Project INF584
Global Illumination using Photon Maps

Irina NIKULINA

April 12, 2020

1 Introduction

One of the important problems of image synthesis is global illumination. Its proper simulation is an essential part of a realistic scene rendering but at the same time it is a very complex task. We often won't even think about how complicated is path of the light we see. Global illumination comprises direct illumination (light coming directly from the light source) and indirect illumination resulting from multiple reflections or refractions of rays that both in their turn depend a lot on surface properties. There are many algorithms of different complexity that allow to simulate illumination with high precision. Theoretical foundation of many rendering approaches was created back in 1980s and 1990s. These algorithms, along with a new ones in combination with modern hardware are closer to a realistic scene simulation that ever.

2 Method description

In this project we consider 2-pass global illumination approach presented in [**1**]. Let's briefly consider its principal steps - photon map construction and rendering.

2.1 Photon map construction

During the first pass a photon map is constructed by emitting photons by light sources and storing the information about particles as they hit a surface. Photon tracing technique is similar to the path tracing one, but in this case we use Russian roulette to decide whether the photon is absorbed and subsequently won't be traced any further or whether it will be reflected and will continue its journey following the direction defined by BRDF of the surface.

The authors propose to create two separate photon maps storing global information and caustics. Global photon map is obtained by emitting photons to all objects of the scene and gives us rather rough representation of light distribution, that is reasonable because this map is not visualized directly but still have a lot of useful information that can be used during rendering. In addition to this, shadow photons are used to reduce number of shadow rays.

It is difficult to track caustics using Monte-Carlo ray tracing and one of possible solutions of this problem is caustics photon map. It contains information about photons emitted with high density towards specular objects and is rendered directly by visualising radiance estimate based on it.

2.2 Rendering

To render the final image rays are traced from the eye into the scene using Monte Carlo ray tracing. The approach described in the article suggest splitting the reflected radiance integral L_r in the rendering equation describing surface radiance (1).

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

where L_e is radiance emitted by the surface, L_i is the incoming radiance in the direction Ψ_i.

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

$$f_r = f_{r,d} + f_{r,s}; L_i = L_{i,l} + L_{i,c} + L_{i,d} \tag{3}$$

where BRDF f_r is split into a sum of diffuse part and a specular part and the incoming radiance L_i includes 3 terms responsible for indirect soft illumination, caustics and light reflected diffusely correspondingly.

There are two different approaches to evaluate the above-mentioned integrals, the accurate one that is used for directly seen objects and thin details and the approximate one for diffusely reflected or low-weighted rays.

For accurate evaluation of direct illumination photon maps are used to identify fully illuminated or shaded areas instead of shadow rays . Approximate evaluation is done by radiance estimation from global photon map. Standard Monte Carlo ray tracing with importance sampling is used to compute specular reflections. Caustics photon map help to evaluate the caustics term, whose radiance is directly visualised.

Finally, soft indirect illumination is also computed using information stored in photon maps. More precisely, accurate evaluation is done using importance sampling, which optimized directions are estimated using photon map with BRDF. Approximate evaluation of corresponding integral is computed according to the following formula:

$$L_r(\mathbf{x}, \Psi_r) = \int_\Omega f_r(\mathbf{x}, \Psi_i; \Psi_r) \frac{d_2\Phi_i(\mathbf{x}, \Psi_i)}{dA d\omega_i} d\omega_i \approx \sum_{p=1}^{N} f_r(\mathbf{x}, \Psi_i; \Psi_r) \frac{\Delta\Phi_p(\mathbf{x}, \Psi_{i,p})}{\pi r^2} \tag{4}$$

where $\Delta\Phi_p$ is a flux corresponding to a photon p arriving at x from direction $\Psi_{i,p}$. Basically the approximation considers a sphere centered at x of radius r that contains N photons, so that ΔA can be approximated as πr^2. To avoid situations where rendering of areas with low photon density gives blurry results a cone filter is proposed. It suggests attaching a weight to each photon, so that:

$$\omega_p = \max(0, 1 - \frac{d_p}{kr}) \tag{5}$$

where d_p is distance from photon to the point x and k is a filter constant.

The method described in the article propose a two pass method of estimation and rendering of different types of illumination using among other things photon maps. It was proved that the latter provides an efficient environment for global illumination [**1**].

3 Implementation

In contrast to the article where the two-pass method was implemented in a program called MIRO, we provide partial but direct implementation of the presented approach using C++.

3.1 Scene description

The scene consists of two side walls, a floor, a cube, a tetrahedron and a light source. All objects are made of the same material and are built using triangular mesh. A light source is an area source with a side of size 0.2. The normal of the light source is pointed to the origin of coordinates that is considered as a center of the scene. The camera is facing negative direction of Z-axis.

3.2 General remarks

To perform ray tracing we generate rays from camera through a pixel into the scene. We emit several rays per pixel that are randomly scattered over its surface and average the received color responses. BRDF is computed as a sum of specular and diffuse components.

3.3 Photon map

A photon map was created by emitting photons from an area light source. Taking into accout the scene arrangement photons were only emitted in the hemisphere facing the center. Each time a photon hits a surface, its position, intensity and incoming direction is stored in the photon map that has form of a vector of Photons (Photon structure is given above). The initial intensity of a photon is 1.0 and it is reduced twice after each reflection. We use Russian roulette to decide whether it will be reflected or absorbed. The direction of reflected photon is computed according to the law of reflection : angle of incidence equals the angle of reflection. For radiance estimation at some point $\mathbf{x}$ (see Section 3.4), in contrast to the article we consider a sphere of fixed size and we use all the photons inside it to compute radiance (that is also a viable solution according to the article). For efficient search of photons within the given sphere a kd-tree is built using a photon map. The kd-tree structure stores positions of photons and they indexes in the photon map.

3.4 Monte-Carlo Path tracing

We use Monte-Carlo path tracing to compute direct and indirect lightening. The path tracing is implemented recursively with the maximum depth of three. Ones ray passed through a pixel intersects a surface, we estimate current color response in the intersection point and we track the new reflected ray in the same manner. The recursion is ended after 2 reflections or if the last ray did not hit any surface. For every pixel resulting color response is equal to a weighted sum of color responses obtained after every intersection. The weight is defined as $1/recursionLevel$, so that a color response from the initial ray has more weight.

We consider two approaches to evaluate radiance - with and without photon map. In the first case that will be further referenced as regular shading routine we use a model of modulated light attenuation, so the radiance is in inverse proportion to the squared distance from the considered point to the light source.

In the second case we take use of photon map to compute the radiance values. Ones ray hits the surface we build a sphere of fixed size centered in the intersection point and look for photons within the sphere. BRDF is computed using average direction of all the photons inside the sphere in contrast to the article where a separate value of BRDF is computed for every photon. Taking into account the chosen approach we add an additional coefficient the obtained radiance so that it would be in inverse proportion to the overall number of photons emitted by a light source to avoid too light or too dark resulting image.

3.5 Experiment

In our experiment in order to build a photon map we emitted $5 * 10^5$ and 1 million photons that after reflection resulted in $6.7 * 10^5$ and $2 * 10^6$ entries in the photon map correspondingly. A visualisation of two photons maps is shown on Fig. 1 and Fig. 2. We can see a significant difference in photon densities on these pictures. There horizontal line on the left wall that divides areas with different photon densities is explained by the not-centered position of area light source that is not parallel to the floor of the scene. We use Russian roulette, that sets probability of reflection equal to 0.7, to decide whether the photon will continue its track after hitting a surface. Also we suppose that the photons with intensity less than 0.01 are not reflected to avoid to many additional computations.

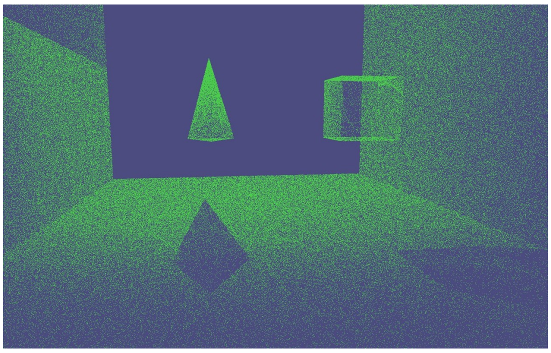

Figure 1: Visualization of a photon map with 500k emitted photons

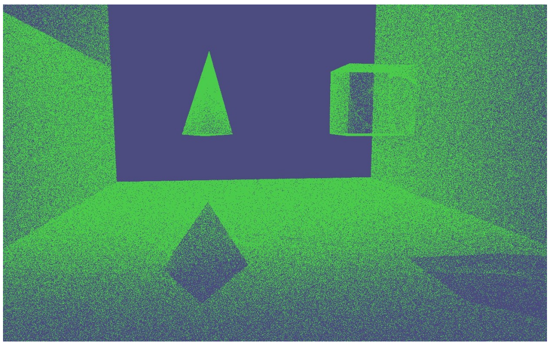

Figure 2: Visualization of a photon map with 1 million emitted photons

We implement Monte-Carlo path tracing with and without photon mapping emitting 5 rays per pixel. The results of the experiment are shown on Fig. 3 and Fig. 4. Using photon maps we were able to obtain an image with globally more realistic illumination. In this example $5*10^5$ photons were emitted and the radius of sphere for the neighbor photon search was 0.5. We also provide images with different value of radius (0.3 and 0.7 on the Fig. 5 and Fig. 6) and with 1 million emitted photos (Fig. 7). The augmentation of sphere radius leads to bigger number of photons concluded in the sphere and consequently to the higher values of computed radiance, so the image corresponding to bigger radius seem lighter.

In the above mentioned examples we supposed that the contribution of photons into the radiance is proportional to their intensity. In addition to these experiments we also implement a cone filter described in the article with constant 1, so the contribution of photons is defined by the corresponding weight (Fig. 8). We can conclude that both approaches give close results on these scene. The advantage of cone filter might be more

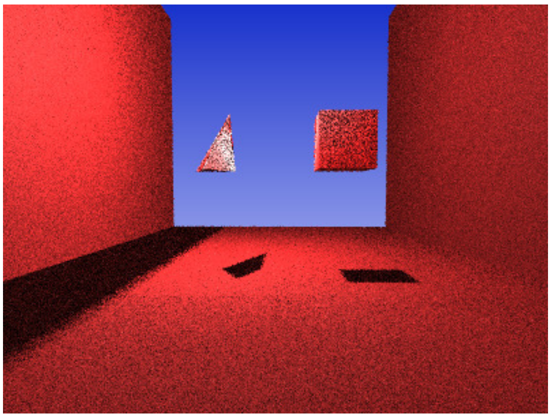

Figure 3: Scene image obtained using Monte-Carlo ray tracing and regular shading routine

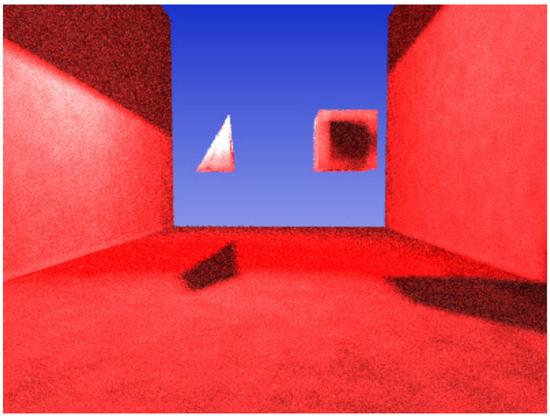

Figure 4: Scene image obtained using Monte-Carlo ray tracing with photon mapping

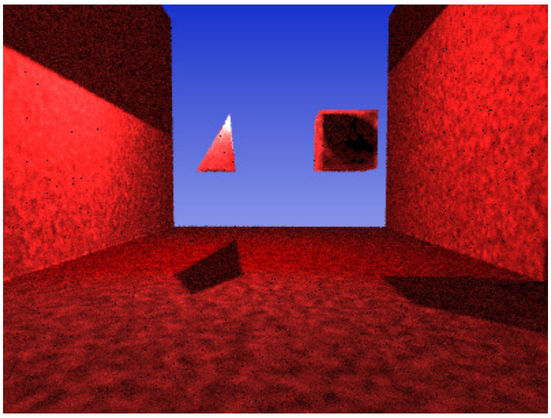

Figure 5: Scene image obtained using Monte-Carlo ray tracing with photon mapping, radius = 0.3

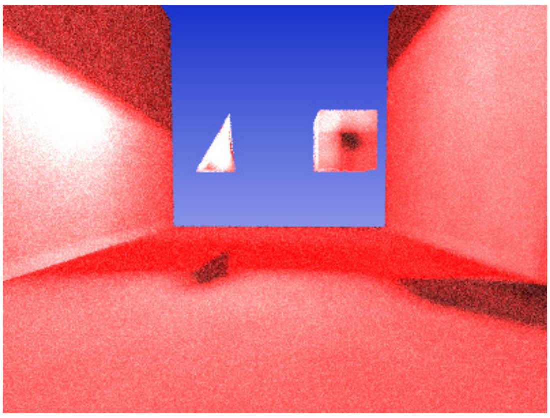

Figure 6: Scene image obtained using Monte-Carlo ray tracing with photon mapping, radius = 0.7

noticeable with more variable density on thin structures within the scene with better resolution.

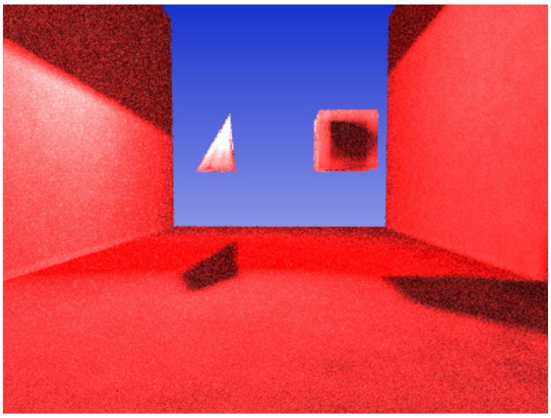

Figure 7: Scene image obtained using Monte-Carlo ray tracing with photon mapping, 1 million emitted photons

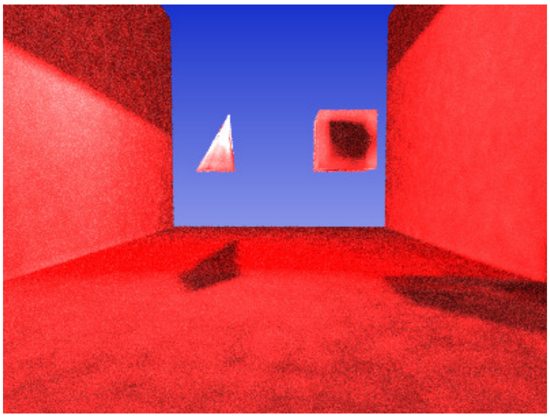

Figure 8: Scene image obtained using Monte-Carlo ray tracing with photon mapping and cone filter

4 Future work

A further extension of the project besides developing more complex scene would be implementing a caustics photon map, so that caustics will be also taken into the account while rendering the scene. However to build such map a large number of photons has to be emitted towards specular objects and defining these directions in C++ will need an accurate choice of a proper approach.Also it would be interesting to trace shadow photons to replace shadow rays as suggested in the article and compare the resulting performance. In addition to this while computing soft indirect illumination using photon map another method to construct a sphere can be applied. In our implementation we considered a sphere of a fixed size, that can be replaced a sphere of variable radius that includes exactly N photons inside it. The visual difference of scene rendering using these two approaches can be better noticed in scenes with areas with variable photon densities. Implementation of importance sampling in ray tracing or photon tracing routines could significantly improve the quality of resulting scene rendering.

5 Conclusion

In this project we have partially implemented a two pass approach for global illumination using photon mapping presented in [**1**]. It was shown that Monte Carlo path tracing combined with photon mapping can help us to effectively render more realistic scene images. The results of suggested implementation in C++ reaffirm the viability of the considered method and demonstrate the importance of proper parametrization.

References

[1] Henrik Wann Jensen, Global Illumination using Photon Maps, The Technical University of Denmark, 1996.

[2] 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.