whereas a mirror will only reflect it. Some materials will absorb most light whereas others will reflect most of it. Current methods used to render realistic images include path tracing and photon mapping.

Introduction and Background

"Realistic image synthesis is the process of creating synthetic images that are indistinguishable from images (such as photographs) of the real world." [1] The goal of this work is to explore methods of rendering an image within a reasonable amount of time, and having that image look real. In order to achieve a realistic look in an image, it is necessary to include global illumination. This illumination is used to simulate the scattering and absorption of light in order to accurately light any point in the model. Global illumination can be divided into four categories: direct illumination, specular reflection, caustics, and indirect illumination. Direct illumination is the light which goes directly to an object from a light source whereas indirect illumination is the light which an object receives, but has bounced at least once before it hits the object. Specular reflection and caustics represent the light that has been reflected from or refracted through a specular object. A caustic results from light rays bending toward a particular area and therefore, concentrating the light in that area. In computer graphics, it is common for this global illumination effect to be generated through ray tracing.

Ray tracing, which is very popular in the area of optics, is very effective in calculating direct illumination and mirror reflection/refraction. In order to achieve indirect illumination and caustics, however, Monte Carlo methods of ray tracing are required. The idea of Monte Carlo ray tracing is to distribute rays stochastically in order to account for all light paths. Monte Carlo ray tracing using a small number of stochastic rays introduces noise into the image. To eliminate this noise, it is necessary to use a large number of rays to calculate such things as indirect lighting. This makes sense because indirect lighting comes from an infinite number of directions included in the hemisphere about the normal of the particular point for which lighting is being calculated. Bidirectional Monte Carlo ray tracing is useful in incorporating shadows into the scene. It is characterized by tracing rays from the light in order to indicate if the object is blocked from the light's point of view, and from the eye in order to indicate if the object is visible.

Using a large number of rays to generate one image becomes a big problem really fast.

The use of a large number of rays is very time consuming. Several strategies have been used in order to reduce noise and/or reduce the number of samples required to render an image. Photon mapping and path tracing are both known methods which are used for rendering indirect illumination and caustics. Path tracing involves tracing a randomly selected path back to the light in order to calculate indirect illumination and caustics. Some paths, however, might become very long before the light is hit. To eliminate long paths, it is common to disregard the incoming light from a path that has become too long. Paths should only be followed for a certain number of bounces in order to minimize the rendering time. Photon mapping, on the other hand, doesn't require tracing a path until it hits the light. With photon mapping, photons are shot out from the light and stored. Indirect light and caustics are determined based on the number of photons stored around a particular point.

Comparison of Methods

Photon Mapping

The idea of photon mapping is to emit light from a light source. Light is represented as photons, which are given a specific power and are shot out from the light source in random directions within the unit hemisphere whose center is the normal of the light source. These photons are stored in a photon map when they hit a diffuse surface, or they are bounced from a specular or diffuse surface and stored when a diffuse surface is hit. A diffuse surface can either absorb (store) a photon, or reflect it. Surfaces such as glass actually split photons. They reflect part of it and refract the other part of it. Henrik Wann Jensen provides C++ code for a photon mapping class in his book on photon mapping[1]. His photon mapping class uses a Kd−tree to keep track of stored photons. He provides a function which gives the irradiance estimate for a specified position in the scene. This function searches for a specified number of photons up to a specified maximum distance away from the given position. The estimate produced by this function is a result of combining the power of all of the photons that it found in its search. Figures 1, 2, and 3, are images of the irradiance estimate. Photons in these images were shot out only towards the two specular objects in the scene. Both spheres were glass and both reflected and refracted the photons. The number of photons that were searched for are 500, 100 and 50 respectively. Searching for 500 photons leads to getting an image that looks pretty smooth, whereas searching for 50 photons leads to getting an image that is less smooth.

These estimates are not directly used in the final illumination of the image. A ray tracer is still used to compute the direct illumination. The indirect illumination, however, does make use of the photon map. Rather than tracing random rays from object to object until the light source is hit, the power of the incoming indirect light is estimated using the photon map. Caustics are calculated using the photon map and added into the final image as well.

Path Tracing

Path tracing is much simpler than photon mapping. Calculation of indirect lighting using path tracing is the same as indirect lighting using photon mapping except, there is no photon map to look up power when an indirect ray hits a surface. Instead we estimate the radiance by stochastically bouncing the ray again in order to find the indirect light on the surface. This is a recursive process. The bouncing of rays continues until a light is hit. Then the function returns from the recursion. Figures 6, 8, and 10 were created using path tracing, but the rays were only

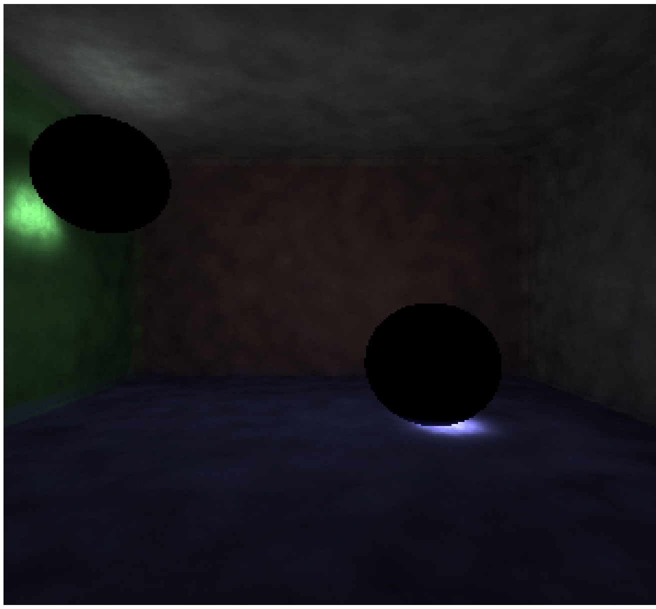

Figure 1: Caustic estimate using 500 photons in the irradiance estimate.

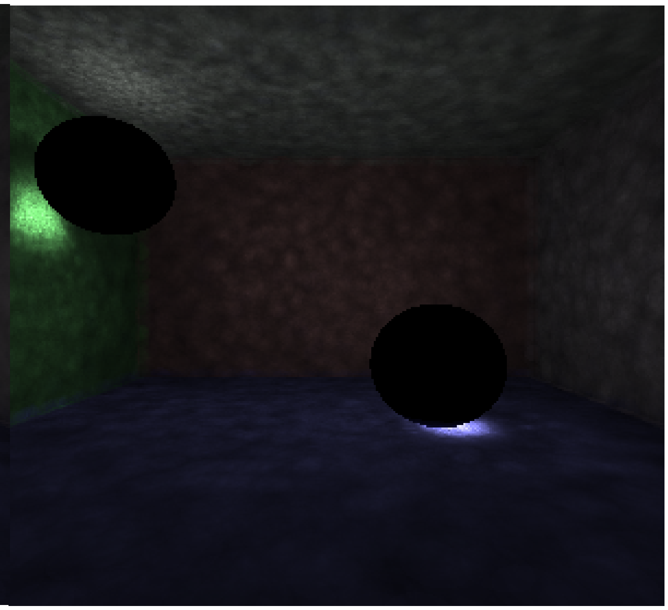

Figure 2: Caustic estimate using 100 photons in the irradiance estimate.

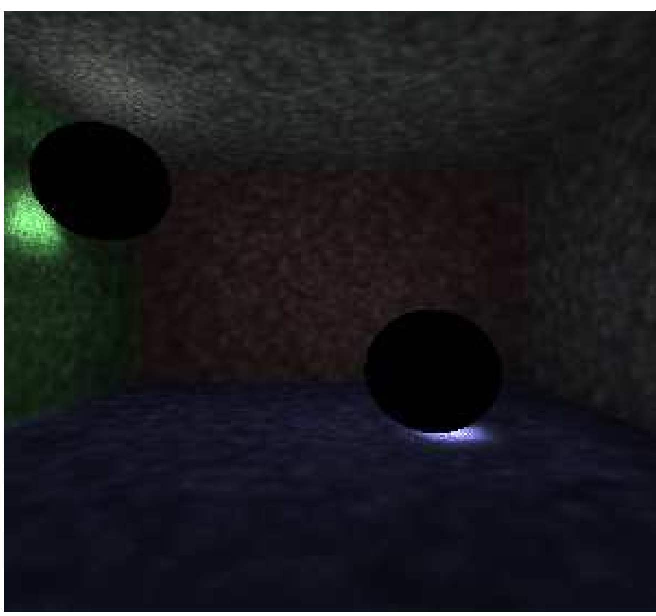

Figure 3: Caustic estimate using 50 photons in the irradiance estimate.

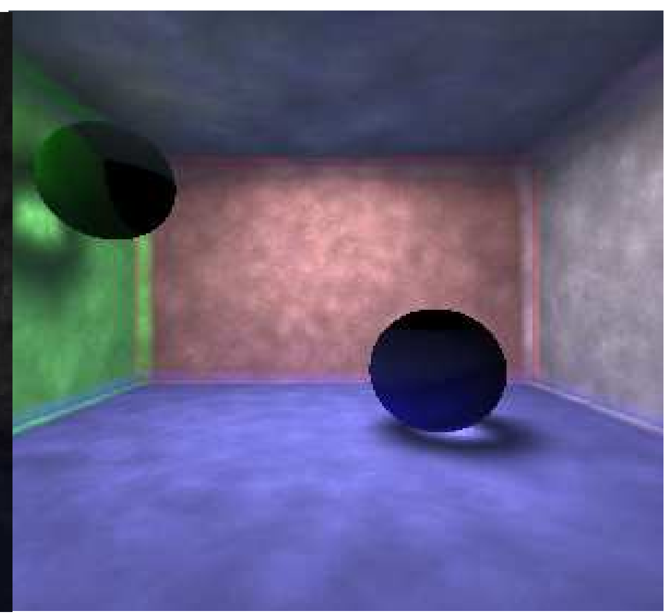

Figure 4: Global illumination using the photon map.

allowed to bounce a total of 5 times. After the 5^{th} bounce, if the ray hasn't hit the light yet, no power is returned. This is a source of noise in the images.

Path Tracing in Comparison with Photon Mapping

In figure 4, it is evident that there is no light source in the global photon map. This is because the light source was excluded from the list of objects. It is, however, included in the rendering using path tracing. This provides for the differences between the brightness of the images create using photon mapping and the images using path tracing. It is noticeable that the path tracing images look brighter.

The photon mapping images tend to have as much static as the path tracing images that were generated using 4 times as many samples as they were. However, the rendering time using 500 photons in the irradiance estimate tended to take about 4 times as long in the photon mapping technique as it took in the path tracing technique. These features seem to have balanced each other out when using 500 photons in the irradiance estimate (photon mapping is not as slow when using less photons in the irradiance estimate). There are, however, other features to be taken into consideration.

Upon examining figures 5 and 6, it can be found that the caustic in image 6 looks sharper than that of image 5. Henrik Wann Jensen has beautiful pictures with very sharp caustics in his photon mapping book[1], but figures 5 and 6 show that photon mapping doesn't always keep sharp edges. The actual look of the caustic, however does depend on the number of photons used in the irradiance estimate and the maximum distance to search for photons. Adjusting those values might yield a better caustic. Photon mapping is supposed to "eliminate noise in the caustic" [1].

Random Ray Generation

When a ray is reflected off of a diffuse surface, it is reflected in a random direction in the unit hemisphere about the direction of the surface normal. In "Realistic Image Synthesis Using Photon Mapping," [1] the method for generating random rays using spherical coordinates is shown. The random rays, however, are generated in a hemisphere biased by the factor of the cosine. This biasing gives a desired effect.

In order to get the hemisphere to be a hemisphere about the surface normal, it is necessary to find two vectors which are orthogonal to each other and the normal and use these in an equation in Henrik's book[1] to get the desired set of random rays. Another way is to take the dot product of the generated ray and the normal and if the dot product is negative, flip the generated ray. When using this dot product method, it is necessary to bias the rays after comparison with the dot product and flipping. Otherwise, the biasing will be backwards for the flipped rays. A third way would be to generate rays randomly on the unit sphere and add the normal to the generated ray and then normalize the result. In this third method, the biasing by a factor of the cosine comes for free (the probability of a ray hitting the center area of the unit hemisphere should be higher than the probability of a ray hitting the edges).

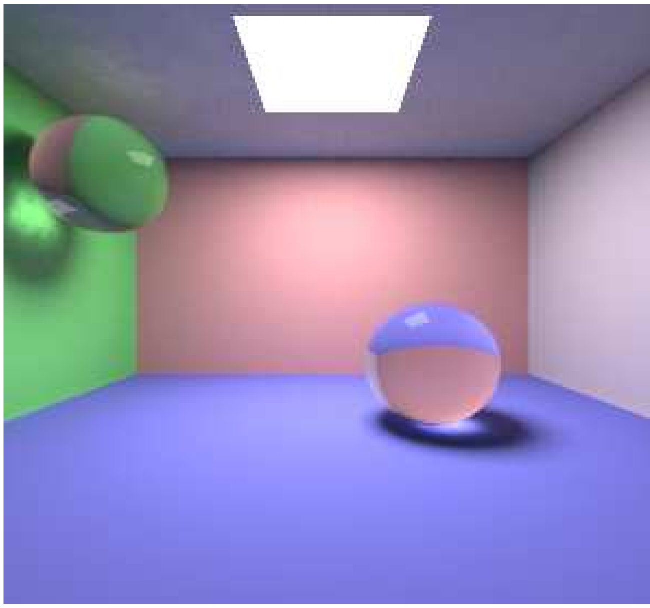

Figure 5: This image was generated using photon mapping with 256 samples.

Figure 6: This image was generated using path tracing with 1024 samples.

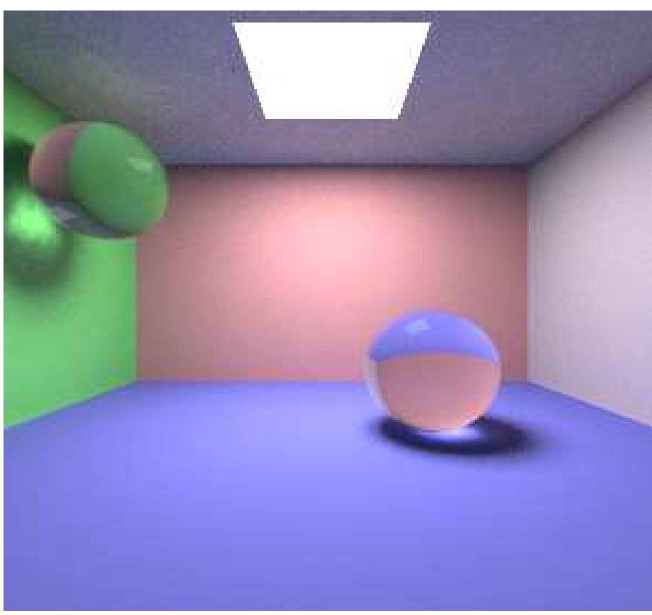

Figure 7: 64 samples.

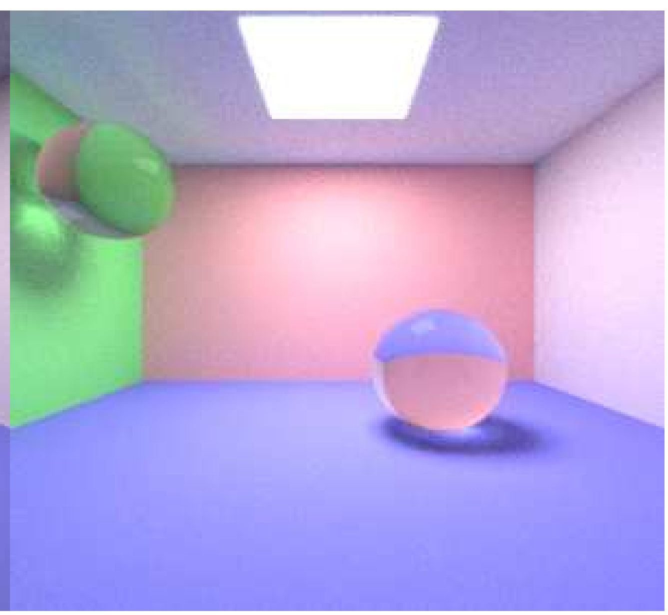

Figure 8: 256 samples.

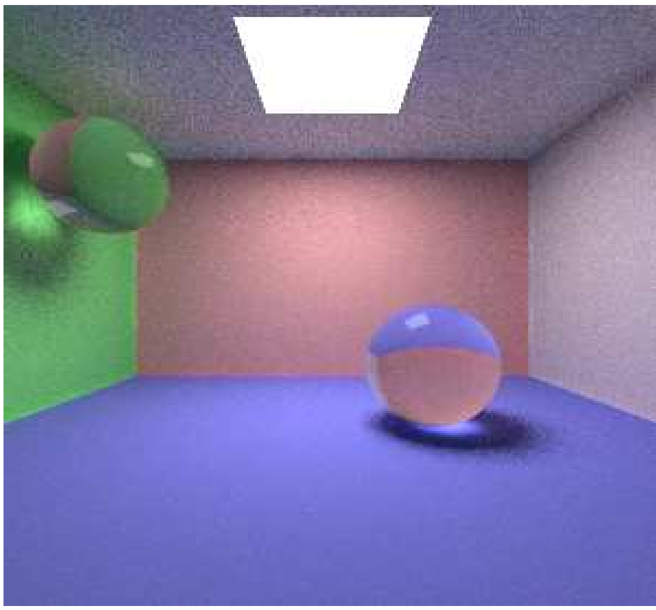

Figure 9: 16 samples.

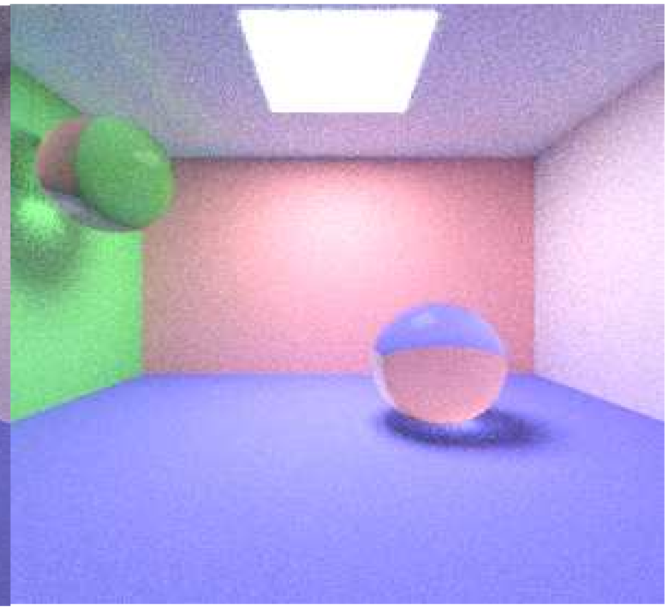

Figure 10: 64 samples.

Future Work and Conclusions

There are definitely tradeoffs between path tracing and photon mapping. Photon mapping has many advantages. One advantage of using photon mapping over other known methods of achieving full global illumination is that it eliminates noise in a caustic. A disadvantage of it is that photon mapping makes it easy to blur the edges of a caustic. However, not all caustics require strict edges. Also, blurred edges of caustics could just mean that something is not right in the implementation of photon mapping. An advantage of path tracing is that the caustic looks pretty exact. A disadvantage is that it takes many samples to eliminate noise in the caustic.

Future work on this project would be to compare ways of generating random rays, adjust some details in the implementation of photon mapping and/or path tracing a bit in order to get the images to look the same. Also, it would be worthwhile to adjust some values to try to get the caustic generated by photon mapping to look a bit sharper.

Acknowledgments

I would like to thank Jeanne Ferrante, University of California San Diego. She was appointed to be my mentor by the Computing Research Association's Distributed Mentor Project. The project was funded by the Computing Research Association through a grant which was provided by the National Science Foundation. I would also like to thank Henrik Wann Jensen, University of California San Diego and Steve Cutchin, San Diego Supercomputing Center for advising me on this project. I would also like to thank the grad students in the graphics lab at

UCSD for the help they offered. Special thanks to Peter Shirley for introducing me to ray tracing and teaching me how to write a ray tracer.

References

[1] Jensen, Henrik Wann. "Realistic Image Synthesis Using Photon Mapping". A.K. Peters, Ltd. Natick, Massachusetts. 2001.
[2] Shirley, Peter. "Realistic Ray Tracing" A. K. Peters, 2003