Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects

Henrik Wann Jensen

Niels Jørgen Christensen

Dept. of Graphical Communication

Technical University of Denmark

fax: +45 45 93 83 17 phone: +45 45 93 41 66

email: igkhwj@unidhp.uni-c.dk or igknjc@unidhp.uni-c.dk

October 20, 1994

Abstract

This paper describes a bidirectional Monte Carlo ray tracing method simulating global illumination in models containing complex objects that do not have to be tessellated. The two pass method combines a first pass light ray tracing (ray casting) with a second pass optimized Monte Carlo ray tracing. In the first pass, the light emitted from the light sources hit objects in the scene and may be reflected or transmitted - a kind of backward path tracing. This step handles all kinds of reflections and not only the specular to diffuse reflections. This turns out to be a valuable optimization. At every object-interaction, energy is stored on the surface of the object. For simple objects an illumination map is used. For complex objects e.g. procedurally based objects like fractals, energy is stored in a photon map. This new concept makes it possible to treat caustics upon such objects without having to parameterize the surface of the objects. The second pass, Monte Carlo ray tracing from the eye, visualizes the scene based upon the result from the first pass. We use the irradiance gradient method to model diffuse reflections seen directly from the eye. All secondary reflections are taken from the photon maps or the illumination maps. Only the caustic part of the ray casting step is visualized directly.

Henrik Wann Jensen

Niels Jørgen Christensen

Department of Graphical Communication

Technical University of Denmark

Building 116, DK-2800 Lyngby, Denmark

1 Introduction

The traditional methods of global illumination: Ray tracing [1] and radiosity [2] do not account for all kinds of indirect illumination in a model. According to the L(S|D)*E classification of light paths from the light source, L, to the eye, E, via diffuse, D, or specular, S, reflections [3], ray tracing only models the LDS*E and LS*E paths. The indirect illumination L(S—D)+DE is only included as a constant ambient term. Radiosity only models LD*E and thus only diffuse surfaces and not mirrors.

Both methods have been extended into the primary domain of the other. By introducing directional form factors the radiosity method was extended to non-diffuse environment [4]. [5] used Monte Carlo techniques and extended ray tracing into a method called path tracing which in theory simulates all types of reflections. However, the radiosity extension uses a huge amount of storage and both methods require a substantial amount of computation time and they have not been of great practical value until now.

Ward et al. [6, 7] introduced a caching scheme which significantly reduced the computational effort of path tracing. This technique is however limited to diffuse surfaces and like path tracing, the method has difficulties simulating caustics adequately. [8] has solved this problem by using a deterministic calculation of caustics. However, this method is limited to polygon mirrors and it is very expensive in models with more mirrors.

The most successful methods today combine radiosity and ray tracing in order to exploit the best from both methods. Unfortunately it is not enough just to make a serial combination of the two methods in which the first pass is radiosity computing indirect diffuse reflections and the second pass is ray tracing visualizing the radiosity solution and computing specular reflections. Such a method would only simulate LD*S*E. Thus lacking for instance caustics. To solve this problem [9, 10, 11, 12] uses mirror form factors. However only [12] simulates caustics on non-Lambertian surfaces. Caustics are not simulated well using radiosity, and more advanced hybrid methods use ray tracing from the light sources (light ray tracing) as introduced by Arvo [13] to catch the high irradiance gradients often seen in caustics. Shirley [14] used light ray tracing to simulate the caustics visualized directly and then he used radiosity with mirror form factors to simulate indirect illumination. The solution is visualized using distributed ray tracing. Chen et al. [15] refined Shirleys method by adding path tracing to the visualization stage. Path tracing is used to calculate all reflections seen directly from the eye. All secondary diffuse reflections are calculated using radiosity. Path tracing is however very expensive to use even though only the primary reflections are calculated. Often a good solution requires 100 or more rays pr. pixel to simulate indirect diffuse illumination properly. This is very expensive when high resolution images are calculated.

Thus the newest hybrid methods can simulate all light paths, L(S|D)*E. However all these methods require that the model is discretized into a finite number of polygons. This discretization affects the final solution and it limits the kinds of objects that can be used with the methods. [15] eliminates the visible artifacts via the path tracing step. None of the methods can deal with procedural objects, like fractals, that are not tessellated. This has justified our development of a new hybrid method.

We want to develop an algorithm capable of simulating global illumination without any restrictions on the objects. The method should be able to efficiently simulate caustics and indirect illumination from all kinds of objects. By introducing photon maps we will also simulate caustics on objects that cannot be parameterized and therefore cannot be used in conjunction with illumination maps (texture maps). We would also like the method to be able to deal with arbitrary bidirectional reflectance distribution functions, BRDF's, like [16, 17]. It should however be more efficient than the classic Monte Carlo methods and not suffer from noisy results and bias on the solution.

2 Bidirectional Monte Carlo ray tracing

Our main goal is to set up a method that does not require tessellation of the objects in the model. Therefore we cannot use the radiosity method. Instead we extend the pure two pass ray tracing technique to account for all kinds of reflections. Thus substituting the radiosity approach used in the newer hybrid methods. Our work has been inspired by the works of [3, 5, 6, 7, 13] and [15].

The idea behind the two pass method is very simple, most Monte Carlo ray tracing methods use most of the time to find the way from the eye to the light sources. Our methods eliminates this expensive search by emitting light from the light sources and then collecting the result by a more efficient Monte Carlo method during visualization.

The first pass is Light Path Tracing, LPT, in which rays from different points on each light source is emitted in accordance with the distribution of emitted energy from the light source. Each light ray is traced through the scene until it is either absorbed or it does not hit any objects. Our LPT approach differs from previous ray casting methods in that it includes diffuse reflections as well as specular reflections. This means that not only caustics are simulated in the ray casting step but in fact a complete (but rough) solution to the rendering equation is produced. Each time the light ray hits an object it deposits energy into the irradiance map on that object. We use the term irradiance map to cover the two kinds of maps that we use. These are illumination maps as introduced by Arvo

[13] and photon maps which are a new more general way of storing light energy on a surface. The quality of the solution in the irradiance maps is not adequate for display and similar to [15] we only use the irradiance maps when calculating secondary reflections. There is one important exception however, the caustics part of the irradiance maps is visualized directly to the eye, and this solution has to be calculated more carefully. The second pass of the method visualizes the model using Monte Carlo ray tracing optimized with the irradiance gradient method [7] and the use of irradiance maps.

Figure 1 illustrates the approach and in the following sections we will describe the two passes in more detail. Notice that we use the term reflection in order to avoid having to write reflection/transmission everywhere — there is nothing in our approach restricting the use of transmission.

HERE IS FIGURE 1

3 Light path tracing — pass 1

In the light path tracing step rays are emitted from different positions on each light source. The number of positions are determined by the size of the light source compared to the dimensions of the scene. At each position a projection map is created. It is a map over the positions of specular and diffuse objects in the model relative to the light source position (see section 3.1). Based on the information in the projection map, rays are emitted into the scene using stratified sampling. If a given direction in the projection map only contains diffuse objects then a certain number of rays are emitted in that direction. If a direction contains specular objects then the number of rays emitted are raised based upon the specularity of the objects seen.

Each ray carry a fraction of the light source energy. This energy is deposited into the irradiance maps as the ray is traced through the scene. Every time the ray hits an object the energy carried by the ray is deposited into the irradiance map at the given location. Then it is decided whether the ray is absorbed or reflected. The approach used is Russian roulette as described by Arvo et al. [18]. This technique helps us avoid bias and at the same time avoid having to trace a ray through an infinite number of reflections. If the ray is reflected we use the BRDF of the surface to generate a direction of the reflected ray.

3.1 Projection maps

The projection map is inspired by the light buffer [19]. It is used to localize specular as well as diffuse surfaces. [14] used feeler rays to solve this problem, however this method might neglect small but important specular objects and edges of mirrors could also risk being ignored. The latter problem was solved by allowing neighbour areas to behave like specular surfaces. We have decided to use an explicit approach to find the important objects in the scene at the cost of a more complicated method.

The projection map is a two dimensional (θ, ϕ) map of the hemisphere at the position on the light source. In this map the directions containing specular and diffuse surfaces are recorded. Unfortunately the projection of objects onto a hemisphere is not simple. Therefore we use the bounding sphere of the object in the construction process. The projection of a sphere onto a sphere results in a circle. However in the (θ, ϕ) space this is no longer the case. We can however use the maximum radius of the circle and based on this information fill information about the object in the correct places in the projection map (see figure 2). This approach results in a clear overestimation of the solid angle of the objects relative to the light source position but most important, nothing is missed.

HERE IS FIGURE 2

3.2 Illumination maps

Illumination maps as introduced by Arvo [13] are texture maps with irradiance information. Arvo only treated the irradiance caused by caustics on diffuse surfaces. We would however like to treat all kinds of irradiance at a surface. Since only caustics are visualized directly we use two illumination maps, one storing the irradiance that is not part of a caustic (has been reflected diffusely once) and another storing the caustics. We name the two maps the diffuse illumination map, DI, and the caustic illumination map, CI. The reason for using two maps is the fact that CI requires high precision while DI can use a much lower precision since it is not seen directly by the eye.

The resolutions of CI and DI are determined using two different approaches. The resolution of DI is based on the relative size with respect to the light sources in the scene while the resolution of CI is determined by a simple pretracing of the scene. It is necessary to pretrace the scene in order to determine the distance between rays going through adjacent pixel and striking the same object. The size of each element in CI should be lower than the smallest of these distances in order to avoid artifacts resulting from elements being to large in the caustics map.

In general the resolution of DI is much lower than CI. Normally CI contains 25-100 times the number of elements in DI. In fact the resolution of CI is a little problematic since these very large texture maps requires a lot of memory. We avoid this problem by not creating the caustic maps until caustics are actually occuring at specific positions. We do this by applying a mini caustic map to each element in DI. Only when an element in DI is illuminated by a caustic is CI created. This is shown in figure 3.

HERE IS FIGURE 3

3.3 Photon maps

Illumination maps cannot be used with all objects. They require that it is possible to describe the surface of an object parametrically. Furthermore, one should be able to determine the area on the surface of the objects corresponding to elements in the illumination map. This can lead to quite complex calculations. For certain kinds of objects like e.g. fractals it is very seldom possible to describe the surface properly.

To solve this problem we introduce a new way of representing irradiance on a surface. For objects which cannot have their surface parameterized we store every photon striking the object. We use two classifications of photons, CP, are those photons that are part of a caustic, DP, are the rest of the photons. With every photon we store the intersection point, the normal, the energy and a bit saying whether this photon is part of a caustic. Such a similar naive approach was rejected by Pattanaik in [20] because of the storage and computational requirements. However, by limiting the photon maps to special objects and using an optimized representation we can make the approach usable. We can limit the storage requirements by each photon to 28 bytes (12 bytes for the irradiance position, 3 bytes for the normal, 3 floats for the enerrgy packed as 4 bytes using Wards real pixels [21], a byte containing the flags and finally two pointers to the remaining data-structure). We store the photons in a multidimensional search tree, kd-tree [22], and this allows relatively efficient look up and storing of the photons. Alternatives like Voronoi diagrams could have been used as well.

A problem that has to be solved is determining the irradiance from the photon distributions. We have chosen a simple and relatively efficient approach. Consider a point x at which we are interested in the irradiance. Around x we create a sphere. The radius of this sphere is extended until the sphere contains n photons and has radius r. The number n is a parameter to the algorithm. We use the area of a disc with radius r as an estimate of the area holding the photons. This leads to the following formula for calculating the irradiance at x:

$$E_x = \frac{\sum_{i=1}^{n} e_i}{\pi r^2}$$

where e_i is the energy belonging to photon no. i. A simple rejection scheme has been applied in order to avoid using wrong photons in the calculation. Photons are only accepted if the dotproduct between their normal and the normal at x is positive.

In the photon map we distinguish between photons giving rise to caustics and other photons. When estimating the irradiance due to caustics we only use the photons that contribute to the caustics irradiance. When estimating the irradiance in general we use all the photons in the map.

The irradiance estimation has one drawback. If the number n is large sharp caustic edges becomes blurry. To avoid this we have observed the following: When adding photons to the estimate and being near an edge the changes of the estimate will be monotonic. That is, if we are just outside a caustic and we begin to add photons to the estimate, then it can be observed that the value of the estimate is growing as we add more photons. On this background we have added differential checking to the estimate which means that we stop adding photons and use the estimate available, if we observe that the estimate is either constantly increasing or decreasing as more photons are added. In this way we avoid blurring the edges.

4 Visualization — pass 2

In the second pass the visualization is carried out. This step requires high precision since the result is visualized directly to the eye.

The model is visualized with Monte Carlo ray tracing which reduces aliasing but unfortunately give rise to noise. The radiance, L_p, through each pixel is determined by shooting one or more rays from the eye through the pixel. If a ray hits an object then the sum of the radiance reflected, L_r, and the radiance emitted, L_e, by the object in that direction is the radiance, L_p, at the pixel:

$$L_p = L_e + L_r$$

The component giving rise to noise in Monte Carlo ray tracing is the term L_r. The noise can be effectively reduced by separating L_r in a diffuse-like component,

$L_{r,d}$, and a specular-like component, $L_{r,s}$:

$$L_r = L_{r,d} + L_{r,s}$$

Notice that this technique is not restricted to ideal diffuse and ideal specular reflection. The reason for splitting L_r is that $L_{r,s}$ is highly directional while $L_{r,d}$ is only weakly directional. $L_{r,s}$ is estimated by recursive Monte Carlo ray tracing. This method is not convenient for $L_{r,d}$ which we instead split in three new components:

$$L_{r,d} = L_{r,d,l} + L_{r,d,s} + L_{r,d,d}$$

where $L_{r,d,l}$ is the light coming from the light sources, $L_{r,d,s}$ is light coming from specular reflection and $L_{r,d,d}$ is light resulting from multiple diffuse reflections. The calculation of each of these terms is discussed in the following sections.

4.0.1 $L_{r,d,l}$

The direct contribution from the light sources is in principle easily determined. However, it can be very time consuming for scenes with many light sources. [23] have optimized this by using a fixed number of shadow rays while [24] uses an adaptive light sampling approach. Both of these methods can be used here. Currently we use simple stratified sampling of each light source.

4.0.2 $L_{r,d,s}$

$L_{r,d,s}$ can be two kinds of light. $L_{r,d,s}$ can either be caustics meaning that we only see one diffuse surface followed by a number of specular surfaces and ending up at a light source. This component is determined directly from the caustics part of the irradiance maps.

$L_{r,d,s}$ can also contain several diffuse reflections following the specular reflection and in this case we use the same approach as described in the following section where the calculation of $L_{r,d,d}$ is explained.

4.0.3 $L_{r,d,d}$

This part of the solution originates from rays which have at least two diffuse reflections. Consequently $L_{r,d,d}$ is soft and only change slowly. This means that the irradiance gradient method, IG, by Ward et al. [7] is well suited for calculating $L_{r,d,d}$. However we have made one important change to IG: All secondary diffuse reflections are fetched directly from the irradiance maps. Specular surfaces and surfaces with complex BRDF's are treated using plain recursive Monte Carlo ray tracing.

5 Implementation and test scenes

The method has been implemented in a program written in ANSI C++. The program has been tested on a PC with respectively Linux and MS-DOS and on a Silicon Graphics Indigo with IRIX v4.01. The memory requirements for normal scenes are less than 16MB. The following results has been produced on a PC 486DX2-66 with 32MB running Linux. All the images have the resolution 640x480 and each pixel has been sampled adaptively based on the variance of each sample. An average of 4 samples pr. pixel is used.

The implementation has been limited to include objects with specular, rough specular and lambertian properties. Furthermore only spherical and rectangular light sources have been implemented.

In order to test the methods three test scenes have been used: Figure 4 shows the first scene which is a desk illuminated by two light sources. The desk is placed against a white wall and on the desk we find some dices, a glass containing a red liquid, and a marble block on which there is a sphere flake. The sphere flake is a procedurally created object which in our case consists of approximately $5 \cdot 10^5$ spheres. The first of the two light sources is an adjustable lamp on the table while the second is fixed above the table. The wood and the marble textures have been created using solid texturing [25] and the dices have been created using constructive solid geometry.

HERE IS FIGURE 4

HERE IS FIGURE 5

HERE IS FIGURE 6

Test scene 2, see figure 6, shows a situation in which only photon maps can be used. The model contains a procedurally generated fractal landscape illuminated by a light source. The model also contains a flat lens-shaped sphere which focuses the light from the light source slightly giving rise to a caustic on the landscape. The landscape represents a discrete sampling of Perlin's turbulence function [25]. The normal is calculated using smoothing which softens the looks of the surface. Consequently it is not immediately possible to assign irradiance values to the surface using illumination maps.

The third scene (illustrated in figure 7) is used to compare the caustics produced by using photon maps, illumination maps and a deterministic method.

HERE IS FIGURE 7

6 Results and discussion

The image in figure 4 (illumination map case) has been produced using illumination maps on all objects except for the sphere flake on which we used photon maps. The image demonstrates the following important visual effects. The light is transmitted through the red liquid and the glass and is giving rise to a caustic on the table. The wall is not perfectly white. It is shaded by light reflected from the table resulting in colour bleeding on the wall. The marble block below the sphere flake is affected by the sphere flake and is coloured blue.

Figure 5 shows a picture of exactly the same scene as figure 4. Here we use photon maps on all objects (photon map case). This was done to compare the visual impression obtained with the two techniques. As one can see the images are almost identical. In the photon map case we can observe a few bright spots near the edge of the desk which is due to the fact that we only used 40 photons in the irradiance estimate. This number is a little to low but on the other hand we can see that the rest of the scene is without any visible artifacts. In the illumination map case we emitted 3 times as many light rays as in the photon map case. This was necessary since our implementation of the illumination map uses a box-kernel to deposit energy. Therefore we had to use a rather large number of rays in order to reduce the noise in the image to an acceptable level. A more advanced kernel combined with perhaps wavefront-tracking (See [26]) could very likely help reducing the number of rays in the illumination maps case significantly.

One of the main purposes of the work was to handle caustics on complex surfaces like fractals. The image of the landscape in figure 6 shows the resulting caustic when using photon maps. Notice that the illumination of the landscape within the caustic is similar to the direct illumination of the rest of the landscape. The irradiance in each point was estimated using 80 photons. Test scene 2 illustrates one major advantage of using the photon maps. We can handle objects which would otherwise require tessellation. The landscape could be triangulized and then one could assign an irradiance value to each triangle. This solution is not always possible and in many situations this also leads to a very huge number of triangles which could cause memory problems. If the number of triangles created is large the solution would also require a large number of light rays in order to get a satisfying result in the diffuse part of the solution, and the caustic part of the solution would suffer from noise in the same way as the use of the box-kernel in the illumination map. One interesting possibility is however examining if it is possible to deposit energy onto a triangle mesh using more refined kernels. This would make the triangulized situation more tractable.

In order to test the concept of photon maps against the illumination maps in more detail, we created the model illustrated in figure 7. It contains a square with a Lambertian surface and an mirror square. The two squares are illuminated by a small light source which results in caustics on the Lambertian square. This model can be treated with both photon maps and illumination maps but it can also be handled using exact calculations. Using the three techniques we sampled the irradiance at discrete points along a line through the caustic area.

HERE IS FIGURE 8

The results are shown in figure 8. The first graph shows the results from the illumination maps versus the exact result. The illumination maps gives a rather noisy result and this noise is visible by the eye. As stated earlier this noise is due to the simple box-kernel used to deposit energy in the illumination map. The second graph shows the results obtained using photon maps with 200 photons pr. estimate with and without the differential checking. The result using the photon maps can in this situation clearly compete with the illumination map. The noise has a high frequency and it cannot be detected easily by the eye. Notice also that the edge detection improves when differential checking is used. Using fewer photons to estimate the irradiance results in a more noisy graph; more photons improve the graph but the result takes longer time to compute.

HERE IS FIGURE 9

Some timings and statistics from the calculation of the different images have been collected in the table in figure 9. Here we notice that the illumination of the landscape was calculated using 341.000 photons. The caustic consists of approx. 150.000 photons. The time it took to calculate the image was primarily affected by the number of photons used in the irradiance estimate and our experiments have shown that the difference in computation time is generally linear in the number of photons used. That is the time it takes to calculate the irradiance estimate is linearly dependent on the number of photons used.

The images in figure 4 and 5 used 4 hours and 5 hours respectively. Notice that this statistics is based on the same number of light rays in the two pictures. Therefore we can conclude that the 25% extra time spent on figure 5 is due to the calculations in the photon map. This extra time is mostly spent in locating the photons. We use a rather primitive search through the kd-tree. Furthermore, the kd-tree is build on the fly during the LPT-pass which means that it is not equally weighted. This also affects the time it takes to locate the photons. A caching scheme combined with a more efficient code should definitely improve the timings.

The number of rays used by the irradiance gradient method in all the images is quite low. In figure 6 this is caused by the fact that there is very little indirect diffuse illumination in the model but for test scene 1 this is no longer true. Here the low number of IG-rays is caused by the optimization in which all secondary reflections are taken directly from the irradiance map. During our implementation of the method we observed a significant speed up due to this. Furthermore, the rather expensive storage of irradiance values with irradiance gradients was also reduced significantly. The quality of the irradiance solution was in some cases even improved with the irradiance maps since the quality of the secondary reflections are not lowered as it is generally the cases when using recursive Monte Carlo sampling.

The number photons used in figure 5 is rather low but still the result is good since the photons are concentrated around the difficult areas, like the caustic from the glass. It is the projection map that enables this automatic concentration of photons. We intend to improve this step using a progressive shooting algorithm in which the photons are concentrated around areas with high gradients etc. This would probably also make the use of even fewer photons possible.

One significant time-consumer in figure 4 and 5 is the light source sampling. As one can see from the table the shadow rays account for approx. 90 % of the rays used. This number is quite difficult to reduce without loss in image quality. In this case there are only two light sources which have a finite size. These light sources are sampled using multiple shadow rays pr. source and the quality of the soft shadow on the wall depends strongly on the number of samples used. It is not enough to restrict the number of samples to a given lower number since this affects the quality of the shadow.

We have tried to solve the problem of sampling the light sources during visualization by extending the information in the illumination map. Every ray that has it's origin at a light source is special. We find all intersections between this ray and the objects in the scene. In the irradiance maps these intersections are characterized as direct hits or shadow hits, in which the direct hit is the intersection nearest the light source. The total number of the two kinds of hits is then used to determine whether it is necessary to sample the light source or not. E.g. if an area in the irradiance map has 10 shadow hits and no direct hits then we consider this area to be fully shadowed. Shadow rays are only used in cases where we find shadow hits and direct hits or when the number of hits is to low. Unfortunately testing this method revealed that small artifacts would appear at the shadow edges and these artifacts were highly noticeable. Therefore we stopped developing the method, which would not work with photon maps either. However optimization of the shadow computation is definitely needed. Currently, we use BSP-trees during the Monte Carlo ray tracing step and this makes the use of the simple light buffer somewhat obsolete.

The idea of storing extra information in the irradiance map especially the photon map is from our point of view very interesting. One could imagine storing the direction from which the photon came. This would give information that would be usable in surfaces with complex BRDF's. If we assume that the n photons found hit the same point, then we can use the photon values instead of sampling the BRDF. This technique could perhaps lead to a less storage intense method when simulating caustics on surfaces with complex BRDF's compared to existing methods based on spherical harmonics and wavelets.

Another potential speed up is using the estimate from the photon maps as an initial guess when sampling the indirect illumination and perhaps also when sampling the light sources. A good estimate would lead to fewer samples being necessary.

The photon maps can also be useful when testing new advanced objects since the only requirements to the object is the ability to intersect the object with a line and determine the normal at the intersection point.

In general the irradiance estimate from the photon map is good considered the simplicity. We have observed problems estimating the correct irradiance at edges. These are not serious faults. They are not correct but they give softer edges which somehow are quite pleasing to the eye. Currently we give uniform weight to the photons. It would probably be better to weigh the photons using a kernel similar to the ones observed with illumination maps. For instance one could expect a Gaussian distribution to give good results, but it is not simple to implement since in theory the photons can be spread throughout the entire sphere which is used to locate the photons in the irradiance estimate.

The current implementation is not very good at scenes with many light sources. In step 1 we have to emit light rays from each light source which will be very time consuming in a room with say 1000 light sources. However in many cases each light source should not have to emit a large number of rays since the light information in such a scene would often be concentrated around each individual light source.

The method also has the same drawback like many of the previous multipass methods in that it does not provide any intermediate feedback as [15]. However some preliminary information could easily be provided by a simple visualization of the information in the irradiance maps perhaps via simple ray tracing of the model.

The timings given indicate that the time it takes to obtain a good solution is measured in hours on a 486DX2-66 PC. If we compared our method with radiosity in a model consisting primarily of triangles then radiosity would clearly be the faster one. The new hierarchical radiosity techniques and also the new clustering techniques are indeed very fast. However our method has one major advantage. It can calculate global illumination in a model containing advanced objects without tessellating any of these.

7 Conclusions and Future Work

We have presented a new Bidirectional Monte Carlo ray tracing method for Global Illumination. The method has one significant improvement over previous methods which is the ability to handle caustics on complex geometries like fractals without any prior tessellation of the objects. The tool for this extension is the concept of photon map which store energy on such complex objects.

The method is also quite efficient even though it is based on Monte Carlo techniques. Furthermore, the images do not contain any artifacts from tessellation and they are not noisy as the images produced by pure path tracing techniques. This is due to the intense use of different importance sampling techniques. Here, one major improvement is the extension of the illumination (and photon) maps with diffuse reflected light. This leads to a solution which - even though it is rough - is so good that we do not have to sample diffuse light recursively. Instead all secondary non-specular reflections are determined directly from the maps (including the contribution from the light sources). This leads to a significant speed up without loss in image quality - in fact image quality is often improved due to the fact that secondary reflections are often calculated with less accuracy.

The emission of light from the light sources can also be considered as an importance sampling of the model since objects that are very bright and thereby important are in most cases near the light sources. These objects will receive more light beams than more distant objects and they are in this way provided with a finer irradiance map.

However, the current implementation is young. Currently we use a lot of computational effort on scenes with complex light sources. We have tried to improve this by adding information to the irradiance maps. The results were unfortunately to noisy.

Currently we try to improve the treatment of surfaces with a complex BRDF. We are also investigating whether we can improve the irradiance gradiant method by reusing information from the irradiance maps.

References

[1] Turner Whitted, An Improved Illumination Model for Computer Graphics. *Comm. of the ACM* **23** (6), 343-349 (1980).

[2] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg and Benneth Battaile, Modeling the Interaction of Light Between Diffuse Surfaces. *Computer Graphics* **18**, 213-222 (1984).

[3] Paul S. Heckbert, Adaptive Radiosity Textures for Bidirectional Ray Tracing. *Computer Graphics* **24** (4), 145-154 (1990).

[4] David S. Immel, Michael F. Cohen and Donald P. Greenberg, A Radiosity Method for Non-Diffuse Environments. *Computer Graphics* **20** (4), 133-142 (1986).

[5] James T. Kajiya, The Rendering Equation. *Computer Graphics* **20** (4), 143-149 (1986).

[6] Gregory J. Ward, Francis M. Rubinstein and Robert D. Clear, A Ray Tracing Solution for Diffuse Interreflection. *Computer Graphics* **22** (4), 85-92 (1988).

[7] Gregory J. Ward and Paul S. Heckbert, Irradiance Gradients. *Global Illumination. ACM SIGGRAPH Course Notes* **18** (1992).

[8] Gregory J. Ward, The RADIANCE Lighting Simulation System. *Global Illumination. ACM SIGGRAPH Course Notes* **18** (1992).

[9] John R. Wallace, Michael F. Cohen and Donald P. Greenberg, A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods. *Computer Graphics* **21** (4), 311-320 (1987).

[10] François Sillion and Claude Puech, A General Two-Pass Method for Integrating Specular and Diffuse Reflections. *Computer Graphics* **23** (3), 335-344 (1989).

[11] Holly E. Rushmeier and Kenneth E. Torrance, Extending the Radiosity Methods to Include Specularly Reflecting and Translucent Materials. *ACM Transactions on Graphics* **9** (1), 1-27 (1990).

[12] François X. Sillion, James R. Arvo, Stephen H. Westin and Donald P. Greenberg, A Global Illumination Solution for General Reflectance Distributions. *Computer Graphics* **25** (4), 187-196 (1991).

[13] James Arvo, Backward Ray Tracing. *Developments in Ray Tracing. ACM Siggraph Course Notes* **12**, 259-263 (1986).

[14] Peter Shirley, A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes. *Proc. Graphics Interface*, 205-212 (1990).

[15] Shenchang Eric Chen, Holly E. Rushmeier, Gavin Miller and Douglas Turner, A Progressive Multi-Pass Method for Global Illumination. *Computer Graphics* **25** (4), 165-174 (1991).

[16] Gregory J. Ward, Measuring and Modeling Anisotropic Reflection. *Computer Graphics* **26** (2), 265-272 (1992).

[17] Xiao D. He, Kenneth E. Torrance, François X. Sillion and Donald P. Greenberg, A Comprehensive Physical Model for Light Reflection. *Computer Graphics* **25** (4), 175-186 (1991).

[18] James Arvo and David Kirk, Particle Transport and Image Synthesis. *Computer Graphics* **24** (4), 53-66 (1990).

[19] E. A. Haines and Donald P. Greenberg, The Light Buffer: A Shadow-Testing Accelerator. *IEEE Computer Graphics and Applications* **6** (9), (1986).

[20] S. N. Pattanaik, Computational Methods for Global Illumination and Visualisation of Complex 3D Environments. *Ph.d. Thesis submitted to Birla Institute of Technology & Science* (1993).

[21] Greg Ward, Real pixels. In *Graphics Gems II*, James Arvo (ed.), Academic Press, 80-83 (1991).

[22] Jon Louis Bentley and Jerome H. Friedman, Data Structures for Range Searching. *Computing Surveys* **11** (4), 397-409 (1979).

[23] Peter Shirley and Changyaw Wang, Luminaire Sampling in Distribution Ray Tracing. *Global Illumination. ACM Siggraph Course Notes* **18** (1992).

[24] Gregory J. Ward, Adaptive Shadow Testing for Ray Tracing. *Eurographics Workshop on Rendering* **2** (1991).

[25] Ken Perlin, An Image Synthesizer. *Computer Graphics* **19** (3), 287-296 (1985).

[26] Steven Collins, Adaptive Splatting for Specular to Diffuse Light Transport. *Eurographics Workshop on Rendering* **5**, (1994).

Direct illumination

$$\underbrace{\mathrm{L} \longleftarrow \mathrm{D} \longleftarrow \mathrm{S}^* \longleftarrow \mathrm{E}}_{\mathrm{MCRT}}$$

Caustics

$$\underbrace{\mathrm{L} \longrightarrow \mathrm{S}+ \longrightarrow \overbrace{D}^{\mathrm{CI/CP}}}_{\mathrm{LPT}} \underbrace{\longleftarrow \mathrm{S}^* \longleftarrow \mathrm{E}}_{\mathrm{MCRT}}$$

Highlights

$$\underbrace{\mathrm{L} \longleftarrow \mathrm{S}+ \longleftarrow \mathrm{E}}_{\mathrm{MCRT}}$$

Multiple diffuse reflections

$$\underbrace{\mathrm{L} \longrightarrow \overbrace{(\mathrm{S}|\mathrm{D})^* \longrightarrow \mathrm{D}}^{\mathrm{CI/CP} \;\&\; \mathrm{DI/DP}}}_{\mathrm{LPT}} \underbrace{\longleftarrow \mathrm{S}^* \longleftarrow \mathrm{D}}_{\mathrm{IG}} \underbrace{\longleftarrow \mathrm{S}^* \longleftarrow \mathrm{E}}_{\mathrm{MCRT}}$$

Figure 1: The 4 combinations of lightreflections used. The figure illustrates the directions in which the light is traced/shoot and the solution technique employed. MCRT = Monte Carlo Ray Tracing, LPT = Light Path Tracing, IG = Irradiance Gradient method.

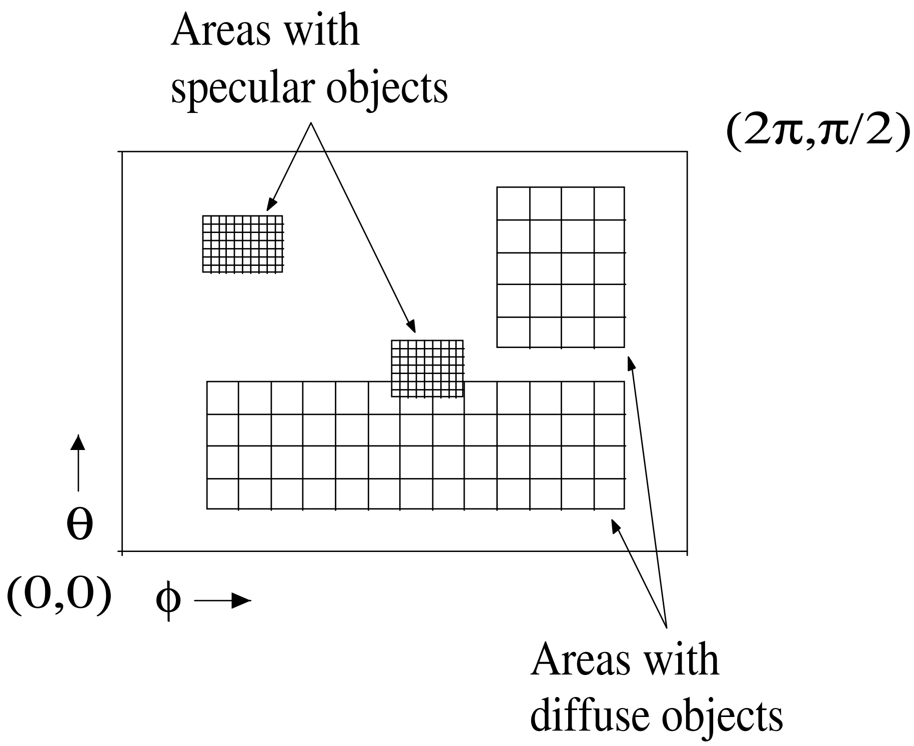


Figure 2: The projection map is filled with information about the surface characteristics of the objects around the light source position.

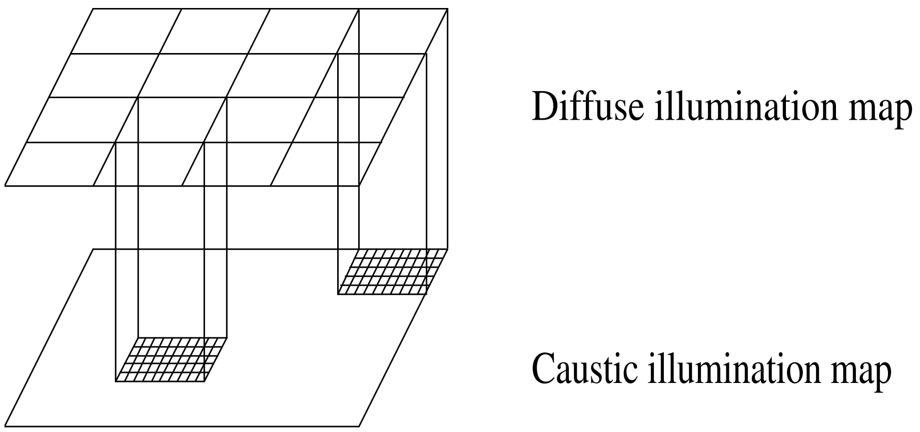


Figure 3: The figure shows how the illumination map is divided into two levels, one for caustics and one for "diffuse" illumination.

COLOUR IMAGE 1

Figure 4: Desk (with illumination maps).

COLOUR IMAGE 2

Figure 5: Desk (photon maps only).

COLOUR IMAGE 3

Figure 6: Fractal landscape with caustics.

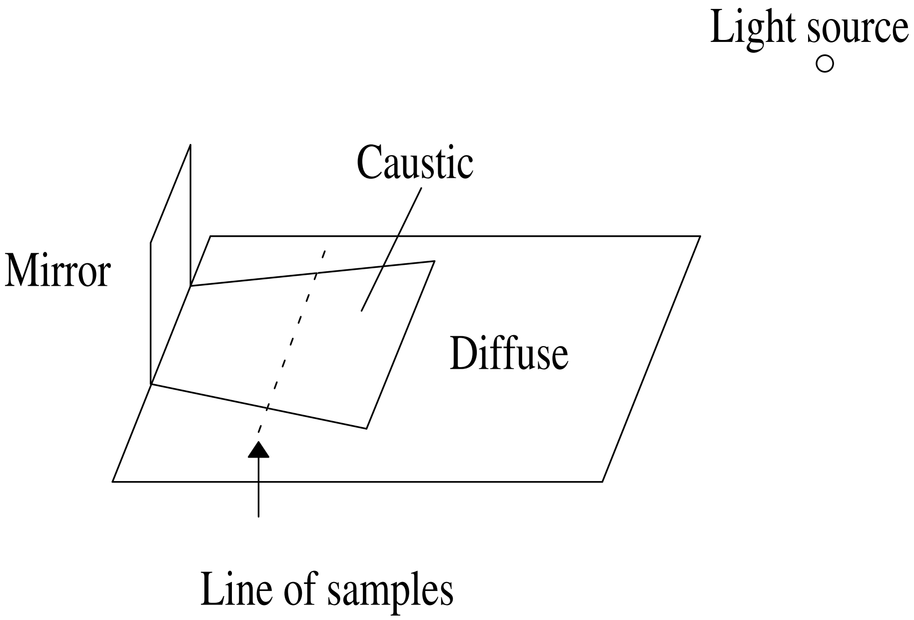
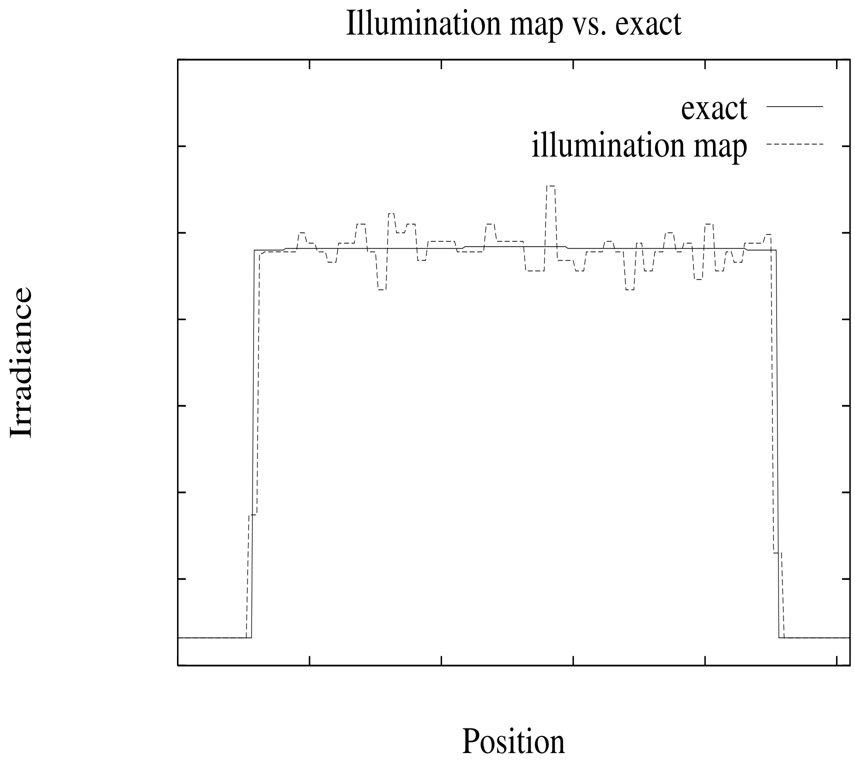


Figure 7: Model used to test illumination maps and photon maps versus a deterministic solution.



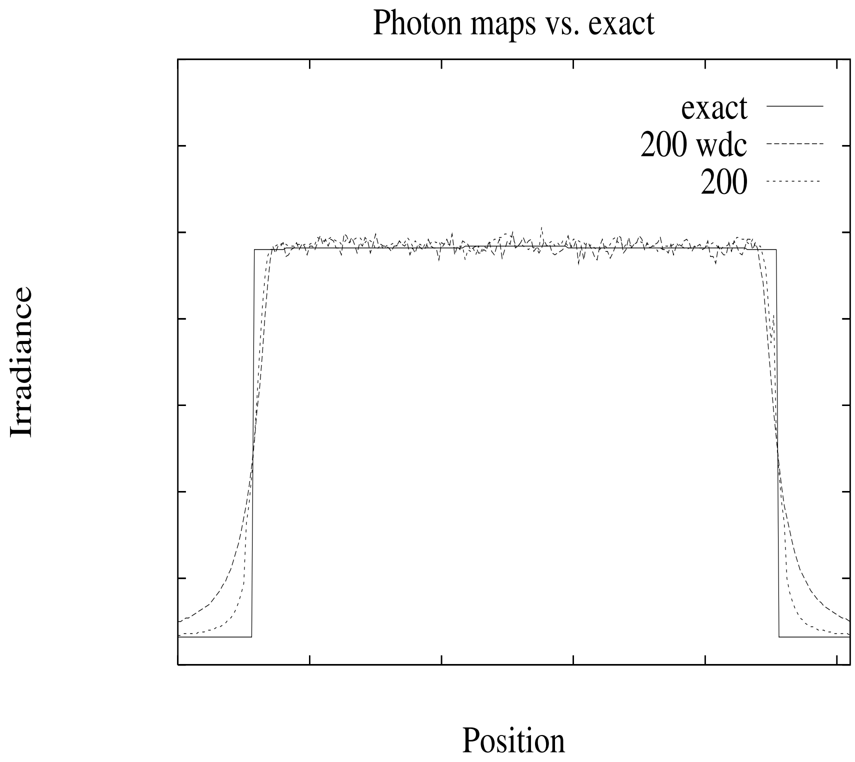


Figure 8: The graphs show the measured irradiance level through a caustic. The top graph shows the result using illumination maps versus an exact solution. The bottom graph shows the result using photon maps versus an exact solution. 200 photons were used in the irradiance estimation. The "200 wdc" version of the photonmaps is without differential checking.

Image	Photons Stored	LPT Rays	IG Rays	Shadow Rays	LPT secs.	Total time hh:mm
Fig. 4†	8253	226254	670447	$29 \cdot 10^6$	240	4:02
Fig. 5	90839	226254	697290	$29 \cdot 10^6$	229	4:53
Fig. 6	341652	$1.1 \cdot 10^6$	29808	721574	585	3:22

† Same number of rays in LPT as figure 5

Figure 9: Timings and statistics