Project Specification: Bidirectional Path Tracing

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

May 2019

1 Background

Classical Path Tracing, as introduced by Kajiya [1986], is an algorithm designed to estimate the Light Transport Equation (LTE). It computes a Monte Carlo estimate by sampling a large number of paths, constructed from the camera and built until a max depth or a light source is found. This technique converges to a correct image, but requires quite a lot of samples in order to reduce noise to an acceptable level.

Many algorithms have been designed since then in order to reduce noise. One of the more successful algorithms is Bidirectional Path Tracing (BDPT), introduced by Lafortune and Willems [1993] and later refined by Veach [1997]. BDPT aims to approximate the LTE by using *multiple importance sampling*, which means that a *multi-sample estimator* is calculated from contributions of many different path sampling strategies, as opposed to the single-sample estimator used for conventional Monte Carlo Integration (which only samples from one path sampling strategy). BDPT has great documented improvements over regular path tracing; the wide array of path sampling strategies that are sampled gives incredible flexibility in BDPT that makes it great in many scenarios.

The aim of this project is to implement a BDPT. Understanding this technique is a step towards understanding more about the strengths and weaknesses of different physically based rendering. Furthermore, this project will help tremendously in increasing my knowledge in physically based rendering.

2 Deliverable

The deliverable implementation will contain the following features:

- A functioning BDPT implementation;
- Multiple Importance Sampling as described in Veach [1997], Pharr et al. [2016];
- Lambertian Surfaces (only diffuse reflection).

The stretch goals (which will be investigated based on available time) are the following features:

- Specular reflection;
- Refraction;

3 Implementation Method

The code will be written in C++11 and will build upon Lab 2 of the Rendering track, because SDL allows for per-pixel manipulations. This means the Cornell Box from Lab 2 will be used as the demonstration model. Further models will not be searched for due to time constraints, however addition of spherical shapes will be done if it helps illustrate certain features of the Bidirectional Path Tracer.

The theoretical background for implementing the different components of a physically based renderer (Sampling, Monte Carlo integration etc.) will be gathered from Pharr et al. [2016], due to their award-winning and intuitive explanations of fundamental topics in physically based rendering. Furthermore, [Veach, 1997, Pharr et al., 2016] are used as references to understand the actual BDPT algorithm.

References

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

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

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

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