Optimizing Photon Mapping Using Multiple Photon Maps for Irradiance Estimates

Bent Dalgaard Larsen
Technical University of Denmark
bdl@imm.dtu.dk

Niels Jørgen Christensen
Technical University of Denmark
njc@imm.dtu.dk

ABSTRACT

The photon mapping method is used extensively in global illumination to render photorealistic pictures. We describe a simple optimization technique for calculating the indirect illumination by modifying the photon mapping method. Using our method the photon maps are divided into several photon maps based on the topology of the polygons in the scene. This modification of the photon mapping method has several advantages compared to the traditional method. We demonstrate that the indirect illumination can be calculated faster using our method.

Keywords

Photon mapping, Global Illumination, kd-tree.

1. INTRODUCTION

Photon mapping is a method used for calculating global illumination effects [Jensen95,Jensen00, Jensen01]. The method has been used to calculate fast and accurate solutions to the global illumination problem. In this paper we will introduce a new simple modification to the existing photon mapping method that makes it faster to calculate global illumination effects without compromising the visual quality. Usually three photon maps are being used in the photon mapping algorithm. One for caustics, one for indirect illumination and one for participating media. In this paper we will only consider the indirect illumination.

2. BACKGROUND

Photon mapping is one way to solve the rendering equation introduced in [Kajiya86]. The outgoing radiance L_o in the point x in the outgoing direction $\vec{\omega}$ can be described as a solution to the equation:

$$L_o(x,\vec{\omega}) = L_e(x,\vec{\omega}) + L_r(x,\vec{\omega})$$

WSCG POSTER proceedings

WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

Copyright UNION Agency – Science Press

Where the reflected radiance $L_r(x,\vec{\omega})$ is described using the following integral:

$$L_r(x,\vec{\omega}) = \int_\Omega f_r(x,\vec{\omega}',\vec{\omega}) L_i(x,\vec{\omega}')(\vec{\omega}'\cdot\vec{n})d\vec{\omega}'$$

Where $L_i(x,\vec{\omega}')$ is the incoming radiance.

Using the photon mapping method the reflected radiance is divided up into several components and these components are then solved differently:

$$\begin{aligned}
L_r(x,\vec{\omega}) &= \int_\Omega f_r(x,\vec{\omega}',\vec{\omega}) L_i(x,\vec{\omega}')(\vec{\omega}'\cdot\vec{n})d\vec{\omega}' \\
&= \int_\Omega f_r(x,\vec{\omega}',\vec{\omega}) L_{i,l}(x,\vec{\omega}')(\vec{\omega}'\cdot\vec{n})d\vec{\omega}' + \\
&\quad \int_\Omega f_{r,S}(x,\vec{\omega}',\vec{\omega})(L_{i,c}(x,\vec{\omega}') + L_{i,d}(x,\vec{\omega}'))(\vec{\omega}'\cdot\vec{n})d\vec{\omega}' + \\
&\quad \int_\Omega f_{r,D}(x,\vec{\omega}',\vec{\omega}) L_{i,c}(x,\vec{\omega}')(\vec{\omega}'\cdot\vec{n})d\vec{\omega}' + \\
&\quad \int_\Omega f_{r,D}(x,\vec{\omega}',\vec{\omega}) L_{i,d}(x,\vec{\omega}')(\vec{\omega}'\cdot\vec{n})d\vec{\omega}'
\end{aligned}$$

Where $L_{i,l}(x,\vec{\omega}')$ is the direct illumination from the light source, $L_{i,c}(x,\vec{\omega}')$ is the caustics, $L_{i,d}(x,\vec{\omega}')$ is the indirect diffuse illumination that has been reflected diffusely at least once, $f_{r,D}(x,\vec{\omega}',\vec{\omega})$ is the diffuse term of the BRDF, and $f_{r,S}(x,\vec{\omega}',\vec{\omega})$ is the specular/glossy term of the BRDF [Jensen01].

In the following we will only consider the indirect illumination which is the last component of the rewritten expression for the reflected illumination. Calculating the irradiance using the photon mapping method implies finding the n nearest photons, summing their energies and dividing by the area they span. This way the reflected radiance can be approximated as [Jensen01]:

$$L_r(x,\vec{\omega}) = \int_{\Omega} f_r(x,\vec{\omega}',\vec{\omega}) \frac{d^2\Phi_i(x,\vec{\omega}')}{dA_i}$$

$$\approx \sum_{p=1}^{n} f_r(x,\vec{\omega}_p,\vec{\omega}) \frac{\Delta\Phi_p(x,\vec{\omega}_p)}{\Delta A}$$

The area is usually found as an intersection of the sphere surrounding the *n* nearest photons and the tangent plane of the point where the irradiance is calculated, although other methods could be used e.g. finding the convex hull around the *n* nearest neighbors.

Many existing optimization techniques exist for increasing the speed of calculating the diffuse illumination using photon mapping. For a good overview see [Jensen01] and [Jensen02]. Many of these techniques are independent of each other and it is possible to use several in conjunction. A method that is of relevance to our optimization technique is the method suggested by Christensen [Christensen00]. The idea is to precalculate the irradiance at all photon positions and when an irradiance estimate is needed for an arbitrary position the nearest photon is located and the precalculated irradiance from this photon is used. Christensen argued that it is not necessary to precalculate the irradiance at all photons and he chooses only to calculate the irradiance in one out of four photons. Even with this time-consuming preprocessing method he reported good speedups. In his algorithm it is necessary to store a normal together with the irradiance since it is important only to choose the nearest photon that has a normal similar to the surface currently considered in order to avoid artifacts. The method suggested in [Christensen00] was used in our implementation. We will in the following argue that several advantages can be achieved by using multiple photon maps instead of just one for the indirect illumination which is the method previously used.

3. THE ALGORITHM

In order to calculate a good approximation for the irradiance a large number of photons are needed. Furthermore it is important that it is truly the nearest photons that are being used. One exception to that is when the photons are located on a surface that does not point in the same direction as the surface where the irradiance is calculated. In this situation usually a disc is used to find the nearest photons on surfaces pointing in the same direction (See Figure 1 and 2).

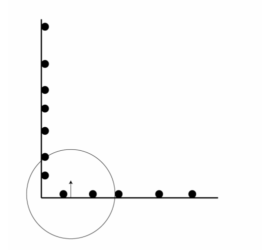

Figure. 1. Photons from one surface leaking to another surface

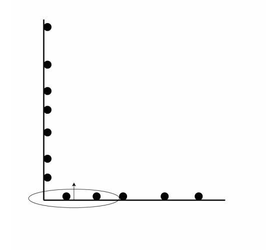

Figure. 2. Leaking avoided by using a disc instead of a sphere

This suggests that it would be advantageous to use different photon maps on adjacent surfaces that have a large angle between them as another way of avoiding leaking.

The important question to answer can therefore be summarized to: When should two adjacent polygons use the same photon map and when should they use a different photon map?

The rule we have chosen is the following:

- If the angle between two adjacent polygons is **below** a predefined threshold (α) they should use the same photon map for storing photons and lookups on the nearest neighbors. An edge between such two polygons is classified as *connected*.
- If the angle between two adjacent polygons is **above** the same predefined threshold (α) they should use different photon maps for storing photons and lookups on the nearest neighbors. An edge between such two polygons is classified as *unconnected*.

This method is very similar to the way hard and soft edges are found in e.g. VRML. Here a variable called *crease_angle* is used to specify whether the normals on an edge should be interpolated between the two polygons that this edge connects [VRML95]. It is noted that the angle between two polygons is the angle between their normals.

The method for assigning a photon map to a polygon can be described using the following pseudo-code:

1. Mark all edges as either *connected* or *unconnected*
2. Assign a unique ID to all polygons
3. If two polygons share a *connected* edge their ID should be made identical
4. Create a photon map for each of the remaining ID’s
5. Each polygon is now connected to a photon map and several polygons can share the same photon map

An example of the resulting photon maps from the algorithm can be seen in Figure 3 and 4.

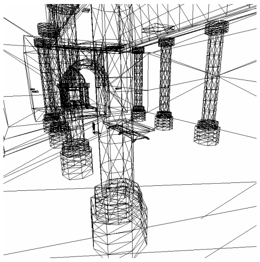

Figure 3. A wireframe polygon model of a scene

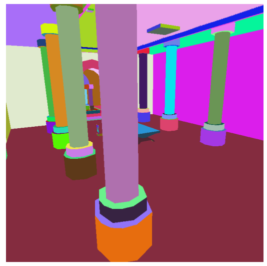

Figure 4. All polygons marked by a color from the photon map they refer to

4. RESULTS

The key issue is answering the question: How big is the advantage of dividing *N* photons into *M* photon maps compared to having one photon map with *N* photons? In an attempt to answer this question we have made two different tests. In the first test we measure the balance time and the time to calculate and irradiance estimate using different sizes of photon maps. For this purpose we use photons that are distributed randomly in space, although this is unlikely in a real scene. Nevertheless it will give an indication of the performance optimization that can be achieved. In the second test we used a simple scene and rendered it both using one photon map as usual and using several photon maps as we propose. All tests were performed on a P4 1.7 MHz Dell Portable with 512 level 2 Cache. The code used to calculate the irradiance is the code made available in [Jensen01].

The results of the first test can be seen in Figure 5, 6 and 7. In Figure 6 the balance time per photon is measured against the number of photons in the photon map. Balancing a binary tree is O(*n*log(*n*)). It is therefore expected that the growth in the diagram is constant when drawn logarithmic. In our implementation each photon uses 40 bytes and with a 512 Kb level 2 cache there is room for approximately 13.000 photons in this cache. We believe this is the reason for the different appearance of the graph before and after 13.000 photons.

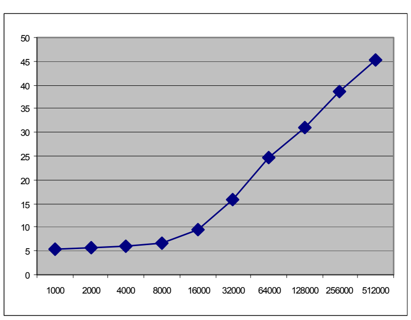

Figure 5. Balance time per photon (in μs) as a function of photons in the photon map

In Figure 6 the cost of calculating an irradiance estimate is shown as a function of the number of photons in the photon map. The savings are not as significant as when balancing the photons, but nevertheless a few hundred percent can still be saved by creating smaller photon maps.

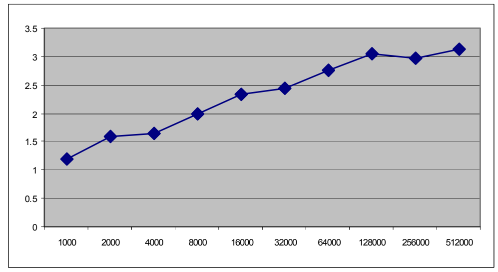

Figure 6. Cost of calculating an irradiance estimate using 500 photons (in ms) using photon maps of different sizes

In Figure 7 we took *n* photons and divided these photons into *m* photon maps.

Photons (*n*)	Maps (*m*), each with *n/m* photons	Balancing the *m* photon maps	Lookup for 500 nearest photons
1000000	1	5.05 s	332 μs
1000000	10	3.08 s	291 μs
1000000	100	0.70 s	213 μs
1000000	1000	0.55 s	116 μs

Figure 7. Comparison of photon map lookups using different amounts of photon maps for storing the same amount photons

All these results found by creating 100 completely random photon maps and then averaging the timings.

In the second test we rendered the same scene using one photon map as usual with the photon mapping method (see Figure 9) and using several photon maps as we propose (see Figure 10). As expected there is no visible difference between the two images. In the scene displayed in figure 3 and 4 the timing difference between using one and several photon maps is substantial (see Figure 8). 50.000 photons were used in this experiment.

	Balancing the photon map(s)	Precalculating the irradiance estimates
One photon map	0.56 s	13.80 s
Several photon maps	0.08 s	5.88 s

Figure 8. Comparison of timings for precalculating irradiance estimates for images in figure 3 and 4

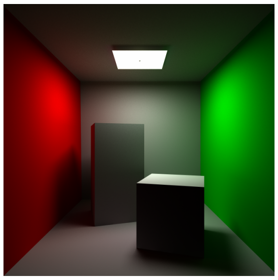

Figure 9. Calculated using one photon map

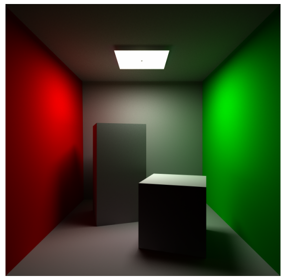

Figure 10. Calculated using several photon maps

It is clear from our results that it is an advantage to distribute the photons into several photon maps.

5. DISCUSSION

Although our proposed solution results in a speedup; using several photon maps instead of one should be done with care. A scene with many small triangles and sharp angles between these will not be a good candidate for this optimization as the angles between the triangles will no longer be a good measure for when to split the photon map. This will typically be the case in scenes generated using fractal algorithms. In general it is important to have a significant amount of photons in each photon map. If dividing the photon map into multiple photon maps violate this property it is not a good idea to use our method. The photon map is created using a left balanced photon map. If photons are added or removed from the data structure it is necessary to rebalance the entire tree. But if the photon map is split up into several photon maps it is only necessary to update the photon maps that have been modified. This property can be very useful in animations in which only some parts of the scene are modified from frame to frame. Furthermore if the radiance has been precalculated as described in [Christensen00] this precalculation can also be reused. In addition, by dividing the photon maps into several photon maps as we suggest it is no longer necessary to store the additional normal introduced by [Christensen00].

To summarize:

The advantage of having several photon maps

- Faster irradiance calculations
- Faster balancing of the photon maps
- No leaking problems in corners
- It may be possible to update a limited number of photon maps when creating animations.

Disadvantage of having several photon maps

- Connectivity has to be calculated
- Does not apply to all scenes

Using our method on the caustics and volume photon maps from the photon mapping algorithm is not as easy as the indirect illumination. This is because it is not as easy to figure out when to split these photon maps. But if the problem of figuring out when to split the photon maps is solved, our optimization applies to them as well.

6. CONCLUSION

We have proposed a simple yet efficient and easy way to optimize the calculation of indirect illumination using the photon map algorithm. We have demonstrated that it gives a speedup. But we have also discussed that our solution will not be a good approach in all circumstances.

7. REFERENCES

[Kajiya86] James T. Kajiya, *The rendering equation*, SIGGRAPH 1986

[VRML95] Gavin Bell, Anthony Parisi and Mark Pesce, *The Virtual Reality Modeling Language*, 1995 http://www.vrml.org/VRML1.0/vrml10c.html

[Jensen95] Henrik Wann Jensen and Niels Jørgen Christensen, *Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects*, Computers & Graphics vol. 19 (2), pages 215-224, March 1995

[Christensen00] Per Henrik Christensen, *Faster Photon Map Global Illumination*, Journal of Graphics Tools, volume 4, number 3, pages 1-10. ACM, April 2000

[Jensen00] Henrik Wann Jensen, Niels Jørgen Christensen, *A Practical Guide to Global Illumination using Photon Mapping*, SIGGRAPH 2000 Course Note #8. ACM, July 2000.

[Jensen01] Henrik Wann Jensen, *Realistic Image Synthesis Using Photon Mapping,* A K Peters, 2001.

[Jensen02] Henrik Wann Jensen, Frank Suykens, Per Henrik Christensen and Toshi Kato, *A Practical Guide to Global Illumination using Photon Mapping*, SIGGRAPH 2002 Course Note #43. ACM, July 2002