Universidade Estadual de Campinas
Instituto de Computação

João Vítor Buscatto Silva

Importance sampling light sub-paths on Bidirectional Path Tracing

Amostragem de importância de subcaminhos de luz em Bidirectional Path Tracing

CAMPINAS
2022

João Vítor Buscatto Silva

Importance sampling light sub-paths on Bidirectional Path Tracing

Amostragem de importância de subcaminhos de luz em Bidirectional Path Tracing

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Dissertação defendida por João Vítor Buscatto Silva e orientada pelo Prof. Dr. Hélio Pedrini.

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Si38i

Silva, João Vítor Buscatto, 1996-
Importance sampling light sub-paths on Bidirectional Path Tracing / João Vítor Buscatto Silva. – Campinas, SP : [s.n.], 2022.

Orientador: Hélio Pedrini.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Computação gráfica. 2. Algoritmos Ray tracing. I. Pedrini, Hélio, 1963-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Amostragem de importância de subcaminhos de luz em Bidirectional Path Tracing
Palavras-chave em inglês:
Computer graphics
Ray tracing algorithms
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Hélio Pedrini [Orientador]
Afonso Paiva Neto
Aurea Rossy Soriano Vargas
Data de defesa: 13-04-2022
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-1676-9485
- Currículo Lattes do autor: http://lattes.cnpq.br/6281010361065162

Universidade Estadual de Campinas
Instituto de Computação

João Vítor Buscatto Silva

Importance sampling light sub-paths on Bidirectional Path Tracing

Amostragem de importância de subcaminhos de luz em Bidirectional Path Tracing

Banca Examinadora:

- Prof. Dr. Hélio Pedrini
  IC/UNICAMP
- Prof. Dr. Afonso Paiva Neto
  ICMC/USP
- Profa. Dra. Aurea Rossy Soriano Vargas
  IC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 13 de abril de 2022

Acknowledgements

I would like to express my appreciation to Dr Chakravarty R. Alla Chaitanya, one of the main authors of the MBDPT paper, for providing the source code of his method. I would also like to thank Ms. Ana Dodik and Mr. Alan Wolfe for their useful insights in earlier stages of the project.

I thank my advisor, Prof. Dr. Hélio Pedrini, for his patience, insights, suggestions, and all the help provided over the past several years, since my first undergraduate research project, in 2015.

I would like to thank my friends Alan Trang, Apollo Xiang, Felipe Bonfietti, Igor Hitzschky, Jesse Patterson, José Ferreira, Julianny Favinha, Kaique Monteiro and Viviane Tonetto, to name a few, for their extended support and patience over the course of my Master's degree.

I would also like to thank my coworkers and friends from the Darmstadt Graphics Group (DGG) GmbH, especially our CTO Max Limper, for their support during the last few months of dissertation writing process.

Last but not least, I would like to thank my parents, Nélida and Valmir, my grandmother, Nathalina, my uncle José and the rest of my family for their support, patience and help ever since my first school year.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

Resumo

Algoritmos de *Ray Tracing* são amplamente utilizados como a principal técnica de renderização de cenas complexas quando se busca realismo e precisão na simulação de luz. As técnicas baseadas em *Bidirectional Light Transport* são conhecidas por se destacarem em cenas com caminhos de luz complexos, o que é possível porque os caminhos que saem da câmera da cena e os caminhos que saem dos emissores de luz são gerados independentemente e, em seguida, conectados deterministicamente. Essas abordagens, no entanto, ainda apresentam questões centrais inerentes à maioria dos métodos de Monte Carlo. As amostras geradas por eles podem ser aglutinadas infinitamente, o que não oferece uma representação precisa do domínio. As conexões determinísticas geralmente são geradas aleatoriamente, em contraste com a forma com que os subcaminhos, tanto da câmera quanto dos emissores, são amostrados de acordo com sua importância para a cena. Pesquisas recentes em algoritmos tradicionais de transporte de luz bidirecional mostraram que outras formulações mais complexas de conexões de caminho são eficazes para permitir a estratificação e amostragem de baixa discrepância no espaço de conexão onde os caminhos são gerados, o que permite uma distribuição uniforme e coerente de caminhos igualmente contribuintes. Mostramos que amostrar a importância da taxa de transferência de subcaminhos de luz resulta em melhor redução de variância sob certas condições. As alterações propostas foram aplicadas ao algoritmo *Bidirectional Path Tracing*, em uma variação que chamamos de *Alias Sampled Bidirectional Path Tracing* (ASBDPT). Experimentos realizados em uma série de cenas 3D para avaliar a eficácia da metodologia mostraram ganhos de redução de variância em cenas com exposição limitada à luz, ao comparar os resultados com métodos do estado da arte da área.

Abstract

Ray tracing algorithms are widely used as the main rendering technique for complex scenes when aiming for realism and precision in the light simulation. Bidirectional light transport based techniques are known to excel in scenes with complex light paths, which is possible because the paths leaving the scene camera and paths that leave the light emitters are generated independently and then connected deterministically. These approaches, however, still have core issues inherent to most Monte Carlo methods. The samples generated by them can be infinitely clumped together, which does not offer an accurate representation of the domain. The deterministic connections are usually generated randomly, compared to how the sub-paths, both from the eye and emitters, are well importance sampled. Recent research in traditional bidirectional light transport algorithms has shown that other more complex formulations of path connections are effective in enabling stratification and low-discrepancy sampling in the connection space where the paths are generated, which allows for an even and coherent distribution of equally contributing paths. We show that importance sampling the throughput of light sub-paths results in improved variation reduction under certain conditions. The proposed changes have been applied to the Bidirectional Path Tracing algorithm, in a variation we call Alias Sampled Bidirectional Path Tracing (ASBDPT). Experiments conducted on a series of 3D scenes to evaluate the effectiveness of the methodology have show variation reduction gains on scenes with limited light exposure, when comparing the results against current state-of-the-art methods.

List of Figures

List of Tables

List of Abbreviations and Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
ASBDPT	Alias Sampled Bidirectional Path Tracing
BDPT	Bidirectional Path Tracing
BSDF	Bi-directional Scattering Reflectance Distribution Functions
CBDPT	Combinatorial Bidirectional Path Tracing
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DDR	Double Data Rate
GI	Global Illumination
GPU	Graphics Processing Unit
IS	Importance Sampling
LDS	Low-Discrepancy Sampling
MBDPT	Matrix Bidirectional Path Tracing
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MIS	Multiple Importance Sampling
MLT	Metropolis Light Transport
MMLT	Multiplexed Metropolis Light Transport
PCBDPT	Probabilistic Connections for Bidirectional Path Tracing
PDF	Probability Density Functions
PMF	Probability Mass Functions
PSSMLT	Primary Sample Space Metropolis Light Transport
PT	Path Tracing
RAM	Random Access Memory
ReSTIR	Reservoir-based Spatio-Temporal Importance Resampling
RIS	Resampled Importance Sampling
RMSE	Root Mean Square Error
SSIM	Structural Similarity Index Measure

Contents

Chapter 1

Introduction

This chapter presents the sub-path importance sampling problem, its challenges, main goals, hypothesis contributions of the dissertation, and the text organization.

1.1 Problem Description

Monte Carlo Ray Tracing has become the most widely used type of photo-realistic rendering technique. The seminal paper "The Rendering Equation" [16] introduced a technique that is known today as Path Tracing (PT). Using Monte Carlo integration, the method had a mathematical formulation that allowed complex light effects, such as motion blur and indirect illumination, to be rendered easily [28, 29]. It is the basis for most light transport algorithms that followed. One of its limitations, however, is the handling of complex light paths, usually from obstructed light sources.

Bidirectional light transport methods attempt to tackle this inefficiency, by sampling partial paths starting both from the light sources, from here on out called light sub-paths, and starting from the camera, or eye sub-paths, followed by a deterministic step that connects all possible pairs. Bidirectional Path Tracing (BDPT) was proposed independently by both Lafortune and Willems [21] and Veach and Guibas [50]. In practice, both proposed a method where paths are traced simultaneously and then connected by shadow rays in a deterministic way.

Veach and Guibas' version of the algorithm had some considerable differences, as it introduced the concept of Multiple Importance Sampling (MIS), allowing the resulting image variance to be considerable reduced. By using MIS, Veach and Guibas were able to sample different Probability Density Functions (PDFs), while keeping the render unbiased. The authors proposed a balance heuristic to weight the PDF values when sampling with MIS that, while not always optimal in practice, produces satisfactory results in most cases.

Figure 1.1 shows examples of the same scene rendered for a specific sub-path connection strategy only. That is, we render the same scene several times, but only connect sub-paths of a certain eye and light length each time. With t standing for the length of the camera sub-path, and s standing for the length of the light sub-path, we can consider connections with $t = 1$ as the 'light tracing' algorithm, where we strictly trace rays of

light from an emitter and attempt to reach the eye or camera. On the other hand, connections with $s = 0$ can be considered as the standard Path Tracing algorithm without next event estimation. This visualization is useful to showcase the main strength of the BDPT algorithm: several details of the rendered image would not be achieved otherwise, or would require a much longer execution.

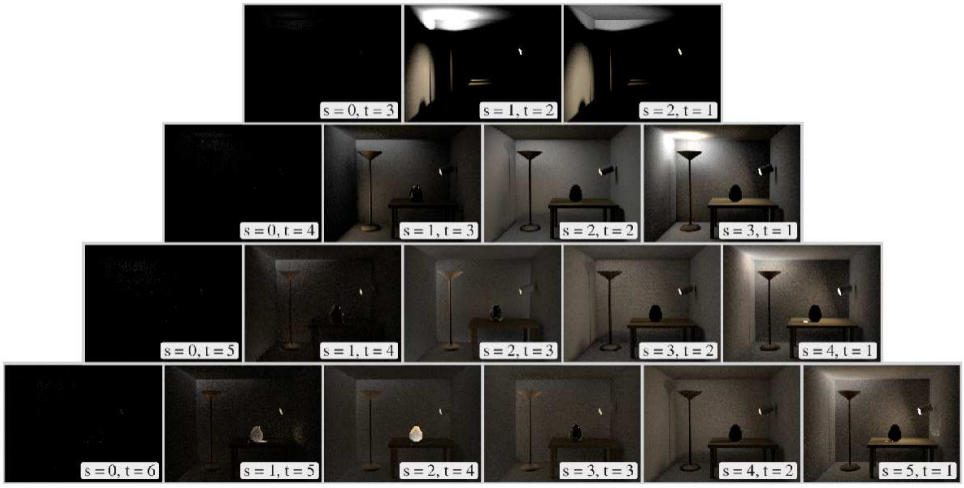


Figure 1.1: Examples of connection strategies rendered individually. Adapted from [41].

While the light and eye sub-paths are well importance sampled, importance sampling the sub-path connections is still mostly considered an open problem. Variations of BDPT, such as the Matrix Bidirectional Path Tracing (MBDPT) [6], Combinatorial Bidirectional Path Tracing (CBDPT) [38] and Probabilistic Connections Bidirectional Path Tracing (PCBDPT) [42], all three described in detail in Section 2.2, have been introduced and attempted to connect the sub-paths in different, more specialized ways than what the standard Veach-styled algorithm does, obtaining better results through a faster variance reduction. They do, however, come with their own limitations in different aspects.

Both PCBDPT and CBDPT reutilize paths to make their specialized connections. While the authors propose different ways to mitigate the variance introduced by sub-path reusal, it is still an undesirable effect. In our proposed technique, however, we attempt to importance sample the light sub-paths without reusal, making sure there is no variance increase or correlation issues. Additionally, we also developed the technique in an easily scalable way, which can be parallelized in CPUs or GPUs, such as the CBDPT.

Several other venues have been explored by Light Transport research in recent years, with special attention being given to real-time Ray Tracing techniques [4, 31, 32], Monte Carlo Path Tracing denoising [9, 24, 58] as well as Differentiable Rendering [18].

Differentiable Rendering specially has had a lot of attention, due to the growing importance of Machine Learning and its applications in the last decade. Generally speaking, this family of algorithms works by backpropagating gradients on a neural network with respect to the rendering output. Notable recent works include Nimier-David et al. [36], Loubet et al. [26], Li et al. [22], Zhang et al. [59], and Hadadan and Zwicker [11].

1.2 Hypothesis

In short, as previously mentioned, rendering algorithms such as the PT, the BDPT and their variations have their own limitations and use cases, as no technique will be optimal in all situations. One of the main challenges of the field is being able to introduce a technique that is robust while being flexible enough in different situations.

This work aimed to answer the following research question:

Is it possible to achieve significant variance reduction by importance sampling sub-path connections with the alias sampling method?

1.3 Objectives and Contributions

The main goal of this work is to investigate state-of-the-art variation reduction techniques through sub-path connections and propose a competitive methodology to address the problem. For this, we developed a new variation of the BDPT rendering algorithm. To achieve this objective, some specific goals were be considered and pursued:

- investigation of state-of-the-art approaches used in variation reduction on sub-path connections.
- proposal of a new sub-path connection sampling technique.
- validation of the method in extensive 3D scenes and models.
- comparison and evaluation of the proposed method against other available approaches.
- publication of the results.

We introduce our new rendering technique, the Alias Sampled BDPT, as a variation of the traditional BDPT algorithm attempting to do that. We employ a new light sub-path sampling method, based on the adoption of a modified version of the alias sampling method.

The alias sampling method generates (natural) numbers with a pre-computed probability by sampling a discrete probability function. That is, given an arbitrary probability function p_x, the algorithm has a probability of p_i to return the integer $0 \leq i \leq n$. The sampling method is efficient, being able to generate the probabilities in a constant, $O(1)$ time, per BDPT iteration. The integer returned by the sampling method will be used to index a list of light sub-paths. The arbitrary probability is based on a metric applied to the mean of the sub-path throughput at its furthest vertex.

Differently from previous methods, we do not reuse paths, which helps the variation to stay under control, but it does, however, force us to generate more paths than traditional techniques under the framework we envisioned. We demonstrate the efficiency and effectiveness of the resulting algorithm, against the standard BDPT, on a series of 3D scenes with different levels of complexity and characteristics.

1.4 Text Organization

Chapter 2 describes the main concepts and works related to importance sampling sub-path connections. Chapter 3 presents the proposed methodology, 3D scenes and models, evaluation metrics and computational resources that will be used in the dissertation. Chapter 4 discuss the results we obtained throughout the proposed method. Chapter 5 provides some concluding remarks, and talks about the strengths and limitations of our method, as well as discussing how these limitations could be tackled, and other areas where we could improve our technique or expand the research topic. Last but not least, Appendix A has additional diagrams that compare the parameters our application can receive as input.

Chapter 2

Background

This chapter is divided into two sections. In the first section, theoretical concepts related to the problem of importance sampling sub-path connections are briefly described. In the second section, relevant methods available for solving the problem are presented, which represent baselines and inspiration for the proposed method presented in Chapter 3.

2.1 Concepts

In this section, we briefly present relevant concepts used in importance sampling sub-path connections to provide a better comprehension of the problem under investigation in this work. For the remainder of this dissertation, we follow the terminology adopted by Haines and Shirley [12].

2.1.1 Path Integral

Monte Carlo (MC) integration techniques aim to find approximate values to integrals where it is hard, or otherwise impossible, to obtain a closed result in practice [5, 14]. This is done by taking random samples of a function over its domain and weighting them. For example, we can estimate a general determinate integral as

$$F = \int f(x)dx \approx \frac{1}{N}\sum_{i=0}^{N}\frac{f(X_i)}{p(X_i)}, \tag{2.1}$$

where $p(X_i)$ is a probability density function (PDF) for the function $f(X_i)$ and N is the number of samples. We can understand a PDF of a random variable as a function that returns us the probability that the random variable (or, in this case, sample) X_i will take a value comprised in a certain interval.

Bringing the problem to the light transport field, we follow Eric Veach's path integral formulation [49] for the remainder of this work. The resulting radiance I_j of a pixel j is defined as the integral of the light throughput, f_j, of all possible light transport paths that connect the pixel j to a light source in the scene, given by the equation

$$I_j = \int_{\Omega} f_j(\overline{\mathbf{x}})\ d\mu(\overline{\mathbf{x}}), \tag{2.2}$$

where Ω is the collection of all possible paths of all lengths k and $\overline{\mathbf{x}}$ represents a path, which we can understand as a list of 3D positions in the form $\overline{\mathbf{x}} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k]$, where $1 \leq k < \infty$. Note that, generally speaking, we can expand a term in respect to a path $\overline{x}$ to a product of terms with respect to their vertices. $d\mu$ is the differential of area-product over the path in question. This means that $d\mu(\overline{\mathbf{x}})$ can also be understood as the product of the area differentials over the path

$$d\mu(\overline{\mathbf{x}}) = \prod_{i=0}^{k} dA(\mathbf{x}_i). \tag{2.3}$$

The throughput f_j of a path $\overline{\mathbf{x}}$, also called measured contribution function, is defined by Veach as the weighted product of all bi-directional scattering reflectance distribution functions (BSDFs) over the path vertices [6], recursively represented as

$$\begin{aligned} f_j(\overline{\mathbf{x}}) = {} & L_e(\mathbf{x}_k \to \mathbf{x}_{k-1}) G(\mathbf{x}_k \leftrightarrow \mathbf{x}_{k-1}) W_e(\mathbf{x}_1 \to \mathbf{x}_0) \\ & \prod_{i=1}^{k-1} f_s(\mathbf{x}_{i+1} \to \mathbf{x}_i \to \mathbf{x}_{i-1}) G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i-1}), \end{aligned} \tag{2.4}$$

with L_e being the emitted radiance, G the geometric term, f_s being the BSDF and W_e being the sensor importance. Note that L_e, which is only different from zero if $\mathbf{x}_k$ is on an emitter, only appears at the emitter end of a path – the recursion itself only takes the BSDF and geometric terms into account.

If we apply this path integral formulation to Kajiya's Path Tracing [16], we are able to estimate each I_j independently by sampling paths $\overline{\mathbf{x}}$ similar to the PDF $p_j(\overline{\mathbf{x}})$, turning Equation 2.2 into

$$I_j = \int_{\Omega} f_j(\overline{\mathbf{x}}) \; d\mu(\overline{\mathbf{x}}) \approx \frac{1}{N} \sum_{i=0}^{N} \frac{f_j(\overline{\mathbf{x}}_i)}{p_j(\overline{\mathbf{x}}_i)}. \tag{2.5}$$

While we offer here a brief introduction where applicable, for more in-depth explanations about Monte Carlo integration and correlate topics in the context of Light Transport, readers may want to refer to the textbook 'Physically Based Rendering: From Theory to Implementation' (PBRT), by Pharr et al..

2.1.2 Importance Sampling

As with any other process that depends on random variables, Monte Carlo integration has a variance value attached to its observations. Formally being defined as the square of the standard deviation, it can be understood as a measure of how much a set of numbers differs from their mean [25, 56].

In the context of Monte Carlo light transport, a higher variance will result in visual artifacts, such as noise. Thus, we intend to reduce the variance of our algorithm. This could be done, for example, by making use of blue-noise sampling techniques [1, 39, 43], sub-path resampling [33], and, more commonly, Importance Sampling (IS), one of the most used families of variance reduction techniques.

Importance Sampling is a variance reduction technique frequently applied to Monte Carlo integration applications [20, 48], such as Path Tracing and BDPT. It makes use of the fact that if we concentrate computational resources where the value of the integrand of Equation 2.5 is higher, we will manage to achieve an accurate estimation faster [41]. Generally speaking, in relation to light transport problems, this happens when ray directions are sampled from distributions that are proportional to other factors of the integrand, such as the BSDF, the illumination distribution and so on, because the PDF will have a similar shape to it.

We can conclude that PDFs that are proportional to different factors of the integrand will importance sample specific aspects of the scene. This can be seen in Figure 2.1, where the same scene is importance sampled in relation to the BSDFs and light sources, respectively. It is also a known fact that finding a single PDF that is a good fit for the entire scene is not an easy task [41] – which brings up the question: can we use multiple different PDFs at the same time? And, if so, how can we optimally mix them together?

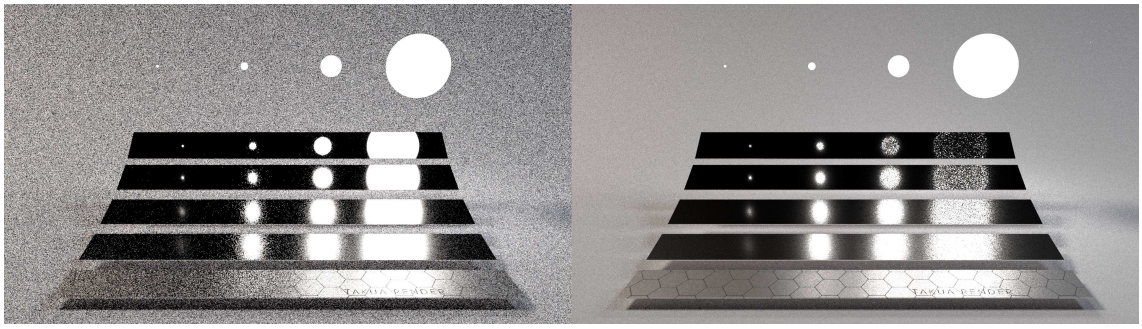


Figure 2.1: Recreation of a scene from Veach's Ph.D. thesis [49], comparing the resulting rendered image when importance sampling the BSDF and the light sources, by Li [23].

2.1.3 Multiple Importance Sampling

Multiple Importance Sampling (MIS), introduced by Veach and Guibas [51], tackles the issue that it is hard to obtain one single PDF that performs well for all situations, but we often know "multiple" different PDFs that locally approximate the integrand value better than others [10]. Then, we can generalize Equation 2.5 as a weighted sum of estimators, which Veach and Guibas call the "multi-sample estimator", given by

$$I = \sum_{i=0}^{n} \sum_{j=0}^{n_i} \frac{w_i(X_{i,j}) f(X_{i,j})}{n_i p_i(X_{i,j})}, \tag{2.6}$$

where n_i denotes the number of samples taken from p_i, where $n_i \geq 1$, and $N = \sum n_i$ represents the total number of samples. Finally, $X_{i,j}$ are the samples taken from technique p_i, for $j = (1, \ldots, n_i)$, and w_i is the weight function of the technique.

In practice, for BDPT, we can use a different and more specialized notation for Equation 2.6, as used by Chaitanya et al. [6], to account for the connection strategies' sub-path lengths s and t, for the light and eye sub-paths respectively. Thus, we can rewrite Equa-

tion 2.5 as

$$I_j = \sum_{i=0}^{N} \sum_{s \geq 0} \sum_{t \geq 0} \frac{w(\overline{\mathbf{x}}_{s,t}) f_j(\overline{\mathbf{x}}_{s,t})}{N p(\overline{\mathbf{x}}_{s,t})}, \tag{2.7}$$

where N is the total number of (independent) samples, $\overline{\mathbf{x}}_{s,t} = [\mathbf{y}_0, \ldots, \mathbf{y}_s, \mathbf{z}_t, \ldots, \mathbf{z}_0]$ is a full, connected path through pixel j, $[\mathbf{y}_0, \ldots, \mathbf{y}_s] = \overline{\mathbf{y}}$ is the eye sub-path, $[\mathbf{z}_0, \ldots, \mathbf{z}_t] = \overline{\mathbf{z}}$ is the light sub-path, $p(\overline{\mathbf{x}}_{s,t})$ is the PDF used to generate $\overline{\mathbf{y}}$ and $\overline{\mathbf{z}}$ and $w(\overline{\mathbf{x}}_{s,t})$ is the MIS weight function.

While there is a number of ways the weights from Equations 2.6 and 2.7 can be obtained, Veach and Guibas showed that the Balance Heuristic is a good choice to reduce variance [44, 51]. The weight function is, then, given by

$$w_s(x) = \frac{n_s p_s(x)}{\sum_i n_i p_i(x)}. \tag{2.8}$$

Note that a naïve implementation of Equations 2.7 and 2.8 would result in an $O(n^4)$ algorithm, but, in practice, implementations of the heuristic usually make use of values gradually calculated and obtained through the construction of the sub-paths to mitigate the computational cost [41].

More recent developments, such as the non-balance heuristic proposed by Sbert et al. [45], the new weighting functions introduced by Ivo et al. [15] and the generalized estimators by Karlík et al. [17], attempt to reduce the variance even further by optimizing the MIS variance estimators in different ways.

2.1.4 Sub-path Connections

The core ideas behind a bidirectional light transport algorithm such as BDPT is how we locally sample paths by generating two sub-paths of varying length starting, separately, from the camera and an emitter and how we connect these sub-paths end-points together. While there are several ways these connections can be made, this is heavily influenced by how the sub-paths are generated.

In the traditional BDPT framework described by Veach [49], for each pixel in the image, and for each sample, we generate a sub-path from the camera and a sub-path from a random emitter in the scene. Then, we iterate throughout their whole lengths, from the starting vertices, to the intermediate and end-point ones, and attempt to make successful connections with each eye and light pair of vertices – in practice, each of these successful connections is called a "connection strategy" and has a specific eye sub-path and light sub-path length. We take the different sub-path connection strategies into account by weighting them with Multiple Importance Sampling. For a connection to be considered successful, we perform a visibility check – we can only connect two vertices if nothing is blocking the view, otherwise this connection results in zero contribution to the sample. An example is shown in Figure 2.2.

Note again that, in the standard BDPT, one light sub-path will be generated for each eye sub-path and then they will be connected together, through the use of several different connection strategies weighted by MIS. Other formulations of BDPT, described

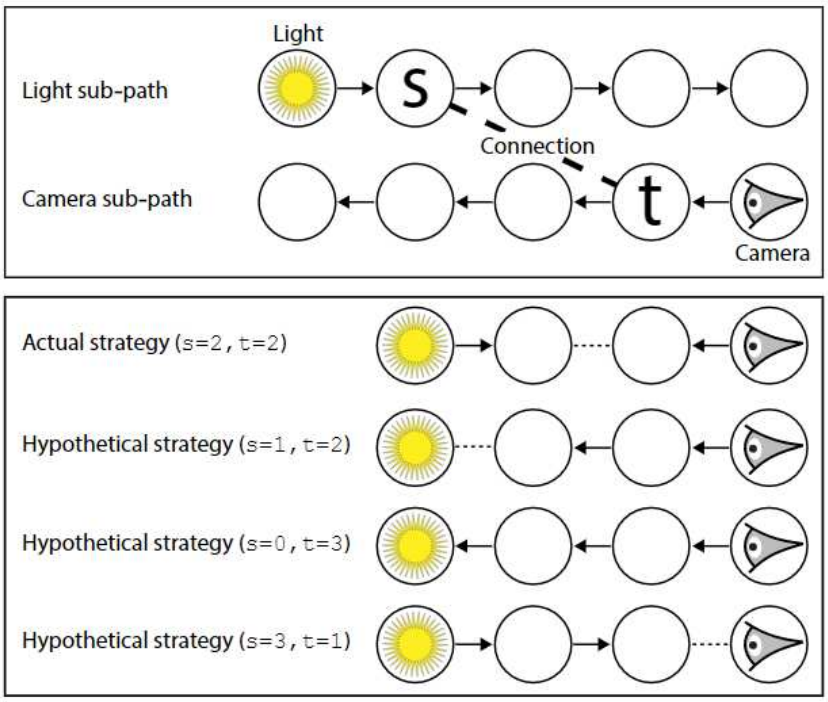


Figure 2.2: Example of connection strategy, with eye sub-path length $t = 2$ and light sub-path length of $s = 2$, followed by other hypothetical strategies that need to be taken into account for an accurate MIS calculation. Adapted from Pharr et al. [41].

in Section 2.2, have investigated different methods to generate these sub-paths and their connection strategies. Ultimately, our goal is, precisely, to find optimal ways to connect generated sub-paths and maximize the variation reduction.

While we focus on BDPT derived techniques here, another (advanced) family of bidirectional algorithms, the Metropolis Light Transport (MLT) [52], also proposed by Veach and Guibas [52], makes use of similar concepts – sub-paths are generated from both emitters and the camera and subsequently connected.

First introduced by Metropolis et al. [30], Metropolis-based methods make use of Markov Chain Monte Carlo (MCMC) integration, a class of sampling algorithms. Through MCMC integration, the Equation (2.1) becomes

$$\int_{\Omega} g(x)dx \approx \frac{b}{N}\sum_{i=1}^{N}\frac{g(x)}{p^*(x_i)}. \tag{2.9}$$

MCMC integration is especially useful in integration problems with a high order of magnitude, which is one of the reasons MLT programs are known to be efficient in scenes with complex light transport paths. MCMC sampling is able to estimate a series of correlated integrals up to the scaling factor b, which is then estimated separately by traditional MC integration, for example.

The samples generated by MCMC sampling are part of a history of states of a Markov chain, as described by Hastings [13]. Based on a conditional probability $q(x_i \rightarrow y)$, a proposal state y is generated given the current state x_i, which is often referred to as mutation in light transport applications. We, then, consider $x_{i+1} = y$ with the probability

$min(a, 1)$, with a defined as

$$a(x_i \to y) = \frac{(p^*(y)q(y \to x_i)}{(p^*(x_i)q(x_i \to y)}, \tag{2.10}$$

or, otherwise, $x_{i+i} = x_i$ and no mutation occurs.

Some variations of the MLT algorithm [3, 10, 19], even make use of BDPT and uni-directional PT techniques underneath them, by applying mutations and iterating over the proposed paths. Other recent applications of MCMC integration to Light Transport differ from previous approaches for using Langevin Monte Carlo sampling, rather than only Metropolis-based integration [27].

2.1.5 The Alias Sampling Method

The alias sampling method [54] is an efficient algorithm to generate random discrete numbers with arbitrary Probability Mass Functions (PMFs) – while similar in principle to PDFs, PDFs refer to continuous distributions and PMFs, as the name suggests, to discrete distributions.

To allow for a fast sampling speed, the algorithm has a pre-processing step that iterates over all p_i probabilities, for $i = 1, \ldots, n$, multiple times to find one that satisfies a defined constraint. Once it is done, p_i is inserted into a bin. Each bin will have, at most, two probabilities, and, depending on the constraints chosen, one probability can appear on more than one bin. To draw a sample from this method, we will roll two random numbers: one to select a bin b_k, with $k = 1, \ldots, K$, and another one to select one of the probabilities of the bin, for instance, p_i. Since we have at most two probabilities on each bin, the comparison is simple and this makes it a $O(1)$ sample generation cost – the pre-process stage, if implemented naively, has $O(n^2)$ cost. Assuming we selected the probability p_i at bin b_k, the sampled random integer we will return is i. An example representing the process can be seen in Figure 2.3.

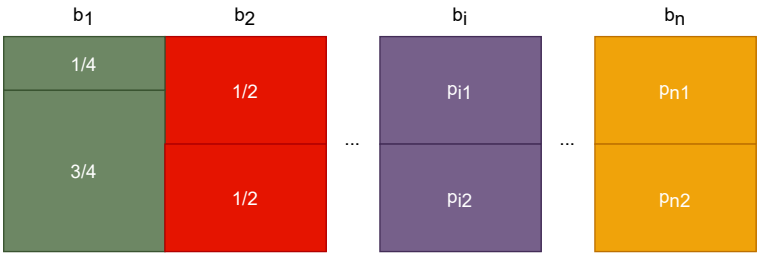


Figure 2.3: Example diagram representing the alias bins and their respective probabilities. According to the method proposed by Walker, we will draw a random number to select one of the b_k bins, followed by another number to select a probability p_i encapsulated in said bin. Note that the sum of the probabilities within a single bin will always be equal to 1.

Vose proposed a variation of the alias method with $O(n)$ complexity to build the probability and alias tables, while keeping the efficient $O(1)$ sample cost [53]. The main difference is, rather than iterating through the list of elements $O(n^2)$ times, we will keep two lists of elements, and iterate through them one single time through the algorithm

execution to build the probability tables. We will build up on Vose's ideas and use a variation of his algorithm to importance sample sub-path connections, as we describe in Section 3.1.2.

2.2 Related Work

In this section, we briefly discuss relevant methods that inspired our work and are relevant for the problem at hand or the field as a whole.

As far as our own method is concerned, some evident common links between it and all of the works described as follows is that they are path-based and most of them are, like ours, bidirectional light transport methods. Several of these methods inspire ours in different areas, such as the specialized sub-path connections of CBDPT, PCBDPT and MBDPT. Our algorithm, however, innovates by performing such special connections through the Alias Sampling method, while avoiding reuse of light sub-paths.

2.2.1 Bidirectional Path Tracing

The Bidirectional Path Tracing algorithm, proposed independently by both Lafortune and Willems, in 1993 [21], and Veach and Guibas, in 1995 [50], came to be from the observation that light transport can be formulated, save rare exceptions, as a symmetric operation: the concepts of light emission and light measurement are mathematically exchangeable [41]. This implies that we can render an image through a forward or backward algorithm and, hypothetically, achieve the same result. Notably, in practice, rendering from the light sources, rather than from sensors/cameras, is a much harder task if done independently, but assuming infinite time constraints and computational resources the result of the algorithm will be the same.

2.2.2 Metropolis Light Transport

Metropolis sampling, based on the Metropolis et al. [30] algorithm for Markov Chain Monte Carlo (MCMC), can further optimize the efficiency of BDPT by focusing computational resources on more relevant light carrying paths. It was first introduced to light transport simulation by Veach and Guibas, in 1997, as Metropolis Light Transport (MLT) [52]. An unconventional application to rendering at the time, the MLT algorithm generates light paths by permuting, or mutating, previous paths, being able to perform a local exploration of the path space. The use of mutation strategies in MCMC integration amortizes the cost of finding a solution to paths that are hard to obtain through traditional Monte Carlo algorithms. It is known, however, as a difficult method to implement, and is usually not considered flexible enough to be used in production rendering.

2.2.3 Primary Space Sample Metropolis Light Transport

Kelemen et al. proposed the Primary Space Sample MLT (PSSMLT) in 2002 [19] as another rendering technique also based on MCMC algorithms. Differently from MLT,

however, PSSMLT does not explore the path space directly. Relevant light paths are initially obtained by a different light transport algorithm, such as BDPT – random initial paths are also valid, but not as effective. The mutation strategies are also applied here, but, usually, much simpler and easier to implement than the ones found in the original MLT algorithm.

A variation of PSSMLT was proposed by Hachisuka et al., in 2014, the Multiplexed Metropolis Light Transport (MMLT) [10]. MMLT shares many similarities to PSSMLT, but it is based on a new MCMC sampling framework proposed by the authors, which allows the algorithm to sample directly from a Multiple Importance Sampling distribution, and introduce other constraints, such as the connection of the sub-paths being limited to a single strategy specified by its current Markov chain state, being able to focus computation efforts on paths that contribute more to the image.

2.2.4 Combinatorial Bidirectional Path Tracing

In an attempt to make use of the increasingly powerful GPUs, while still keeping the flexibility of a CPU-based algorithm, Pajot et al. introduced a novel formulation of a hybrid implementation of BDPT [38]. The author attempted to harvest the strengths of both processing unit models, while avoiding their downsides – for example, while CPUs lack behind in parallelism when compared to GPUs, GPUs, when used alone, are limited when it comes to material, light or camera model complexity compared to CPUs.

Differently from the traditional BPDT, where a light sub-path and an eye sub-path are combined individually, Combinatorial BDPT (CBDPT) connects *sets* of light and eye sub-paths, using the GPU power to its maximum without increasing the sampling cost. Considering the light and eye sub-paths are connected in all possible ways, path reusal will clearly be present. To prevent correlation patterns and the consequent increase in variance, CBDPT shuffles the image-space coordinates, stored in an array, used by each eye path, on a stratified scheme over the entire image. The authors also managed to interleave the GPU and CPU workload, in order to always keep them busy.

2.2.5 Probabilistic Connections Bidirectional Path Tracing

While Bidirectional Path Tracing with Multiple Importance Sample is one of the most versatile rendering methods available, BDPT still generates many low contributing paths. Popov et al. observed on Probabilistic Connections Bidirectional Path Tracing [42] that importance sampling the sub-path connections is still an open issue and could result in improved variation reduction. The authors proposed a new framework that makes connections probabilistically, which is done by storing light paths and estimating PMFs out of them.

They also introduced an efficient caching scheme for the light paths, chosen by low-discrepancy sequences, which manages to avoid expensive calculations otherwise used by interpolating the store paths. This approach brings some issues, however, as the reuse of (light) paths increases the variance. They discuss the effects of such path reuse and introduce specialized sample weights to reduce the variance increase.

2.2.6 Resampling-aware Weighting Functions for Bidirectional Path Tracing using Multiple Light Sub-Paths

When using Multiple Importance Sampling with BDPT, Veach's balance heuristic is able to provide a minimal upper bound for the variance, weighting each full path in proportion to its probability. While when resampling paths their probability can change, the standard MIS functions don't account to that.

Expanding on the work of Popov et al., Nabata et al. [33] introduced a new, precise formulation of the variance, as well as derivations of weighting functions that take the probability changes into account. The authors show that the functions depend on the size of the pre-sampled light sub-path set, with the standard balance heuristic becoming less efficient as more paths are resampled. Their new functions are adaptable to the size of the path resampling set and are derivable by applying Resampled Importance Sampling (RIS) to the resampling of a light sub-path from a set of previously sampled light sub-paths.

2.2.7 Two-stage Resampling for Bidirectional Path Tracing with Multiple Light Sub-paths

While Popov et al. [42] and Nabata et al. [33] showed that the resampling of a small subset of the generated light sub-paths can lead to a more efficient variance reduction, the reuse of paths lead to correlation and in turn visual artifacts. On the other hand, using more pre-sampled light sub-paths would increase the execution time. Nabata et al. [34] introduced a new two-stage resampling method for BDPT making use of multiple light sub-paths.

The first stage is composed by an efficient resample of important light sub-paths from a large number of pre-sampled light sub-paths, followed, during the second stage, by the resampling of these important paths from the first stage, resulting in a single sub-path. The authors introduce a specialized function for their two-stage resampling method, which, coupled with their resampling algorithm, manages to to obtain stronger variance reduction than previously introduced BDPT methods making use of multiple light sub-paths.

2.2.8 Matrix Bidirectional Path Tracing

In traditional Monte Carlo ray tracing algorithms, paths sampled can be clumped together, frequently resulting in redundancy or low contribution to the end result, and this is even worse for Bidirectional methods, such as BDPT, as the connections might be clumped as well.

Chaitanya et al. [6] introduced a new matrix formulation for Bidirectional Path Tracing algorithms, through the Matrix Bidirectional Path Tracing, to enable stratification and allowing efficient use of low-discrepancy sampling, which results in faster variance reduction when compared to other BDPT methods in most situations, except in cases that the authors call "implicit path sampling techniques", meaning Path Tracing without next-event estimation, with eye sub-path length $s = 0$, and Light Tracing, with light sub-path

length $t = 1$. In these cases, they recommend making use of traditional algorithms.

2.2.9 Ellipsoidal Path Connections for Time-gated Rendering

Time-of-flight (ToF) image sensors are used in a broad array of applications, such as robotics, medicine and human-computer interaction. Unlike traditional intensity sensors, ToF sensors take into account that the speed of light is finite and record data about the time it takes for photons to travel from a light emitter to itself. Ray-tracing rendering techniques, such as the standard BDPT, can be used to simulate time-gated applications, after adaptations are introduced. BDPT is not, however, efficient for this time of problem, as several paths end up being rejected according to certain conditions set up by the simulation. Pediredla et al. [40] proposed a novel algorithm based on BDPT with specialized path connections, the Ellipsoidal Path Connections for Time-gated Rendering.

In short, their algorithm initially selects a target path length τ through a path length importance function W_τ. Then, they used the traditional BDPT to trace sub-paths of max length equal to τ. The connections are, then, made by a novel technique that will connect every pair of vertices in the light and eye sub-paths through an additional vertex, rather than directly, with the new vertex, called the ellipsoidal vertex, selected to make the total path length equal to τ. The proposed connection strategy, for this application, results in less rejected paths, managing to increase the performance of the algorithm against traditional BDPT.

2.2.10 Caustic Connection Strategies for Bidirectional Path Tracing

While BDPT is one of the most robust rendering methods, it is known to have poor performance when sampled paths involve specular light transport, such as refraction through dielectrics. Speierer et al. [46] introduced the Caustic Connection Strategies for Bidirectional Path Tracing. The method has specialized connection strategies that are adapted to specular events, based on Manifold Next Event Estimation (MNEE) techniques, to deterministically connect vertices across refractive surfaces. These new techniques as especially useful when we consider that, for caustic sub-paths other strategies frequently fail, resulting in specific areas of the image to be overly noisy. The authors also introduced a recursive scheme to compute MIS weights during the creation of the paths, keeping it efficient.

2.2.11 Path Graphs: Iterative Path Space Filtering

One of the main issues of traditional Path Tracing based methods is the usual need for a high number of samples to get a noise-free image. Bidirectional algorithms, while usually more efficient in cases where the illumination is indirect and incoming from an external area to the scene, have trouble solving paths that pass through windows, for example. These issues can be mitigated by techniques that explore path guiding, as well as image denoising. They are more effective, however, when more information about the scene (i.e.,

'features') are available rather than just the resulting light data. Deng et al. [8] introduced a novel technique that operates on a Path Graph.

Normally, ray or path based methods operate on a tree, where one vertex is connected with one edge going towards the light source, while the other heads to the camera view, recursively. In practice, each pixel in the resulting image is the starting point of one such tree. The Path Graph approach described by Deng et al. [8] takes the union of these trees and add extra edges which share information to vertices in the spatial vicinity, effectively making clusters of vertices and connecting them. With such a graph in hand, the radiance of each pixel can be estimated by aggregating the radiance at each node, and propagating it to nearby vertices, which in practice is an iterative way of reusing paths, followed by a final gather step. The results can, then, be reconstructed and denoised. The authors argue the extra steps of this method are relatively cheap, but manage to obtain considerable variance reduction compared to denoised images generated by standard techniques.

2.2.12 Optimised Path Space Regularisation

Despite Path Tracing being by far the most popular approach in simulating light transport, some effects such as caustics are especially hard for it to render. Other rendering approaches such as the Bidirectional Path Tracing and Photon Mapping can solve many of these hard-to-render effects, but Path Tracing is still often preferred due to his flexibility and ease to integrate. Production renderers commonly apply a form of regularisation called 'roughening', which, in short, means adding roughness to materials to ease the sampling of hard paths. This effectively adds a bias, or 'error', to the method, but allows it to be more efficient, on a trade-off. Weier et al. [57] introduced a new algorithm, the Optimised Path Space Regularisation, aiming to introduce only the necessary amount of roughness to reduce variance while keeping the added error as low as possible.

Weier et al. employed differentiable rendering [22, 35] to "model the problem as a joint bias-variance minimisation", allowing them to choose which vertices to normalize on specific paths during the rendering. While their new estimator is biased, the authors demonstrate that it is consistent when "mollifiers" are present. Mollifiers are sequences of smooth, positive distributions that converge to a Dirac delta distribution as the mollification bandwidth approaches zero. The proposed regularization scheme works on arbitrary scenes without having to be re-optimized, allowing for a straight forward implementation.

2.2.13 Reservoir-based Spatio-Temporal Importance Resampling

Monte Carlo-based methods have trouble efficiently sampling dynamic direct light when the number of emitters is big, on the order of millions of sources, even if we consider offline methods only. The problem, however, is considerably more complex when attempting to do it in real time. Bitterli et al. [4] introduced a new algorithm that renders light sources interactively and without requiring complex data structures to stay efficient, the Reservoir-based Spatio-Temporal Importance Resampling (ReSTIR).

The algorithm resamples "candidate light sources" repeatedly, as well as resampling relevant samples spatially and temporally. Their method samples one-bounce direct lighting

from a vast number of light sources and is suited for real-time ray tracing implementations on dynamic scenes, building on Resampled Importance Sampling (RIS) by selecting samples from one distribution and then weighting another subset of them, chosen through another distribution that better matches the function to be integrated which reduces the variation [47].

Differently from previous applications, however, the authors use a small "reservoir" data structure that only stores samples that were accepted. In practice, Bitterli et al. [4] introduce a new, unbiased MC estimator for that, showing that their method can obtain results up to 60 times faster than state-of-the-art methods. An biased version of the estimator can reduce variance further, and is even faster, "at the cost of some energy loss".

2.2.14 ReSTIR GI: Path Resampling for Real-Time Path Tracing

While recent advancements in GPU architectures allow for hardware-accelerated, real-time Ray Tracing applications, only a small number of samples can be draw at a time. That represents a challenge, even with state-of-the-art denoising algorithms. Expanding on the principles introduced by Bitterli et al. [4] on ReSTIR, which samples direct lighting, Ouyang et al. [37] proposed a new approach that is effective at sampling indirect lighting on scenes with millions of light sources, the ReSTIR GI, or ReSTIR Global Illumination.

ReSTIR GI manages to efficiently resample multi-bounce indirect lighting paths resulting from Path Tracing, which allows for information about relevant contributing paths to be shared over time and space (i.e., pixels). Differently from the original ReSTIR, the ReSTIR GI algorithm samples directions in the local sphere around shading points, rather than placing them in a global light space, with the resulting rays intersecting surfaces in the scene. Their RIS weights are, then, determined by the throughput of light each intersection scatters back toward the ray origin. The authors observed a strong variance reduction, when comparing the method against the traditional Path Tracing in similar conditions and scenes.

Chapter 3

Material and Methods

This chapter describes the methodology proposed in this work, as well as the 3D scenes and models, evaluation metrics and computational resources that will be used in the experiments during the development of the method.

3.1 Methodology

In this section, we describe the main steps that constitute the proposed methodology for importance sampling sub-path connections on a Bidirectional Path Tracer algorithm. A general architecture is illustrated in Figure 3.1, where each stage is shown as an enumerated module.

Module 1 in the architecture diagram represents the application initialization step, which includes user input and the construction of auxiliary structures for the alias sampling. The input of the algorithm is a 3D scene and a set of parameters to configure different aspects of the application. This leads to the sample generation step of the algorithm – note that, here, sample and "iteration" can be used interchangeably, illustrated by Module 2. From that step of the application, we have two independent steps related to the camera and the light sources: the eye and light sub-paths generation, respectively Modules 3 and 4. This will be discussed in more detail in Section 3.1.1.

Following the light sub-path generation step, we apply the alias sampling method to importance sample sub-path connections, illustrated by Module 5, which will make use of the structure constructed on Module 1. Our take on this algorithm, along with some pseudo-code and implementation and performance notes, will be discussed in Section 3.1.2.

From the importance sampling of connections, we will connect the N alias sampled light sub-paths with the N eye sub-paths, on Module 6. The contribution of these connections will be averaged for each pixel (that is, for each eye sub-path) they correspond to, resulting in one complete BDPT iteration, given in the diagram by Module 7. This process is repeated S times, with the image samples being accumulated and averaged to obtain the final rendered image, illustrated by Modules 8 and 9.

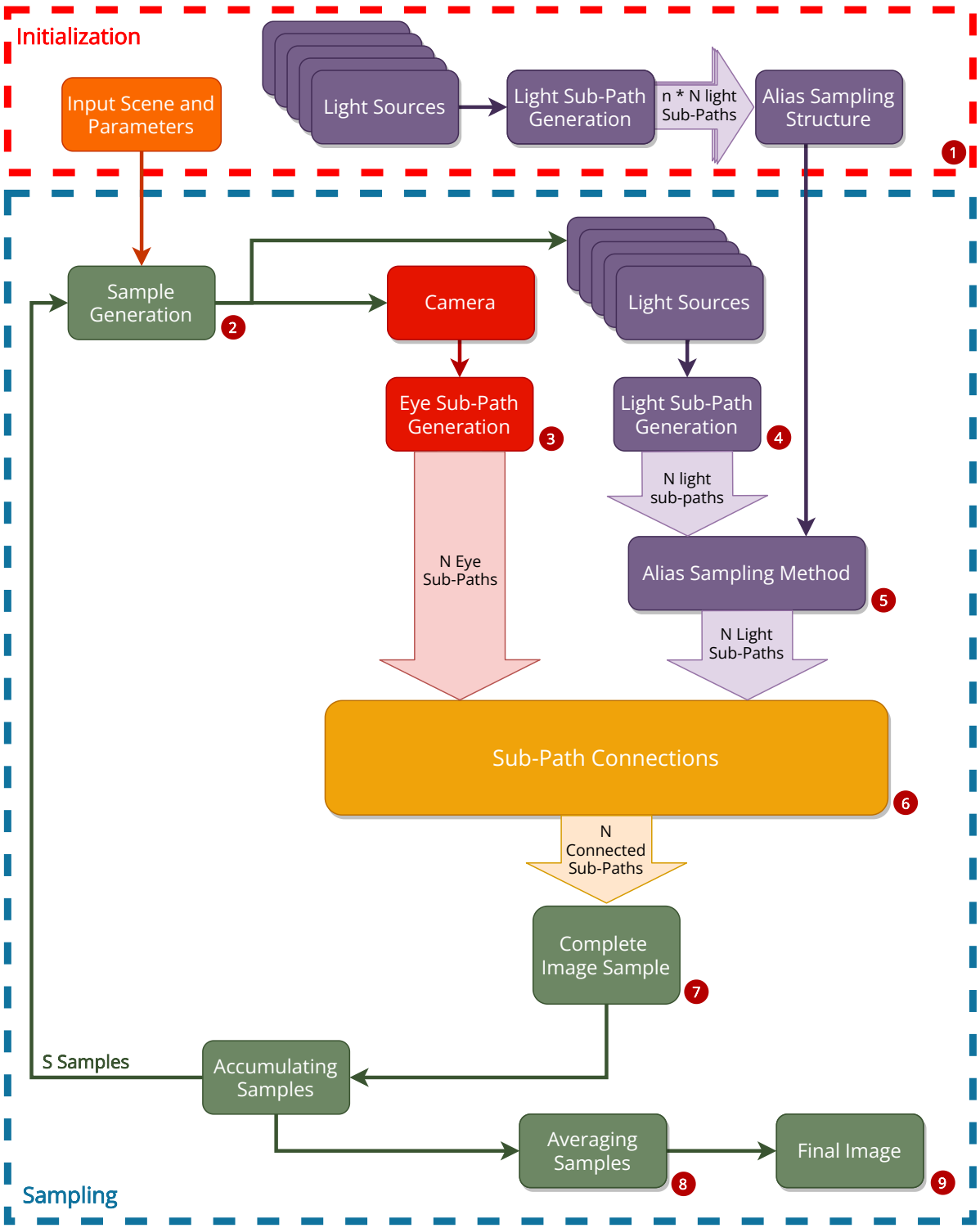


Figure 3.1: Proposed architecture for importance sampling sub-path connections on a BDPT algorithm.

3.1.1 Sub-path Generation

The generation of light and eye sub-paths are essential steps to any Bidirectional Path Tracer algorithm. While in the original BDPT algorithms by Veach and Lafortune the light sub-path generation was pixel-based, with a light sub-path being generated and connected to the respective eye sub-path generated for a certain pixel, in other formulations,

such as the MBDPT and the PCBDPT, the sub-path generation is independent.

In our implementation, to generate the eye sub-paths, for each sample of the BDPT algorithm, we will iterate through all the pixels of the image. For each pixel, we will use jittering to allow us to apply super sampling anti-aliasing to the image rendering, by averaging the several samples we take from the scene. Here jittering will refer in practice to an intentional random deviation applied to the ray origin's in the camera, starting from the pixel center. This will allow us to sample parts of the image that would not be possible otherwise, and help us achieve a smoother and more accurate image at a low cost. From there, we will follow a "random walk" process, as it is done by any traditional ray tracer or path tracer algorithm, with the path contribution being affected by the bounces in the scene surfaces that the ray hits. The random walk stops if the ray hits a light source or a specified number of bounces is achieved.

On the other hand, to generate light sub-paths, we will first pick a random light source, or emitter, of the scene. Then, we will sample a random direction parting from the light surface. Assuming this is a point light, this is simply a random direction with origin in the light's center. If the emitter in question is an area light, infinite or not, the direction is sampled according to its respective geometric form [41]. After the initial direction is picked from the light source, a random walk process occurs analogously to the one described for the eye sub-path generation.

3.1.2 Importance Sampling with the Alias Method

To importance sample the sub-path connections with the alias method, we will implement a variation of Vose's method [53], briefly described in Section 2.1.5. First, we have a pre-processing step to generate structures needed for the sample step, as described in Algorithm 1. We will generate $n \cdot N$ light sub-paths, where n, a positive integer, is a parameter we give the application with the other input variables, and keep these paths in a vector.

Algorithm 1: Constructing Alias Structures

Result: Bin Data Structure for the Alias Method
Generate $n \cdot N$ light sub-paths each and add them to a vector `V`;
Divide the $n \cdot N$ elements of `V` into alias bins;
for *bin* `b` *in bins* **do**
  assign probabilities to elements in b, e.g.: $p_{e_0} = \frac{{\beta_{e_0}}^2}{{\beta_{e_0}}^2 + {\beta_{e_1}}^2}$;
end

We will, then, divide the elements into a series of alias bins. Note that each bin will have exactly two light sub-paths, and that one given light sub-path should only appear in a single bin, for simplicity. The paths are randomly split into bins. For each of the bins, we will assign probabilities to their elements as

$$p_{e_0} = \frac{{\beta_{e_0}}^2}{{\beta_{e_0}}^2 + {\beta_{e_1}}^2}, \tag{3.1}$$

where p_{e_i} is the probability of the element i of the current bin, with i being either 0 or 1, which means each bin will have a total probability of 1. β_{e_i} is the unshadowed path throughput of the same element. Here, to obtain the probability with a higher degree of certainty we apply the balance heuristic to weight the path throughputs. Additionally, note that each element of the bin will be related to a certain sub-path that, when chosen by the alias method, in practice, will return the sub-path variable itself.

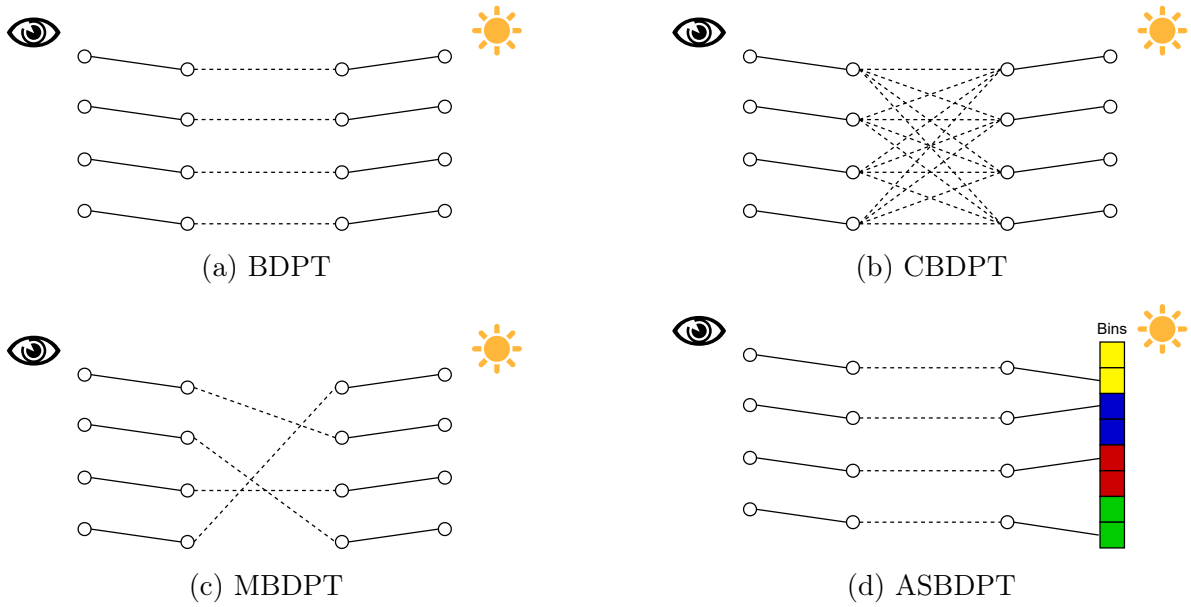
For the sampling step, shown in Algorithm 2, we will first generate N light sub-paths and keep them in a queue. We will sample a random variable and choose a bin, which has a complexity $O(1)$. Then, we sample another random variable u in $[0, 1)$, also in complexity $O(1)$. If $u < p_{e_0}$, we will return the element e_0 of the bin, otherwise we will return e_1. After we return a sub-path variable, we will replace the path variable in the bin with a new path, generated as normal in a common BDPT iteration.

Algorithm 2: Importance Sampling Sub-path Connections Based on the Alias Method

Result: Vector `V` with N Importance Sampled Paths
Create an empty vector `V` with length N;
while n `<` N **do**
  choose a bin b with low-discrepancy sampling: $b := g[rnd() * nOfBins]$;
  sample a variable in $[0, 1)$: $u := rnd()$;
  use u to choose an element in b: e `= if` $u > p_{e_0}$ `?` e_0 `:` e_1;
  add e to `V`;
  replace the chosen element e with a newly generated one;
  n++;
end

Note that we will not discard these structures – they will be kept throughout the application. At each new BDPT iteration, we will generate N new light sub-path variables. The sampling process itself will, then, have an $O(1)$ cost. The core difference here when compared to the traditional BDPT is that we will generate $(n-1) \cdot N$ "extra" light sub-paths after the execution of the whole algorithm is completed, which is equivalent to the amount of light sub-paths consumed by $(n-1)$ iterations. We intend to keep these "extra" paths at a minimum to reduce the use of computational resources while keeping the quality of the importance sampling process, so n should be a reasonably small number compared to the total number of iterations of the application.

To showcase the strengths of the algorithm we are proposing, we compared our connection strategies to the standard BDPT, described in Chapter 2. In our algorithm, we connect one path to one other path, managing to keep the same ratio of path connections as BDPT and MBDPT. Also just like MBDPT and BDPT, we manage to avoid sub-path reuse, which happens in both CBDPT and PCBDPT, avoiding the increased variance that comes with it. Most of our proposed implementation, especially the sub-path generation, can be parallelized, which could ultimately result in a hybrid renderer, much like CBDPT. Figure 3.2 visually compares the number and logic behind the connection strategies of our method, and others from the field.



(a) BDPT (b) CBDPT

(c) MBDPT (d) ASBDPT

Figure 3.2: Diagram demonstrating connection strategies used, respectively, by the standard BDPT (a), CBDPT (b), MBDPT (c) and our technique (d). We can also compare the amount of connections performed by each of the algorithms.

3.2 Evaluation Metrics

When researching Monte Carlo-based light transport algorithms, several metrics, such as Structural Similarity Index Measure (SSIM) [55] and S-CIELAB [60], are commonly used to assess the performance of the methods. We opted, however, to make use of Root Mean Square Error (RMSE). The RMSE metric is commonly used to evaluate a proposed algorithm's convergence rate, when compared to other state-of-the-art techniques [7].

We consider that the lowest the RMSE for a set execution time, the better an algorithm performs at reducing the variance of the rendering. The RMSE value for the resulting image of the execution of each algorithm, for each amount of samples, on each test scene, is given by Equation 3.2.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{N}} \tag{3.2}$$

where x_i refers to the pixel i of a test image, and y_i refers to the pixel i of a reference image.

Reference images are rendered for multiple days, and are considered "ground truths", and thus used for comparisons like this. In our case, we rendered each of our reference images for 16384 samples of the standard BDPT algorithm implementation of PBRT v3. Due to their long execution time, we consider these "ground truth" images to have successfully converged, having negligible error for the sake of evaluating rendering methods.

We will, also, generate heatmaps based on the RMSE values of each method, on a pixel-by-pixel basis. After obtaining the RMSE values of each pixel, the resulting numbers are divided by the maximum number obtained, normalizing them for 8-bit colors. The

results, in our case, are displayed on a heatmap that goes from blue, for the lowest error encountered, to green, for the highest error.

In addition to RMSE measurements and its heatmaps, the execution time will also be evaluated against other techniques, over all our test cases. In short, the RMSE will be our efficacy and quality measure, while the execution time allows us to gauge the algorithm's efficiency against other existing methods.

3.3 3D Models and Scenes

For the experiments and evaluation of our methods, we have used well known 3D models and scenes to the literature with hard-to-render light paths, such as Veach's Ajar Room and variations of the Cornell Box. Additional scenes include the Bathroom scene and the Living Room scene. The scenes that we used are available freely on Bitterli's rendering resources [2] scene database.

While we used different criteria to choose each of the selected scenes, one characteristic is common to them all: the presence of hard-to-render light paths. Light paths are considered "hard-to-render" when the path between a surface and a light source is indirect. That is, when multiple bounces are required for the surface to be illuminated. This could happen, for example, on a scene with two independent rooms, as shown in Figure 3.3. The camera is on the left room, the sole light emitter is in the right room and there is a small hole in the wall between them.

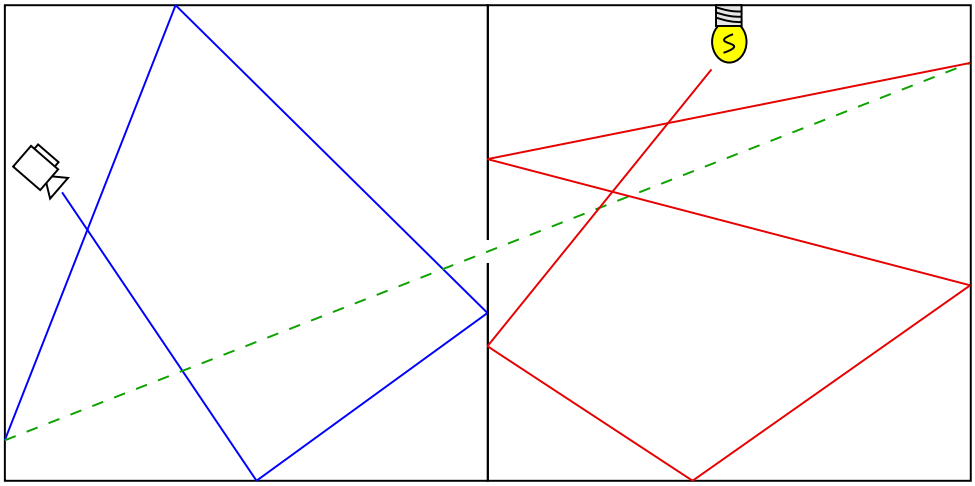


Figure 3.3: Example diagram showing a typical hard-to-render light path. The blue lines represent eye sub-paths, the red eye represents light sub-paths and the green dotted line represents the sub-path connection strategy.

This type of light path is especially useful to showcase limitations and strengths of offline rendering techniques, even more so when the techniques in question are bidirectional, such as BDPT and our ASBDPT.

3.4 Computational Resources

Our method was implemented in the PBRT Renderer [41], based on the third edition of the homonymous book, using C++ programming language. Using this open source and easily extendable framework takes away the hurdle of implementing a completely new renderer from the ground, as several reusable features, such as scene loading, geometry and material definitions, are already implemented and largely out of the scope of this work. PBRT supports several techniques that we used as a basis to our work, such as BDPT. Using this framework also enables us to deploy our new algorithm to the range of operational systems supported by the framework.

The experiments were performed on a machine equipped with a CORE i5-8250U, 1.6GHz processor, 8GB of RAM and an Nvidia GeForce 930MX video card, with 384 CUDA cores, 2GB DDR3 standard memory and 900MHz clock.

Chapter 4

Results

In this chapter, we report and analyze the results obtained when testing our ASBDPT algorithm against BDPT across different test scenes. We also discuss the different parameters that our application can receive as input, and their impact on the efficiency and efficacy of the method.

4.1 Parameters

Our algorithm has two parameters that can be adjusted, according to the application: the number of paths to be distributed in the bins, and a set number of iterations after which all the bins are discarded and remade from scratch, here called the 'restart limit'. We discuss these parameters for one of our test scenes, Veach's Ajar. Additional diagrams with comparisons for the other scenes can be found in Appendix A. For the comparisons we make in the current section, we fix one of the parameters while varying the other one. However, note that generally speaking the behavior we see in these test cases is observable regardless of the value of the fixed parameter.

Figure 4.1 shows the comparison of convergence rate for different values of paths per bin, while keeping the path restart limit fixed at 32. We can see that, for a higher number of samples, the change in the amount of paths per set of bins doesn't impact the efficacy of the algorithm, but the execution time increases significantly.

On the other hand, Figure 4.2 shows the comparison of convergence rate for different values of restart limits, while keeping the amount of paths per bin fixed at 8. We can see that, for a higher number of samples, a lower number of restart limit tend to result in a noisier image, when the restart limit equals to 0, for instance, disabled, or with higher execution time, when the restart limit is equal to 8. Generally speaking, higher restart limits, in our tests ranging from 16 to 32 samples, achieve better results, more efficient and less noisy.

4.2 Test Cases

All tests performed in the remainder of this chapter use the following set of parameter combinations: 8 paths per set of bins, path age of 24 iterations and a general restart age

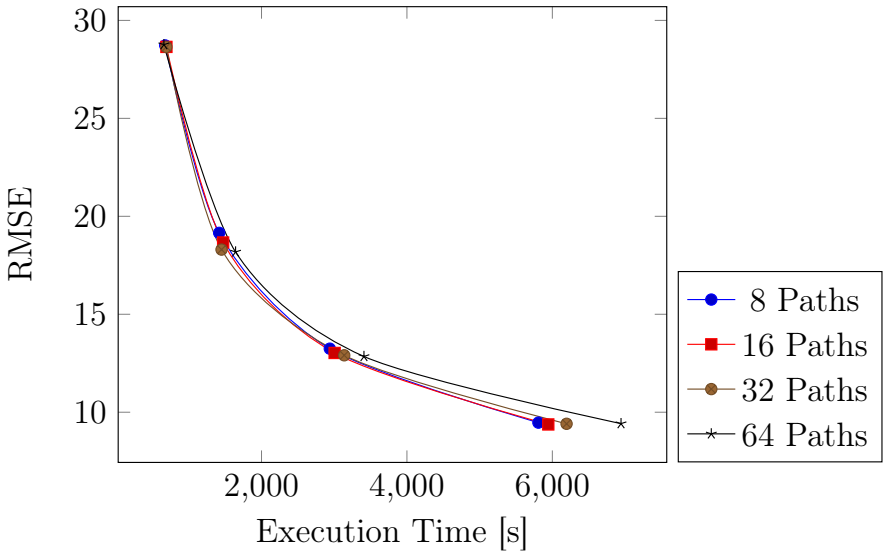


Figure 4.1: Convergence plot showing RMSE value versus the rendering time, in seconds, for a varying number of paths per bin for ASBDPT, for the Veach's Ajar scene, while keeping the restart limit fixed to 32.

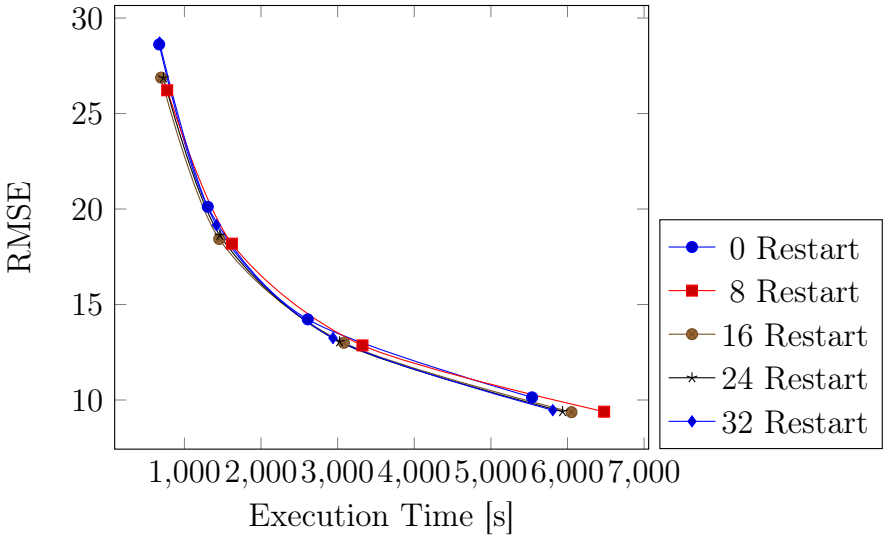


Figure 4.2: Convergence plot showing RMSE values versus the rendering time, in seconds, for a varying restart limit for ASBDPT, for the Veach's Ajar scene, while keeping the number of paths per bin fixed to 8.

of 32 iterations. We found out that, in practice, it is a combination of parameters that manages to obtain results that are either the best, or at least comparable to the best, in most of the test cases we apply our method to.

4.2.1 Veach's Ajar

Figure 4.3 shows a comparison of the noise produced by ASBDPT and BDPT for the Veach's Ajar scene, through RMSE heat maps. This scene, characteristically, contains only one light source which is completely out of direct reach from the camera. Figure 4.4

compares the RMSE values of each technique against their execution times, for the Veach's Ajar scene. Table 4.1 also display the RMSE value of each technique, for each number of samples.

Table 4.1: Comparison of RMSE values for ASBDPT and BDPT for the Veach's Ajar scene.

Samples	ASBDPT	BDPT
32	28.3020	28.7123
64	19.1206	20.1449
128	13.2645	14.2145
256	9.4453	10.1481
512	6.9799	7.2419

4.2.2 Cornell Box

We believe the level of visibility of a light source to the camera directly impacts the quality results of ASBDPT, in comparison to BDPT. Figure 4.5 shows a comparison of the noise produced by ASBDPT and BDPT, as well as their convergence rates on Figure 4.6, for the traditional Cornell Box scene, where the scene has only one light source completely visible to the camera. We can see, both from the convergence plot and the heatmaps, that ASBDPT performs noticeably worse than the standard BDPT. This corroborates our theory that the quality of our light sub-path sampling technique is tied to the level of visibility of the light sources in relation to the camera.

We also believe this scene showcases an additional, expected, limitation of our technique as it stands: a light sub-path bin, in a way, act as a cache of light sub-paths. Paths from the bins are consumed throughout the BDPT iterations, discarded and replaced without any changes to the other bins, degrading the quality of the algorithm results, and by default their convergence, over time.

4.2.3 Modified Cornell Box

To further test how ASBDPT performs against the standard BDPT under different light conditions for a given 3D scene, we modified the Cornell Box scene by partially obstructing the light source. The results of this experiment can be seen in Figure 4.8. Similarly to the original scene, there is still only one light source, but a mesh is close to the light source, making the direct connection of paths from the light source to the camera harder, but not completely impossible, as seen on the rendered results of Figure 4.7.

We are able to conclude that, while our algorithm still is not able to outperform BDPT on this new test scene, not only our method performed better in relation to the original Cornell Box scene, but also BDPT performed worse. In other words, the partial obstruction of the light source made our algorithm perform better – for instance, it has less noise when compared to the reference image – while the standard BDPT is noisier. This

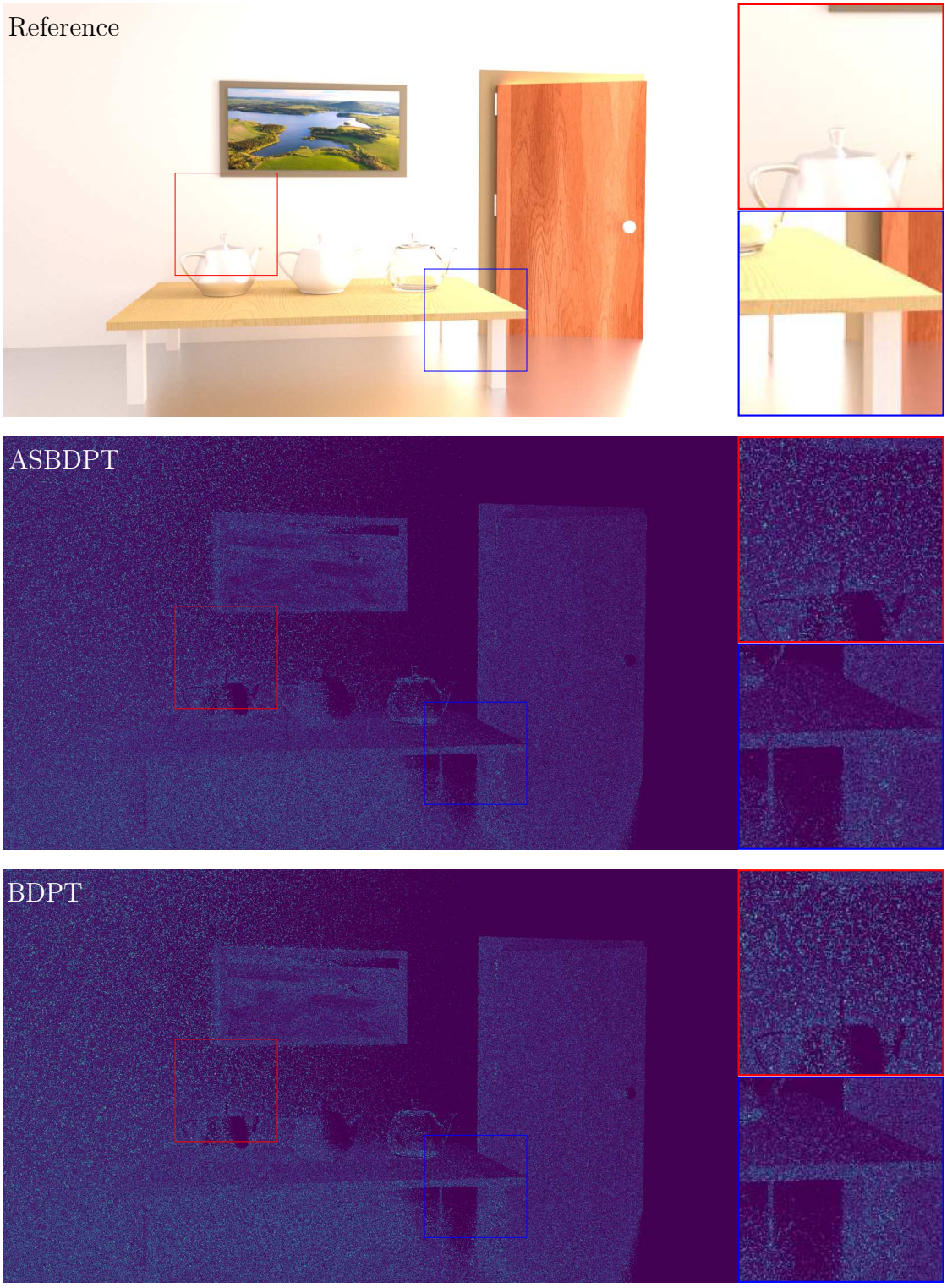


Figure 4.3: Comparison of the heatmap of the RMSE values of Veach's Ajar scene rendered with ASBDPT and BDPT for 64 samples, as well as the ground-truth image rendered with BDPT for 16384 samples. The level of noise produced by each method can be visualized and compared through the zoom-ins.

indicates that the ASBDPT efficiency increases when light sources are, at least, partially blocked, when compared to BDPT. We can see a comparison between the RMSE values

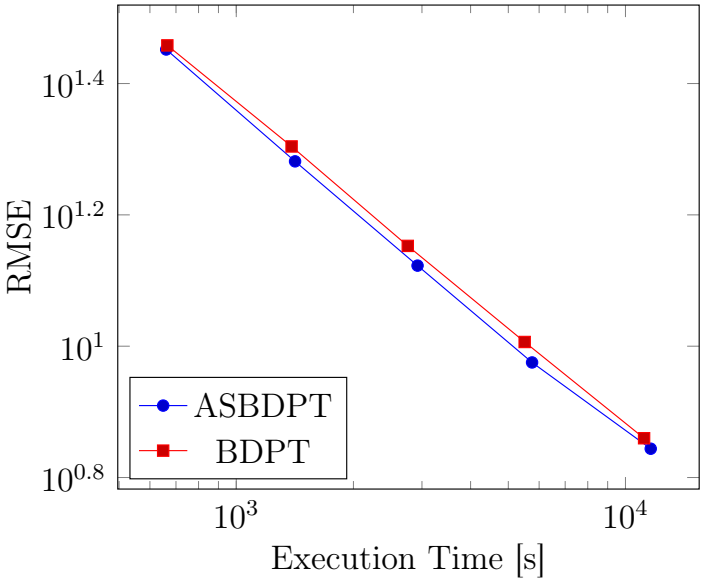


Figure 4.4: Convergence plot showing RMSE value versus the rendering time, in seconds, for ASBDPT and BDPT. Both axes are on a logarithmic scale. Although close, ASBDPT manages to consistently converge faster than BDPT, for the Veach's Ajar scene.

of both techniques for the traditional Cornell Box scene as well as its modified version in Table 4.2.

Table 4.2: Comparison of RMSE values for ASBDPT and BDPT on the traditional Cornell Box scene and its modified version, with limited light exposure. We see that ADBPT achieves lower error against the modified version of the scene, while BDPT had an increase in its error.

Scene		Cornell	Modified Cornell
32 Samples	ASBDPT	2.8022	3.3398
	BDPT	2.7836	3.3369
64 Samples	ASBDPT	2.2055	2.4268
	BDPT	1.9842	2.3851
128 Samples	ASBDPT	1.9630	1.8763
	BDPT	1.4369	1.7241
256 Samples	ASBDPT	1.8629	1.5464
	BDPT	1.0609	1.2604
512 Samples	ASBDPT	1.8169	1.3567
	BDPT	0.8086	0.9453

4.2.4 Bathroom

Figure 4.10 shows the results of the tests we performed with the Bathroom scene. We can see, both by looking at the convergence plot and the heatmaps, on Figure 4.9, that the ASBDPT managed to obtain comparable performance to BDPT throughout the whole execution process. It is worth noting, as well, that the only light source of the scene is

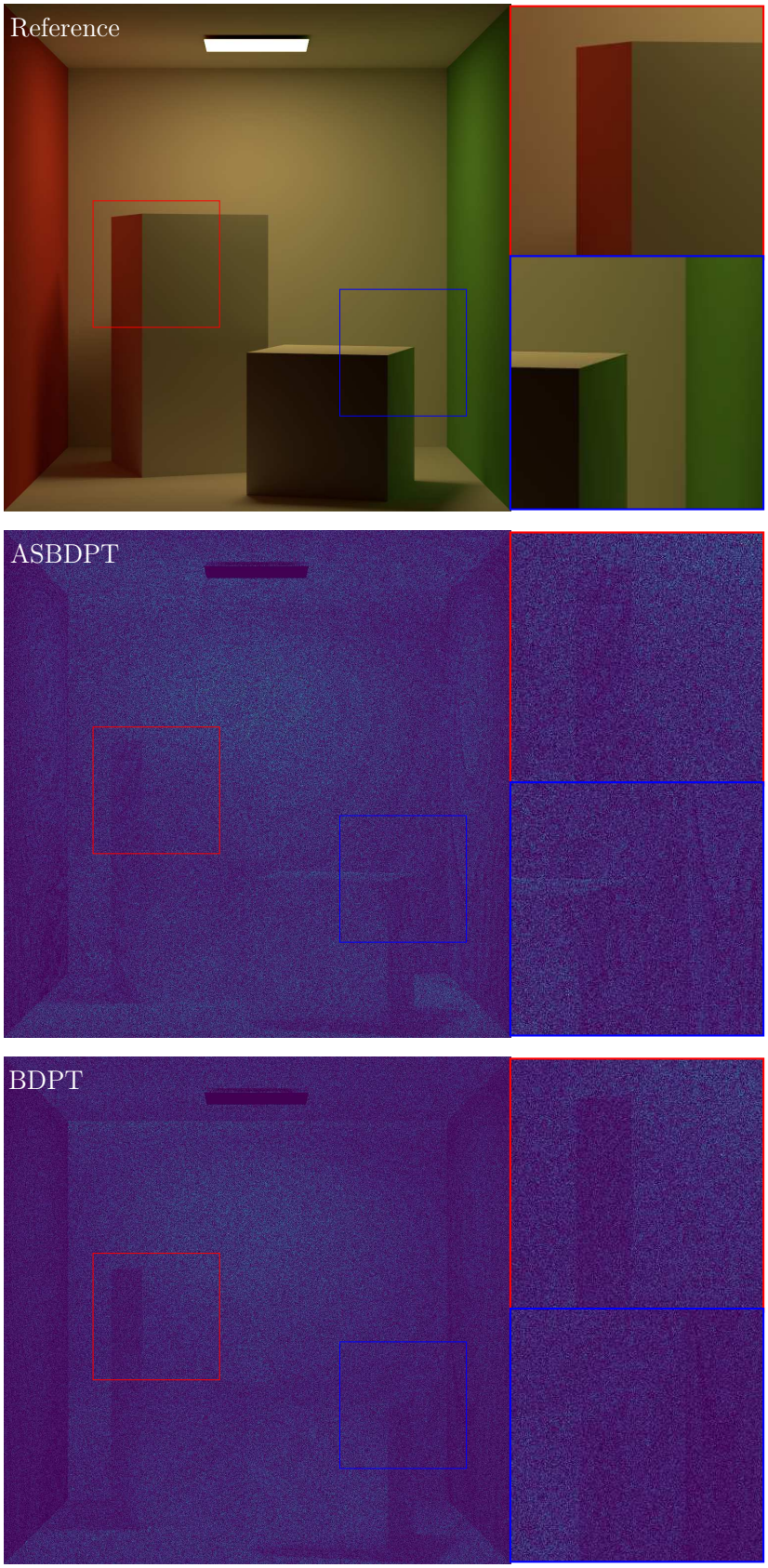


Figure 4.5: RMSE heatmaps for ASBDPT and BDPT, for the traditional Cornell Box scene.

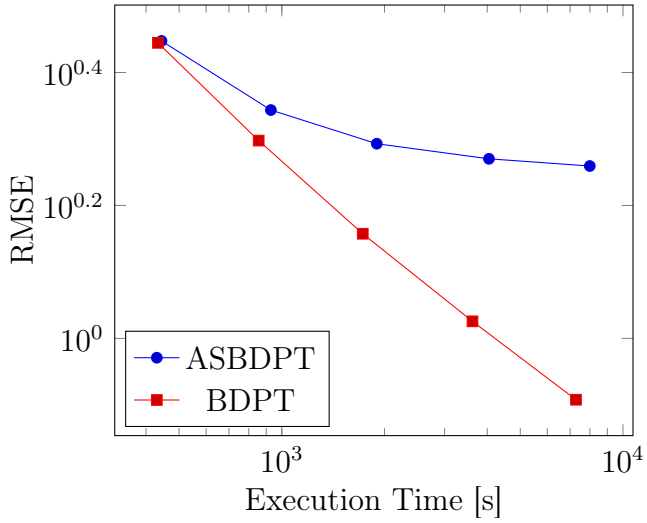


Figure 4.6: Convergence plot showing RMSE value versus the rendering time, in seconds, for ASBDPT and BDPT, for the traditional Cornell Box scene. Both axes are on a logarithmic scale. Generally speaking, ASBDPT is considerably worse than the standard BDPT for the traditional Cornell Box scene.

outside the window, and partially blocked by the window blinds. Table 4.3 also shows the numerical RMSE and time comparison, for the different execution times and samples rendered for the scene.

Table 4.3: Comparison of RMSE values for ASBDPT and BDPT on the Bathroom scene.

Samples	ASBDPT	BDPT
32	17.3230	17.3743
64	13.8195	13.9254
128	11.7632	11.8313
256	10.6083	10.6455
512	10.0122	10.0004

4.2.5 Living Room

In Figure 4.12, we can see the results of tests performed against the Living Room scene. While relatively close to each other, as also seen in the error heatmap in Figure 4.11 and in Table 4.4, our method got outperformed by BDPT towards later samples of the execution. We believe this is another case of the quality degradation we experienced in tests performed with the Cornell Box scene, due to the cache-like nature of our alias bins.

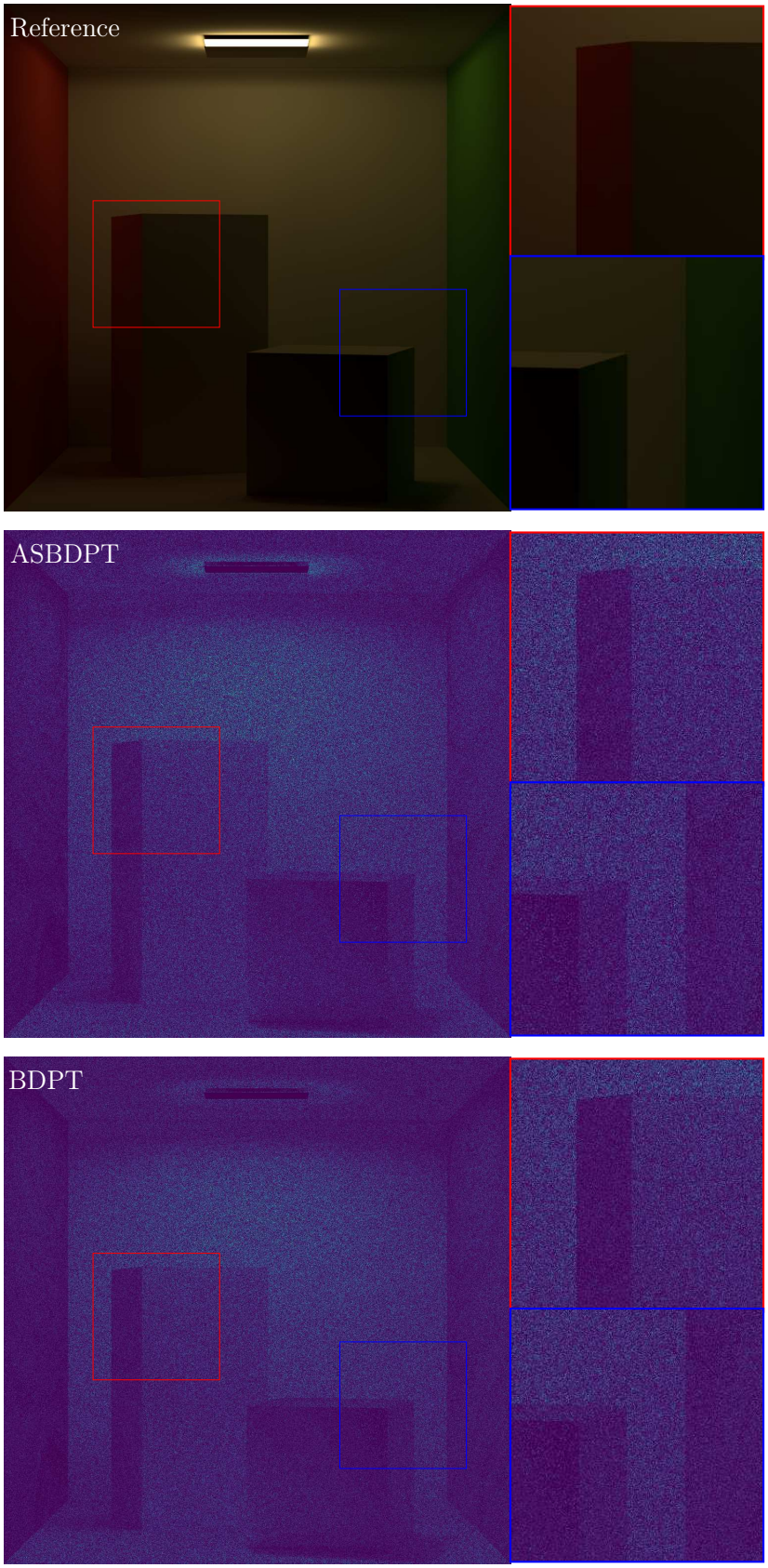


Figure 4.7: RMSE heatmaps rendered for 64 samples for ASBDPT and BDPT, as well as the reference image, for the modified Cornell Box scene.

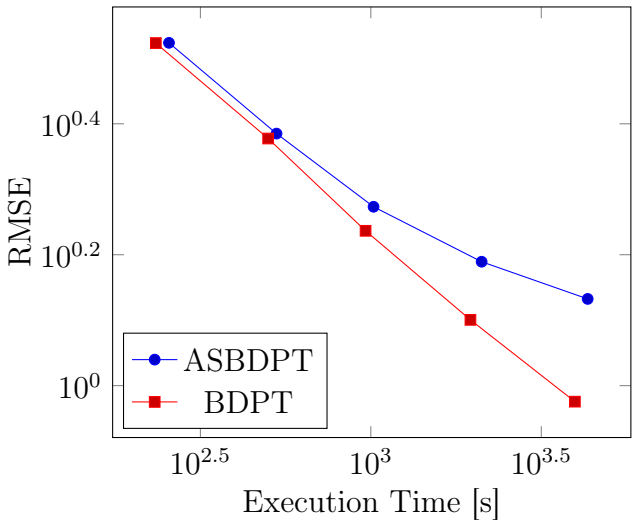


Figure 4.8: RMSE convergence plot showing RMSE value versus the rendering time, in seconds, for ASBDPT and BDPT for the modified Cornell Box scene. Both axes are on a logarithmic scale. The changes made to the traditional scene made BDPT perform worse, resulting in a noisier image than before. Our algorithm, while still not outperforming BDPT, managed to obtain a less noisy image when the light source is partially obstructed.

Table 4.4: Comparison of RMSE values for ASBDPT and BDPT for the Living Room scene.

Samples	ASBDPT	BDPT
32	8.2173	8.2646
64	5.8870	5.9254
128	4.3207	4.2720
256	3.2946	3.0800
512	2.6155	2.2604

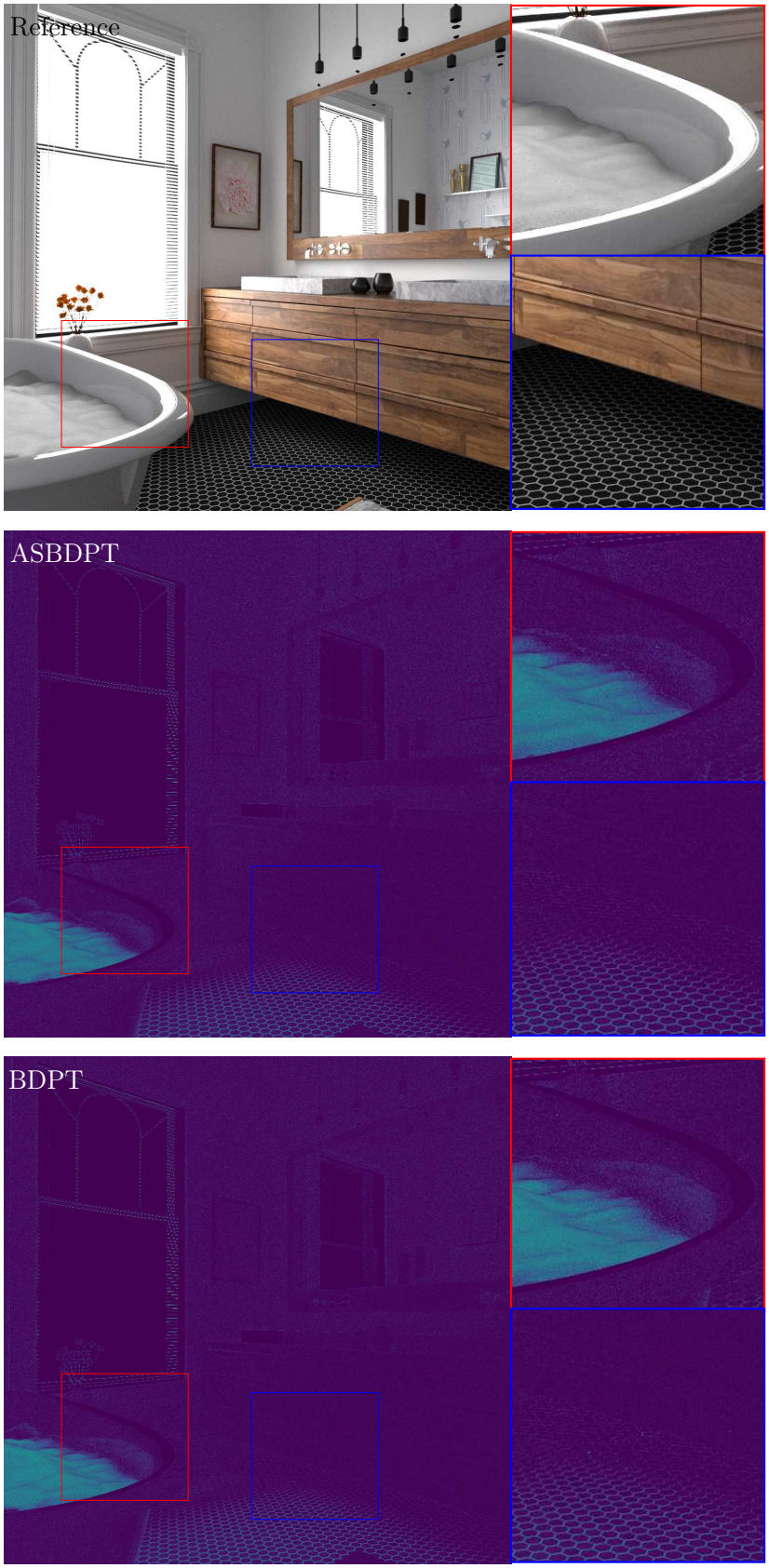


Figure 4.9: RMSE heatmaps rendered for 512 samples for ASBDPT and BDPT, as well as the reference image, for the Bathroom scene.

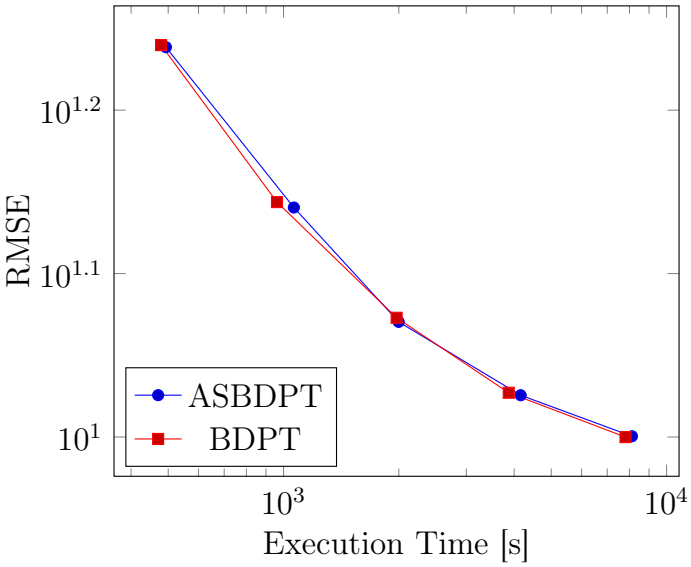


Figure 4.10: Convergence plot showing RMSE value versus the rendering time, in seconds, for ASBDPT and BDPT, for the Bathroom scene. Both axes are on a logarithmic scale. For this test case, both algorithms performed similarly.

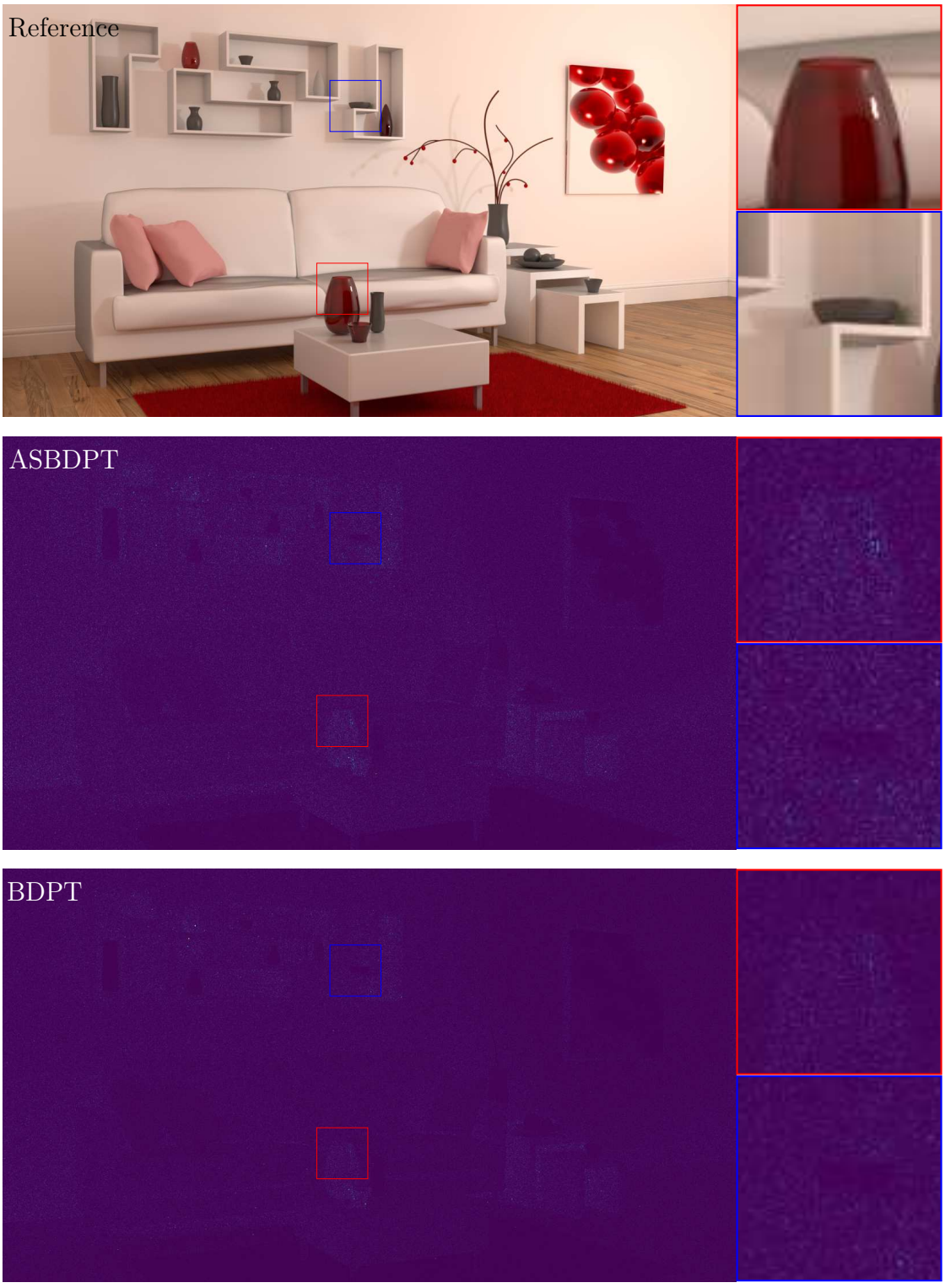


Figure 4.11: RMSE heatmaps, for 128 samples, for ASBDPT and BDPT, for the Living Room scene. We can see that both techniques achieved similar results, with ASBDPT having a noisier result around a specular highlight area, visible on the error heatmap zoom-in in green.

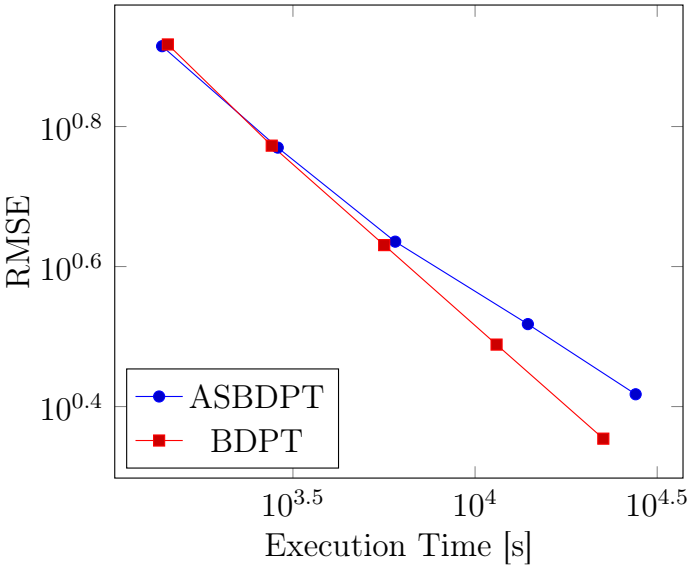


Figure 4.12: Convergence plot showing RMSE value versus the rendering time, in seconds, for ASBDPT and BDPT, for the Living Room scene. Both axes are on a logarithmic scale. We can see that both techniques are close throughout the test case, with BDPT outperforming ASBDPT when more samples are drawn.

Chapter 5

Conclusions and Future Work

Existing light transport methods attempt to improve on the variation reduction of traditional rendering methods, such as Path Tracing and Bidirectional Path Tracing, but they come with their own set of limitations, such as reusing of light paths which increase variance.

This study investigated the problem of importance sampling light sub-paths on Bidirectional light transport methods and whether it was possible to achieve significant variation reduction through it, avoiding eventual issues other methods encountered.

We had, as a main research question:

Is it possible to achieve significant variance reduction by importance sampling sub-path connections with the alias sampling method?

An extensive literature review was performed in order to find promising approaches to explored. We introduced a light sub-path sampling technique that, to the best of our knowledge, is the first to make use of the Alias Sampling method is such a way, selecting sub-paths for connection based on their throughputs. After analyzing the results of our tests in different scenes and light conditions, we showed that the ASBDPT is efficient against environments with limited light exposure, especially when compared against the traditional BDPT method.

While the use of the alias sampling method is efficient for the sake of our application in specific conditions, it comes with some limitations. The quality of the paths in the alias bins degrade over time, as expected. Considering the paths that remain in the bins are, over time, the ones that did not get chosen after each interaction, or the ones that replaced chosen paths, the throughput of the remaining paths tend to be, in the worst case, lower, which according to our technique will tend to result in higher variation.

While constantly refreshing the bins and their paths is not feasible due to performance constraints, resetting the alias paths and rebuilding our bins every certain number of iterations could generate resulting images with lower variance without increasing the execution time too much.

Returning to our main research question, we believe we managed to introduce a new technique that provides a relevant amount of variance reduction with comparable execu-

tion time when conditions are met, such as restricted but not completely blocked paths between the camera and the light sources.

There is room to make the ASBDPT more efficient, both algorithmically and technically. Our method generates more paths than the standard BDPT and other derived approaches and would therefore benefit greatly from more aggressive parallelization approaches, especially on GPU. Recent developments on GPU architectures allow fast generation of paths on hardware level, for example, on Nvidia's RTX GPUs and their RT Cores. The RT Cores make it possible for intersections, with triangles or custom geometry, to be processed at a much faster speed than standard CPU hardware or even non-RTX GPUs. Note that the efficient introduction of parallelization in our technique would be even more desirable and relevant if the reset of alias bins discussed previously in this section were implemented, as the number of generated light sub-paths would increase.

On the algorithm side, we could apply other concepts introduced by Chaitanya et al. [6] in the MBDPT, such as the low-discrepancy sampling (LDS) made possible by their matrix formulation. Adapting their technique, it would be viable to reorder our light path vectors by a metric such as their L2 distance, before splitting up the paths into bins with the use of LDS. This, in turn, would allow for a more equally distributed sampling over the generated paths, avoiding issues such as parts of the scene or image not being sampled enough due to poor sample placement, further improving the variance reduction of our technique.

Bibliography

[1] A. Ahmed, H. Perrier, D. Coeurjolly, V. Ostromoukhov, J. Guo, D. Yan, H. Huang, and O. Deussen. Low-Discrepancy Blue Noise Sampling. *ACM Transactions on Graphics*, 35(6):247:1–247:13, Dec. 2016.

[2] B. Bitterli. Rendering Resources, 2016 (visited on 02/15/2022). `https://benedikt-bitterli.me/resources/`.

[3] B. Bitterli and W. Jarosz. Selectively Metropolised Monte Carlo Light Transport Simulation. *ACM Transactions on Graphics*, 38(6):153:1–153:10, Nov. 2019.

[4] B. Bitterli, C. Wyman, M. Pharr, P. Shirley, A. Lefohn, and W. Jarosz. Spatiotemporal Reservoir Resampling for Real-Time Ray Tracing with Dynamic Direct Lighting. *ACM Transactions on Graphics*, 39(4):148:1–148:17, July 2020.

[5] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo Methods. *Acta Numerica*, 7: 1–49, 1998.

[6] C. R. A. Chaitanya, L. Belcour, T. Hachisuka, S. Premoze, J. Pantaleoni, and D. Nowrouzezahrai. Matrix Bidirectional Path Tracing. In *Eurographics Symposium on Rendering: Experimental Ideas & Implementations*, pages 23–32, Goslar, DEU, 2018. Eurographics Association.

[7] F. Dekking, C. Kraaikamp, H. Lopuhaä, and L. Meester. *A Modern Introduction to Probability and Statistics: Understanding Why and How*, pages 305–306. Springer Texts in Statistics. Springer, 2005.

[8] X. Deng, M. Hašan, N. Carr, Z. Xu, and S. Marschner. Path Graphs: Iterative Path Space Filtering. *ACM Transactions on Graphics*, 40(6), Dec. 2021.

[9] M. Gharbi, T.-M. Li, M. Aittala, J. Lehtinen, and F. Durand. Sample-based Monte Carlo Denoising Using a Kernel-Splatting Network. *ACM Transactions on Graphics*, 38(4):125:1–125:12, 2019.

[10] T. Hachisuka, A. S. Kaplanyan, and C. Dachsbacher. Multiplexed Metropolis Light Transport. *ACM Transactions on Graphics*, 33(4):100:1–100:10, July 2014.

[11] S. Hadadan and M. Zwicker. Differentiable Neural Radiosity. *CoRR*, abs/2201.13190, 2022. URL `https://arxiv.org/abs/2201.13190`.

[12] E. Haines and P. Shirley. *Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and other APIs*, chapter 1, pages 7–14. Apress, Berkeley, CA, 2019.

[13] W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and their Applications. *Biometrika*, 57(1):97–109, 1970.

[14] J. Hughes, A. Van Dam, M. McGuire, J. Foley, D. Sklar, S. Feiner, and K. Akeley. *Computer Graphics: Principles and Practice*. The Systems Programming Series. Addison-Wesley, 3rd edition, 2014.

[15] K. Ivo, P. Vévoda, P. Grittmann, T. Skřivan, P. Slusallek, and J. Křivánek. Optimal Multiple Importance Sampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)*, 38(4):37:1–37:14, July 2019.

[16] J. T. Kajiya. The Rendering Equation. In *13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, pages 143–150, New York, NY, USA, 1986. Association for Computing Machinery.

[17] O. Karlík, M. Šik, P. Vévoda, T. Skřivan, and J. Křivánek. MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling. *ACM Transactions on Graphics*, 38(6), Nov. 2019.

[18] H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon. Differentiable Rendering: A Survey. *arXiv*, pages 1–20, 2020.

[19] C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Computer Graphics Forum*, 21(3):531–540, 2002.

[20] T. Kloek and H. K. van Dijk. Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo. *Econometrica*, 46(1):1–19, 1978.

[21] E. P. Lafortune and Y. Willems. Bi-directional Path Tracing. In *3rd International Conference on Computational Graphics and Visualization Techniques*, Compugraphics, pages 145–153, 1993.

[22] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Transactions on Graphics*, 37(6):222:1–222:11, Dec. 2018.

[23] Y. K. Li. Multiple Importance Sampling, 2015 (visited on 02/15/2022). `https://blog.yiningkarlli.com/2015/02/multiple-importance-sampling.html`.

[24] W. Lin, B. Wang, L. Wang, and N. Holzschuch. A Detail Preserving Neural Network Model for Monte Carlo Denoising. *Computational Visual Media*, 6:157–168, 2020.

[25] M. Loeve. *Probability Theory I*. Graduate Texts in Mathematics. Springer, 1977.

[26] G. Loubet, N. Holzschuch, and W. Jakob. Reparameterizing Discontinuous Integrands for Differentiable Rendering. *ACM Transactions on Graphics*, 38(6):228:1–228:14, Nov. 2019.

[27] F. Luan, S. Zhao, K. Bala, and I. Gkioulekas. Langevin Monte Carlo Rendering with Gradient-Based Adaptation. *ACM Transactions on Graphics*, 39(4):140:1–140:6, July 2020.

[28] A. Marrs, P. Shirley, and I. Wald. *Ray Tracing Gems II*. Apress, 2021. `http://raytracinggems.com/rtg2`.

[29] S. Marschner and P. Shirley. *Fundamentals of Computer Graphics*, chapter 2, pages 357–382. CRC Press, 5th edition, 2021.

[30] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[31] T. Möller, E. Haines, and N. Hoffman. *Real-Time Rendering*, chapter 26. CRC Press, 4th edition, 2018 (visited on 02/15/2022). URL `https://www.realtimerendering.com/Real-Time_Rendering_4th-Real-Time_Ray_Tracing.pdf`.

[32] P. Moreau, M. Pharr, and P. Clarberg. Dynamic Many-Light Sampling for Real-Time Ray Tracing. In M. Steinberger and T. Foley, editors, *High-Performance Graphics - Short Papers*, pages 21–26, Strasbourg, France, 2019. Eurographics Association.

[33] K. Nabata, K. Iwasaki, and Y. Dobashi. Resampling-Aware Weighting Functions for Bidirectional Path Tracing Using Multiple Light Sub-Paths. *ACM Transactions on Graphics*, 39(2):15:1–15:11, Mar. 2020.

[34] K. Nabata, K. Iwasaki, and Y. Dobashi. Two-stage Resampling for Bidirectional Path Tracing with Multiple Light Sub-paths. *Computer Graphics Forum*, 39(7):219–230, 2020.

[35] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Transactions on Graphics*, 38(6), Nov. 2019.

[36] M. Nimier-David, S. Speierer, B. Ruiz, and W. Jakob. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *ACM Transactions on Graphics*, 39(4):146:1–146:15, July 2020.

[37] Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni. ReSTIR GI: Path Resampling for Real-Time Path Tracing. *Computer Graphics Forum*, 40(8):17–29, 2021.

[38] A. Pajot, L. Barthe, M. Paulin, and P. Poulin. Combinatorial Bidirectional Path-Tracing for Efficient Hybrid CPU/GPU Rendering. *Computer Graphics Forum*, 30 (2):315–324, Apr. 2011.

[39] J. Pantaleoni. Blue-Noise Dithered QMC Hierarchical Russian Roulette. *arXiv e-prints*, art. arXiv:1907.12343, July 2019.

[40] A. Pediredla, A. Veeraraghavan, and I. Gkioulekas. Ellipsoidal Path Connections for Time-Gated Rendering. *ACM Transactions on Graphics*, 38(4):38:1–38:12, July 2019.

[41] M. Pharr, W. Jakob, and G. Humphreys. *Physically Based Rendering: From Theory to Implementation (3rd ed.)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, Oct. 2016.

[42] S. Popov, R. Ramamoorthi, F. Durand, and G. Drettakis. Probabilistic Connections for Bidirectional Path Tracing. *Computer Graphics Forum*, 34(4):75–86, July 2015.

[43] D. Russell. An Exploration of Improving Sampling Within Monte Carlo Ray Tracing Using Adaptive Blue Noise. Master's thesis, Bournemouth University, 2019.

[44] M. Sbert and V. Elvira. Generalizing the Balance Heuristic Estimator in Multiple Importance Sampling. *Entropy*, 24(2), 2022.

[45] M. Sbert, V. Havran, and L. Szirmay-Kalos. Multiple Importance Sampling Revisited: Breaking the Bounds. *EURASIP Journal on Applied Signal Processing*, 2018 (1):15, Dec. 2018.

[46] S. Speierer, C. Hery, R. Villemin, and W. Jakob. Caustic Connection Strategies for Bidirectional Path Tracing, 2018 (visited on 02/15/2022). Pixar Technical Memo #18-01. `http://graphics.pixar.com/library/CausticConnections/`.

[47] J. Talbot, D. Cline, and P. Egbert. Importance Resampling for Global Illumination. In K. Bala and P. Dutre, editors, *Eurographics Symposium on Rendering (2005)*. The Eurographics Association, 2005.

[48] H. K. van Dijk and T. Kloek. Experiments With Some Alternatives For Simple Importance Sampling In Monte Carlo Integration. Econometric Institute Archives 272281, Erasmus University Rotterdam, May 1983.

[49] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1998.

[50] E. Veach and L. Guibas. Bidirectional Estimators for Light Transport. In *Photorealistic Rendering Techniques*, pages 145–167, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[51] E. Veach and L. J. Guibas. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, pages 419–428, New York, NY, USA, 1995. Association for Computing Machinery.

[52] E. Veach and L. J. Guibas. Metropolis Light Transport. In *24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, pages 65–76, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[53] M. D. Vose. A Linear Algorithm for Generating Random Numbers with a Given Distribution. *IEEE Transactions on Software Engineering*, 17(9):972–975, Sept. 1991.

[54] A. J. Walker. An Efficient Method for Generating Discrete Random Variables with General Distributions. *ACM Transactions on Mathematical Software*, 3(3):253—256, Sept. 1977.

[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004.

[56] L. Wasserman and L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer, 2004.

[57] P. Weier, M. Droske, J. Hanika, A. Weidlich, and J. Vorba. Optimised Path Space Regularisation. *Computer Graphics Forum*, 40(4):139–151, 2021.

[58] B. Xu, J. Zhang, R. Wang, K. Xu, Y.-L. Yang, C. Li, and R. Tang. Adversarial Monte Carlo Denoising with Conditioned Auxiliary Feature Modulation. *ACM Transactions on Graphics*, 38(6):224:1–224:12, Nov. 2019.

[59] C. Zhang, B. Miller, K. Yan, I. Gkioulekas, and S. Zhao. Path-Space Differentiable Rendering. *ACM Transactions on Graphics*, 39(4):143:1–143:19, July 2020.

[60] X. Zhang, D. Silverstein, J. Farrell, and B. Wandell. Color Image Quality Metric S-CIELAB and its Application on Halftone Texture Visibility. In *IEEE International Computer Conference, Digest of Papers*, pages 44–48, 1997.

Appendix A

Parameter Comparison Diagrams

Note that, for some of these diagrams, not all samples for all combinations of parameters have been rendered. Nevertheless, the data here displayed corroborates with the findings of Section 4.1.

A.1 Cornell Box

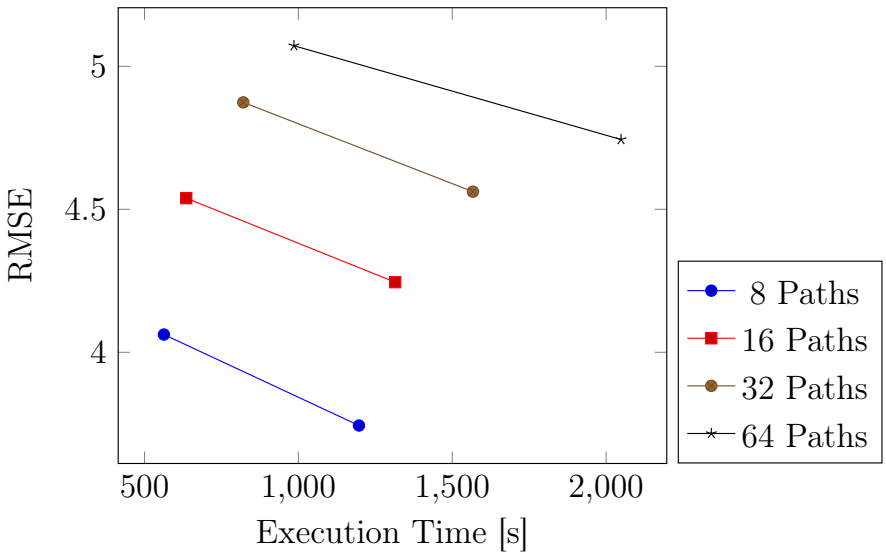


Figure A.1: Convergence plot showing RMSE value versus the rendering time, in seconds, for a varying number of paths per bin for ASBDPT, for the traditional Cornell Box scene, while keeping the restart limit fixed to 8.

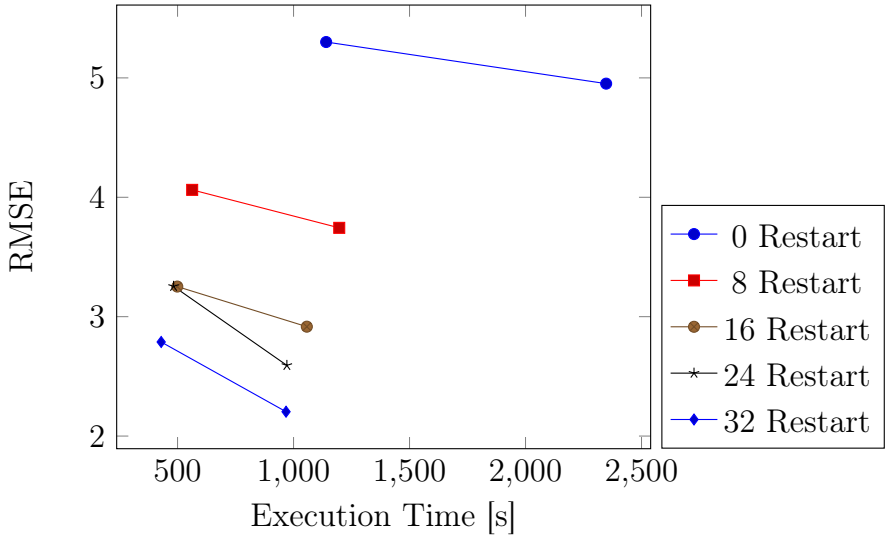


Figure A.2: Convergence plot showing RMSE values versus the rendering time, in seconds, for a varying restart limit for ASBDPT, for traditional Cornell Box scene, while keeping the number of paths per bin fixed to 8.

A.2 Modified Cornell Box

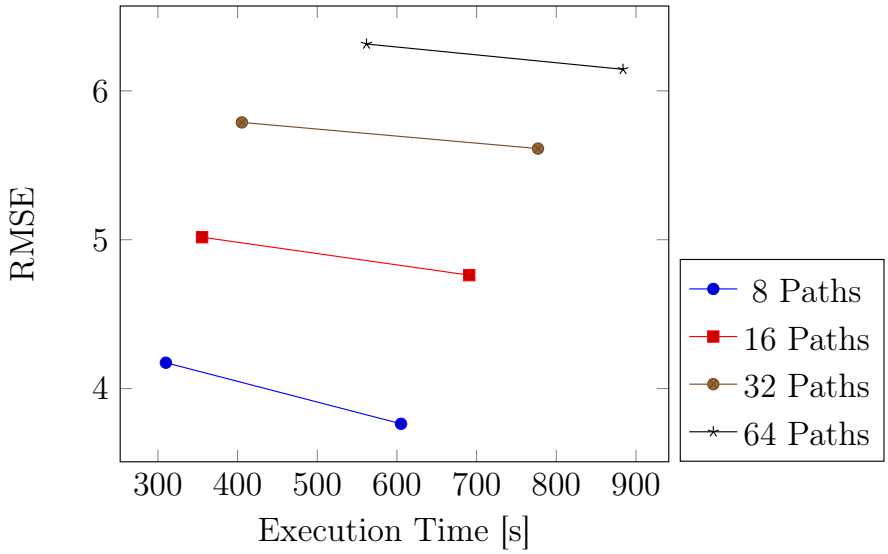


Figure A.3: Convergence plot showing RMSE value versus the rendering time, in seconds, for a varying number of paths per bin for ASBDPT, for the modified Cornell Box scene, while keeping the restart limit fixed to 8.

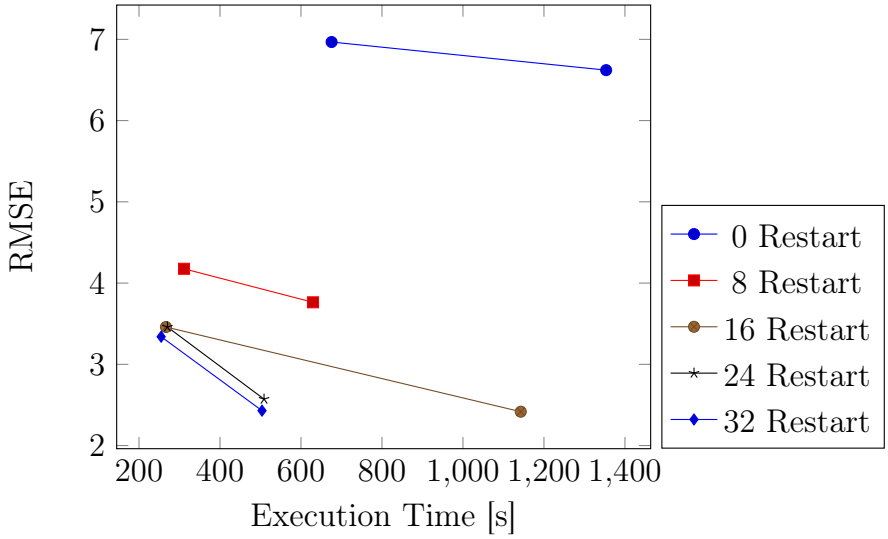


Figure A.4: Convergence plot showing RMSE values versus the rendering time, in seconds, for a varying restart limit for ASBDPT, for the modified Cornell Box scene, while keeping the number of paths per bin fixed to 8.

A.3 Bathroom

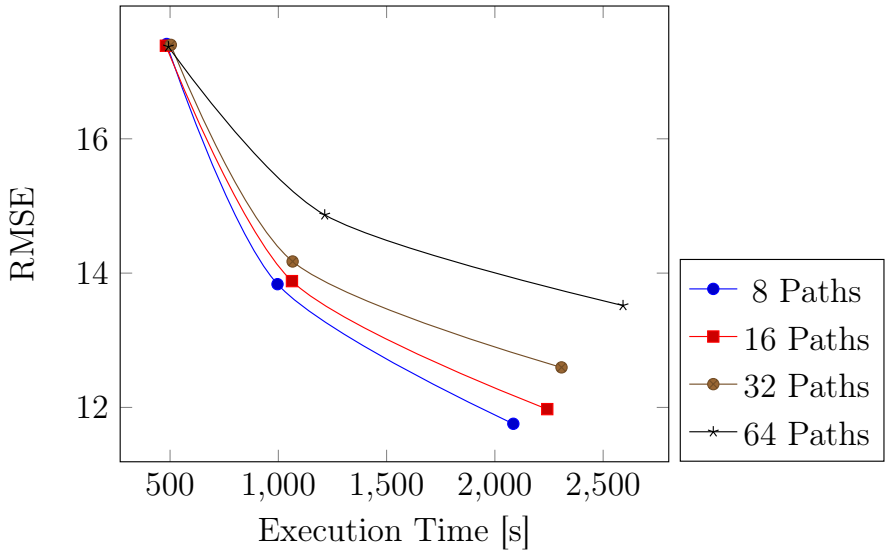


Figure A.5: Convergence plot showing RMSE value versus the rendering time, in seconds, for a varying number of paths per bin for ASBDPT, for the Bathroom scene, while keeping the restart limit fixed to 32.

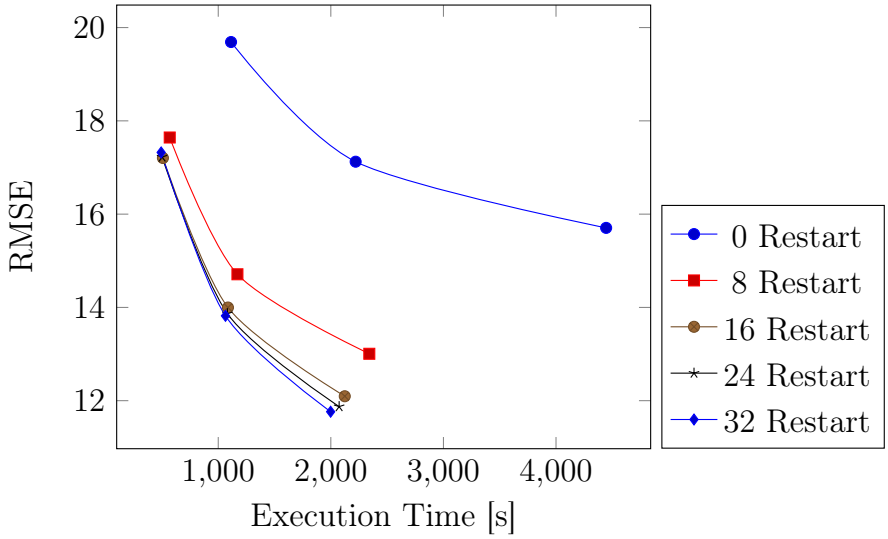


Figure A.6: Convergence plot showing RMSE values versus the rendering time, in seconds, for a varying restart limit for ASBDPT, for the Bathroom scene, while keeping the number of paths per bin fixed to 8.

A.4 Living Room

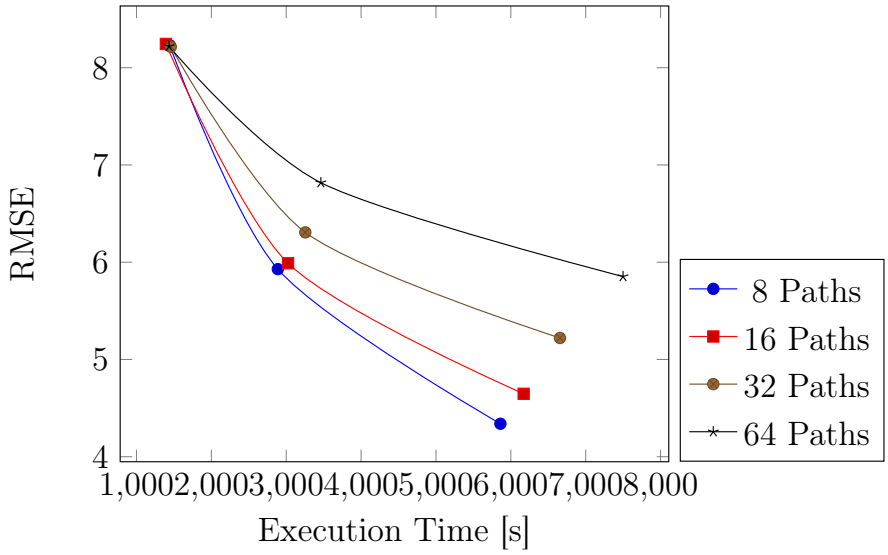


Figure A.7: Convergence plot showing RMSE value versus the rendering time, in seconds, for a varying number of paths per bin for ASBDPT, for the Living Room scene, while keeping the restart limit fixed to 32.

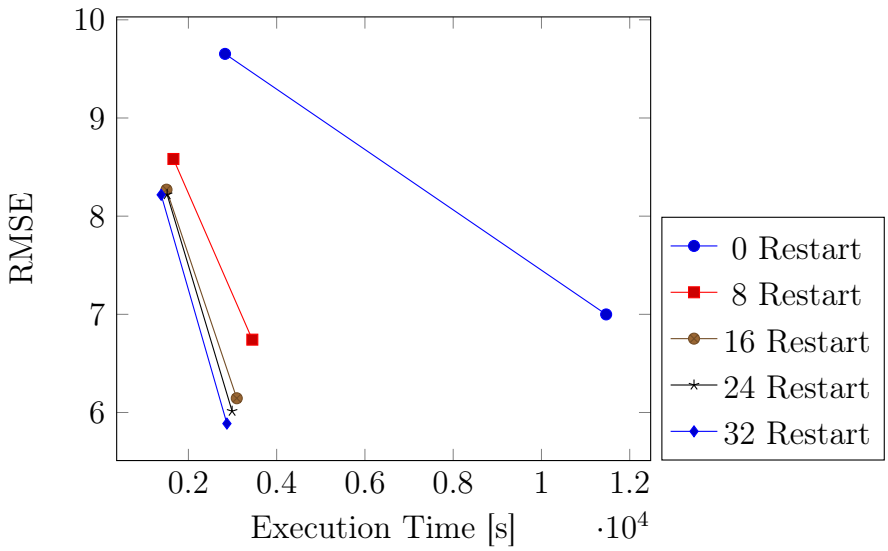


Figure A.8: Convergence plot showing RMSE values versus the rendering time, in seconds, for a varying restart limit for ASBDPT, for the Living Room scene, while keeping the number of paths per bin fixed to 8.