

Bidirectional Photon Mapping

Jiří Vorba

Supervised by: Jaroslav Krivánek

Charles University, Prague

Abstract

This paper introduces a method for optimal combination of light paths generated from the camera and from the light sources in the photon mapping algorithm used for computing global illumination. Our method is based on Multiple Importance Sampling, a general approach, introduced by Veach, for adaptive path connection in bi-directional path-tracing. Our goal is to examine this method in connection with the biased algorithm of photon mapping and to improve the ineffective final gather heuristic used in the original version of this algorithm. This heuristic is usually problematic when applied to the scenes where highly glossy materials prevail.

1 Introduction

Bi-directional methods for computing global illumination generate light paths both from light sources and from camera and afterwards they connect them together. A provably good strategy for connecting light paths introduced by Veach [1] is known for bi-directional path tracing (BDPT) [2]. However, the algorithm is not put to use that often in practice because of the slow convergence of some parts of the light transport.

On the other hand, for popular bi-directional method like photon mapping [3] the optimal strategy for path connection is not known. The consequence is poor image quality in scenes containing many glossy materials. There are two reasons why objectionable artifacts usually occur when using photon mapping on glossy objects. First, radiance estimate on highly glossy materials suffers from high variance. Second, distribution rays cast during final gathering will usually hit the scene too close to each other because of the narrow BRDF peak. This results in highly correlated radiance estimates and the desired error averaging of rough information in the photon map is not achieved. Both of these issues result in the objectionable artifacts especially visible in the corners and on glossy surfaces.

The contribution of our paper consists in formulating the algorithm of *bi-directional photon mapping* (BDPM) capable of handling various scenes with prevailing highly glossy materials without exhibiting the aforementioned artifacts. We do not attempt to address the former issue by increasing the number of photons. Such approaches are described for instance in [4, 5]. Instead, we deal with

the latter by replacing the final gather heuristic by a more principled approach. The original photon mapping performs the radiance estimate from the photon map only at the end of the final gather rays and differ between "global" and "caustic" photon map while we use a combination of various path connection strategies corresponding to a photon map estimate performed at different vertices of the full camera path. Our approach is inspired by Veach's multiple importance sampling technique [1] for adaptive light path connection used in bi-directional path-tracing.

Figure 1 shows an example of two different strategies used for computing the light transport of length three and the result of their combination by BDPM. Image 1a) demonstrates the high variance of radiance estimate on glossy surfaces while the image 1b) shows how this variance exhibits itself as a grainy noise when one level of distribution ray-tracing is used to render diffuse surfaces. Image 1c) demonstrates the superior image quality produced by our bidirectional photon mapping.

The BDPM algorithm is essentially the combination of path-tracing algorithm which follows the light paths from camera and with photon mapping algorithm which trace light paths from light sources. The photon map query is performed in every vertex along the path from the camera. To avoid computing the same light transport multiple times weighting functions summing to unity are used with each photon-camera path pair. This is analogous to the approach taken in BDPT [1, 2].

In the following section we review the Photon Mapping algorithm. To be able to combine various strategies for computing the same light transport we need to treat each connection of a single photon with a path from the camera as a single path. The weighted contribution to the image pixel is computed along that path. To be able to do that, section 2.1 gives a formulation of photon mapping consistent with Veach's path integration framework [1, chapter 4.A]. This formalism allows us not to think about the photon mapping in terms of radiance estimate but rather in terms of individual paths which can be sampled from various strategies. Based on this fact we derive the formula describing the BDPM algorithm in section 3. Section 4 gives an overview of the algorithm and specifies some details about computing path weights. Finally, in section 5 we present our results.

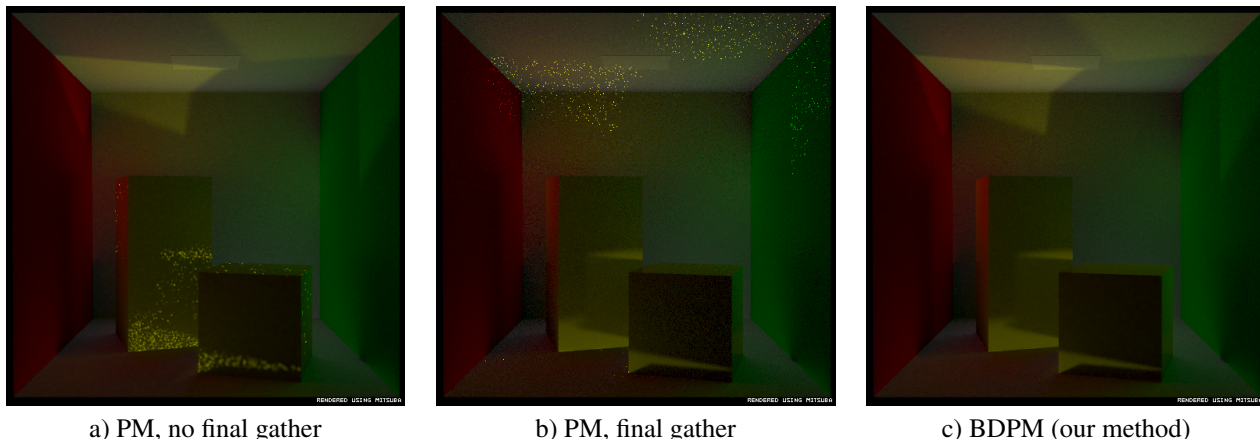


Figure 1: All images show rendering of the same scene and the same light transport of length exactly 3. The scene consists of diffuse walls and two highly glossy objects. The image to the left was rendered without final gather while the image in the middle uses final gather everywhere. Note how each strategy works well for different parts of illumination. The image to the right demonstrates combination of both strategies by our bidirectional photon mapping method.

2 Background: Photon Mapping

The original photon mapping algorithm [3] consists of two passes. In the first pass, little quanta of energy called photons are emitted from the light sources and traced through the scene. When the photon hits a surface we store the information about the hit into a data structure called photon map. At the end of the first pass the information stored in the photon map approximates the overall global illumination of the scene.

In the second pass we cast camera rays into the scene through the image plane to compute the pixel values and whenever the ray hits the surface we can either compute the reflected radiance by means of distribution ray-tracing, i.e. estimating the integral over the hemisphere by casting secondary rays, or we can exploit the information in the photon map and estimate the outgoing radiance value. According to the formula

$$L(x, \omega_o) \approx \frac{1}{\pi r^2} \sum_{p=1}^N f_r(x, \omega_o, \omega_p) \Phi_p(x, \omega_p) \quad (1)$$

where we estimate the outgoing radiance from x in the direction ω_o , the radiance estimate procedure can be interpreted as expanding the sphere around the point x until we catch N photons within the disk of radius r [3]. Each photon coming from the direction ω_p found within the disk contributes by its energy Φ_p multiplied by the BRDF f_r evaluated right at point x . This interpolation step introduces bias into the image which exhibits itself as a low-frequency noise observed as blurriness in the image.

2.1 PM in Terms of Particle Tracing

We describe our bidirectional photon mapping algorithm within the framework introduced by Veach [1]. This allows us not to think in terms of an aggregate radiance estimate but rather in terms of contributions of individual

paths sampled from various strategies. Under these conditions we are able to apply multiple importance sampling for weighting the contribution of various paths. In this section we relate the radiance estimate procedure described by Jensen to the particle tracing characterization described by Veach.

In [1, chapter 4.A] Veach describes particle tracing as a method which generates a set of N sample particle paths ρ_p with their corresponding weights denoted as α_p . From now on to avoid any confusion we will rather use a term photon instead of particle. Sample paths are constructed by following a random walk of a photon through the scene. Their corresponding weights are computed by multiplying the initial energy of a photon in each bounce by the appropriate BRDF value and cosine term and divided by the probability of sampling a new direction and probability of terminating the particle path.

In general each path ρ has its own importance $W_e(\rho)$ for the measurement which the photon is used for. Using Monte Carlo estimation general measurement can be expressed as a weighted sum over sample paths

$$I = E \left[\frac{1}{N} \sum_{p=1}^N W_e(\rho_p) \alpha_p \right]. \quad (2)$$

We implement this measurement as a radiance function value estimation. To estimate $L(x, \omega_o)$ we define $W_e(x, \omega_p) := f_r(x, \omega_o, \omega_p)$. ω_p denotes the direction of the incident photon in point x :

$$L(x, \omega_o) = E \left[\frac{1}{N} \sum_{p=1}^N f_r(x, \omega_o, \omega_p) \alpha_p \right]. \quad (3)$$

In this formulation, photons which end their path out of the point x will yield no contribution.

Last step towards the photon mapping within Veach's formulation is using the biased estimator in form of den-

sity estimation method [6] instead of the unbiased one. Introducing the kernel

$$\kappa(d) = \begin{cases} 0 & \text{if } d > r \\ \frac{1}{\pi r^2} & \text{if } d \leq r, \end{cases} \quad (4)$$

where r is the radius, the formula for the radiance estimation now reads

$$L(x, \omega_o) \approx E \left[\sum_{p=1}^N \kappa(\|x - x_p\|) f_r(x, \omega_o, \omega_p) \alpha_p \right]. \quad (5)$$

The point x_p represents the last vertex on the photon path ρ_p (i.e. the photon position) with corresponding weight α_p . This equation suggests that only those photon paths which end within the support of kernel κ will contribute to the radiance estimate.

Of course, we ended basically with the same formula as we have already stated in (1). The purpose of this section was to put the radiance estimate into the context of Veach's light transport framework that serves as basis for our BDPM formulation.

3 BDPM formulation

Veach in his dissertation presented the path integral formulation which is the measurement equation describing the whole light transport in the form of one non-recursive equation where the computation is concentrated around a geometric primitive - the path [1]. In this framework he derived Bi-Directional Path Tracing originally introduced by Laforge [2], an unbiased algorithm in which he combines paths from light sources and paths from the camera. After connecting one path from the camera to one path from the light source, the contribution of the resulting path is weighted so that no light transport is taken into account more than once. Each strategy, defined by the length of the light and camera paths, is better at computing some parts of the whole light transport while being worse at computing the other. Nevertheless, each of them converges to the correct result. Their combination just exploits their respective advantages and makes the overall light transport computation faster and more robust.

In Photon Mapping we can interpret the radiance estimate procedure as connection of one path from camera to N paths from the light source as opposed to BDPT where paths are connected in a one-to-one manner. To be able to apply Multiple Importance Sampling we need to separate these N paths which share the same camera prefix and then weight each path alone.

In the following section we show the derivation of the formula which describes the BDPM algorithm itself. Overview of the algorithm is given in section 4.

3.1 Derivation

We aim to combine the path-tracing algorithm with the photon mapping. To describe path-tracing we use a stan-

dard Monte Carlo quadrature which describes the measurement of discrete pixel value I_j of pixel j

$$I_j = E \left[\frac{1}{M} \sum_{l=1}^M T_l L_e(x, \omega_o) \right] \quad (6)$$

where T_l is a throughput of the whole camera path ρ_l^* and is computed by multiplying the initial importance value of that path in every bounce by appropriate BRDF and cosine term and divided by the probability of sampling the new direction and by the probability of terminating the path. Point x is the last vertex on the camera path ρ_l^* and ω_o is the direction pointing towards the direction from which the last segment was sampled. L_e is the source radiance function. Note the obvious duality to the particle tracing described in previous section.

Let $M = 1$ so that we have the following one-sample estimator

$$I_j = E [T L_e(x, \omega_o)] \quad (7)$$

and let us consider the scenario where we have only photons which bounced along the paths ρ_p of the same length s segments and its corresponding weights are α_p^s . From the camera point of view we will take into account only one path from camera ρ^* of t segments with its throughput T^t .

Under these conditions we can perform connection of light sub-paths by replacing L_e in the equation (7) by equation (5) so that we compute only the pixel value I_j^n due to the light transport of length $n = s + t$

$$I_j^n \approx E \left[T^t \sum_{p=1}^N \kappa(\|x - x_p\|) f_r(x, \omega_o, \omega_p) \alpha_p^s \right]. \quad (8)$$

We cannot write the equal sign because using the radiance estimate introduces bias into the pixel value estimation. The equation describes the process of tracing the path of fixed length from the camera and then querying the photon map in point x . In the photon map query only those photons are taken into account which traveled along the path of fixed length. End points x and x_p as well as the directions ω_o and ω_p depend on the sampled sub-paths and for simplicity will be omitted in the following formulas. By simple multiplication we get the biased estimator for I_j^n

$$I_j^n \approx E \left[\sum_{p=1}^N \kappa T^t f_r \alpha_p^s \right]. \quad (9)$$

In this formula we put the term T^t inside the sum to emphasize that the formula treats all N paths in the estimator as N separate light paths sharing the same camera sub-path.

Next we will introduce the following notation

$$X_{s,t} := \rho_p^s \rho^{*t} \quad (10)$$

to depict the path consisting of s segments from light source and t segments from camera.

Now we can apply Veach's multiple importance sampling technique (MIS) to combine more estimators in form of (9) so that we account for contributions of paths with various length of sub-paths from camera and from light source. Our multi-sample estimator looks as follows:

$$I_j^n \approx E \left[\sum_{i=1}^n \sum_{p=1}^N w(X_{n-i,i}) \kappa T^i f_r \alpha_p^{n-i} \right], \quad (11)$$

where $w(X_{n-i,i})$ is MIS weight function for path of length n and index i runs through n various strategies where i means the number of segments on the path from camera before it is connected with paths from light.

Knowing how to weight the path contributions we can rearrange the quadrature so that we can use the N -nearest neighbour query into the photon map after tracing the path from camera:

$$I_j^n \approx E \left[\sum_{i=1}^n T^i \left(\sum_{p=1}^N \kappa w(X_{n-i,i}) f_r \alpha_p^{n-i} \right) \right]. \quad (12)$$

The expression in the round brackets is the radiance estimate restricted to fixed photon path length and enriched by MIS weight function which takes into account even the sub-path traced from camera.

It was necessary to fix the sub-paths length to show how to weight the individual path samples. If we consider the fact that

$$I_j = \sum_{i=1}^{\infty} I_j^i \quad (13)$$

and that the sub-path length both from camera and from light are random variables which in the implementation are determined by using Russian Roulette then we can remove the restriction of equal photon path length in the radiance estimate and simplify the formula to

$$I_j \approx E \left[T \left(\sum_{p=1}^N \kappa w(X_{s(p),t}) f_r \alpha_p \right) \right]. \quad (14)$$

Using Russian Roulette for controlling sub-path lengths gives non-zero probability of sampling the light path of any finite length.

In the next section we give an overview of the algorithm which is essentially the evaluation of the formula (14) and we describe how to deal with evaluating the MIS weight function w .

4 Overview

Our algorithm is very similar to the original photon mapping (PM). It consists of two passes. In the first pass we collect photon histories in one photon map (i.e. unlike in PM, there is no special "caustic" photon map and "global" photon map) with additional information about probabilities of sampling photon paths. In the second pass the distribution ray-tracing and the final gather heuristic, as it

was described by Jensen [3], is replaced by standard path-tracing. The special radiance estimate procedure involving weighting each photon contribution is performed gradually at every vertex of the path traced from camera.

The described algorithm is based on the formula (14) and in this section we especially focus on evaluating the weight function w . In next two sub-sections we describe how to compute probabilities needed for evaluating w . Evaluation itself is then described in sub-section 4.3.

4.1 First Pass

We start shooting and tracing the photons as in original PM including usage of Russian Roulette. The only difference is that we need to store probabilities of generated photon paths which are used during the second pass for computing the MIS weights.

Let y_0, \dots, y_s denote vertices of a photon path ρ and let y_0 be a point on a light source. Then we define probability p_k^L of sampling the path y_0, \dots, y_k for $0 < k \leq s$ as

$$\begin{aligned} p_1^L &= p_A(y_0) \cdot p_D(y_0 \rightarrow y_1) \\ p_i^L &= p_{i-1}^L \cdot p(y_{i-1}, y_{i-1} \rightarrow y_i); \quad \forall i \quad 2 < i \leq k \end{aligned}$$

where $p_A(z)$ is the probability of sampling a point z on the light source, $p_D(\omega)$ describes the directional properties of the light source and finally $p(z, \omega)$ is the probability of sampling the direction ω from the point z .

We store the probability p_k^L with each photon hit record in the point y_k . Another information that we need to retain for the second pass is the track of photon bounces within single photon path. For example for a given point y_i we need to be able to backtrack the probabilities $p_{i-1}^L \dots p_1^L$ to evaluate the path weight during the second pass.

4.2 Second Pass

The backbone of the second pass is the path-tracing algorithm. We incrementally construct the path ρ^* from camera using Russian Roulette for controlling the path length. In every vertex we perform a special radiance estimate as described by formula (14) in the round brackets. To be able to evaluate the weight function w we need to determine the probability of sampling the current camera path.

Let us suppose that we have already constructed the path x_0, \dots, x_k and updated the throughput T of that path. Then we need to compute the probability p_k^E of sampling the path for $0 < k \leq t$ as follows:

$$\begin{aligned} p_1^E &= p_A(x_0) \cdot p_D(x_0 \rightarrow x_1) \\ p_i^E &= p_{i-1}^E \cdot p(x_{i-1}, x_{i-1} \rightarrow x_i); \quad \forall i \quad 2 < i \leq k. \end{aligned}$$

Similarly as in the case of light source p_A and p_D depends on the type of camera. Usually when using pinhole camera we set $p_1^E = 1$. The $p(z, \omega)$ is the probability of sampling the direction ω from the point z .

```

proc secondPass(j) ≡
  Ij := 0;
  while (ray := generateRay(j)) do           Ray through the pixel j
    T := Wej(ray);                          source importance of the ray
    p1E := 1;                               1 for case of pinhole camera
    k := 0;
    while ¬absorption(xk) do                apply RR
      k := k + 1;                            prolong the path
      xk := intersect(ray);
      pE = array(p1E, ..., pkE);
      comment: compute pixel contribution
      comment: R is weighted radiance estimate procedure
      Ij := Ij + T · R(radius, k, pE);
      comment: Generate new ray with probability "pdf"
      ray := sampleBRDF(xk, var pdf);
      pdf := pdf * RRpdf;                    includeRRprobability
      comment: update throughput and compute path probability
      T = T * BRDF(xk) * cosθ / pdf;
      pk+1E = pkE * pdf;
    od
  end

```

Figure 2: Pseudo-code for the second pass of our bi-directional photon mapping algorithm. Procedure computes the value I_j of pixel j .

After updating the throughput T and determining the probability p_k^E we can perform the special weighted radiance estimate according to formula (14) and add the contribution to overall pixel value I_j .

If the camera path is not terminated due to application of Russian Roulette we extend the path about new vertex x_{k+1} by tracing the ray from the point x_k and we apply the same procedure as for the vertex point x_k . We prolong the path in the same manner until the absorption of path occurs. The pseudo-code for the second pass is shown in Figure 2.

4.3 MIS Weighted Radiance Estimate

The procedure for estimating the radiance from the photon map is basically the same as in the classical photon mapping algorithm. The biggest difference is that each photon contribution is multiplied by MIS weight dependent both on the given photon sub-path and on the current camera sub-path. In the rest of this section we describe how the MIS weight is computed.

During the algorithm we take path samples from various strategies. The probability of sampling some path x from the strategy where we connected camera sub-path with t segments to photon sub-path with s segments is

$$P_{s,t}(x) = p_t^E p_s^L, \quad (15)$$

where p_t^E is probability of sampling first t segments of path x from camera and p_s^L is probability of sampling the rest s segments from a light source.

For sampling the paths of length n we use $n - 1$ strategies in our implementation since for now we neglect two

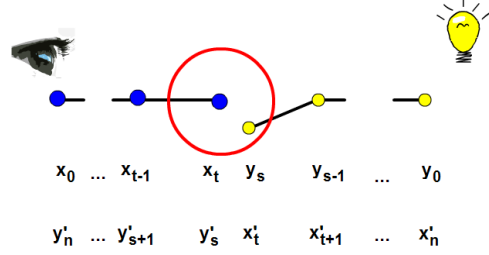


Figure 3: This figure clarifies the extended notation of a sampled path used for defining the sub-path probabilities. The red circle depicts the kernel centered in the point x_t .

corner cases where there is no segment on a camera sub-path or no segment on a light sub-path (see section 6). Using balance heuristic for MIS [1] every strategy has its corresponding weight function

$$w_{s,t}(x) = \frac{P_{s,t}(x)}{\sum_{i=1}^n P_{n-i,i}(x)}. \quad (16)$$

For the path $X_{s,t}$ we simplify the notation to

$$w(X_{s,t}) = w_{s,t}(X_{s,t}) \quad (17)$$

so that we are consistent with section 3.

In previous section we described how to compute sub-path probabilities p_1^E, \dots, p_t^E and p_1^L, \dots, p_s^L . To be able to evaluate MIS function w obviously we need to evaluate $p_{t+1}^E, \dots, p_{n-1}^E$ and $p_{s+1}^L, \dots, p_{n-1}^L$.

Let assume that we have sampled one path

$$X_{s,t} = x_0, \dots, x_t, y_s, \dots, y_0$$

as it was denoted in the preceding text. The camera sub-path starts at x_0 and we currently perform the radiance estimate in its end point x_t . Photon sub-path starts at y_0 and ends at y_s where it was found within the support of the kernel κ . We introduce the redundant notation

$$\begin{aligned} x_0 = y'_n, \dots, x_{t-1} = y'_{s+1}, x_t = y'_s, \\ y_s = x'_t, y_{s-1} = x'_{t+1}, \dots, y_0 = x'_n \end{aligned}$$

so it is possible to easily express how to compute the rest of sub-path probabilities. This notation is clarified in Figure 3.

Then we take special care about probabilities

$$\begin{aligned} p_{s+1}^L &= p_s^L \cdot p(x_t, y_{s-1} \rightarrow y_s, y'_s \rightarrow y'_{s+1}) \\ p_{t+1}^E &= p_t^E \cdot p(x_t, x_{t-1} \rightarrow x_t, x'_t \rightarrow x'_{t+1}) \end{aligned}$$

where we compute the probability in point x_t in both cases. The reason is that the radiance estimate is performed in point x_t and thus the BRDF f_r is evaluated at x_t for the given photon. Note that in the limit case when there was infinite number of photons the points x_t and x_s would merge into one point.

Other sub-path probabilities

$$p_{s+i}^L = p_{s+i-1}^L \cdot p(y'_{s+i-1}, y'_{s+i-2} \rightarrow y'_{s+i-1} \rightarrow y'_{s+i})$$

for $i \in \langle 2, t \rangle$

$$p_{t+i}^E = p_{t+i-1}^E \cdot p(x'_{t+i-1}, x'_{t+i-2} \rightarrow x'_{t+i-1} \rightarrow x'_{t+i})$$

for $i \in \langle 2, s \rangle$

can be computed all in the same manner. They can be even pre-computed and stored in a cache during tracing the camera sub-path and tracing photons respectively because the outgoing and incoming directions are known at that time.

To evaluate MIS weight we need to compute all $\sum_{i=1}^n P_{n-i}(X_{s,t})$. It is easy to evaluate $P_{s,t}(X_{s,t})$ for the current path because from given photon history and camera path-tracing we immediately get p_s^L and p_t^E respectively. The rest sub-path probabilities can be evaluated starting from these ones.

In this stage we know everything to evaluate the weight for a given single path according to formulas (16) and (15) and thus to evaluate the weighted radiance estimate. With this knowledge we can compute the contribution to the pixel value I_j according to formula (14) as it is described in pseudo code in figure 2.

5 Results

We provide renderings of a challenging scene in terms of the number of glossy surfaces and light settings. All test images were rendered on a 1.6GHz Intel Core i7 Q720 using 4 physical cores but 8 logical cores. We ran one thread per logical core.

Figure 5 shows a model of a kitchen where almost every material is glossy. A number of spot lights are used as the only source of the light in the scene. They are all placed on the opposite site of the kitchen behind the camera and all light reaching the shot went through at least one indirect bounce, so there is no direct lighting in the particular shot. To make the conditions even more difficult there are few bottles made of frosted glass placed on the counter. In all images we restricted the path length to a maximum of 7 segments.

These renderings demonstrate the robustness of the BDPM which is its main advantage over the state-of-the-art. PM with final gather will always sample paths only from the one and only strategy and thus it can always be given some scene (as in our case) on which it will perform very poorly.

The image 5a) shows result of path-tracing using 131 072 samples per pixel. Even though it is still noisy it can serve as the reference image to judge which reflections in the images 5b) and 5c) are correct and which are artifacts.

The image 5b) was rendered by standard photon mapping with final gather heuristic. We used one level of distribution ray-tracing (final gather) casting 128 secondary

rays and 32 samples per pixel. This means performing $128 \times 32 = 4096$ radiance estimates per pixel.

To get comparable results we used 512 samples per pixel for rendering the image 5c) which was rendered by BDPM so we would perform $512 \times 7 = 3584$ radiance estimates and thus took approximately the same number of measurements in both 5a) and 5b) and run the algorithms for the same amount of time. In both cases we used 2 million photons, nearest-neighbour density estimate in the photon map query with the conical kernel [3].

We can see that in the image 5c) the severe noise on the cupboard vanished completely. Important improvement is also observable on the tea pot. Unlike the image 5c) on the image 5b) in the counter there is not visible the reflection of the main reflection visible on the tea pot. Given the circumstances this will not get better in the classical photon mapping algorithm even if we used more samples in the distribution ray-tracing. The reason is almost specular surface of the tea pot. We would have to dramatically increase the number of photons in the map to get better results in 5b). Another noticeable issue are some completely missing reflections.

In figure 6 we present renderings of a Cornell box-like scene. Diffuse materials are used for walls and highly glossy materials for the blocks. The images were rendered with paths of length exactly three (i.e. $n = 3$). Image 6a) represents the strategy where $t = 1$ and $s = 2$ while the image 6b) shows the result of using strategy where $t = 2$ and $s = 1$. Images 6c) and 6d) were rendered by bidirectional photon mapping where both of the latter strategies were combined together. We used 20 million photons to render the combined image and 512 samples per pixel. In image 6c) we used fixed radius for every photon map query while in the image 6d) we used the nearest-neighbour estimation method and conical kernel. This demonstrates that using varying radius through the scene does not yield objectionably worse results.

6 Conclusions and Future Work

We have presented a robust bidirectional photon mapping algorithm for computing global illumination. The algorithm is capable of rendering scenes with many glossy materials where the original photon mapping algorithm produces objectionable artifacts. The algorithm takes advantage of combining various strategies for computing the same light transport.

In our implementation we ignore strategies where the path is generated entirely from the light source or from the camera. Taking these paths into account will be addressed in future work.

One reason for artifacts in glossy scenes is the high variance of radiance estimate on glossy surfaces. This can be improved by shooting an enormous number of photons. Nevertheless, original photon mapping is limited by physical memory boundaries so we cannot use sufficient num-

Algorithm	Samples per Pixel	Final Gather Samples	Resolution	Rendering Time
PT	131 072	-	320x320	1.4 day
PM	32	128	320x320	4 hours
BDPM	512	-	320x320	4.1 hours

Figure 4: Information about renderings in figure 5.

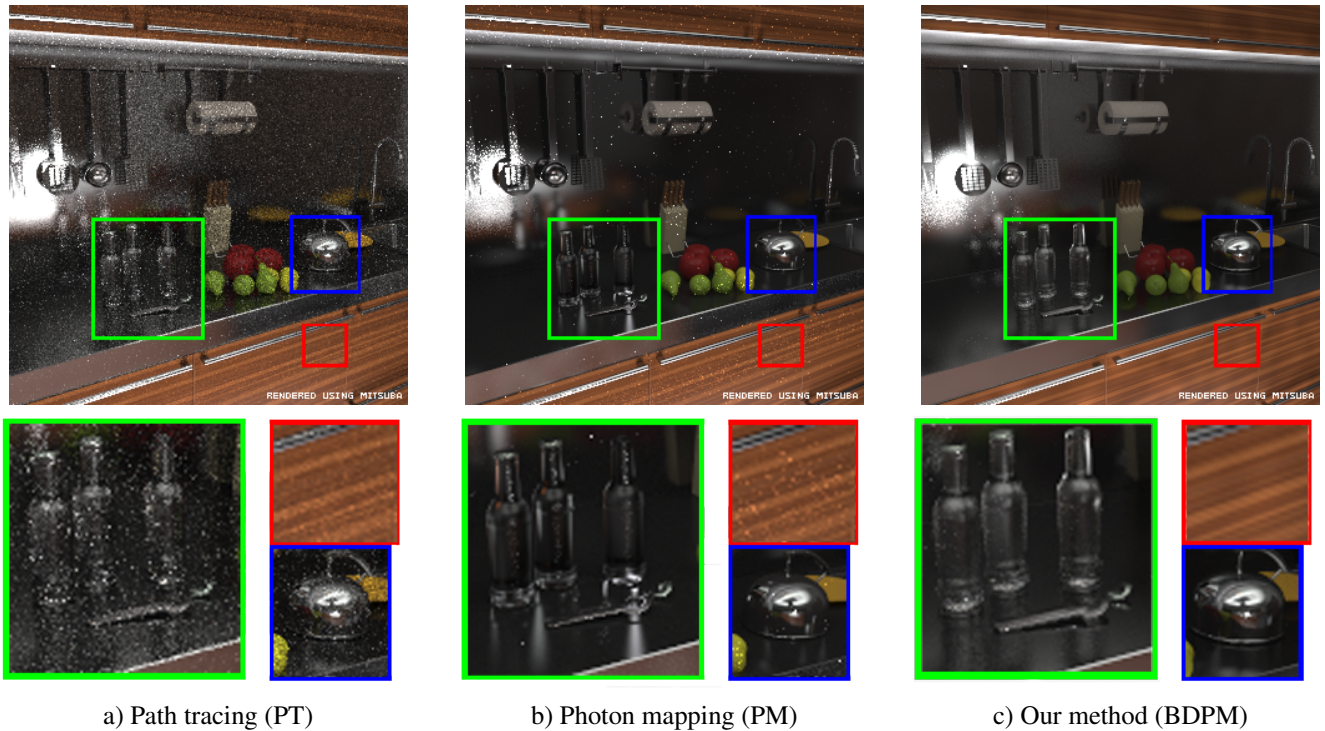


Figure 5: There is a comparison of rendering the same scene by PT, PM and BDPM. We let PM and BDPM run for the same time. Both PM, BDPM use 2 million photons, nearest-neighbour density estimate and conical kernel. Noisy reference image was rendered by PT with 131 072 samples per pixel.

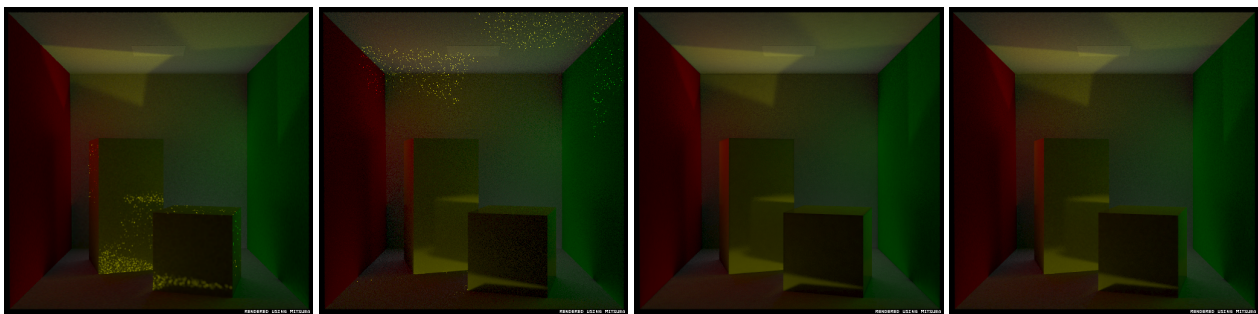


Figure 6: All images show rendering of the same scene and the same light transport of length 3. There are diffuse walls and two highly glossy objects. The image a) was rendered without final gather while the image b) uses final gather everywhere. Note how each strategy works well for different parts of the illumination. Images c) and d) demonstrate combination of both latter strategies by bidirectional photon mapping method. In c) we used fixed radius in density estimate through the whole scene while in d) we allowed varying radius and used conical kernel.

ber of photons. These boundaries are addressed by progressive photon mapping approaches [4, 5]. In future work we would like to examine the possibilities of bidirectional photon mapping within these progressive frameworks.

Acknowledgements

Many thanks to Wenzel Jakob for providing his Mitsuba Renderer. I am no less grateful to Jaroslav Krivánek for his supervision, numerous advices, ideas and text corrections.

References

- [1] Eric Veach, *Robust Monte Carlo Methods For Light Transport Simulation*. Ph.D. Dissertation, Stanford University, 1997.
- [2] Eric P. Lafortune, Yves D. Willems, *Bi-Directional Path Tracing*. Proceedings of Compugraphics '93, Alvor, Portugal (December 1993), pp. 145-153.
- [3] Henrik Wann Jensen, *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.
- [4] Toshiya Hachisuka, Shinji Ogaki, Henrik Wann Jensen, *Progressive Photon Mapping*. ACM Trans. Graph. 27,5, 2008.
- [5] Toshiya Hachisuka, Henrik Wann Jensen, *Stochastic Progressive Photon Mapping*. ACM SIGGRAPH Asia, 2009.
- [6] Shirley P., Wade B., Hubbard P. M., Zareski D., Walter B., Greenberg D. P. *Global illumination via density-estimation*. Eurographics Rendering Workshop 1995 Proceedings, pp. 219–230.
- [7] Y. Tokuyoshi, *Photon density estimation using multiple importance sampling*, ACM SIGGRAPH ASIA 2009 Posters, pp. 37:1–37:1.