DH2323 Computer Graphics and Interaction
Bidirectional Path Tracer
Project Report

Simon Sirak (`ssirak`) (`ssirak@kth.se`)

Abstract

In this paper, a Bidirectional Path Tracing (BDPT) implementation is presented in detail. The system features spherical area lights and lambertian surfaces, and can render spheres and triangle-meshes. The BDPT uses a balance heuristic for multiple importance sampling. The implementation showed good results, requiring relatively few samples to achieve an image of good quality. The system can be further improved by adding support for specular reflection and refraction, and by using a lens-based camera model. This would allow for caustics and depth-of-field effects to be properly rendered.

1 Introduction

In computer graphics, physically based rendering is the task of approximating (global) illumination using physically based models, for instance based on geometric optics. One well-known equation that is often approximated is the Light Transport Equation, originally introduced by (Kajiya, 1986), which describes the outgoing radiance on a surface point as the sum of the emitted radiance and scattered radiance at the surface point.

More formally, for each pixel j of an image, we can calculate the radiance incident to that pixel using the Path Integral formulation of the Light Transport Equation (LTE). The equation is defined as:

$$I_j = \int_{x \in \Omega} f_j(x)\, d\mu(x) \tag{1}$$

Where Ω is the space of all possible paths (of all possible lengths) in the scene, f_j is the measurement contribution function, and $d\mu(x)$ is the differential area-product measure. f_j for a certain path is defined as follows:

$$\begin{aligned} f_j(x) = {} & L_e(x_0 \to x_1) G(x_0 \leftrightarrow x_1) W_e(x_{k-1} \to x_k) \\ & \prod_{i=1}^{k-1} f_s(x_{i-1} \to x_i \to x_{i+1})\, G(x_i \leftrightarrow x_{i+1}) \end{aligned} \tag{2}$$

Where k is the length of the path x, G is a geometric term and W_e is the importance function of pixel j (W_e for a certain pixel is related to how much of the light reaching a point on the pixel's film on the camera is actually registered by the film).

Bidirectional Path Tracing is a physically based global illumination technique that aims to approximate the LTE. It generates two subpaths; one from the camera (eye path) and one from a light source (light path). These are then connected in different ways, resulting in many different eye-to-light path samples where each sample correspond to its own path sampling strategy. The sample contribution $C_{s,t}$ from each path sample is then calculated and added together using multiple importance sampling, resulting in a multi-sample estimator F of the LTE integrand:

$$\begin{aligned} F &= \sum_{s \geq 0} \sum_{t \geq 0} C_{s,t}(x_{s,t}) \\ &= \sum_{s \geq 0} \sum_{t \geq 0} w_{s,t}(x_{s,t}) \frac{f_j(x_{s,t})}{p_{s,t}(x_{s,t})} \end{aligned} \tag{3}$$

Where $w_{s,t}$ is the weighting function for the sampling strategy (s,t), which has a corresponding probability density function $p_{s,t}$ for sampling from this strategy. A path from path sampling strategy (s,t) can be denoted $x_{s,t}$ and has s vertices on the eye path and t vertices on the light path. An approximation of the LTE is then obtained by applying the estimator F to Monte Carlo Integration:

$$I_j \approx \frac{1}{N} \sum_{i=1}^{N} F_i \tag{4}$$

2 Related Work

Below are the most important references that were used when implementing and studying concepts regarding physically based rendering, sampling and BDPT.

(Pharr et al., 2016) present multiple topics related to physically based rendering in an intuitive order. The award-winning book has served as a useful source of information on the fundamentals of physically based rendering throughout this project, clarifying topics such as Radiometry, Sampling and the Light Transport Equation. It was also used as a reference for the implementation of the weighting heuristic used for multiple importance sampling in the BDPT implementation of this project.

(Veach, 1997) thoroughly presents the core concept of multiple importance sampling and its relation to Monte Carlo Integration and BDPT. The results of Veach have served as the main source for understanding multiple importance sampling as a concept. The robustness of his results has led to application of multiple importance sampling in many state-of-the-art renderers even today.

(Lafortune and Willems, 1993) present the first instance of a BDPT. The report was used as an initial source for understanding which implementation details are generally involved in a BDPT.

3 Implementation

Below is a description of the different choices made during implementation. For the curious reader, a more thorough explanation of the implementation process (including theory summaries) can be found in (Sirak, 2019).

3.1 Algorithm Outline

The pseudo code below shows the overall outline for the BDPT algorithm.

```
for i = 1 to N:
    for each pixel:
        eyePath = GenerateEyePath()
        lightPath = GenerateLightPath()
        F = connect(eyePath, lightPath)
        color[pixel] = (color[pixel]
                        * (i-1) + F) / i
```

The outer for loop iterates through N evaluations of the multi-sample estimator. The inner for loop iterates through each pixel. For each pixel, the eye paths and light paths are generated. The connect() procedure then computes a multi-sample estimator. This is done by computing the sample contribution for all of the path sampling strategies that can be generated by connecting the subpaths in different ways, and then sums them according to the multi-sample estimator shown in section 1 (connections always occur by adding a single edge and taking the only eye-to-light path created). The multi-sample estimator calculated is then added to the current pixel color in the last line. This line is essentially the Monte Carlo Integration, but restructured in order to show the convergence of the BDPT in each iteration.

Below are some decisions/issues that needed to be made/resolved when generating the subpaths and computing the multi-sample estimator.

3.2 Path Sampling Strategies

The path sampling strategies considered were based on the limitations of the camera model used. Because the BDPT was implemented using a pinhole camera, any path sampling strategy that connects the eye subpath with the light subpath directly from the camera vertex would make little sense (only a very specific ray shot towards the eye can actually contribute to the pixel, namely the ray we shoot when generating an eye path). As such, this path sampling strategy was not considered (other BDPT implementations with more complex implementations include this path sampling strategy, allowing for added effects to be rendered (Veach, 1997) (Pharr et al., 2016)).

For the BDPT implementation of this project, the path sampling strategies considered for calculating a multi-sample estimator were the following (s is the number of vertices from the eye path used, t is the number of vertices from light path):

1. $s \geq 2, t = 0$, i.e any path concatenated some portion of the eye path and none of the light path. This essentially only contributes to the multi-sample estimator if the last vertex of the eye path is indeed a light source, and has the important effect of rendering the actual light source onto the scene (if it part of a path).

2. $s \geq 2, t \geq 1$, i.e any path concatenated using some portion of the eye path and some portion of the light path. This captures both direct lighting ($s = 2, t = 1$) and indirect lighting. Some implementations use specialized direct lighting procedures for those cases (for optimization purposes), but I decided to treat them all the same way (this does not harm correctness, only slows down convergence of the image).

3.3 Probability Density Functions

The PDF:s for all directional choices made along the eye path and light path were according to uniform hemisphere sampling. If I had implemented specular reflection, another useful PDF to use would have been a cosine-weighted hemisphere sampling. Cosine-weighted hemisphere sampling favors generated directions that are similar to a specified axis, which can be useful in specular reflection in order to reduce noise. However, due to time constraints, I did not have time to implement specular reflection.

3.4 Weighting Heuristic

The weighting heuristic used was the balance heuristic, as described in chapter 9 of (Veach, 1997):

$$w_{s,t} = \frac{n_{s,t} p_{s,t}(x)}{\sum_{i=1}^{s+t} n_{i,s+t-i} p_{i,s+t-i}(x)}$$

Because BDPT takes 1 sample per path sampling strategy, this is reduced to:

$$w_{s,t} = \frac{p_{s,t}(x)}{\sum_{i=1}^{s+t} p_{i,s+t-i}(x)} \tag{5}$$

The balance heuristic is provably good in the sense that any other weighting function results in at most a constant term of less noise compared to the balance heuristic.

3.5 Bidirectional Reflectance Distribution Functions

Currently, the BDPT implementation considers only lambertian surfaces. This means the BRDF of a surface is given by its reflectance ρ:

$$f_r(x, w_o, w_i) = \frac{\rho}{\pi}$$

Specular reflection could not be implemented due to time constraints, but it is relatively straight forward to add this to the BDPT.

3.6 Calculating the Sample Contribution

The implementation of calculating the sample contribution was heavily inspired by the description in chapter 16.3 of (Pharr et al., 2016).

As seen in equation 3, the sample contribution can be written as:

$$C_{s,t}(x_{s,t}) = w_{s,t}(x_{s,t}) \frac{f_j(x_{s,t})}{p_{s,t}(x_{s,t})} \tag{6}$$

$p_{s,t}$ is the probability of sampling a path according to path sampling strategy (s, t), and is defined by the product of the probabilities of the choices made to generate the path (e.g directional samples). In my implementation, the sample contribution $C_{s,t}$ is pre-calculated incrementally by doing the following when adding a new vertex in the procedure for generating subpaths:

1. Multiply the sample contribution constructed up to this vertex with the BRDF at this intersection.
2. Multiply the sample contribution at this vertex with the geometry term G between this and the previous vertex.
3. Divide the sample contribution at this vertex by the probability of choosing a certain direction (the probability should be converted to an area-measured probability using a conversion factor, in order to be compatible with the differential in the LTE integral).

At the end of the subpath generation, each vertex therefore has an unweighted sample contribution as constructed up to that vertex. (1) and (2) have the effect of calculating the expanded series of multiplications in the f_j (see equation 2). (3) has the effect of dividing the measurement function by the probability density function of that particular path sampling strategy. The ”base cases” in pre-computing f_j are calculated using the camera and light endpoints (just as specified in f_j). For this project, the importance function W_e for each pixel was set to 1, since the starting ray from the eye path was computed in a deterministic manner to always go through the same point on the film of a given pixel. Other implementations, particularly those with more realistic lens-based camera models, have a more complex importance function which allow added effects to be rendered (Pharr et al., 2016).

In order to calculate the final unweighted sample contribution of a particular path, we need to multiply the pre-calculated contributions from the two points that connected the eye and light subpaths, and multiply that by the BRDF of the two vertices as if they were connected to each other (since those BRDF:s cannot be pre-calculated). We then multiply that with the geometry term G between the two connecting vertices. This yields the unweighted sample contribution, which is then multiplied by the weighting function in order to yield the final sample contribution. The full calculation of the sample contribution is therefore computed in the path-connection procedure according to:

$$\begin{aligned} C_{s,t} = vs.C \cdot vt.C \cdot G(vs \leftrightarrow vt) \\ f(vs.prev \to vs \to vt) \\ f(vs \to vt \to vt.prev) \\ w_{s,t}(x) \end{aligned} \tag{7}$$

Where x is the path consisting of the eye subpath y and light subpath z, $vs \in y, vt \in z$ are the last vertices on each subpath and $prev$ is the second-to-last vertex generated for each respective subpath.

3.7 Calculating the Weighting Function

At each vertex, the probability of hitting the next vertex is stored (a “forward” probability). Each vertex also stores the probability of the next vertex hitting the current vertex (a “reverse” probability). Both of these probabilities are stored as area-measured probabilities, in order to be compatible with the differential in the LTE integral. The reverse probability is naturally calculated on the next iteration of path construction, when we actually have access to the “next” vertex. The forward and reverse probabilities are then used to calculate the balance heuristic defined in section 3.4.

The actual calculation is done as specified in chapter 16.3 of (Pharr et al., 2016). The technique involves rewriting the weighting function and utilizing a recursive relation between the two probabilities stored for each vertex. This leads to a relatively fast evaluation of the balance heuristic.

4 Results

This section describes some resulting renders that were achieved using the BDPT. All images approximate LTE by Monte Carlo Integration on 75 evaluations of the multi-sample estimator, with a 400x400 resolution, on an Intel i7-7700K CPU (on Ubuntu). With this setup, a full render took approximately 150 seconds.

Figure 1 shows two images only varying in the size of the light source. It can be seen that the larger light source produces a small amount of fireflies in the resulting render, most likely due to obscure light ray directions shot from the surface of the spherical light. For a smaller light source, the direction can approximately be viewed as being taken from the center of the sphere. Therefore, a smaller light source generates less fireflies.

Also note that the overall level of illumination between the images in figure 1 is about the same, suggesting that the BDPT works correctly. A regular Path Tracer would render a significantly darker image for a smaller light source since it only samples paths from the eye. A BDPT implementation also includes strategies that connect light sources to the eye path, which efficiently spreads the light from the light sources onto the scene.

Figure 2 shows a render with multiple light sources. Again, the overall illumination level is consistent with the previous image, suggesting implementation correctness. There are also no fireflies present. Furthermore, the light interaction can be seen in the shadows that are cast; shadows for the boxes now appear in multiple places, but are lighter than in the previous images. This is because each light source casts some light onto a shadow created by the other light source.

5 Perceptual Study[1]

Although Bidirectional Path Tracing is often considered a well-balanced and consistent rendering algorithm, there are certain lighting scenarios where other rendering techniques excel in terms of reduced amount of noise. Identifying the scenarios in which renders from a BDPT are perceived as noisy could be a good way to find new path sampling strategies that could included in the multiple importance sampling.

One way to identify for which scenes a BDPT-rendered image is perceived as noisy or flawed is to consult individuals through a perceptual study. Below is a description of a potential perceptual study that investigates this.

5.1 Design

The perceptual study will be based on a within-subject, controlled experiment, where subjects will observe a reference image and a BDPT-rendered image and report whether a noticeable difference was observed.

[1] I participated in Maha El Garfs perceptual study on virtual characters and personalities!

5.1.1 Stimuli

The study described here will focus on investigating the consistency in achieving perceived realism using BDPT for four scenarios:

1. Closed room, one visible (small) light source, only diffuse reflection.
2. Closed room, one occluded (small) light source, only diffuse reflection.
3. Closed room, one visible (small) light source, a mix of diffuse and non-diffuse reflection.
4. Closed room, one occluded (small) light source, a mix of diffuse and non-diffuse reflection.

Further scenarios could be investigated in other studies, such as outdoor scenes or scenes with refraction.

5.1.2 Hypothesis

The perceived realism from scenes rendered by BDPT can vary depending on the lighting and the materials involved in the scene.

5.1.3 Metrics

The metric used to evaluate the results of each scenario in the study is the number/ratio of subjects that observed a difference between the two images. In order to reduce experimenter and response bias, the question posed should be the same for all scenarios, and should not be suggestive or aggressive. One potentially good question for each scenario could be:

> Was there a noticeable difference between the two images?

5.2 Procedure

The perceptual study will be based on a within-subject controlled experiment. Each subject will observe multiple different scenes. For each scene, the subject will observe a reference image and an image rendered by BDPT using a predetermined number of samples per pixel (using the same number of samples for each scenario). Then, the subject will mark whether they observed a difference between the two images.

For each scenario, the images will be shown on two different monitors (of the same resolution) spaced 1 meter apart. Furthermore, the subject will sit about 1 meter away from the plane of the two monitors. The subject will view the image for 5 seconds before that scenario ends. These precautions allow the subject to view the images clearly, while ensuring that the subject observes the images in a more natural way instead of deliberately looking for minute differences.

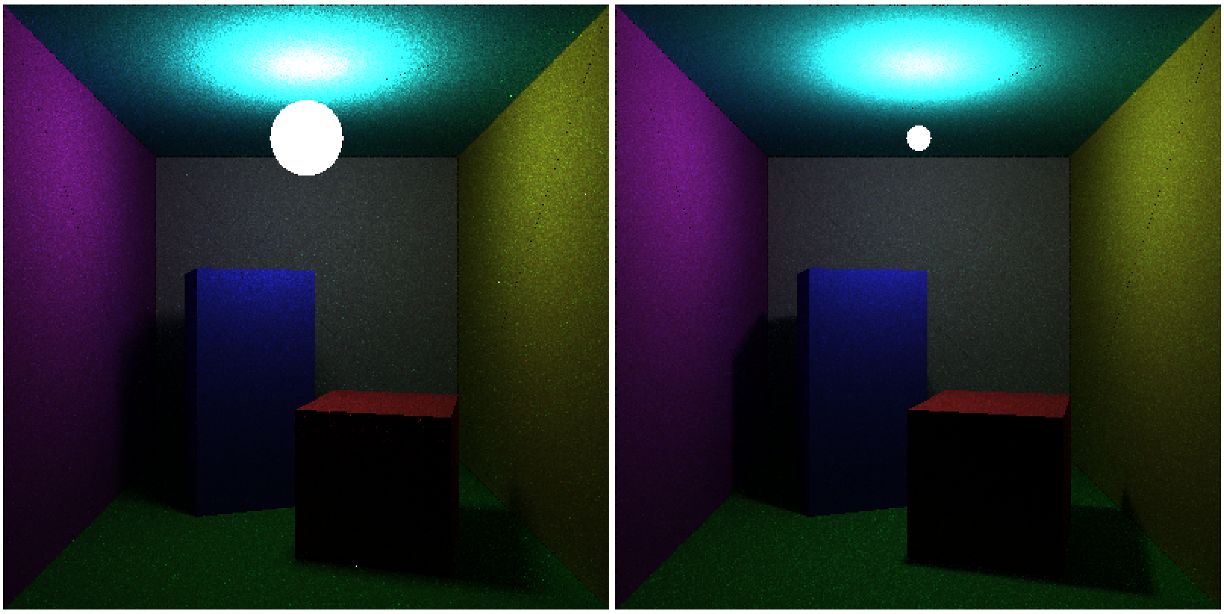

Figure 1: From left to right; Cornell box rendered with large and small light source respectively.

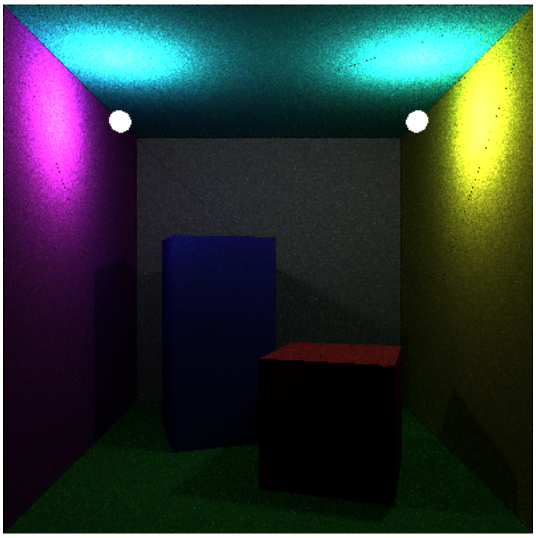

Figure 2: Cornell box rendered with two small light sources.

5.3 Expectations

Scenes with indirect illumination (through occluded light sources) and diffuse surfaces are more likely to be perceived as different from a reference image. Scenes with visible light sources are less likely to be perceived as different from a reference image. This hypothesis is drawn from personal experience during implementation of this BDPT algorithm.

6 Future Work

The BDPT implemented in this project can be extended in many ways. One extension could be to add materials that exhibit more than lambertian properties, such as specular reflection and refraction or a spatially varying BRDF. This would allow the system to render caustics, as seen in other BDPT implementations (Lafortune and Willems, 1993) (Vlnas, 2018). Another extension could be to implement a more sophisticated, lens-and-film-based camera model instead of the pinhole camera model currently used. This would allow for effects such as depth of field to be modeled accurately, and would also provide a realistic interface for users to configure the camera with certain settings (a realistic camera interface and its impact on physically based rendering can be seen in chapters 6, 7 and 16 of (Pharr et al., 2016)). A third extension could be connected to the perceptual study described in the project specification, i.e to find more path sampling strategies that would be beneficial to include in the multi-sample estimator for BDPT.

References

James T Kajiya. 1986. The rendering equation. In *ACM SIGGRAPH computer graphics*, volume 20, pages 143–150. ACM.

Eric P Lafortune and Yves D Willems. 1993. Bi-directional path tracing.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically based rendering: From theory to implementation*. Morgan Kaufmann.

Simon Sirak. 2019. DH2323 Project: Bidirectional Path Tracer. `https://bidirectionalpathtracer.blogspot.com/`. Accessed: 2019-05-25.

Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation*, volume 1610. Stanford University PhD thesis.

Michal Vlnas. 2018. Bidirectional path tracing.