EFFICIENTLY RENDERING SHADOWS USING THE PHOTON MAP

Henrik Wann Jensen
Niels Jørgen Christensen

Dept. of Graphical Communication
Technical University of Denmark
e-mail: igkhwj@unidhp.uni-c.dk
url: http://www.gk.dtu.dk/home/hwj/

ABSTRACT

This paper presents a method that reduces the number of shadow rays used in ray tracing. This reduction is achieved by extending the concept of the photon map. Here the photons are characterized as either shadow photons or direct illumination photons and in the visualization step this information is used to eliminate shadow rays in areas that are either completely illuminated or completely shadowed. The method handles both sharp and soft shadows and it handles complex objects (ie. procedurally based objects). We have integrated the method into a two-pass global illumination renderer and our results indicate that we are able to reduce the number of shadow rays with more than 90 %

Key Words: Shadows, Ray Tracing, Photon Map, Monte Carlo

INTRODUCTION

Accurate rendering of shadows is essential in the generation of photorealistic images. In ray tracing [Whitted80] the computation of shadows is elegantly performed by using shadow rays. Whenever we want to examine whether a point within the scene is illuminated we send a shadow ray to the light source. If the shadow ray intersects an object before hitting the light source then the point is in shadow otherwise it is illuminated. This approach is very simple but unfortunately also very time consuming.

Several papers have been presented that optimizes the computation of the direct illumination by reducing the number of shadow rays [Wooet al. 90]. Haines et al. [Haines-Greenberg86] introduced the light buffer. A light buffer is created for each light source in the scene. The buffer contains lists of objects as seen from different directions of the light source. These lists are used either to eliminate shadow rays or to reduce the number of objects that must be tested. The method is not very useful in complex scenes as every object must be represented a least once for each light source. Woo et al. [Woo-Amantides90] uses a global octree in which he precomputes the visibility of each voxel as seen from the light source. Every voxel is tagged as either illuminated, shadowed or as having complex illumination. Shadow rays are only required for objects within voxels that have complex illumination. However, this approach is limited to polygons and is not very useful in complex scenes since the entire scene must be projected into the direction of every light source.

A different kind of strategy is presented by Shirley et al. [Shirley-Wang91, Shirley et al. 96] and Ward [Ward91]. They mainly focus on scenes with a large number of light sources and they use probabilistic methods to distribute the shadow rays. These methods are well suited for complex scenes with many light sources. In scenes with only few light sources there is almost no benefit from the methods since they use shadow rays — even in the simplest situations.

The idea of probabilistic sampling has been pushed even further by Zimmerman et al. [Zimmerman-Shirley95]. They combine the probabilistic sampling scheme with a preprocessing step in which they use a statistically based approach to precompute visibility estimates regarding the light sources as seen from different volumes (cubes) within the scene. These visibility estimates are computed using up to 100 sample rays that are sent between random positions within the cube to random positions on the light sources. If each one of the rays intersect other objects it is assumed that nothing within the cube can be illuminated by the particular light source. This assumption is very critical when only a small part of the cube is illuminated and it can give rise to errors that are very easily seen by the eye - such as jagged shadow edges. Futhermore the preprocessing step in which the visibility estimates are computed is rather time consuming even compared to the rendering time. The method does however provide substantial reductions in the number of shadow rays used and it is particularly suited for rendering high resolution images.

Recently Collins [Collins95] presented a paper in which he introduced the concept of negative splatting. This approach uses illumination maps and it is similar to the simulation of caustics. However, instead of splatting positive energy into the illumination map, negative energy is used. During visualization everything is considered fully illuminated and shadows are created as the negative energy from the illumination map is used. This approach gives nice looking soft shadows. It is however limited to plane surfaces and it requires a texture map for each surface in order to be able to represent shadows.

Currently there is a lack of techniques suitable for complex scenes. The probabilistic techniques [Shirley-Wang91, Shirley et al. 96, Ward91, Zimmerman-Shirley95] use at least some shadow rays in any region. However, in most scenes large areas are either illuminated or shadowed and if these areas could be detected then we could restrict the use of shadow rays to the critical areas around the shadow boundaries.

Previously we have presented a method which uses the concept of the photon map in a two-pass global illumination method [Jensen-Christensen95a]. This method is particularly useful in complex scenes with complex objects (ie. procedurally based objects). In the first pass the photon map is created by emitting a large number of photons from the light sources into the scene. This photon map is used during the visualization step to characterize the incoming light and thus provide more efficient sampling [Jensen95].

In this paper we present a method which uses the photon map to identify shadow boundaries within the scene. The method is probabilistic and it is based upon an extension of the photon map by the concept of shadow photons. These shadow photons can be used to either completely eliminate the use of shadow rays or to restrict the use of shadow rays to regions along the shadow boundaries.

EXTENDING THE PHOTON MAP

The photon map [Jensen-Christensen95a] represents a distribution of photons (particles) throughout the scene and it is created by emitting a large number of photons from each light source into the scene based upon the emissive characteristics of the light source. Each photon is traced through the scene using a technique similar to path tracing [Kajiya86]. Every time a photon

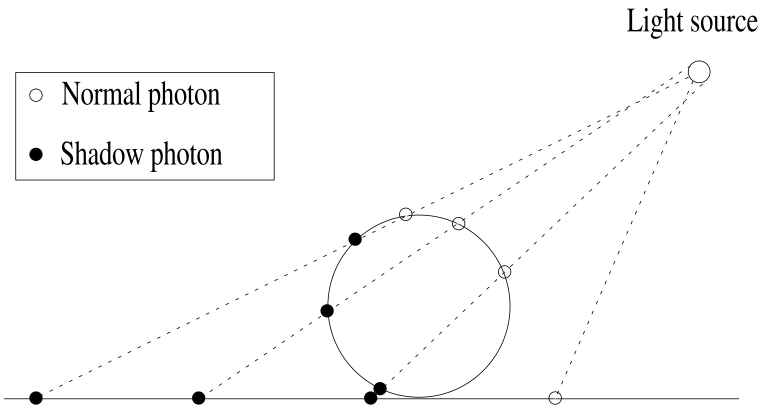

Figure 1: Creating shadow photons

hits a surface two things happen: Firstly, if the surface is diffuse the photon is stored in a global data structure. Secondly, Russian roulette is used to determine whether the photon should be reflected or absorbed. If the photon is reflected the bidirectional reflection distribution function of the surface is used to generate a new reflected direction.

In order to capture shadow information we extend the concept of the photon map and introduce shadow photons (see figure 1). Shadow photons are created by tracing rays with origin at the light source through the entire scene. At the first intersection point a standard photon (with positive energy) is created and at all of the following intersection points a shadow photon (with negative energy) is created. This approach is similar to the negative splatting performed in [Collins95]. In this way we obtain three types of photons in the extended photon map: The normal photon, the illumination photon (the part of the normal photons that represents direct illumination) and the shadow photon.

This extended photon map can be used in several ways to optimize the calculation of the direct illumination. In order to obtain information about the direct illumination at an intersection point, $\mathbf{x}$, we locate the nearest photons around $\mathbf{x}$ as it is done in [Jensen-Christensen95a, Jensen95]. We only use the photons that are part of a ray with origin at a light source — as illustrated in figure 1. This gives us a number of shadow photons, n_s, and a number of photons representing direct illumination, n_i. Based upon these two numbers we obtain three classifications of $\mathbf{x}$:

1. $n_i = 0 \rightarrow$ All photons are shadow photons $\rightarrow \mathbf{x}$ is shadowed

2. $n_s = 0 \rightarrow$ All photons are illumination photons $\rightarrow \mathbf{x}$ is illuminated

3. $n_s \neq 0$ and $n_i \neq 0 \rightarrow$ We have a mixture of shadow photons and illumination photons $\rightarrow \mathbf{x}$ is part of or near a shadow boundary.

This classification can be used to eliminate shadow rays. In case 1 the direct illumination is zero and we do not have to use any shadow rays. In case 2 we just have to add the contribution from the light source. It is only in case 3 that the use of shadow rays is beneficial. We could use the photons to predict the visibility of the light source:

$$\text{visibility} = \frac{n_i}{n_s + n_i} \qquad (1)$$

This technique does unfortunately require a large number of photons to get accurate estimates and it is particularly problematic at sharp shadow boundaries. More accurate results would be obtained by using shadow rays. In this way using fewer photons will not affect the quality of the shadow but instead create larger areas along the shadow boundary in which shadow rays are used.

In order to handle shadows from multiple light sources each photon must be tagged with the light source that created the photon. In the computation of shadows, photons from different light sources are seperated into different classes. For each light source, l, we would have to determine $n_{s,l}$ and n_{i_l} (ie. the number of shadow photons and illumination photons from that particular light source). These two numbers could either be used to predict the visibility of light source l or to determine whether it is necessary to use shadow rays.

An alternative strategy could be obtained by visualizing the results from the photon map directly (like it is done by [Shirley et al. 95]). This can be done in two ways. We could either visualize all of the photon map directly — this will require a very large number of photons — or we could create shadows by adding the negative energy from the photon map. In this way we would always assume that the light sources are visible and shadows are created simply by adding the negative energy carried by the shadow photons. The advantage of this approach is that it is no longer necessary to distinguish between photons from different light sources.

We are not using any of these last two strategies since our use of the photon map is based upon a rough distribution of the light within the scene. We create the photon map in the first pass of a two pass method. During visualization the photon map is only used to improve the quality of the visualization without increasing the number of rays.

RESULTS AND DISCUSSION

We have implemented the shadow optimization in a rendering program called MIRO. All the results have been produced on a 100 Mhz Pentium PC running Linux.

We have tested the algorithm using two test scenes. The first test scene is shown in fig. 2. It is a marble sphere illuminated by a small spherical light source which gives rise to a rather sharp shadow. The image contains 4 different renderings of the scene. A reference image computed using path tracing with 100 samples per pixel. A normal rendering using the two pass method. A normal rendering using shadow rays only in the regions that contains a mixture of shadow photons and illumination photons. Finally the image contains a rendering in which no shadow rays have been used — the visibility estimate (equation 1) is used directly.

The visual inspection of the images shows that the quality of the path tracing image is the best (in particular in the highlights). Most important however is the fact that there is no difference between the shadows in the normal rendering and the rendering in which shadow rays are only used along the shadow boundary — by using only 3420 shadow photons and 30 photons to make predictions regarding the direct illumination we were able to reduce the number of shadow rays with more than 90 % without any artifacts. Eliminating the shadow rays and using the visibility estimate results in a shadow that is too blurred. To obtain sharp shadows from the visibility estimate requires a much higher number of shadow photons.

The second test scene shown in fig. 3 is a variant of the Cornell box demonstrating a significant amount of indirect illumination and also large areas with soft shadows due to the large square-shaped light source. This test scene is also visualized using the 4 different rendering techniques.

Again we can see that the quality of the shadows are quite good in the first three images. The shadows in the image using the visibility estimate are not very good. There are errors in the areas around the cube and the shadow photons below the cube generate false shadows along the side of the cube. We used 10690 shadow photons, 176.000 normal photons (both direct and indirect illumination) and 50 photons to estimate the direct illumination. Using more than 50 photons in the

Marble sphere			
Scene	Time	Shadow rays	Rays total
Path tracing	2100 s.	8.574.150	30.325.200
Two pass rendering	31 s.	248.259	452.185
Shadow optimization	28 s.	13.104	246.718
Visibility estimate	28 s.	0	234.001

Box Scene			
Scene	Time	Shadow rays	Rays total
Path tracing	16 hours	269.361.500	656.126.500
Two pass rendering	915 s.	2.033.070	3.073.938
Shadow optimization	847 s.	182.556	1.226.560
Visibility estimate	843 s.	0	1.044.132

Table 1: Rendering statistics

estimate would make the shadows even more perfect in image 3 since the region along the shadow boundary becomes larger. The visibility estimate would probably be worse since the false shadows would become even larger.

We collected some statistics while rendering the images. A few important numbers are given in table 1. As we can see significantly fewer shadow rays are used when the information in the photon map is used. In both scenes the number of shadow rays were reduced with more than 90 % This also clearly affected the total number of rays used since shadow rays represents a large percentage of the rays used. The rendering time was not reduced to the same degree. This is due to the fact that most of the rendering time is spent computing indirect illumination and reusing this information using the irradiance gradient method [Ward92]. For complex scenes where the intersection tests takes long time (ie. with procedurally based objects) the benefit of using our approach would definitely become larger.

We have only tested the method in scenes with one light source as the current implementation of the algorithm is mostly suited for scenes with only few light sources. Many light sources will require large photon maps and more photons would have to be used in the estimates. In this situation it would probably be better just to visualize the photon map directly. Another approach could be to divide the scene into zones according to the light sources. A photon map generated by a specific light source would only generate photons in the light source zone. In this way the use of shadow rays in each zone could be restricted to the light sources belonging to the zone and the sampling of these light sources could be done based upon the information in the photon map within the zone.

A problem introduced with the shadow optimization scheme is the use of shadow photons. In the case of very complex scenes (ie. a tree) a significant amount of shadow photons might be created. In this case it would probably be better to use some kind of probabilistic spreading of the shadow photons — we could use a maximum of 10 shadow photons per ray and distribute these 10 shadow photons among the intersection points with the ray. This strategy is however quite complicated and it might introduce errors in the sampling scheme.

Due to the probabilistic nature of the method it might miss shadows from small objects. If an object is small and not intersected by any photon then the object wont create any shadow photons and this might result in a missing shadow. The only way to solve this is to emit larger number of photons but there is no guaranty that the photons will hit all objects — this is also the case in nature.

One benifit of the photon map is that is varies almost continuesly within the scene. This property is very nice when points along shadow edges are classified. When fixed structures are used to precompute visibility as in [Zimmerman-Shirley95] problems can occur along the shadow edges. The photon map is very flexible and errors along shadow edges can easily be avoided by using more photons to classify the point of interest. In this way a larger area will be considered. This also means that more shadow rays will be used since chances are that these larger areas contains both shadow photons and illumination photons. In general we have found that using only 20 photons is adequate for most situations.

CONCLUSION

In this paper we have demonstrated how the photon map can be used to provide significant reductions in the number of shadow rays used to compute the direct illumination. By introducing shadow photons we were able to detect regions around the shadow boundaries within the scene and restrict the use of shadow rays to these boundaries. In our test scenes this resulted in a reduction in the number of shadow rays of more than 90 %

REFERENCES

Collins, S.: "Reconstruction of Indirect Illumination from Area Luminaires". In "Rendering Techniques '95". Eds. P.M. Hanrahan and W. Purgathofer, *Springer-Verlag*, pp. 274-283, 1995

Haines, E., Greenberg, D.P.: "The Light-Buffer: A Shadow Testing Accelerator". *IEEE CG&A*, vol. 6, no. 9, pp. 6-16, Sep. 1986

Jensen, H.W., Christensen, N.J.: "Photon maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects". *Computers and Graphics* **19** no. 2, pp. 215-224, 1995

Jensen, H.W.: "Importance Driven Path Tracing using the Photon Map". In "Rendering Techniques '95". Eds. P.M. Hanrahan and W. Purgathofer, *Springer-Verlag*, pp. 326-335, 1995

Kajiya, J.T.: "The Rendering Equation". *Computer Graphics* **20** (4), pp. 143-149, 1986

Shirley, P., Wang, C.: "Direct Lighting Calculation by Monte Carlo integration". *Eurographics Workshop on Rendering* **2**, Barcelona 1991

Shirley, P., Wang, C., Zimmerman, K.: "Monte Carlo Techniques for Direct Lighting Calculations". To appear in *ACM Transactions on Graphics*.

Shirley, P., Wade, B., Hubbard, P., Zareski, D., Walter, B., Greenberg, D.P.: "Global Illumination via Density Estimation". In "Rendering Techniques '95". Eds. P.M. Hanrahan and W. Purgathofer, *Springer-Verlag*, pp. 219-230, 1995

Ward, G.J.: "Adaptive Shadow Testing for Ray Tracing". *Eurographics Workshop on Rendering* **2**, Barcelone 1991

Ward, G.J., Heckbert, P.S.: "Irradiance Gradients". *Eurographics Workshop on Rendering* **3**, Bristol 1992

Whitted, T.: "An Improved Illumination Model for Computer Graphics". *Comm. of the ACM*, vol. 23, no. 6, pp. 343-349, June 1980

Woo, A., Poulin, P., Fournier, A.: "A Survey of Shadow Algorithms". *IEEE Computer Graphics*, pp. 13-32, nov. 1990

Woo, A., Amantides, J.: "Voxel Occlusion Testing: A Shadow Determination Accelerator for Ray Tracing". *Graphics Interface 90*, pp. 213-220, May 1990

Zimmerman, K., Shirley, P.: "A Two-Pass Solution to the Rendering Equation with a Source Visibility Preprocess". In "Rendering Techniques '95". Eds. P.M. Hanrahan and W. Purgathofer, *Springer-Verlag*, pp. 284-294, 1995

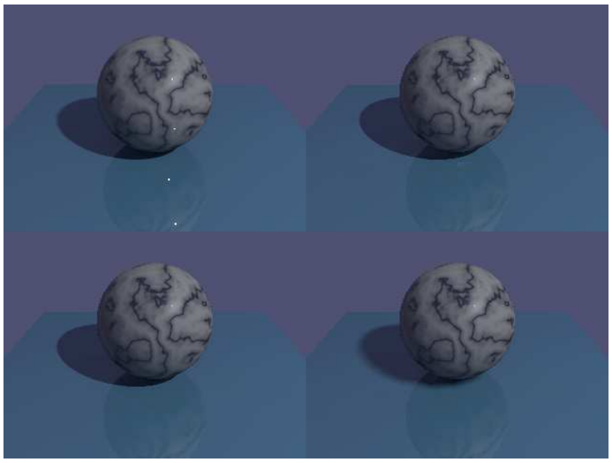

Figure 2: A marble sphere illuminated by a small spherical light source. 33.000 photons were used. Upper left: Path tracing reference (100 samples per pixel), upper right: Normal rendering with all shadow rays, lower left: Shadow rays are only used at shadow boundaries, lower right: No shadow rays only visibility estimate.

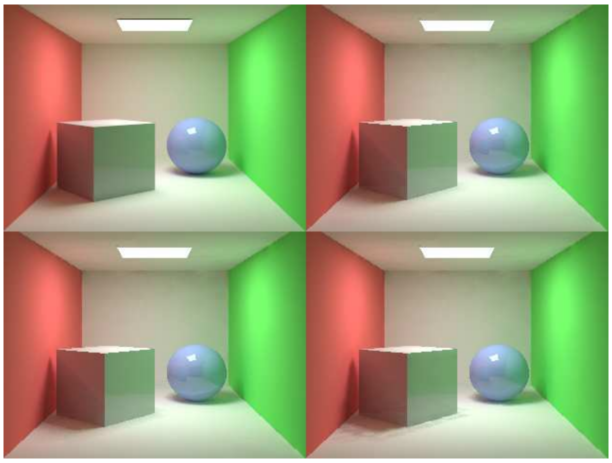

Figure 3: Box scene illuminated by a large area light source. 176.000 photons were used. Upper left: Path tracing reference (1000 samples per pixel), upper right: Normal rendering with all shadow rays, lower left: Shadow rays are only used at shadow boundaries, lower right: No shadow rays only visibility estimate.