Ray Tracing and Photon Mapping

Zhengnan Zhu  
 University of Southern California  
 zhengnanz@usc.edu

Song Lu  
 University of Southern California  
 songlu@usc.edu

Yinghao Li  
 University of Southern California  
 liyingha@usc.edu

ABSTRACT

Ray tracing and photon mapping is an accurate and efficient algorithm applied in global illumination [1]. This two-pass algorithm first builds photon maps by emitting photons from the light source, and then it uses the stored photon maps to render images with ray tracing methods. We implement this algorithm with JavaScript, which could be used to create high quality global illumination images and models in online graphic presentation.

1 Introduction

Global illumination simulation is a complex task. Prior to photon mapping, ray tracing is widely used in the field combined with radiosity and Monte Carlo techniques. However, by using photon maps in global illumination, we can get significant improvement with respect to speed, accuracy and versatility. We complete our project based on this two-pass photon mapping algorithm with JavaScript. The dataset is from Cornell Box, which includes five planes and two simple spheres.

Firstly we construct the ray tracing algorithm without using photon maps. Given the light, we trace the radiance from light source to different objects differently, and we calculate the pixel colors after getting the objects normal. In this way, our rendering result is an image without photons.

Secondly, we render the images with photons. We set up the photons map by emitting photons from the light source. These photons are stored in balanced KD-trees. When we compute colors, we choose a batch of nearest photons to the pixels, sum these selected photons to get the color results.

In the end, we integrate the single ray tracing part and photon maps part to our system demonstration. Users can drag the objects on the online demonstration to see the instantaneous rendering results. We can conclude that photon mapping methods give us more accurate and realistic images.

2 Models

We will describe our models in detail.

2.1 Data

We implement this method based on the Cornell Box data. We change the image a little so that each plane is composed of two triangles, which is similar to what we learned in class. There are total two spheres and five planes, simulating objects in a cube but leaving the missing plane as the view.

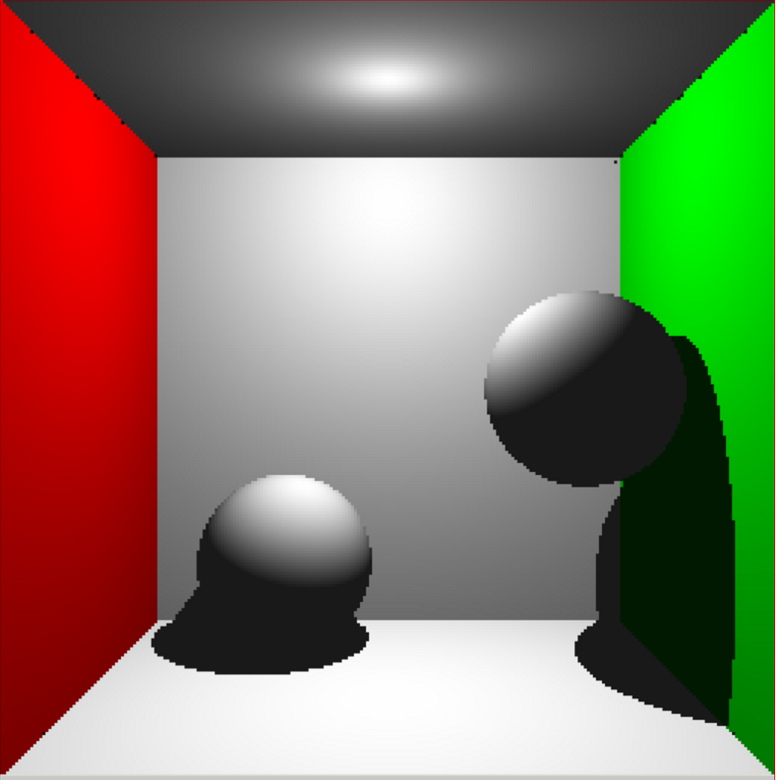
2.2 Ray Tracing

In this part, we compute the pixel color directly to get rendered images. Given specific image resolution, we first need to convert the pixels to image plane coordinates. After getting the ray coordinates, we trace them to global objects, both planes and spheres.

Until the ray intersects with objects, we record current states. Every time the ray intersects with some object, we will calculate its intersected points and distance, and iteratively choose the nearest intersected point and update the distance.

Then we apply standard illumination models. We trace the ray and compute object colors that are not in shadow. For shadows, we choose the ambient light.

Simple ray tracing rendering result is Figure 1; we can find that only using ray tracing method will not give us good result:



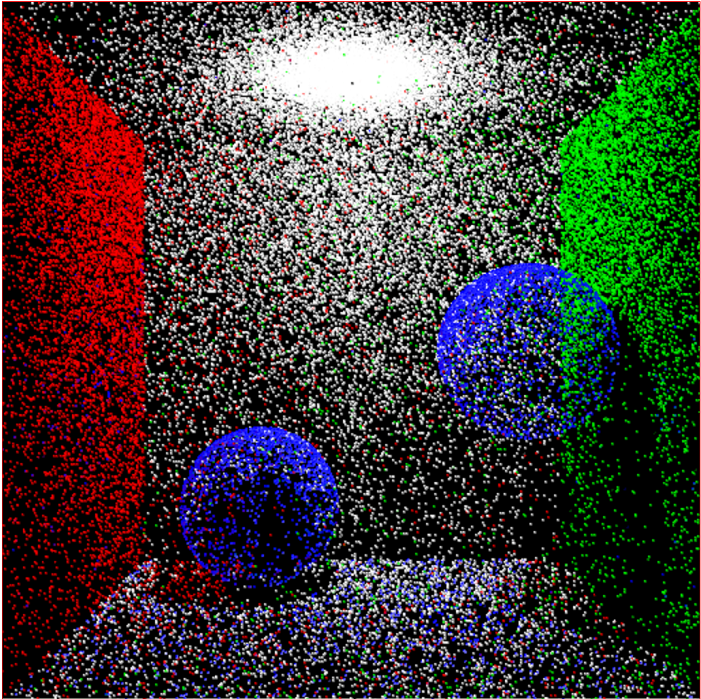
2.3 Photon maps

In order to build photon maps, we simulate emitting photons from the light source. In the program, we fix the total energy of photons and change the number of photons to be emitted. We make the number to be 50,000. Using the ray tracing method in part 2.1, we trace the photons path. But differently, photons can intersect with one object and then it is reflected to another. In this case, we trace every object for each photon intersects with and store it in kd-tree. Photons can’t reflect permanent so we set a ratio to decide if the photon is reflected or it vanishes. In our system demonstration, we can show the all the photons recorded in the image.

kdtree?

With constructed photon maps, we will turn to rendering image with photons. For each ray above, we can have a bunch of photons around the point. We will choose certain nearest number photons as good photons. Since each photon already carries important information like colors, now we can sum theses near photons by some weights. The weights are decided by XXX. This way gives us rendering results with photon maps.

We can visualize the emitted photon maps and draw these photons to our canvas, the result is Figure 2:



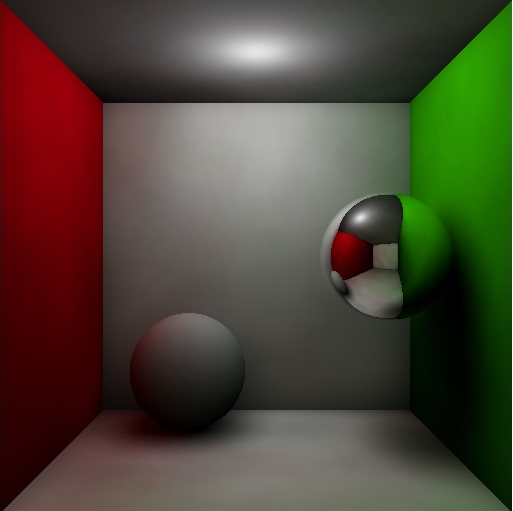
2.4 Web

ZZZZZZZ

3 Experiments

We run our systems with following parameters. The resolution of image is 512\*512. The number of emitted photons is 50,000, which is enough to render a more accurate image than single ray tracing. Refection ratio of the photons is 0.5, that means once a photon intersects with some object, there is 50% probability that it vanishes and 50% probability to reflect to continue tracing.

After adding photon maps, our result is in Figure 3:



4 Conclusions

We have realized the global illumination method based on photon maps with javascript. The information in the global photons map is used to generate optimized sampling directions, to reduce the number of shadow rays and to limit the number of reflections traced by providing an approximate radiance estimate.

Besides, we successfully implemented this algorithm on online graphic models, where the systems can interacts with users and show high quality global illumination images to people.

ACKNOWLEDGMENTS

“Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.”

REFERENCES

[1] Jensen, Henrik Wann and Niels Jørgen Christensen: ”Photon maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects”. Computers and Graphics 19 (2), pp. 215-224, 1995.

[2]. Wojciech Jarosz, Henrik Wann Jensen, and Craig Donner. 2008. Advanced global illumination using photon mapping. In ACM SIGGRAPH 2008 classes (SIGGRAPH '08). ACM, New York, NY, USA, Article 2, 112 pages.

[3] Shirley, Peter: ”A Ray Tracing Method for Illumination Calculation in DiffuseSpecular Scenes”. Proceedings of Graphics Interface ’90, pp. 205-212, 1990.

[4] Bentley, Jon Louis: ”Multidimensional Binary Search Trees Used for Associative Searching”. Comm. of the ACM 18 (9), pp. 509-517, 1975.

[5] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY..

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00