

This paper is a slightly extended version of the paper in *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 21–30, 1996

Global Illumination using Photon Maps

Henrik Wann Jensen

Department of Graphical Communication
The Technical University of Denmark
hwj@gk.dtu.dk, <http://www.gk.dtu.dk/~hwj>

Abstract

This paper presents a two pass global illumination method based on the concept of photon maps. It represents a significant improvement of a previously described approach both with respect to speed, accuracy and versatility. In the first pass two photon maps are created by emitting packets of energy (photons) from the light sources and storing these as they hit surfaces within the scene. We use one high resolution caustics photon map to render caustics that are visualized directly and one low resolution photon map that is used during the rendering step. The scene is rendered using a distribution ray tracing algorithm optimized by using the information in the photon maps. Shadow photons are used to render shadows more efficiently and the directional information in the photon map is used to generate optimized sampling directions and to limit the recursion in the distribution ray tracer by providing an estimate of the radiance on all surfaces with the exception of specular and highly glossy surfaces.

The results presented demonstrate global illumination in scenes containing procedural objects and surfaces with diffuse and glossy reflection models. The implementation is also compared with the Radiance program.

Key words: Global Illumination, Photon Maps, Monte Carlo Ray Tracing

1 Introduction

Simulating global illumination in general environments is a complex task. Currently the most successful approaches combine radiosity and ray tracing [24, 5, 34]. Even

though ray tracing has been extended with Monte Carlo techniques [14, 19, 27, 29, 32] and radiosity has been extended with directional capabilities [13, 26, 3, 11, 6] neither of the two methods precludes the use of the other. In general Monte Carlo ray tracing is very time consuming and gives noisy results while radiosity uses a lot of memory to store directional information and it cannot handle specular reflection properly.

Most radiosity implementations use a simplified ray tracing algorithm to render the result in order to simulate specular reflections seen by the eye [28, 26, 25, 6]. Shirley [24] noticed that the ray tracing method could be used to render shadows as well since radiosity has problems at discontinuities. He also introduced the use of light ray tracing [1] to render caustics. Chen et al. [5] went even further and used path tracing to render all diffuse reflections seen directly by the eye in order to eliminate all visible artifacts from the radiosity algorithm. They only used the radiosity algorithm to model soft indirect illumination. Rushmeier et al. [22] concluded that the radiosity solution could be simplified since the path tracing algorithm would hide most of the artifacts in it and they introduced geometric simplification in which the radiosity algorithm is performed on a simple geometric approximation of the original model. Their motivation was the fact that radiosity becomes very time and memory consuming as the number of surfaces in the model grows.

This paper introduces a two pass method in which we simplify the representation of the illumination instead of simplifying the geometry. We obtain this simplification by using the photon map introduced in [15]. We combine the extensions to the photon map presented in recent papers [16, 17, 18] in order to render the scene more efficiently. The photon map is used to generate optimized sampling directions, to reduce the number of shadow rays, to render caustics and to limit the number of reflections traced as the scene is rendered with distribution ray tracing.

2 Overview of the Method

The first pass in the method is constructing the photon map by emitting photons from the light sources in the model and storing these in the photon map as they hit surfaces. The result is a large number of photon hits stored within the scene. This information can be seen as a rough representation of the light within the model.

Ward [29, 32] uses a comparable strategy storing irradiance values at surface points. Our approach does however differ significantly in several aspects. The creation of the photon map is light driven and it supplements the eye-driven rendering step very well. Effects like caustics that are very difficult to compute using traditional Monte Carlo ray tracing are easily obtained with the photon map. Furthermore we store incoming flux (photons) which is much simpler and less accurate

than irradiance values. Our motivation for doing so is that we obtain a very flexible environment with a lot of useful information that can be applied in the rendering step. The use of photons allows us to estimate surface radiance at surfaces with arbitrary BRDF's. The information can also be applied in an unbiased fashion to optimize the rendering step. The photons can be used to compute improved control variates [20] or as demonstrated in [16] to generate optimized sampling directions.

Our rendering engine is a distribution ray tracer in which rays are traced from the eye into the scene. The information from the photon map is applied during rendering in two different ways. We distinguish between situations where we need an accurate computation and situations in which an approximate estimate can be applied. As the rays are traced through several reflections their contribution to the final pixel radiance becomes lower and we apply the approximate estimate which for all surfaces equals a radiance estimate obtained from the photon map. For highly glossy surfaces we do however trace additional sample rays since reasonable radiance estimates for these surfaces require a large number of photons. The accurate computation is applied at surfaces seen directly by the eye or via a few specular reflections. This computation is performed using importance sampling where the information about the incoming flux is integrated with the BRDF to provide optimized sampling directions. Furthermore we use information about shadow photons to reduce the number of shadow rays. Accurate computation of caustics is done by visualizing a radiance estimate obtained using a separate caustics photon map which has a high density of photons.

3 Pass 1: Constructing the Photon Maps

The photon maps are constructed by emitting a large number of photons (packets of energy) from the light sources in the scene. Each photon is traced through the scene using a method similar to path tracing. Every time a photon hits a surface it is stored within the photon map and Russian roulette [2] is used to determine whether the photon is absorbed or reflected. The new direction of a reflected photon is computed using the BRDF of the surface.

Unlike previous implementations we use two photon maps: A *caustics photon map* and a *global photon map*. The caustics photon map is used only to store photons corresponding to caustics and it is created by emitting photons towards the specular objects in the scene and storing these as they hit diffuse surfaces. Caustics are rendered by visualizing a radiance estimate based on the caustics photon map directly and this requires a high density of photons.

The global photon map is used as a rough approximation of the light/flux within the scene and it is created by emitting photons towards all objects. It is not visu-

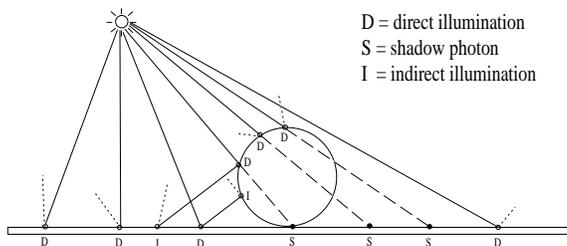


Figure 1: The photons in the global photon map are classified to optimize the rendering of shadows

alized directly and therefore it does not require the same precision as the caustics photon map. We use the extension presented in [17] and create shadow photons by tracing rays with origin at the light source through the entire scene. At the first intersection point a normal photon is stored and at the following intersection points we store shadow photons. These shadow photons are used during the rendering step to reduce the number of shadow rays (see figure 1).

The fact that we have two separate photon maps has improved both the speed, reduced the memory requirements and improved the accuracy of the method. Rendering caustics is faster since the caustics photon map contains only photons related to caustics. Locating photons in the global photon map is also faster since it has fewer photons and these photons have energy levels that are more similar since it does not contain the mixture of caustics photons with high density and low energy and normal photons with low density and high energy. This significantly improves the accuracy of the radiance estimate.

The photons are stored in a balanced kd-tree [4]. This data-structure is both compact and efficient. The fact that the tree is balanced guarantees that the time it takes to locate M photons in a tree with N photons is $O(M \cdot \log_2(N))$. In practice the search is much more efficient since the photons are located in the same parts of the tree. The use of a balanced kd-tree makes the rendering more efficient as demonstrated in [18] but just as important it reduces the memory requirements for each photon hit and allows us to represent each photon using only 20 bytes.

4 Pass 2: Rendering

The final image is rendered using Monte Carlo ray tracing in which the pixel radiance is computed by averaging a number of sample estimates. Each sample consists of tracing a ray from the eye through the pixel into the scene. The radiance returned

by each ray is computed at the first surface intersected by the ray and it equals the surface radiance, $L_s(\mathbf{x}, \Psi_r)$, leaving the point of intersection, \mathbf{x} , in the direction, Ψ_r , of the ray. $L_s(\mathbf{x}, \Psi_r)$ is computed using the rendering equation [14]:

$$L_s(\mathbf{x}, \Psi_r) = L_e(\mathbf{x}, \Psi_r) + \int_{\Omega} f_r(\mathbf{x}, \Psi_i; \Psi_r) L_i(\mathbf{x}, \Psi_i) \cos \theta_i d\omega_i \quad (1)$$

Where L_e is radiance emitted by the surface, L_i is the incoming radiance in the direction Ψ_i , f_r is the BRDF and Ω is the sphere of incoming directions. L_e is taken directly from the surface definition and needs no further calculation. The value of the integral, L_r , depends on the radiance values in the rest of the scene and it can be solved directly using Monte Carlo techniques like path tracing. This is however a very expensive method and a more efficient approach can be obtained by using the photon map in combination with our knowledge of the BRDF and the incoming radiance.

The rendering equation (1) can be split into a sum of several components. We omit the position and direction parameters for clarity, and express L_r as

$$\begin{aligned} L_r = & \int_{\Omega} f_r L_{i,l} \cos \theta_i d\omega_i + \\ & \int_{\Omega} f_{r,s} (L_{i,c} + L_{i,d}) \cos \theta_i d\omega_i + \\ & \int_{\Omega} f_{r,d} L_{i,c} \cos \theta_i d\omega_i + \\ & \int_{\Omega} f_{r,d} L_{i,d} \cos \theta_i d\omega_i \end{aligned} \quad (2)$$

where

$$f_r = f_{r,s} + f_{r,d} \quad \text{and} \quad L_i = L_{i,l} + L_{i,c} + L_{i,d}$$

In this equation the incoming radiance has been split into contributions from the light sources, $L_{i,l}$, contributions from the light sources via specular reflection (caustics), $L_{i,c}$ and indirect soft illumination, $L_{i,d}$ (light which has been reflected diffusely at least once). The BRDF has been separated into a diffuse part, $f_{r,d}$, and a specular part, $f_{r,s}$. The diffuse part represents all reflection models from Lambertian to slightly glossy while the specular part are highly glossy and ideal specular reflection models (examples are presented in section 6).

Equation 2 is used to compute the radiance leaving a surface. In the following sections we discuss the evaluation of each of the parts in the equation in more detail. We distinguish between two different evaluations of the integrals: An accurate and an approximate.

We use the accurate computation if the surface is seen directly by the eye or perhaps via a few specular reflections. We also use the accurate computation if the distance between the ray origin and the intersection point is below a small threshold value — otherwise we might risk inaccurate colour bleeding effects in corners. The approximate evaluation is used if the ray intersecting the surface has been reflected diffusely since it left the eye or if the weight of the ray is low (it contributes only little to the pixel radiance).

4.1 Direct Illumination

The first term in (2) represents the contribution via direct illumination by the light sources. This term is normally computed by sending shadow rays towards all light sources to check for visibility. We compute the contribution differently depending on whether we need an accurate or an approximate evaluation.

In the accurate evaluation of the contribution we use the observation that most scenes have large areas that are either fully illuminated or in shadow. We can use the information in the photon map to identify these areas in order to avoid using shadow rays. We only use shadow rays in situations where the nearest photons in the global photon map contains a mixture of direct illumination photons and shadow photons or if the number of illumination and shadow photons located is too low. This strategy is described in more detail in [17].

The approximate evaluation is simply the radiance estimate obtained from the global photon map (no shadow rays or light source evaluations are used).

4.2 Specular Reflection

The second term in (2) is radiance reflected of specular and highly glossy surfaces. This value is computed using standard Monte Carlo ray tracing. By using importance sampling based on the BRDF the computation can in most cases be done using only a limited number of sample rays.

4.3 Caustics

The third term in (2) represents caustics on diffuse and slightly glossy surfaces. We evaluate this term using the information in the caustics photon map (see section 5). We never compute caustics via Monte Carlo sampling since this is almost impossible in most situations. This means that the radiance estimate based on the caustics photon map is visualized directly and this is the reason why the number of photons in the caustics photon map must be high.

4.4 Soft Indirect Illumination

The fourth term in (2) is incoming light which has been reflected diffusely at least once since it left the light source. This light is then reflected diffusely by the surface (using $f_{r,d}$) and consequently the resulting illumination is very “soft”.

The approximate evaluation of this integral is the radiance estimate based on the global photon map (see section 5).

In the accurate evaluation we use importance sampling to compute the indirect illumination. As described in [16] we combine the information in the photon map with the BRDF in order to generate optimized sampling directions. At Lambertian surfaces we also use the irradiance gradient caching scheme [32]. This means that we only compute indirect illumination on Lambertian surfaces if this information cannot be interpolated from previously computed values.

5 Estimating Radiance using the Photon Map

The information in the photon map can be used to compute the radiance leaving a surface in a given direction. Since the incoming direction is stored with each photon we can integrate the information with any BRDF. In practice the approximation is limited to surfaces ranging from Lambertian to slightly glossy. To compute radiance leaving highly glossy surfaces a very large number of photons is needed. There is nothing in our algorithm preventing this approach. We have however found that highly glossy surfaces can be treated efficiently using Monte Carlo ray tracing and we use this strategy in order to limit the memory requirements.

To compute the radiance, L_r , leaving an intersection point \mathbf{x} at a surface with BRDF f_r , we locate the N photons with the shortest distance to \mathbf{x} . Based on the assumption that each photon p represents flux $\Delta\Phi_p$ arriving at \mathbf{x} from direction $\Psi_{i,p}$ we can integrate the information into the rendering equation as follows

$$L_r(\mathbf{x}, \Psi_r) = \int_{\Omega} f_r(\mathbf{x}, \Psi_r, \Psi_i) \frac{d^2\Phi_i(\mathbf{x}, \Psi_i)}{dA d\omega_i} d\omega_i \approx \sum_{p=1}^N f_r(\mathbf{x}, \Psi_r, \Psi_{i,p}) \frac{\Delta\Phi_p(\mathbf{x}, \Psi_{i,p})}{\pi r^2} \quad (3)$$

We use the same approximation of ΔA as [15] where a sphere centered at \mathbf{x} is expanded until it contains N photons and has radius r . ΔA is then approximated as πr^2 .

An alternative could be using a sphere of a fixed size and use all the photons within this sphere. We have tested this technique and it improves the estimate slightly since ΔA is kept constant. It does however fail in scenes with a high variation in the density of the photons since it either gives bad estimates in areas with few photons or blurry estimates in areas with a high photon density. We have

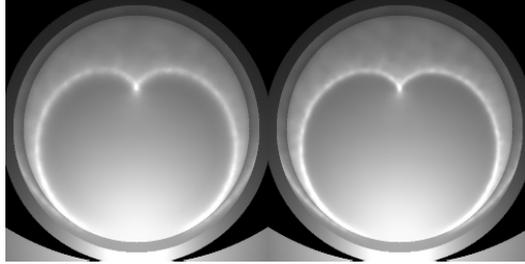


Figure 2: The effect of the cone filter. The left image is unfiltered and the right image is filtered using the cone with 1 as the filter constant

considered a number of adaptive strategies for computing the necessary size of the sphere based on the local photon density. We did however find that the payoff with respect to quality could not compensate for the extra computing time.

In situations where the density of the photons is too low the radiance estimation strategy can give blurry results. To compensate for this situation we have successfully applied a cone-filter to the estimate. In the cone-filter a weight is attached to each photon based on the distance, d_p , between \mathbf{x} and the photon p . This weight is:

$$w_p = \max(0, 1 - d/(kr)) \quad (4)$$

where k is a filter constant characterizing the filter. To normalize the filter we need some knowledge on the distribution of the photons. Since we use a sphere to locate the photons it would be natural to assume that the distribution of the photons is 3 dimensional and related to the sphere. However, photons are stored at surfaces which are 2 dimensional. Furthermore the area estimate is also based on the assumption that photons are located on a surface. Our normalization is therefore based on a 2d-distribution of the photons and it becomes $1 - \frac{2}{3k}$. The filtered radiance estimate can thus be expressed as

$$L_r(\mathbf{x}, \Psi_r) \approx \frac{\sum_{p=1}^N f_r(\mathbf{x}, \Psi_r, \Psi_{i,p}) \Delta \Phi_p(\mathbf{x}, \Psi_{i,p}) w_p}{(1 - \frac{2}{3k}) \pi r^2} \quad (5)$$

In figure 2 we have showed the effect of the cone filter as it is applied to the well known cardioid caustic. We used only 12000 photons to render this caustic and the result is that the traditional radiance estimate looks blurry. Applying the cone filter significantly reduces this blur. In the figure we use a filter constant $k = 1$. This value generally works very well.

6 Results and Discussion

We have implemented the two pass method in a program called MIRO on a 100MHz Pentium PC with 32MB RAM running Linux.

Our first test scene is the *museum* shown in fig. 3. It has 5000 normal objects (spheres, triangles etc.) and 1 procedural object (the sphere flake). All important combinations of light reflections can be found in this scene. We have caustics from the glass sphere onto the rough surface, caustics from the glossy cylinder on the wall and also on the procedural sphere flake object. The sphere flake object has been rendered using Schlick’s reflection model [23] with a diffuse-specular parameter of 0.1. Other important reflections include the colour bleeding effect between the walls, the glossy reflection of the metallic teapot (using Ward’s anisotropic model [31]), the transmission of light through the glass sphere and the specular reflection in the floor and the teapot. The scene is illuminated by two small spherical area light sources. We used 289.000 photons in the caustics photon map and 165.000 photons in the global photon map to render this scene. This corresponds to approx. 9MB memory. The image was rendered in the resolution 1280x960 and the rendering time was 51 min. The photon map was constructed in 5 min. The most time consuming part of the scene is the computation of reflection of the teapot into the glossy cylinder since the number of reflections traced by each ray is not limited by using the photon map.

To demonstrate how the photon map actually works we have visualized the radiance estimate directly in fig. 4. The radiance estimate is shown for all diffuse surfaces (include the sphere flake) and it is based only on the 165.000 photons in the global photon map. An average of 80 photons have been used to estimate the radiance per surface intersection. Notice how all important types of reflections are included even though they are blurry. The blur in the photon map is actually an advantage since it reduces noise in the final gathering step where Monte Carlo sampling is used to render the initial reflections accurately.

Our second test scene shown in fig. 5 demonstrates the looks of a caustic from a

Scene	Resolution	Caustic photons	Global photons	Pass 1	Rendering
Diffuse Cornell Box	1280x960	21.162	286.489	67 sec.	8 min
Diffuse Cornell Box [R]	1280x960	-	-	-	60 min
Glossy Cornell Box	2560x1920	0	382.598	56 sec.	50 min
Glossy Cornell Box [R]	5120x3840	-	-	-	360 min
The Museum	1280x960	389.755	165.791	298 sec.	51 min
The Cognac Glass	1280x960	224.316	3095	27 min.	65 min

Table 1: Rendering statistics. **[R]** indicates the images rendered with Radiance

cognac glass onto a rough (fractal) surface approximated by 500.000 triangles. The reflection model for the surface is Schlick’s reflection model with a diffuse-specular component of 0.6 — we found that this value makes the sand look more realistic. The caustic was created using 224.000 photons. These photons represent both the red caustic and the illumination of the surface below the cognac glass. If this model is rendered without caustics the surface below the cognac glass would be black. We also measured the advantage of using shadow photons in this scene and by using only 216 shadow photons (\approx 5KB extra information) we were able to reduce the number of shadow rays with more than 70 %

To test the performance of our method we have compared it with the **Radiance** program — a superb global illumination program developed over the last 10 years by Greg Ward [33]. It is based on a significantly optimized Monte Carlo ray tracing scheme and it performs very well even compared with newer hierarchical radiosity techniques [6].

We have used our photon map implementation and **Radiance** to render two variations of the *Cornell box*: One in which the floor is Lambertian and one in which the floor is highly glossy (using the Anisotropic reflection model). We adjusted the parameters in both programs in order to obtain good quality within a reasonable time. The rendering times are shown in table 1.

The version with the diffuse floor was rendered in 1280x960 with both programs and as it can be seen from the table. With photon maps the rendering time is 6 times faster than **Radiance**. The primary reason is that we avoid the recursive sampling of the indirect illumination and the fact that we use fewer shadow rays to sample the area light source. In this scene the average depth of the image sampling rays is very close to 2 since most of the radiance computations beyond the first diffuse reflection is handled by the photon map.

The Cornell box with the glossy floor was rendered in 2560x1920 with **MIRO** and 5120x3840 with **Radiance**. We had to increase the resolution in **Radiance** since it uses path tracing to render glossy surfaces. Both images have been reduced to the resolution 640x480 and this means that **Radiance** uses 64 samples per pixel (ie. 64 samples to sample the indirect illumination on the first glossy surfaces seen through a pixel). **MIRO** uses distribution ray tracing and the only reason why we increased the resolution was in order to obtain the same level of anti-aliasing as **Radiance**. The distribution ray tracer in **MIRO** spawns a maximum of 6 sample rays at the glossy surface. Combined with the 16 samples per pixel this gives a maximum of 96 samples used to compute the indirect illumination on the first glossy surface seen through a pixel. This also means that the glossy surface looks less noisy in the version rendered by **MIRO**. The two images are shown in fig. 6 and fig. 7 and as we can see they look very similar. There is a slight differences in the overall illumination caused by different tone reproduction functions (gamma correction). The timing results in

table 1 shows that MIRO renders the glossy Cornell box approximately 7 times faster than Radiance. The number of pixels is 4 times higher in the Radiance version but this number cannot be used directly due to the different sampling schemes used by the two programs. It is more correct to look at the number of samples spawned at the glossy surface since this is the actual reason why rendering these images takes so relatively long.

In table 1 we have collected some statistics showing the memory and time required to render the test images. As it can be seen the number of photons used is in the range 200.000-500.000. We have not carefully optimized these numbers since the rendering time is only affected slightly by the number of photons. Instead we use an appropriate number of photons and for our test images we have found that 2-500.000 photons gives nice results. In more complex scenes it would probably be necessary to use a higher number of photons. It is however important to notice that the necessary number of photons is not directly related to the number of objects in the scenes. It is instead related to the complexity of the flux within the scene. We would probably be able to render the Cornell box with detailed stone walls made of millions of triangles using the same number of photons as we did with the simple Cornell box. If we render a scene with too few photons we get low frequency noise in the caustics - this kind of noise is less disturbing than the high frequency noise that is normally seen in Monte Carlo ray tracing algorithms. The effect on the remaining parts of the illumination is more subtle and it depends on the rendering parameters. But if too few photons are used it means that we have to use more sample rays to compute the indirect illumination and we might get "poor statistics" in the radiance estimates which in our implementation results in recursive Monte Carlo sampling.

We believe that the results can be improved even further by using the photon map more intelligently. As an example we might use the photons to answer questions regarding the number of samples necessary to use for a pixel. Another interesting use of the photon map would be reclassification of light sources as done in [5]. The photon map could also be used to represent flux within participating media. This should be straightforward to implement since nothing prevents photons from being stored within a volume.

Currently we have only rendered scenes containing a few light sources (less than 10). Rendering scenes with many light sources makes the use of photon maps more complicated since naive emission of photons from every light source will generate a very large number of photons. We might use some kind of radiosity-like importance to distribute the photons more intelligently within the scene. It would be very interesting to make a 3-pass method in which an initial simple ray tracing pass is used to generate importance information that can be used when emitting the photons. This might also help answering the difficult question of the necessary

number of photons. The current strategy is just to use enough photons (what the available memory permits).

A very important aspect of the photon map is the fact that it is easy to integrate into existing ray tracing programs since it only requires the existence of intersection routines for each object. The scene does not have to be tessellated and the photon map structure is completely separated from the geometric representation. The photon map code can be provided in a separate module that contains the necessary functions (e.g. a function that given a position and a surface definition returns the radiance in a given direction).

7 Conclusion

We have presented a general two-pass global illumination method based on photon maps. We integrate information from an accurate caustics photon map and a less accurate global photon map into a distribution ray tracer. Caustics are rendered by visualizing a radiance estimate from the caustics photon map directly. The information in the global photon 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.

We have used the method to simulate global illumination in scenes containing procedural objects and surfaces with diffuse and glossy reflection. Comparisons with existing global illumination techniques indicate that the photon map provides an efficient environment for global illumination.

8 Acknowledgment

The author wishes to thank Greg Ward who provided invaluable help with the Radiance program. Thanks also to Niels Jørgen Christensen and the reviewers for their helpful comments.

References

- [1] Arvo, James: "Backward Ray Tracing". *Developments in Ray Tracing. ACM Siggraph Course Notes* **12**, pp. 259-263, 1986.
- [2] Arvo, James and David Kirk: "Particle Transport and Image Synthesis". *Computer Graphics* **24** (4), pp. 53-66, 1990.
- [3] Aupperle, Larry and Pat Hanrahan: "A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection". *Computer Graphics* **27** (4), pp. 53-66, 1993.

- [4] Bentley, Jon Louis: "Multidimensional Binary Search Trees Used for Associative Searching". *Comm. of the ACM* **18** (9), pp. 509-517, 1975.
- [5] Chen, Eric Shenchang; Holly E. Rushmeier, Gavin Miller and Douglass Turner: "A Progressive Multi-Pass Method for Global Illumination". *Computer Graphics* **25** (4), pp. 164-174, 1991.
- [6] Christensen, Per Henrik: "*Hierarchical Techniques for Glossy Global Illumination*". Ph.d. thesis, University of Washington, 1995.
- [7] Collins, Steven: "Adaptive Splatting for Specular to Diffuse Light Transport". In *proceedings of 5. Eurographics Workshop on Rendering*, pp. 119-135, Darmstadt 1994.
- [8] Cook, Robert L.: "Distributed Ray Tracing". *Computer Graphics* **18** (3), pp. 137-145, 1984.
- [9] Glassner, Andrew S.: "*Principles of Digital Images Synthesis*". Morgan Kaufmann Publishers Inc., 1995.
- [10] Goral, Cindy M.; Kenneth E. Torrance; Donald P. Greenberg and Benneth Battaile: "Modeling the Interaction of Light Between Diffuse Surfaces". *Computer Graphics* **18**, pp. 213-222, 1984.
- [11] Gortler, Steven J.; Peter Schröder; Michael F. Cohen and Pat Hanrahan: "Wavelet Radiosity". *Computer Graphics* **27** (4), pp. 221-230, 1993.
- [12] Heckbert, Paul S.: "Adaptive Radiosity Textures for Bidirectional Ray Tracing". *Computer Graphics* **24** (4), pp. 145-154, 1990.
- [13] Immel, David S.; Michael F. Cohen and Donald P. Greenberg: "A Radiosity Method for Non-Diffuse Environments". *Computer Graphics* **20** (4), pp. 133-142, 1986.
- [14] Kajiya, James T.: "The Rendering Equation". *Computer Graphics* **20** (4), pp. 143-149, 1986.
- [15] 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.
- [16] Jensen, Henrik Wann: "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.
- [17] Jensen, Henrik Wann and Niels Jørgen Christensen: "Efficiently Rendering Shadows using the Photon Map". In *Proceedings of Compugraphics 95'*, pp. 285-291, 1995.
- [18] Jensen, Henrik Wann: "Rendering Caustics on non-Lambertian Surfaces". To be presented at *Graphics Interface '96*, Toronto 1996.
- [19] Lafortune, Eric P.; Yves D. Willems: "Bidirectional Path Tracing". *Proceedings of CompuGraphics*, pp. 95-104, 1993.
- [20] Lafortune, Eric P.; Yves D. Willems: "The Ambient Term as a Variance Reducing Technique for Monte Carlo Ray Tracing". In *proceedings of 5. Eurographics Workshop on Rendering*, pp. 163-171, Darmstadt 1994.

- [21] Lafortune, Eric P.: "*Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*". Ph.d. thesis, Katholieke University, Leuven, Belgium 1995.
- [22] Rushmeier, Holly; Ch. Patterson and A. Veerasamy: "Geometric Simplification for Indirect Illumination Calculations". *Proceedings of Graphics Interface '93*, pp. 35-55, 1994.
- [23] Schlick, Christophe: "A Customizable Reflectance Model for Everyday Rendering". In *proceedings of 4. Eurographics Workshop on Rendering*, pp. 73-84, Paris 1993.
- [24] Shirley, Peter: "A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes". *Proceedings of Graphics Interface '90*, pp. 205-212, 1990.
- [25] Shirley, Peter; Bretton Wade; Phillip Hubbard; David Zareski; Bruce Walter and Donald P. Greenberg: "Global Illumination via Density Estimation". In "Rendering Techniques '95". Eds. P.M. Hanrahan and W. Purgathofer, *Springer-Verlag*, pp. 219-230, 1995.
- [26] Sillion, François X.; James R. Arvo; Stephen H. Westin and Donald P. Greenberg: "A Global Illumination Solution for General Reflectance Distributions". *Computer Graphics* **25** (4), pp. 187-196, 1991.
- [27] Veach, Eric and Leonidas Guibas: "Optimally Combining Sampling Techniques for Monte Carlo Rendering". *Computer Graphics* **29** (4), pp. 419-428, 1995.
- [28] Wallace, John R.; 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), pp. 311-320, 1987.
- [29] Ward, Gregory J.; Francis M. Rubinstein and Robert D. Clear: "A Ray Tracing Solution for Diffuse Interreflection". *Computer Graphics* **22** (4), pp. 85-92, 1988.
- [30] Ward, Greg: "Real pixels". In *Graphics Gems II*, James Arvo (ed.), *Academic Press*, pp. 80-83, 1991.
- [31] Ward, Gregory J.: "Measuring and Modeling Anisotropic Reflection". *Computer Graphics* **26** (2), pp. 265-272, 1992.
- [32] Ward, Gregory J. and Paul S. Heckbert: "Irradiance Gradients". In *Proceedings of the Third Eurographics Workshop on Rendering*, pp. 85-98, Bristol 1992.
- [33] Ward, Gregory J.: "The RADIANCE Lighting Simulation and Rendering System". *Computer Graphics* **28** (4), pp. 459-472, 1994.
- [34] Zimmerman, Kurt and Peter Shirley: "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. 284-295, 1995.



Figure 3: The Museum scene



Figure 4: Direct visualization of the global photon map in the Museum scene



Figure 5: A Cognac glass on a fractal surface

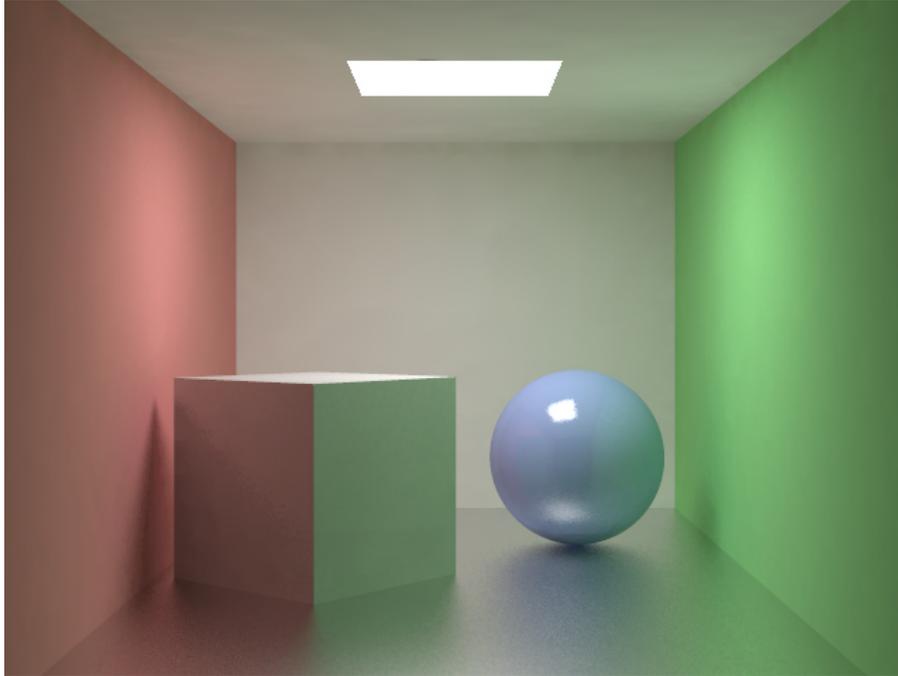


Figure 6: The glossy Cornell box rendered with photon maps

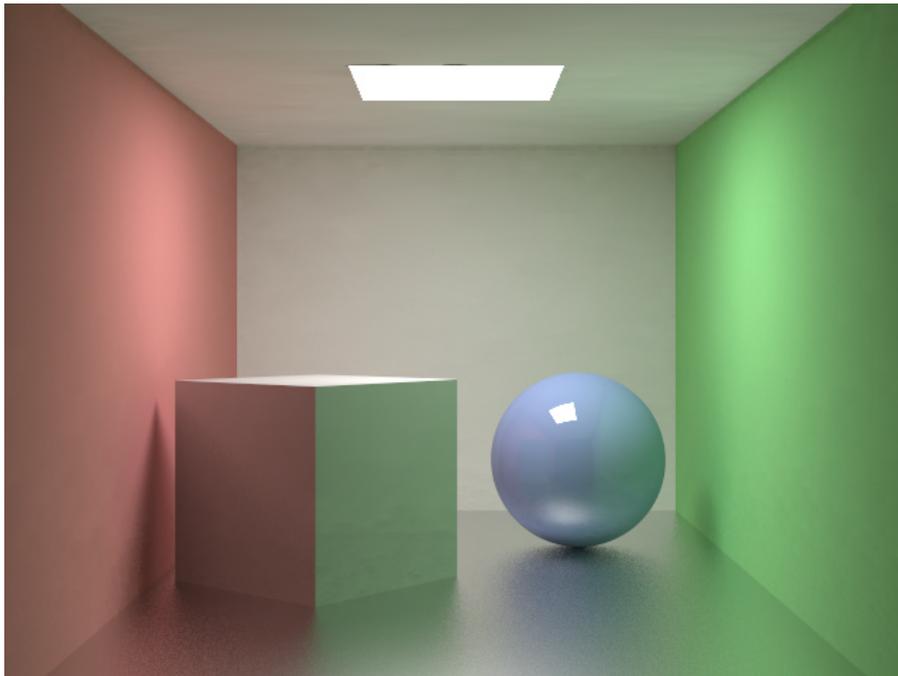


Figure 7: The glossy Cornell box rendered with Radiance