TU WIEN Informatics

Advanced Importance Sampling Techniques for Virtual Ray Lights

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Michael Oppitz, BSc
Registration Number 01227129

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Assistance: Dr.techn. Károly Zsolnai-Fehér, B.Sc. M.Sc.

Vienna, 18th April, 2022

Michael Oppitz

Michael Wimmer

Informatics

Fortgeschrittene Stichprobenentnahme nach Wichtigkeit für Virtuelle Lichtstrahlen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Michael Oppitz, BSc
Matrikelnummer 01227129

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Mitwirkung: Dr.techn. Károly Zsolnai-Fehér, B.Sc. M.Sc.

Wien, 18. April 2022

Michael Oppitz

Michael Wimmer

Erklärung zur Verfassung der Arbeit

Michael Oppitz, B.Sc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. April 2022

Michael Oppitz

Acknowledgements

I would like to thank Károly Zsolnai-Fehér who supported and guided me through the creation of this thesis. I would also like to thank Professor Michael Wimmer for his support and feedback. Both gave me the necessary time to complete this work, even if it took a little longer. Additionally, I would like to thank Professor Derek Nowrouzezahrai who review my initial ideas and provided valuable feedback.

I would also like to sincerely thank my parents, family and friends who supported and encouraged me to complete this work. Even in difficult times, they always strengthened and encouraged me.

Danksagung

Ich möchte Károly Zsolnai-Fehér danken, der mich bei der Erstellung dieser Arbeit unterstützt und begleitet hat. Ich möchte auch Professor Michael Wimmer für seine Unterstützung und sein Feedback danken. Beide gaben mir die nötige Zeit, diese Arbeit abzuschließen, auch wenn es etwas länger gedauert hat. Außerdem möchte ich Professor Derek Nowrouzezahrai danken, der meine ersten Ideen überprüft und wertvolles Feedback gegeben hat.

Ich möchte mich auch ganz herzlich bei meinen Eltern, meiner Familie und meinen Freunden bedanken, die mich unterstützt und ermutigt haben, diese Arbeit abzuschließen. Auch in schwierigen Zeiten haben sie mich immer wieder gestärkt und ermutigt.

Abstract

This thesis provides new importance sampling techniques for Virtual Ray Lights for Rendering Scenes in Participating Media. We will discuss the foundations of rendering scenes with participating media first, to get an understanding of this topic. Furthermore, we will provide an overview of different approaches that can be used for the rendering of these scenes.

Virtual Ray Lights is an algorithm that traces light rays from the light source through the scene, which are then evaluated individually for each camera ray. Importance sampling is used on both rays to get a samples for each ray for which their contribution can be calculated.

As a solid understanding of the original algorithm is needed to understand the new approaches that we introduce, a focus is laid on explaining the mathematical foundations of the approach. We highlight the shortcomings that we found for rendering anisotropic participating media and introduce our solutions to solve them more efficiently.

We provide two different solutions to the problem that we evaluated in the original algorithm. Our solutions are explained mathematically, via pseudo code and are evaluated with a multitude of tests. The goal of this thesis is to provide new, simple, robust and fast solutions to rendering scenes with anisotropic participating media.

Kurzfassung

In dieser Arbeit stellen wir neue Techniken zur Stichprobenentnahme nach Wichtigkeit für virtuelle Lichtstrahlen (Virtual Ray Lights) in Partikelwolken vor. Wir werden zunächst die Grundlagen des Renderings von Szenen mit Partikelwolken diskutieren, um ein Verständnis für dieses Thema zu bekommen. Außerdem werden wir einen Überblick über verschiedene Ansätze geben, die für das Rendering von diesen Szenen genutzt werden können.

Virtual Ray Lights ist ein Algorithmus, der Lichtstrahlen von der Lichtquelle durch die Szene verfolgt und dann für jeden Kamerastrahl einzeln auswertet. Die Stichprobenentnahme nach Wichtigkeit wird auf beiden Strahlen verwendet, um eine Abtastung für jeden Strahl zu erhalten, für die ihr Beitrag berechnet werden kann.

Da ein gutes Verständnis des ursprünglichen Algorithmus erforderlich ist, um die von uns vorgestellten neuen Ansätze zu verstehen, liegt ein Schwerpunkt auf der Erläuterung der mathematischen Grundlagen des ursprünglichen Ansatzes. Wir heben die Unzulänglichkeiten hervor, die wir beim Rendern anisotroper Partikelwolken gefunden haben, und stellen unsere Lösungen vor, um diese effizienter zu lösen.

Wir stellen zwei verschiedene Lösungen für das Problem, das wir im ursprünglichen Algorithmus bewertet haben vor. Unsere Lösungen werden Mathematisch und mittels Pseudocode erklärt und mit einer Vielzahl von Tests evaluiert. Das Ziel dieser Arbeit ist es, neue, einfache, robuste und schnelle Lösungen für das Rendern von Szenen mit anisotropen Partikelwolken bereitzustellen.

Contents

Abstract **xi**

Kurzfassung **xiii**

Contents **xv**

1 Introduction **1**
1.1 Motivation and Problem Statement . . . 1
1.2 Aim of the Work . . . 3
1.3 Contribution . . . 4
1.4 Structure of the Thesis . . . 5

2 Light Transport **7**
2.1 The Rendering Equation . . . 7
2.2 Monte Carlo Estimator . . . 8
2.3 Participating Media Properties . . . 8

3 Related Work **13**
3.1 Photon Approaches . . . 13
3.2 Virtual Light Approaches . . . 18

4 Original Approach **23**
4.1 Methodology . . . 23
4.2 Algorithm . . . 24
4.3 Importance Sampling . . . 26

5 New Approaches **37**
5.1 Sum of two Cauchy Distributions . . . 38
5.2 Mixture Model . . . 41

6 Implementation **45**
6.1 Original Approach - Isotropic Sampling . . . 45
6.2 Original Approach - Anisotropic Sampling . . . 46
6.3 Sum of Cauchy Distributions . . . 49

6.4 Mixture Model 51

7 Results & Comparison 53
7.1 Equal Error 54
7.2 Equal Time 55
7.3 Equal Sample 57

8 Conclusion & Future Work 59
8.1 Conclusion 59
8.2 Future Work 60

A Full Results 61
A.1 Cornell Box 63
A.2 Unidirectional 71
A.3 Bidirectional1 79

B Other Approaches 95
B.1 Product of two distributions 95
B.2 Product of two Cauchy distributions 95
B.3 Other Distributions for the Mixture Model 96

List of Figures 97

List of Tables 101

List of Algorithms 103

Acronyms 105

Bibliography 107

CHAPTER 1

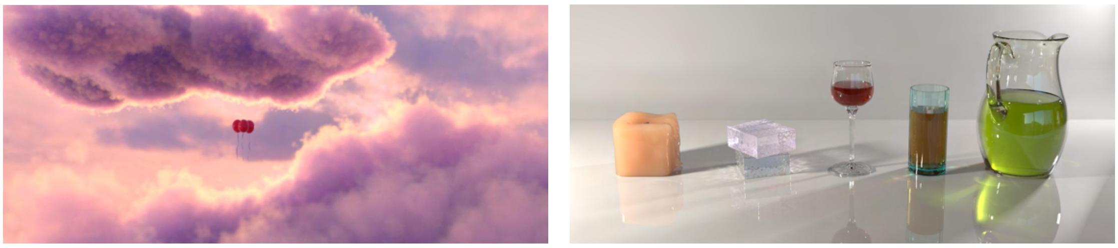
Introduction

1.1 Motivation and Problem Statement

In computer graphics, there are many different challenges to overcome when simulating real-world phenomena. Some of these challenges can be solved rather easily, whereas others require complex ideas to approximate the real world. One of the challenges that requires complex ideas is light simulation. For realistic imagery, the calculation of light plays an important role. Therefore, the topic of light simulation is extensively studied. The issue with light simulation is that an infinite number of light rays has to be traced through a scene to get realistic results. At an intersection with a surface, the surface properties determine how the rays will be adjusted for every direction possible. According to the rendering equation, the incoming radiance from all other points affects the exiting radiance from a point. Therefore, approximations are needed, as an analytic solution is impossible.

While the calculation of light is already a complex task, one of the phenomena that make this task even harder to solve is participating media. There are many different examples for participating media, but the most common ones are fluids and smoke, as can be seen in Figure 1.1. In participating media, the light rays can also intersect with the media itself, not just a surface. For every intersection in the medium, the properties of the medium at that point determine how the new light rays are traced further. As again, the contribution of all incoming radiance is needed to calculate the exiting radiance at a point, a new layer of complexity is added. In most cases, light simulation in participating media requires complex algorithms, or otherwise, it will take a lot of time to get realistic results.

For even the simplest of scenes, an infinite number of calculations would be needed to achieve correct results. Therefore, approximations have to be made. In many cases, random samples are taken and evaluated and the average of these samples converges to



VPM BRE

Figure 1.1: Different variations of participating media. Source: [NSJ14] (left), [KGH+14] (right)

the correct solution. This is called the Monte Carlo method [PJH16, Chapter 13]. Most approaches work with statistical methodologies to evaluate which calculations yield a higher importance to the final result. This method is called importance sampling, as just randomly picking samples would require much more time, and in some cases, may not even converge in any reasonable time frame. Therefore, importance sampling is used as a way to determine which calculations are necessary. Importance sampling is a technique that uses a probability function to estimate which samples should be taken in various scenarios.

As an example, a light ray that traveled through thick media may have lost nearly all of its power before arriving at a point that we want to evaluate. The radiance that this light ray adds can be so low that it does not add anything to the scene that would be visible to the naked eye. One scenario to use importance sampling is the calculation of the direction of a light ray after a surface intersection, another is the calculation of where a light ray intersects with a medium. A great introduction to this topic was provided by Steve Marschner from Cornell University [Mar15], in which he describes the mathematical and pseudo-code formulations of Volumetric Path Tracing.

There are different ways of simulating light in participating media. Out of all these approaches, there are two groups that introduce common structures to the approximation of light. One group approximates the incident light via photons. In these approaches, photons are traced from a light source through the scene, and then their contribution for points of interest is evaluated. For the second group, virtual lights are created instead of photons. For a point of interest, these groups of algorithms evaluate the many lights that were created in the step before. Both of these groups of approaches have different shortcomings that new variations of them try to correct.

The approach that we will have a look at in detail is Virtual Ray Lights (VRL) [NNDJ12b]. It is part of the group of virtual light approaches, and in contrast to other approaches it creates virtual ray lights, instead of virtual point lights or virtual lights in other geometric variations. In this algorithm, light rays are cast from the light source through the scene to be stored in a data structure. In the next step, camera rays are cast to evaluate the light rays that were created in the previous step. These first two steps are straightforward, and only the calculation of the new virtual light rays is an innovation compared to virtual light approaches that were introduced before. The idea is to always evaluate a pair of rays, one light, and one camera ray, at a time. First, a point on the light ray, and secondly, a point on the camera ray, is chosen. Then, the contribution of the point on the light ray is calculated corresponding to the point on the camera ray.

Evaluating which points are chosen is the main focus of the VRL paper. The selection process is done via importance sampling. Novák et al. evaluated that the inverse squared distance, and, in the anisotropic case, the product of the phase functions need to be evaluated to create a conclusive sampling function. For the light ray, the sample is chosen according to the distance between the two rays. For this process, the closest points between the two rays are calculated and with the help of the inverse squared distance between them, a point on the virtual light ray can be sampled. For the sample on the camera ray, there is a case separation to be made. In the easier case, the media is isotropic, which means that the light is scattered uniformly in all directions. Therefore, a very practical approach by Kulla et al. [KF11] can be used to sample the light ray. In this approach, the inverse squared distance can be used again to create a function for importance sampling. The probability density function (pdf) is therefore proportional to the inverse squared distance

$$\mathrm{pdf}(u_i, v_i) \propto \frac{1}{w(u,v)^2}, \tag{1.1}$$

where $\mathrm{pdf}(u_i, v_i)$ is the probability function for choosing the point u_i and v_i.

Unfortunately, the case of anisotropic media is significantly more complex. We have to factor in the direction of both rays, as otherwise the anisotropy of the media is not properly accounted for. Whereas in the isotropic case, a function approximate to the inverse squared distance was enough to achieve correct results, in the anisotropic case, the function has to be approximate to the inverse squared distance times the product of the phase function f_s at both points, u_i and v_i

$$\mathrm{pdf}(u_i, v_i) \propto \frac{f_s(u_i) f_s(v_i)}{w(u,v)^2}. \tag{1.2}$$

As there is no closed-form solution to this problem, Novák et al. propose to construct a piecewise-linear function from which a corresponding piecewise-quadratic function can be computed which has to be inverted to create samples. Although this solution creates accurate results, it is, compared to other importance sampling techniques, notably more complex, which reduces the comprehensibility and impairs the performance.

1.2 Aim of the Work

The aim of this work is to provide a new importance sampling technique that solves the anisotropic case of VRL. As the original technique already creates a highly accurate approximation of the target function, the idea is to find representations that may approximate the target functions slightly worse, but are easier and faster to calculate. Therefore, more samples can be taken within the same time frame, as a result of performance improvements with an easier methodology. Additionally, the new importance sampling approach should also be less complex than the original approach. For these goals, there are three rules that we have set ourselves as guidelines.

1. **The new approach has to be simpler than the original one.** As simplicity is difficult to measure, we will compare the complexity via two different methods. One method is the comparison of the mathematical formulation. This comparison may not be of scientific importance, but it can be seen as a way of estimating how easy it is to understand this approach.
2. **All cases that were solved by the original approach, also have to be solved by the new approach.** The new algorithm has to be robust against all changes of settings the original approach can calculate. If the new approach could only calculate a subset of the settings that the original approach can, a comparison of complexity or performance would be invalid.
3. **The new approach has to be faster than the old one when comparing the time it takes to take a single sample.** Additionally, although it is not the main intention, the new algorithm should produce more accurate results, in less time. This is especially important the more time is given to render.

1.3 Contribution

We evaluated different ways on how to simplify anisotropic importance sampling in VRL. Some ideas were discarded as they would have been similarly complicated, others were just too slow, and some did not converge to satisfying results in various cases. However, we found two methods that actually lead to improvements. Both of them replace the piecewise approximation of the product of the phase functions by an approximation.

First, we present the idea of using the sum of two Cauchy distributions instead of the product of the phase function. This solution can even be computed in a closed form, which results in a clearly defined sampling function. This approach still works in a similar fashion as the original importance sampling method, therefore it is possible, with just minimal changes, to replace the original version of the importance sampling with the new approach.

Second, we present our findings of different mixture models of distributions. In our search for distributions that can be solved in a closed form, we found that the mixture model of two probability distributions itself already provides acceptable results. As in the first approach, the mixture models replace the product of the phase functions. The advantage of this idea is that mixture models are very common and can therefore be expected to be understood quite easily. As mixture models are also used in various other fields, there is a lot of material available online.

In Appendix B we shortly describe other approaches that we tried and why they did not succeed.

The implementation of our contribution is made available via a GitHub project.

1.4 Structure of the Thesis

In this thesis, we will first provide some insight into light transport and the notations and terms of it in Chapter 2. After that, we use this knowledge to discuss photon-based and virtual light approaches (Chapter 3). With this we want to create a general understanding of the topic and the challenges that lie ahead. Afterwards, in Chapter 4, we will provide a detailed explanation of VRL, as this is the basis of understanding the problem. The focus is hereby laid on the mathematical formulation of the importance sampling techniques, as this is the foundation of our new ideas.

In Chapter 5 we describe our new approaches. This chapter gives a detailed explanation and mathematical evaluation, whereas the next chapter targets implementation details (Chapter 6). After showing the implementation details, we compare our results with the original approach in Chapter 7. This is done mostly by comparing renderings by equal sample, equal time and equal error comparisons. Additionally, we will compare the complexity of algorithms by their mathematical description and implementation length. In the end we will conclude what has been provided in this thesis and evaluate future work in Chapter 8.

CHAPTER 2

Light Transport

A key to understanding the complex approaches that will be discussed in the following chapters is a sound foundation of the basic concepts of rendering. Therefore, this chapter focuses on the general mathematical formulations that are required for light transport (Section 2.1). As we show the difficulties in solving this equations, we also highlight a method of approximating them in Section 2.2. Important participating media properties are explained in detail in Section 2.3. As this is on the side of general knowledge, many parts of this chapter follow the theory of excellent lectures [ZF18, Mar15], books [PJH16] and theses [Jar08, Nov14].

2.1 The Rendering Equation

There are many different ways of calculating light transport in a scene, but the underlying concept is always the same, the rendering equation [Kaj86, ICG86]. The rendering equation is a physically correct method of calculating the behavior of light. Therefore, it is the key for creating realistic imagery.

In 1986 James Kjiya [Kaj86] and David Immel et al. [ICG86] published the rendering equation simultaneously and it is still one of the most important equations in computer graphics. The equation can be written as

$$L_o(x, \vec{\omega}) = L_e(x, \vec{\omega}) + \int_\Omega L_i(x, \vec{\omega}') f_r(x, \vec{\omega}, \vec{\omega}') \cos\theta \, d\vec{\omega}', \tag{2.1}$$

which calculates the exiting radiance L_o at a point x in a direction $\vec{\omega}$. The emission L_e at the point x in the direction $\vec{\omega}$ is added to the sum of all incoming radiance from all directions $\vec{\omega'}$, which is written as the integral of these directions $\int_\Omega \dots d\vec{\omega}$. For every incoming direction $\vec{\omega'}$, the incoming radiance L_i at this point, the bidirectional reflectance distribution function (BRDF) f_r at this point and the angle between both directions, the outgoing direction $\vec{\omega}$ and the incoming direction $\vec{\omega'}$, have to be evaluated.

This already shows the main problem with the rendering equation, it is not solvable in a closed form. As we need the incoming radiance from all other points to evaluate the exiting radiance from a point, we would need the result of our calculation in advance, to calculate the incoming radiance. Therefore, it is only possible to approximate the rendering equation, but not solve it. Many variations of approaches approximate the rendering equation, like Recursive Ray Tracing [Gla89, Whi79], Path Tracing [LW93], Metropolis Light Transport [VG97] and Photon Mapping [Jen01].

2.2 Monte Carlo Estimator

A way of approximating these complex equations that is important to our case, is the approximation by the Monte Carlo method, which is used in Virtual Ray Lights. This methodology, also called the Monte Carlo integration, approximates an integral by means of an estimator, the Monte Carlo estimator. This estimator takes samples and averages their results to converge to the correct result after many samples taken. For an integral $F(x)$ of a function $f(x)$ the Monte Carlo estimator is written as

$$F(x) \approx \frac{1}{N} \sum_{N}^{i=1} \frac{f(x)}{\mathrm{pdf}(x)}, \tag{2.2}$$

where N is the number of samples, x is the current sample, $f(x)$ is the result of the function with the current sample and $\mathrm{pdf}(x)$ is the probability density function (pdf) of the sample. The pdf can be chosen freely, but the closer it fits to the target function, the faster the estimator converges. Unfortunately, having an accurate pdf is not enough, as it also has to be fast to process on a computer. This already describes the goal of this thesis, finding a pdf for a function that fits closely and is fast to compute.

2.3 Participating Media Properties

There are a few important properties that need to be explained to get a better understanding for participating media. These properties define how the media interacts with light. This is only a general overview to get an idea of the various concepts. In Equation 4.1 of Section 4.1 light transport in participating media is evaluated further, while in this section the basic participating media properties are discussed. The participating media properties that are explained in the following sections are visualized in Figure 2.1.

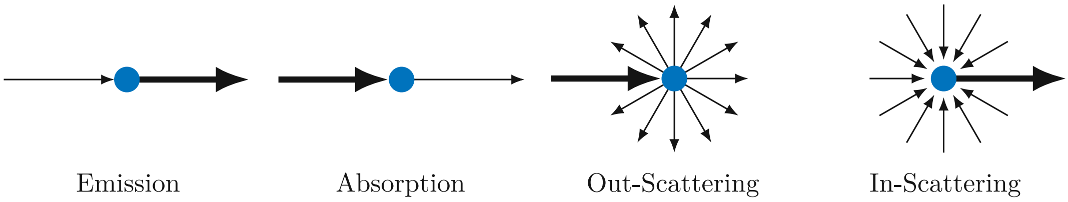


Figure 2.1: The four properties of participating media.

2.3.1 Emission

A participating medium can add radiance to a light ray that travels through it, this is called emission. The emission of a medium is defined by the emission coefficient $\varepsilon(x)$ that defines the amount of radiance that is added at a specific point x. Many participating media like smokes or most fluids do not emit light, but there are a few that do like fire or fluorescent liquids.

2.3.2 Absorption

As a medium can add radiance, it can also reduce the radiance. This phenomenon is called absorption and is defined with the coefficient $\sigma_a(x)$. The absorption coefficient is defined as the percentage of which the radiance is reduced over a unit length distance. An absorption coefficient of 0.5 reduces the radiance to fifty percent after traveling a unit distance.

2.3.3 Out-Scattering

As with the absorption coefficient, the scattering coefficient $\sigma_s(x)$ defines a reduction of the radiance. This time, the radiance is not absorbed by the medium, but it scattered into other directions. Similarly to the absorption coefficient, the scattering coefficient defines a percentage of radiance lost over a unit distance.

2.3.4 Extinction

The absorption and the scattering coefficient can be combined, as they both represent a loss of radiance. This coefficient is called the extinction coefficient $\sigma_t(x)$ and is defined as $\sigma_t(x) = \sigma_a(x) + \sigma_s(x)$. As it is sometimes needed to know the loss of light after a distance t, the absorption coefficient can be expressed with the transmission function

$$T_r(x, x_t) = e^{-t\sigma_t(x)}. \tag{2.3}$$

2.3.5 In-Scattering

The In-Scattering is the exact opposite of the Out-Scattering. Instead of the radiance that is lost into other directions, the radiance that is coming from these other directions is evaluated and added. It is not straightforward how the light is distributed from a point, therefore a phase function $f_s(x, \vec{\omega}, \vec{\omega}')$ is used to evaluate this distribution.

In the easiest case, the distribution is even and all directions are used equally. This is called an isotropic phase function which is always

$$f_s(x, \vec{\omega}, \vec{\omega}') = \frac{1}{4\pi}. \tag{2.4}$$

A visual representation of this function can be seen in Figure 2.2 (middle).

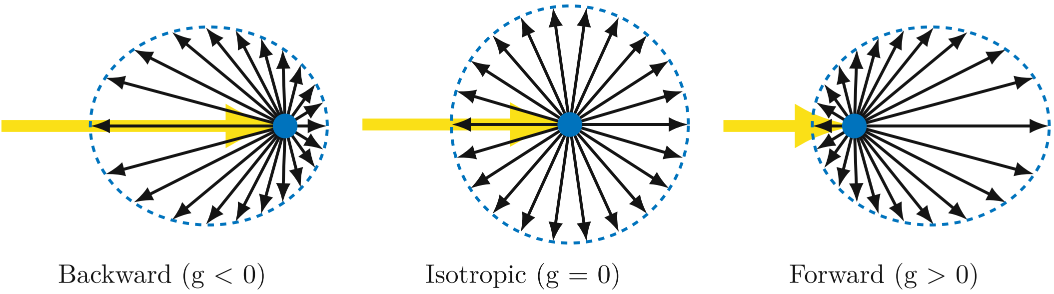


Figure 2.2: The possibilities of scattering in a participating media. The light ray (yellow) is scattered into various directions depending on the media properties.

In the case of other scattering, there are various options of phase functions to use like the Henyey-Greenstein phase function [HG41], the Rayleigh phase function [Buc95], the Mie-Debye phase function [FW59] and many more.

As an example, the Henyey-Greenstein phase function introduces the variable $g \in [-1, 1]$ as a way of adjusting the scattering. A negative g-value creates backwards scattering (Figure 2.2 left) and a positive g-value creates forward scattering (Figure 2.2 right). The Henyey-Greenstein phase function for an angle θ is written as

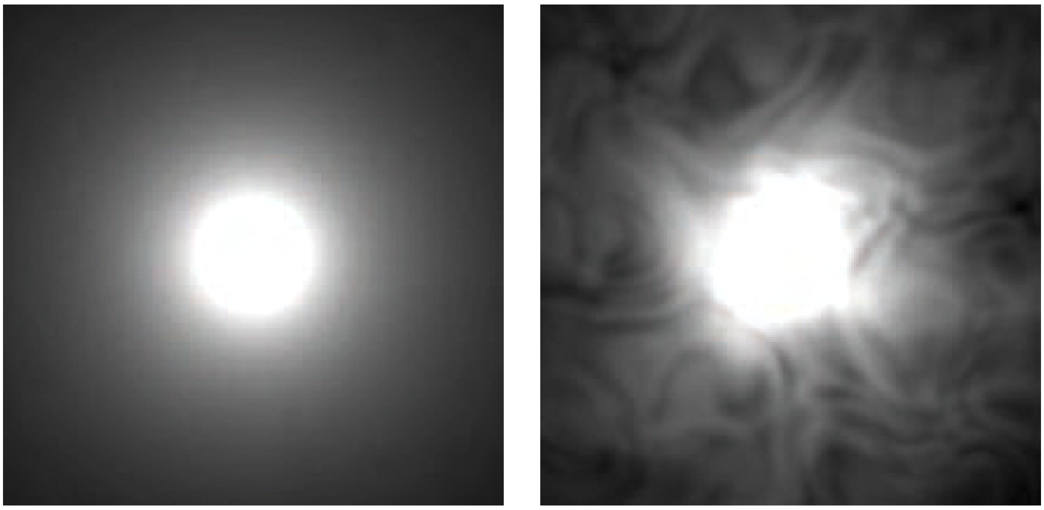
$$f_s(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g\ \cos\theta)^{1.5}}. \tag{2.5}$$

2.3.6 Homogeneous and Heterogeneous Media

In the real world, two points in a participating medium rarely share the same properties with each other. There are always differences between the properties, however small they may be. This type of medium is called heterogeneous (Figure 2.3 (right)). In computer graphics, the process of evaluating heterogeneous media takes much effort, as the medium has to be evaluated at all of its points and no shortcuts can be taken. Therefore, in many cases, homogeneous media (Figure 2.3 (left)) is used to simplify the calculations. A homogeneous medium has constant properties that can therefore be evaluated much easier. In this thesis, if not otherwise stated, we work with homogeneous media.

2.3.7 Single and Multiple Scattering

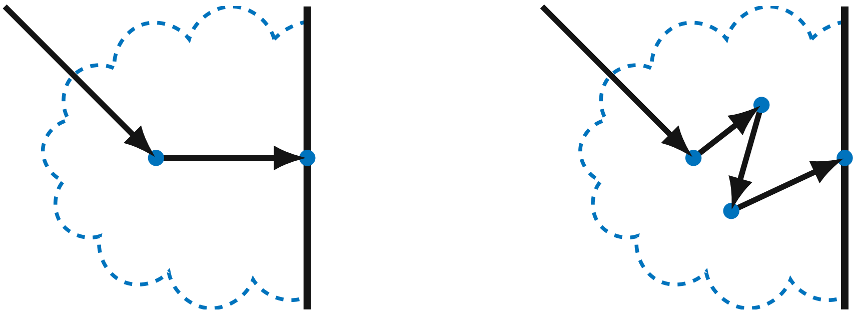
The scattering inside a medium is an important factor in how realistic the final result looks. So for a perfectly realistic image, that would mean that we should use as much scattering as possible. Unfortunately, the calculation of the scattering events and the further evaluation takes a lot of time. We separate here between single (Figure 2.4 (left)) and multiple (Figure 2.4 (right)) scattering. For multiple scattering, the number of bounces inside the medium can still be adjusted and experimented with. In most cases it makes sense to find a number of bounces that still creates results in a reasonable amount of time while still maintaining the visual benefits.



Homogeneous Media

Heterogeneous Media

Figure 2.3: Different types of participating media. Source: [Jar08]



Single Scattering

Multiple Scattering

Figure 2.4: A visual comparison of single and multiple scattering.

CHAPTER 3

Related Work

There are various ways of approaching the rendering of participating media. In this chapter, we look at two groups of approaches that are very similar. One group contains the photon-based approaches (Section 3.1), which cast photons through the scene and estimate the illumination with these photons. In the second group, virtual lights are created and the contribution of these lights is calculated. This group will be discussed in Section 3.2. It will be shown that both of these groups share a lot of common ideas within the development of new approaches. This will guide us towards Virtual Ray Lights and newer approaches that were developed since then.

3.1 Photon Approaches

Photon Mapping (PM) is a well-known approach for surface rendering, with variations of it being used in many different areas. The approach was developed by Henrik Wann Jensen in his PhD thesis [Jen96b] and was first published in 1995 [Jen96a]. PM is a two-pass approach. In the first pass, the photon map is constructed by casting a predefined number of photons from the light source through the scene. In the second pass, Ray Tracing (RT) is used to evaluate the illumination from the previously cast photons. A visual representation of photons that were created and a reference point around which the photons are sampled can be seen in Figure 3.1.

For this approach, many different variations were developed. Some of the most popular are Progressive Photon Mapping (PPM) [HOJ08] and Stochastic Progressive Photon Mapping (SPPM) [HJ09]. This section focusses on further variations that can calculate the light transport within participating media.

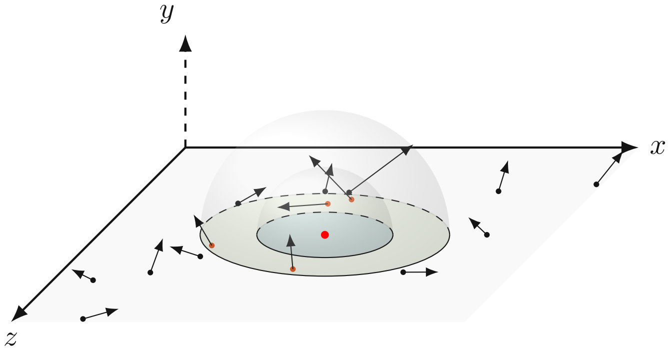


Figure 3.1: To search for a specific number of photons (in this case four) around a point. The sphere is expanded until enough photons are enclosed by it. In the first step (blue, inner sphere) no photons are found within the area. The sphere is therefore expanded (green, outer sphere) to find four photons inside of it.

3.1.1 Volumetric Photon Mapping (VPM)

The first variation of PM that was developed for participating media is Volumetric Photon Mapping [JCKS02]. In PM a photon map is used to store the photons that were traced through the scene. For VPM, evaluating this photon map can already be an issue. As photons are not only stored on surfaces, but within the medium as well, an efficient data structure has to be used. To evaluate the photons in the medium, a process called ray marching is used. Ray marching evaluates the medium step by step with adjustable step sizes.

As with the original PM algorithm, VPM still suffers from the same disadvantages. Both approaches are only theoretically consistent, as only with an infinite number of photons, there is no error. Unfortunately, the storage of the photons requires a lot of memory, and the evaluation of them takes a lot of time. Therefore, a compromise has to be found how many photons are stored and evaluated. Other approaches solve this problems to a part as they are progressive. With these approaches, the photon-tracing step can be repeated multiple times and evaluated again.

3.1.2 Beam Radiance Estimate (BRE)

One of the flaws of VPM is the usage of ray marching. Ideally, the steps taken would be infinitely small to evaluate all points along the ray, but this is not feasible, as this many evaluations per ray would have a negative effect on the performance of the algorithm. Choosing a larger step size leads to noise and artefacts. Therefore, the Beam Radiance Estimate [JZJ08] was introduced by Jarosz et al. to gather the photons over the whole camera ray. This is formulated as

$$L(x,\vec{\omega}) = \int_0^s T_r(x,x_t)\sigma_s(x_t)L_i(x_t\vec{\omega})\,dt. \tag{3.1}$$

For a practical implementation, this formulation has to be slightly adjusted. As currently, a photon would have to lie directly on the ray, which is practically impossible, the ray has to be expanded into a beam to evaluate an area around the ray.

The BRE solves multiple problems of VPM. In VPM it is possible that a photon is evaluated multiple times. Consider a low step size, with a wider area of evaluation, in this case a photon may be multiple times in the area of evaluation. For the BRE this cannot be the case, as it evaluates the photons all at once for every beam. Additionally, the noise is reduced and the convergence is faster. A comparison of these two techniques can be seen in Figure 3.2.

Reference 02:38:57 VPM 00:04:21 BRE 00:04:15

Figure 3.2: A comparison of VPM, the BRE and a reference solution that was rendered with VPM and a small step size. The render time is provided in hh:mm:ss. Source: [JZJ08]

3.1.3 Photon Beams (PB)

An important contribution to the field of photon-based approaches was made by Jarosz et al. [JNSJ11]. They introduce a new way of presenting photons. Instead of storing a single point, a whole ray is stored. This ray existed beforehand, but was ended early on by scattering events that produced the photon. Figure 3.3 shows a comparison on how photons are stored.

With this new way of storing photons, the data density is increased. Areas that previously would not have had any data in it, can now be populated by data that was already evaluated in the tracing step, but is now actually stored. The beams, which are now additional information sources, can be used to produce results with faster convergence. Figure 3.4 shows a comparison between VPM and PB. One practical byproduct of this

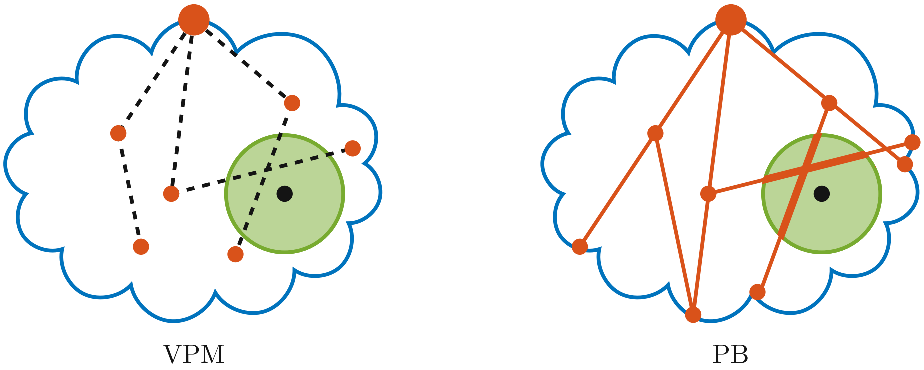


Figure 3.3: A visual comparison of the ideas behind VPM and PB.

approach is that the number of traces does not change, but the density of the data still increases.


Reference
BRE 00:00:31
PB 00:00:25

Figure 3.4: A comparison of BRE and the PB, to a reference solution. The render time is provided in hh:mm:ss. Source: [JNSJ11]

3.1.4 Progressive Photon Beams (PPB)

As for PM, PB is an unbiased approach, but only in the theoretical sense. For it to be unbiased, an infinite amount of beams are needed, as only then the blur kernel gets an infinitely small width. Progressive Photon Beams by Jarosz et al. [JNSJ11] solves this problem by adapting the original algorithm to allow multiple passes that can be combined. This change has a close correlation as the adaptations that were made from PM to PPM.

3.1.5 Unifying Points, Beams and Paths (UPBP)

Most of the previous approaches are used only for certain use cases or are mostly just of scientific importance. Therefore, Unifying Points, Beams and Paths [KGH^{+}14, Vév15] has such a high significance to the field. In this paper, Krivanek et al. create an estimator that chooses between different approaches, depending on various media and scene properties.

As can be seen in Figure 3.5, the candle, to the right of all examples, can be rendered best with the BRE, whereas other participating media, or just areas of the participating

VPM BRE

PB BPT

Figure 3.5: A comparison of UPBP, VPM, BRE, PB and BPT. The images for VPM, BRE, PB and BPT are weighted by their contribution to the overall scene. Source: [KGH$^+$14]

media, can be rendered better with other approaches. Therefore, UPBP gathers and uses the best parts of all approaches. Due to the scope of this thesis, there cannot be a thorough evaluation of this work, but it is highly recommended to the reader.

3.1.6 Higher-Dimensional Photon Representations

From the following approaches, a clear trend can be seen. The dimensionality of the photon representation has been expanded, to create a bigger area of data density. Bitterli et al. [BJ17] take this idea even further and expand the photon representation to planes and volumes. They determine that when the increase in dimensionality already was an improvement initially, a further improvement can be made by adding further dimensions. Although the evaluation of these higher dimensional photon representations can be costly, it is still an interesting idea that can spark further improvements to the area. A comparison between renderings with different photon dimensionality can be seen in Figure 3.6.


PB PP PV

Figure 3.6: A comparison of photon beams, photon planes and photon volumes. All of these images were created within the same render time. Source: [BJ17]

3.2 Virtual Light Approaches

Only a few years after the invention of PM Alexander Keller introduced Virtual Point Lights [Kel97] for surface rendering. As described before, PM creates photons that are evaluated in a second pass. For VPL instead of photons, virtual lights are created that are evaluated in the next steps. This is the main difference between photon-based approaches and Virtual Lights (VL)-approaches.

3.2.1 Virtual Point Lights (VPL)

In 1997 Alexander Keller introduced the idea to evaluate the traced data as VLs. This can be interpreted as rendering the scene many times with a different VL every time. The final result is given by the sum of all the results with as single VL. This method was originally developed for surface rendering, but can be extended to participating media [NNDJ12b]. The radiance at a point is then given as

$$L(x,\vec{\omega}) \approx L_e(x,\vec{\omega}) + \sum_{i=1}^{N} V(x_i)G(x_i)L_i(x_i,\vec{\omega}_i), \tag{3.2}$$

where $L_e(x,\vec{\omega})$ is the emitted radiance from all non-VL sources, N is the number of VLs, $V(x)$ is the visibility at a point x, $G(x)$ the geometric term at a point x and $L_i(x,\vec{\omega})$ the in-scattered radiance at a point x in the direction $\vec{\omega}$. This in-scattered radiance is calculated with

$$L_i(x,\vec{\omega}) \approx \frac{\Phi f_s(\theta_p) f_s(\theta_u) T_r(w)}{w^2}, \tag{3.3}$$

where Φ is the power of the light source, $f_s(\theta_p)$ is the phase function at the position of the virtual light, $f_s(\theta_u)$ is the phase function at the position of estimation, $T_r(w)$ is the transmission and w is the distance from the light to x_u.

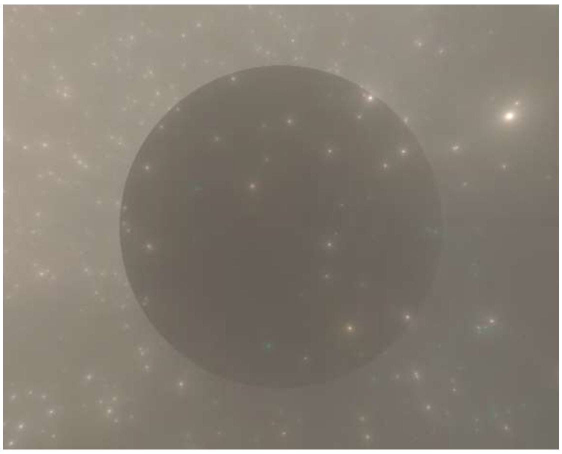


Figure 3.7: A scene rendered with VPL. The singularities in the medium can be seen as there are not enough point lights to illuminate the scene evenly. Source: [NNDJ12b]

One of the main advantages of this method is the speed of evaluation. Rendering a scene with a single point light is very fast compared to the evaluation of photons. Unfortunately, local singularities can occur, which can be seen in Figure 3.7. These singularities can be resolved by either clamping or blurring, as will be discussed in the next section.

3.2.2 Virtual Spherical Lights (VSL)

To solve the issue of singularities occurring, one idea is to blur the single point light over a certain distance. This results in a sphere around that point, and the corresponding algorithm is therefore called Virtual Spherical Lights [HKWB09]. VSL was originally developed for surface rendering, but can be extended to participating media in a similar way as VPL. The usage of spheres instead of points reduces the number and intensity of the singularities significantly. How this method works in theory can be seen in the visual comparison of Figure 3.8.

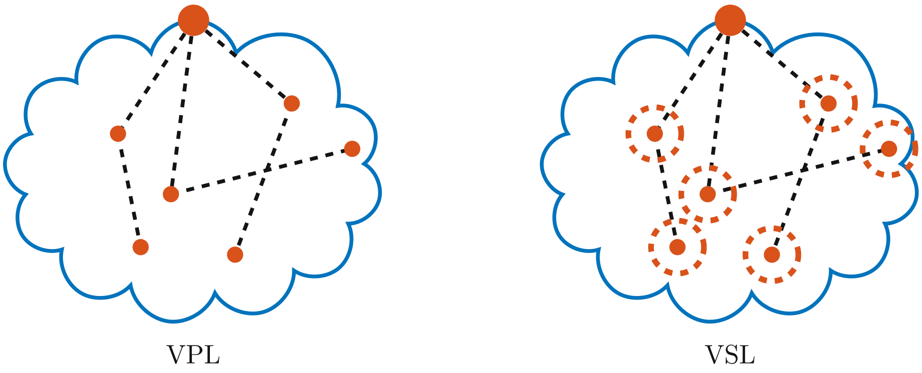


Figure 3.8: A visual comparison of the ideas behind VPL and VSL.

To create the VSLs the VPLs are expanded for the specified radius. The properties, like the calculation of normals, the BRDF and phase functions stay exactly the same as for the original corresponding VPLs.

For the evaluation of the radius of the VSL the density of the lights is used. Therefore, for an infinite number of lights, the radius converges to zero and the approach becomes identical to VPL. This form of consistency of this algorithm is therefore identical to the consistency for PM, only in a theoretical sense, with an infinite amount of light or photons, respectively. Additionally, the blurring of the VPLs introduces bias.

3.2.3 Virtual Ray Lights (VRL)

As PM was further developed to PB, a similar development took place for VLs. In 2012, Novák et al. developed Virtual Ray Lights [NNDJ12b]. This algorithm, and improvements on it, are the main focus of this thesis. A comprehensive analysis of it is provided in Chapter 4, whereas this section focuses only on the general contributions for the completeness of this chapter.

As PB expanded PM from a 1D photon representation to a 2D photon representation, the same is true regarding VLs in VRL. As the name of the algorithm already reveals, VRL uses the whole ray instead of a single point. A visual comparison of VRL and VPL, and the change in data representation, can be seen in Figure 3.9.

Additionally, the BRE is used for the sampling of the camera ray. Therefore, the sampling takes place from one ray, the camera ray, to another ray, the light ray. The disadvantage of VPL of having local singularities can be significantly reduced by using the whole ray instead of a single point. The mathematical formulations and descriptions are omitted in this chapter, but are evaluated in detail in Chapter 4.

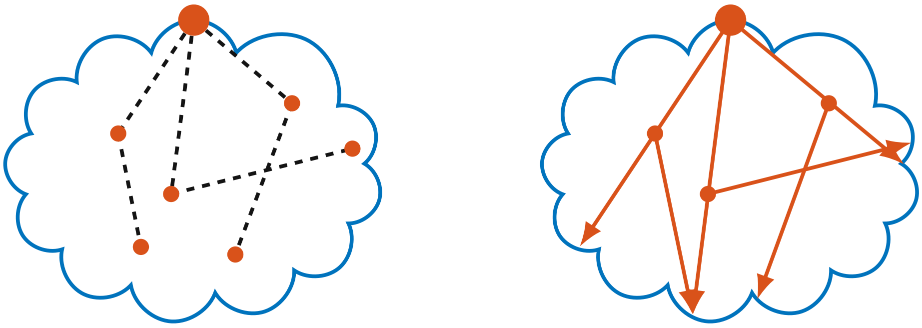


Figure 3.9: A visual comparison of the ideas behind VPL and VRL.

One of the main problems is the sampling of the two rays. Another major contribution of Novák et al. is the introduction of a product importance sampling technique for the camera and light ray. Especially for anisotropic media, a performant sampling is difficult. The chosen points on the light ray can be seen as VPLs, which therefore corresponds to a final gather pass over the light ray. Additionally, VRL is an unbiased method, which therefore allows progressive updates trivially. A comparison between VPL, PPB and VRL can be seen in Figure 3.10.


VPL PPB VRL

Figure 3.10: A comparison of VPL, PPB and VRL. All of these images were created within the same render time (600 seconds). Source: [NNDJ12b]

3.2.4 Virtual Beam Lights (VBL)

Although VRL already reduces the intensity of singularities significantly, a further reduction is needed for some cases. Especially on glossy surfaces the intensity of single light rays can stand out compared to others. Therefore, Novák et al. introduce the idea of Virtual Beam Lights [NNDJ12a]. Instead of using just the light ray, the ray is blurred into a beam. The properties within the beam stay consistent to the original ray.

This idea is similar to the change from VPL to VSL. Again, the radius of the beam corresponds to the density of the rays and once again, the radius converges to zero if an infinite amount of light rays is represented in the scene. Although bias is introduced, the progressiveness can be maintained by reducing the beam radius for different passes. Comparison images of VRL and VBL can be seen in Figure 3.11.

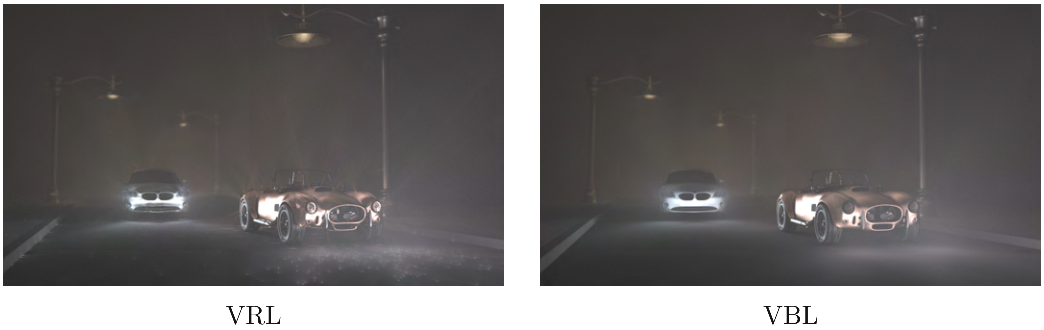


Figure 3.11: A comparison of VRL and VBL. Both of these images were created within the same render time (186 seconds for volume to volume events, 363 seconds for volume to surface events). Source: [NNDJ12a]

CHAPTER 4

Original Approach

To understand the new approaches that are discussed in the following chapter, a sound foundation of the original approach is needed. Virtual Ray Lights (VRL) [NNDJ12b] was developed by Novák et al. in 2012. It introduces Virtual Lights (VL) in the representation of rays to also gather the data along the sampled distance, not only at the intersection points.

4.1 Methodology

How the light data is represented is one of the main changes of the VRL algorithm compared to VPL. Figure 4.1 shows that not only the intersection points are now evaluated, but the whole ray is used for the evaluation. Therefore, the data density is increased marginally.

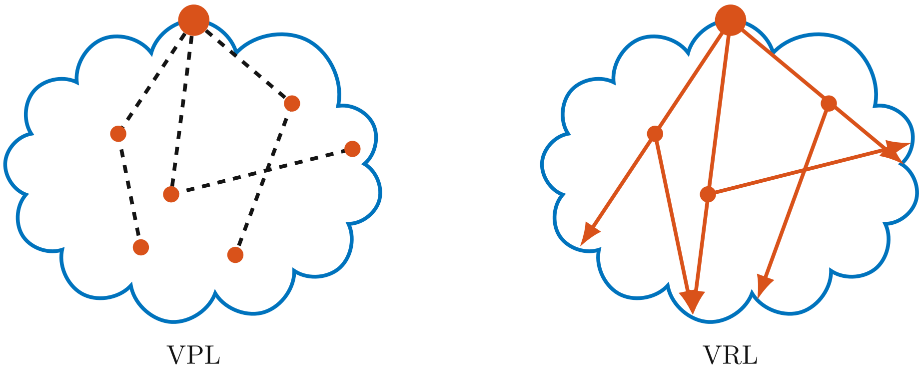


Figure 4.1: A visual comparison of the ideas behind VPL and VRL.

The incoming radiance at a point x in a direction $\vec{\omega}$ can be calculated with

$$L(x,\vec{\omega}) = T_r(s)L_s(x_s,\vec{\omega}) + L_m(x,\vec{\omega}), \tag{4.1}$$

Symbol	Description
v	a sampled point on the light ray
θ_v	the angle of the light ray and the direction from v to the surface point
Φ	the power of the light ray
$\sigma_s(v)$	the scattering coefficient at the position v
$f_s(\theta_v)$	the phase function of the angle θ_v
f_r	the BRDF at the surface point
$T_r(w_u(v))$	the transmission from the point v to the surface point
$T_r(v)$	the transmission from the start of the light ray to the point v
$V_u(v)$	the binary visibility from the point on the surface to the point v
$\frac{1}{w_u(v)^2}$	the inverse squared distance from the point v to the surface point

Table 4.1: The terms of the equation for the surface radiance.

where s is the distance to the nearest surface, x_s is the nearest surface location when going back along the ray at $x_s = x - s\vec{\omega}$, $T_r(s)$ is the transmission from x to x_s, L_s is the surface radiance (Equation 4.2) and L_m the radiance from the medium (Equation 4.3). For greater clarity, the notation of Novák et al. [NNDJ12b] was used in this equation. This equation is visualized in Figure 4.2, additionally, the properties of participating media are discussed in Section 2.3.

$$L_s \approx \Phi \int_0^t \frac{\sigma_s(v) f_s(\theta_v) f_r T_r(w_u(v)) T_r(v) V_u(v)}{w_u(v)^2} \, dv, \tag{4.2}$$

Equation 4.2 shows the evaluation of the surface radiance where the light ray is evaluated over its whole length, described by the integral $\int_0^t$. The variables of this equation can be seen in Table 4.1.

For the medium radiance

$$L_m \approx \Phi \int_0^s \int_0^t \frac{\sigma_s(u)\sigma_s(v) f_s(\theta_u) f_s(\theta_v) T_r(u) T_r(v) T_r(w) V}{w(u,v)^2} \, dv \, du, \tag{4.3}$$

not only the whole length of the light ray $\int_0^t$ has to be evaluated, but also the length of the camera ray $\int_0^s$. A visual representation of this equation can be seen in Figure 4.3, where additionally, h is the closest distance between both rays and the v_h and u_h are the closest points along each ray. The other terms of the equation and figure are described in Table 4.2.

4.2 Algorithm

The main part of the algorithm is relatively simple. First, light paths are traced from the light sources through the scene, where the light rays are the segments of the path. The number of light paths that are traced for each light source can be freely adjusted. For each camera path segment, all light rays are evaluated. Therefore, the evaluation

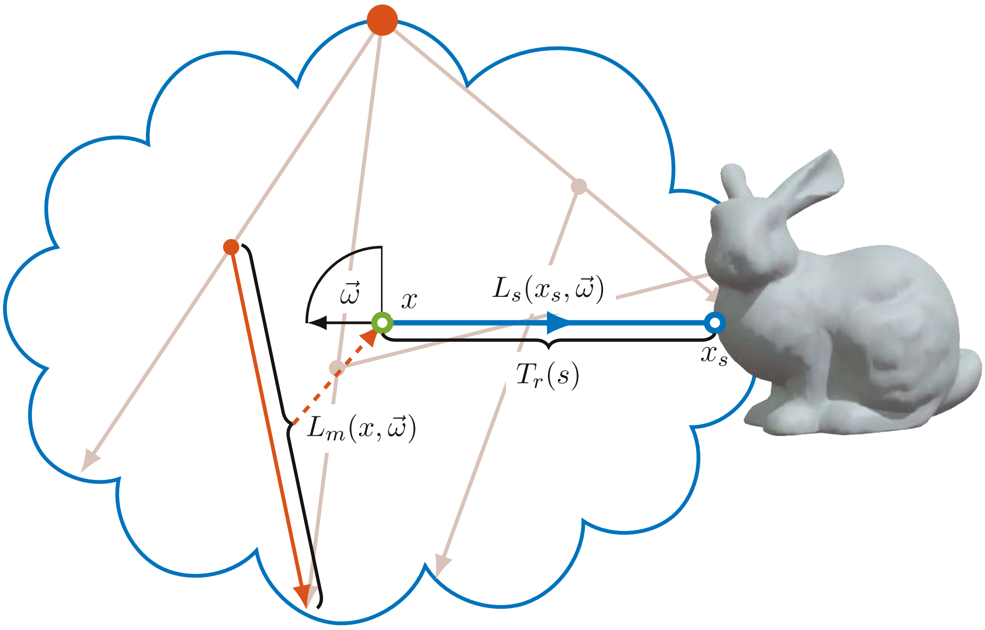


Figure 4.2: A visualization of the terms of Equation 4.1. The incoming radiance at the point x in a direction $\vec{\omega}$ is the sum of the radiance from the nearest surface $L_s(x_s, \vec{\omega})$ when going back along the ray and the radiance from the medium $L_m(x, \vec{\omega})$. The radiance from the surface is attenuated due to the transmittance $T_r(s)$. For the radiance from the medium, only the gathered radiance from one light path segment is visualized in this Figure, although all light path segments need to be evaluated.

Symbol	**Description**
v	a sampled point on the light ray
u	a sampled point on the camera ray
θ_v	the angle of the light ray and the direction from v to u
θ_u	the angle of the camera ray and the direction from v to u
Φ	the power of the light ray
$\sigma_s(u)$	the scattering coefficient at the position of the point u
$\sigma_s(v)$	the scattering coefficient at the position of the point v
$f_s(\theta_u)$	the phase function of the angle θ_u
$f_s(\theta_v)$	the phase function of the angle θ_v
$T_r(u)$	the transmission from the start of the camera ray to the point u
$T_r(v)$	the transmission from the start of the light ray to the point v
$T_r(w)$	the transmission between the point v to the point u
V	the binary visibility from the point v to the point u
$\frac{1}{w(u,v)^2}$	the inverse squared distance from the point v to the point u

Table 4.2: The terms of the equation for the radiance from the medium.

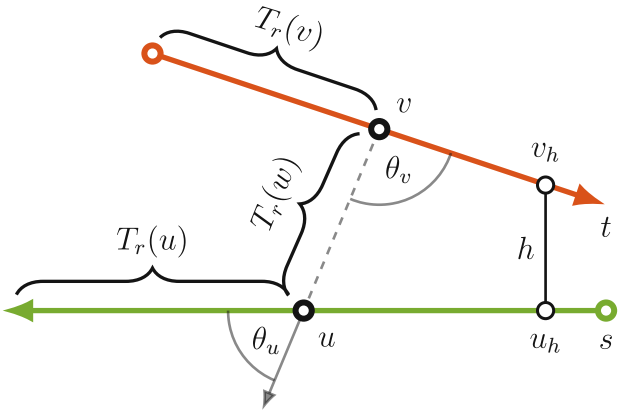


Figure 4.3: A visualization of the terms of Equation 4.3. The green ray is the camera ray that has the length s. The orange ray is the light ray with the length t. In this case, the contribution of the point light at the position v on the light ray is evaluated for the position u on the camera ray.

can always be done pair-wise at path segments from the light path and the camera path. The equations that were discussed previously can unfortunately not be solved in a closed form, therefore a Monte Carlo integrator is used.

The surface radiance (Equation 4.1) can be solved by only taking a single sample, on the light ray. For the medium radiance (Equation 4.2) an additional sample has to be taken on the camera ray. As the idea for the sampling of the surface radiance can be derived from the idea of the medium radiance, only the sampling approach of the medium radiance is discussed in detail.

To create a Monte Carlo estimator, we take the integrand

$$g(u,v) = \frac{\Phi\sigma_s(u)\sigma_s(v)f_s(\theta_u)f_s(\theta_v)T_r(u)T_r(v)T_r(w)V}{w(u,v)^2} \tag{4.4}$$

of Equation 4.3 and use it in the unbiased Monte Carlo estimator

$$L_m \approx \frac{1}{N}\sum_{i=1}^{N}\frac{g(u_i,v_i)}{\mathrm{pdf}(u_i,v_i)}. \tag{4.5}$$

In this estimator, N is the number of samples and $\mathrm{pdf}(u_i, v_i)$ the probability of choosing theses two samples.

4.3 Importance Sampling

Choosing the samples is one of the main difficulties of this approach. A naïve technique, like uniform sampling, would take a long time to produce acceptable results. Therefore, how the samples are chosen is of utmost importance. A case separation is made to account for differences of the medium. Section 4.3.1 describes the approach for isotropic media and 4.3.2 for anisotropic media.

4.3.1 Isotropic Sampling

In the isotropic case, the pdf

$$\mathrm{pdf}(u_i, v_i) \propto \frac{1}{w(u,v)^2} \tag{4.6}$$

has to be proportional to the inverse squared distance term as it causes the most variation.

Fortunately, there is already a sampling method in place that solves this kind of problem effectively. Kulla et al. [KF11] created the equi-angular sampling which is a simple, but very effective, method of sampling a ray corresponding to a point. The method works in the angular domain, by taking equal steps along the angle, therefore the name, equi-angular. Figure 4.4 shows a visualization of this method.

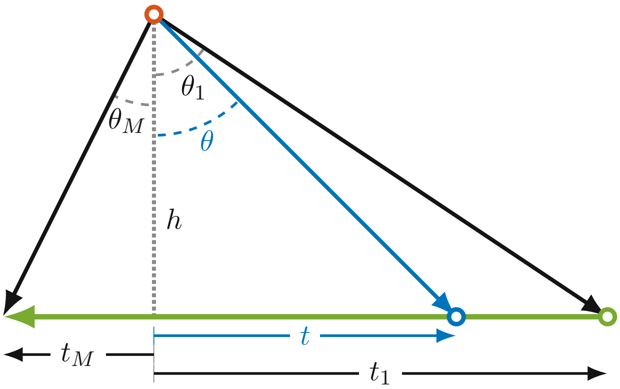


Figure 4.4: A visualization of the importance sampling by Kulla et al. [KF11]. The pdf of this technique is proportional to the inverse squared distance.

The first step of this method is to find the closest point on the ray to the point of interest. This point is used as the center of measurement, therefore the distances are measurements from this point to both ends of the ray. Therefore, the distances t_1 and t_M can now be negative.

To get the probability of choosing a distance t along the ray the pdf can be calculated with

$$\mathrm{pdf}(t) = \frac{h}{(\theta_M - \theta_1)(h^2 + t^2)}, \tag{4.7}$$

where h is the distance between the point of interest and the closest point on the ray, θ_1 is the angle to the start of the ray and θ_M the angle to the end of the ray.

The inverse transform sampling method

$$\mathrm{cdf}^{-1}(\xi) = h \tan\left((1-\xi)\theta_1 + \xi\theta_M\right) \tag{4.8}$$

is used to get a sampled distance along the ray, where ξ is a random number in the interval $[0,1)$ which will be remapped to the desired distribution $\mathrm{pdf}(t)$ with this function.

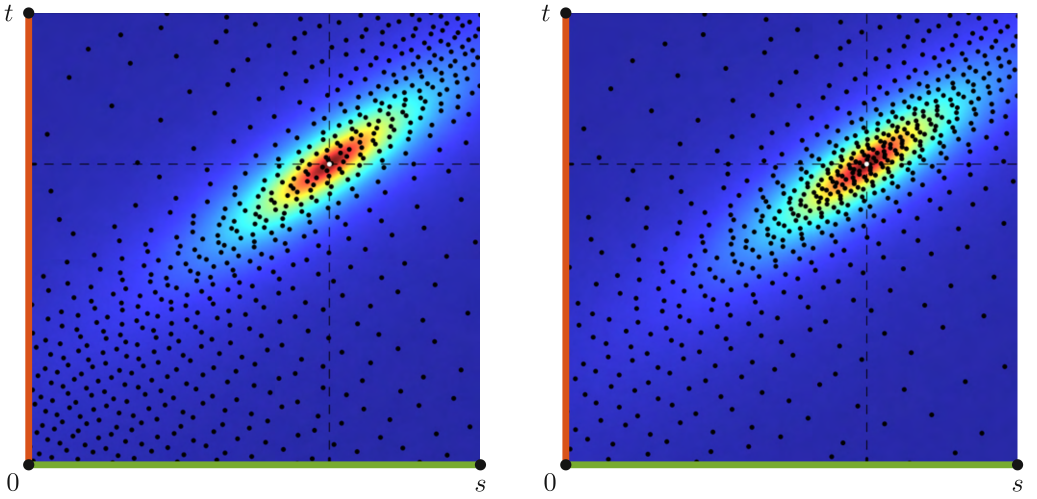
To convert the distance to an angle, the equation

$$\theta_x = \tan^{-1}\left(\frac{x}{h}\right), \tag{4.9}$$

is used. The inverse is

$$x = h\tan(\theta_x). \tag{4.10}$$

The approach that was just described can only sample from one ray and a specified point of interest. Novák et al. tried to sample the light ray uniformly first and then sample the camera ray with the described method and the light sample as a point of interest. As can be seen in Figure 4.5 (left), the distribution of these samples is already close, but still leaves room for improvement as the distribution is not ideally matched to the distance to the camera ray.



Uniform & Equi-Angular Sampling

Joint Distribution Sampling

Figure 4.5: A comparison between choosing a uniformly distributed sample on the light ray and an equi-angular sample on the camera ray (left), to choosing these samples with a joint distribution (right). Source: [NNDJ12b]

Therefore, a joint sampling distribution is created to also account for the distance to the camera ray. For this joint distribution, a marginal pdf samples the light ray first and is used within the conditional pdf for the camera ray. The variance of the sampling method is therefore greatly reduced.

The marginal pdf is first computed to get a light sample. For this, a few variables have to be computed first, u_h and v_h, the points with the closest distance to the other rays and h, the distance between those points. These variables are adapted similar as before in the equi-angular case, as they are now used as a center of computation. The distance $\hat{u}$ is calculated via $\hat{u} = u - u_h$ and $\hat{v}$ is calculated via $\hat{v} = v - v_h$.

The same adaptations have to be made on the light ray with the start and end point v_0 and v_1. The calculation of the marginal pdf can be done with

$$\mathrm{pdf}(\hat{v}, \hat{v}_0, \hat{v}_1) = \frac{\int_{\hat{u}_1}^{\hat{u}_0} w(\hat{u}, \hat{v}, h, \theta)^{-2} d\hat{u}}{\int_{\hat{v}_1}^{\hat{v}_0} \int_{\hat{u}_1}^{\hat{u}_0} w(\hat{u}, \hat{v}, h, \theta)^{-2}\, d\hat{u}\, d\hat{v}}, \tag{4.11}$$

where the denominator of this equation just serves for the normalization of the pdf. $w(\hat{u}, \hat{v}, h, \theta)^{-2}$ can be written as $h^2 + \hat{u}^2 + \hat{v}^2 - 2\hat{u}\hat{v}\cos\theta$ with the help of the law of cosines. $\cos\theta$ is the angle between the camera ray and the light ray. Currently, there does not exist an analytic solution for this equation, that means that simplifications must be made. Therefore, the camera ray is assumed to be infinite. With this the pdf can be calculated with

$$\mathrm{pdf}(\hat{v}, \hat{v}_0, \hat{v}_1) = \frac{\sin(\theta)}{\sqrt{h^2 + \hat{v}^2 \sin^2(\theta)}(A(\hat{v}_1) - A(\hat{v}_0))}, \tag{4.12}$$

where the term A is used as $A(x) = \sinh^{-1}\left(\frac{x}{h}\sin(\theta)\right)$. The cdf can be calculated with

$$\mathrm{cdf}(\hat{v}, \hat{v}_0, \hat{v}_1) = \frac{A(\hat{v}_0) - A(\hat{v})}{A(\hat{v}_0) - A(\hat{v}_1)}. \tag{4.13}$$

For the cdf^{-1}, the cdf has to be inverted

$$\mathrm{cdf}^{-1}(\xi, \hat{v}_0, \hat{v}_1) = \frac{h \sinh((1 - \xi)A(\hat{v}_0) + \xi A(\hat{v}_1))}{\sin(\theta)}. \tag{4.14}$$

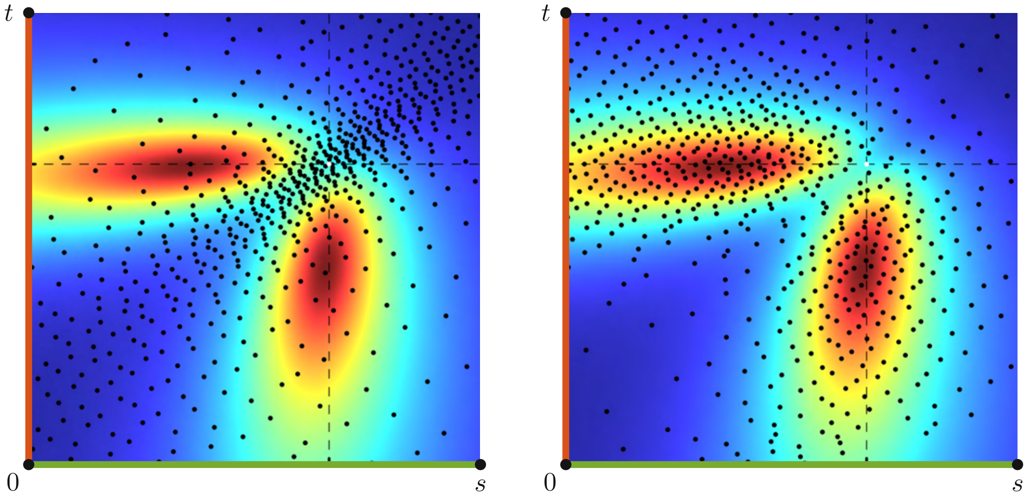
Equation 4.14 samples a point on the light ray with the help of a random number $\xi \in [0, 1)$ as input. The second step is now exactly as described in the beginning of this section. The equi-angular sampling method can be used with the just created sample as a point of interest. This corresponds to a VPL where a corresponding sample is created on the camera ray. This approach will be called the isotropic approach in the following sections.

4.3.2 Anisotropic Sampling

For anisotropic media, if we only account for the inverse squared distance, the results are not convincing. Specifically, with high anisotropy, the sampled results get further and further away from our target. Therefore, a new pdf was introduced. This pdf

$$\mathrm{pdf}(u_i, v_i) \propto \frac{f_s(u_i) f_s(v_i)}{w(u, v)^2} \tag{4.15}$$

also takes into account the phase functions f_s. As can be seen in Figure 4.6, the new pdf (right) represents our target distribution a lot better than the old, equi-angular approach



Equi-Angular Sampling of Camera Ray Piecewise-Linear Sampling of Camera Ray

Figure 4.6: A comparison between the isotropic approach (left), to the anisotropic approach (right). Source: [NNDJ12b]

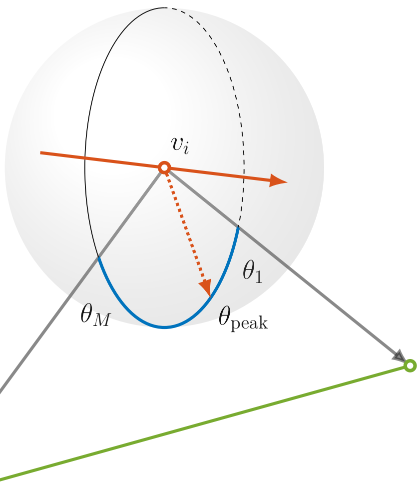
(left). The inverse squared distance remains an important factor and stays a part of this pdf.

To construct the new conditional pdf, the equi-angular sampling is advanced further. As this approach can be seen as uniformly distributing samples along the angle, the uniformity of this approach is now replaced with a distribution function that is fitted proportional to the product of the phase functions.

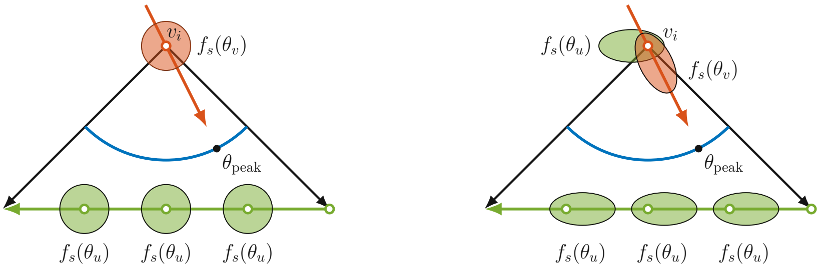
For a visual representation of this problem, Figure 4.7 (top) shows the camera ray (green) and a light ray (orange). They are encapsulated by a unit sphere to help showcase the construction of the arc. The light ray is projected onto this arc. The higher values will always occur either at the projection of the light ray, or the start or end points. It is therefore important to evaluate these areas with a higher importance.

The bottom left of Figure 4.7 shows the isotropic case. As the results of the phase function are constant at all points, they do not have to be considered. Compared to the anisotropic (bottom right), it can be seen that the complexity changes when the evaluation of the production of the phase functions has to be accounted for. This is the main reason why the isotropic approach does not lead to sufficient results in all cases.

The representation of the arc as an x-axis where the y-axis represents the result of the phase function can be seen in two cases in Figure 4.8. The green line represents the result of the phase function along the camera ray, the orange line along the light ray and the blue line is the product of both. The phase function along the camera ray will always be monotonic, it can only have its peak on either end of the spectrum. Differently, the phase function along the light ray can have the peak anywhere, it can even be outside



Angular Domain around v_i



Isotropic Case

Anisotropic Case

Figure 4.7: An illustration of the spherical domain around v_i as well as the arc that is generated by the projection of the camera ray (green) onto the sphere (top). Furthermore, the projection of the light ray into the angular domain can be seen in the same figure. The figures on the bottom show the isotropic (left) and anisotropic (right) cases. In the isotropic case, the product of the phase functions at θ_{peak} is equal to the result at all other angles.

of the spectrum. The product of the phase functions is our target, of which we want to have a pdf proportional to it. Novák et al. suggest taking a number of samples that are distributed around the areas of interest (the ends and the peak) more densely. The dotted line represents the result of this piece-wise reconstruction.

Distribution of Samples Along the Arc

As previously described, there are between two and three points that the distribution of the samples is focused on. First, we need to evaluate a few directions to make some case separations about the peak. The directions $\vec{a}$ and $\vec{b}$ are from the light sample to the start and the end of the camera ray, respectively. The plane that is constructed with these directions has the normal $\vec{c} = \text{norm}(\vec{a} \times \vec{b})$. The projected light direction can be

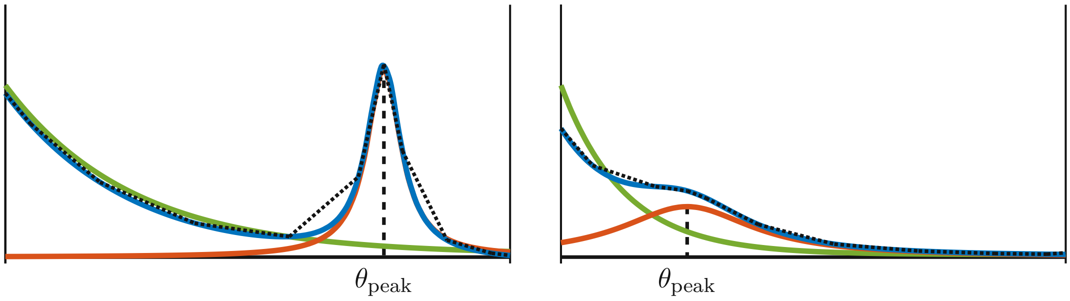


Figure 4.8: Two examples of the product of the phase functions (blue). The green line is the phase function along the camera ray and the orange line the phase function along the light ray.

calculated with

$$\vec{e} = \text{norm}((\vec{c} \times \vec{d}) \times \vec{c}). \tag{4.16}$$

The case separation can now be made by evaluating if $\vec{e}$ lies between $\vec{a}$ and $\vec{b}$. If the peak does not lie within these directions, the samples are distributed between the start and end point in cosine-warped uniform spacing.

If the peak does lie within these directions, the index of the sample of the peak has to be determined. If we would just put half of the samples on either side, the samples might be distributed unevenly (if the peak is close to the start or the end). The equation for the evaluation of this sample is

$$j_{\text{peak}} = \left\lfloor \frac{\theta_{\text{peak}} - \theta_1}{\theta_M - \theta_1} M - 0.5 \right\rfloor , \tag{4.17}$$

where the angles θ are to the corresponding points are used in the evaluation. The samples are then again distributed between the points in cosine-warped uniform spacing. Novák et al. suggest using 10 points for the reconstruction as this guarantees a relatively tight fight and is not getting too costly to compute at the same time.

Construction of the pdf for the Product of the Phase Functions

After distributing the samples, the first part is the evaluation of the samples at the specified positions. This has to be done to construct the piecewise linear function that is the reconstruction of the target function. For this reconstruction we first need the angle θ_u, which is the angle between the camera ray and the direction from the sampled position on the light ray to the sampled position along the arc. Similarly, we need the angle θ_v, which is the angle between the light ray and the direction from the sampled position on the light ray to the sampled position along the arc. With these two angles, we can calculate the phase function along the camera ray $f_s(\theta_u)$ and along the light ray $f_s(\theta_v)$. For further calculations we need the product of these phase functions, therefore we introduce a simplification in parameter. From here on, the product of the phase functions at an angle θ_j on the arc is written as $f_{\mathrm{uv}}(\theta_j)$.

The mathematical derivations that follow are not part of the description of the original paper from Novák et al. In their paper, they only describe the approach formally. Originally, they were part of the Bachelor Thesis of Michael Oppitz [Opp18] and are reformulated in this chapter.

The piecewise linear reconstruction is represented in the well-known slope-intercept form $y = kx + d$ used per segment. In this formulation, k is the slope of the line and d is the coordinate where the line crosses the y-axis. For our scenario, the x-axis is used for θ and the y-axis for the product of the phase functions $f_{\mathrm{uv}}(\theta)$.

The slope-value k_j for the segment j can be calculated with

$$k_j = \frac{f_{\mathrm{uv}}(\theta_{j+1}) - f_{\mathrm{uv}}(\theta_j)}{\theta_{j+1} - \theta_j}, \tag{4.18}$$

whereas the coordinate d_j at which the slope crosses the y-axis per segment j can be calculated with

$$d_j = f_{\mathrm{uv}}(\theta_{j+1}) - k_j\theta_{j+1} = f_{\mathrm{uv}}(\theta_j) - k_j\theta_j. \tag{4.19}$$

For every segment, θ_j is the start of the segment and θ_{j+1} the end of it.

The resulting piecewise linear function already creates our non-normalized pdf $\widetilde{\mathrm{pdf}}_{f_{uv}}(\theta)$

$$\widetilde{\mathrm{pdf}}_{f_{uv}}(\theta) = \begin{cases} k_1\theta + d_1, & \theta \in [\theta_1, \theta_2] \\ k_2\theta + d_2, & \theta \in (\theta_2, \theta_3] \\ \vdots & \\ k_N\theta + d_N, & \theta \in (\theta_N, \theta_M] \end{cases}. \tag{4.20}$$

The normalization of this function $\widetilde{\mathrm{pdf}}_{f_{\mathrm{uv}}}(\theta)$ can be achieved by dividing it through the corresponding cdf $\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta)$. $\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta)$ is a piecewise quadratic function that can be written as

$$\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta) = \begin{cases} \int_{\theta_1}^{\theta_2} k_1\theta + d_1\, d\theta, & \theta \in [\theta_1, \theta_2] \\ \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_2) + \int_{\theta_2}^{\theta_3} k_2\theta + d_2\, d\theta, & \theta \in (\theta_2, \theta_3] \\ \vdots & \vdots \\ \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_N) + \int_{\theta_N}^{\theta_M} k_N\theta + d_N\, d\theta, & \theta \in (\theta_N, \theta_M] \end{cases}. \tag{4.21}$$

The evaluation of the integral of a segment j can be done via

$$\int_{\theta_j}^{\theta_{j+1}} k_j\theta + d_j \, d\theta = \left(\frac{k_j}{2}\theta_{j+1}^2 + d_j\theta_{j+1}\right) - \left(\frac{k_j}{2}\theta_j^2 + d_j\theta_j\right). \tag{4.22}$$

The result of the cdf over the whole length of the area is now on simply called $\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}$. We therefore get the normalized pdf $\mathrm{pdf}_{f_{\mathrm{uv}}}(\theta)$

$$\mathrm{pdf}_{f_{\mathrm{uv}}}(\theta) = \begin{cases} \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}^{-1}(k_1\theta + d_1), & \theta \in [\theta_1, \theta_2] \\ \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}^{-1}(k_2\theta + d_2), & \theta \in (\theta_2, \theta_3] \\ \cdots\cdots\cdots\cdots & \cdots\cdots \\ \vdots & \vdots \\ \cdots\cdots\cdots\cdots & \cdots\cdots \\ \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}^{-1}(k_N\theta + d_N), & \theta \in (\theta_N, \theta_M] \end{cases}, \tag{4.23}$$

with the corresponding cdf

$$\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta) = \begin{cases} \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}^{-1} \int_{\theta_1}^{\theta_2} k_1\theta + d_1 \, d\theta, & \theta \in [\theta_1, \theta_2] \\ \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_2) + \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}^{-1} \int_{\theta_2}^{\theta_3} k_2\theta + d_2 \, d\theta, & \theta \in (\theta_2, \theta_3] \\ \cdots\cdots\cdots\cdots & \cdots\cdots \\ \vdots & \vdots \\ \cdots\cdots\cdots\cdots & \cdots\cdots \\ \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_N) + \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}^{-1} \int_{\theta_N}^{\theta_M} k_N\theta + d_N \, d\theta, & \theta \in (\theta_N, \theta_M] \end{cases}. \tag{4.24}$$

To create samples from the pdf, we have to invert the cdf. The cdf^{-1} can then be used for inverse transform sampling. Here, a special case has to be considered, as in cases where the slope of the segment is 0, the evaluation is different. The piecewise function for this evaluation is

$$
\mathrm{cdf}_{f_{\mathrm{uv}}}^{-1}(\xi) =
\begin{cases}
\left(\theta_1 d_1 + \xi\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}} - \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_1)\right) d_1^{-1}, \\
\qquad k_1 = 0,\ \xi \in [\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_1), \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_2)] \\
\left(\sqrt{k_1(\theta_1^2 k_1 + 2\xi\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}} - 2\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_1)) + 2\theta_1 d_1 k_1 + d_1^2} - d_1\right) k_1^{-1}, \\
\qquad k_1 \neq 0,\ \xi \in [\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_1), \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_2)] \\
\left(\theta_2 d_2 + \xi\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}} - \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_2)\right) d_2^{-1}, \\
\qquad k_2 = 0,\ \xi \in [\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_2), \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_3)] \\
\left(\sqrt{k_2(\theta_2^2 k_2 + 2\xi\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}} - 2\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_2)) + 2\theta_2 d_2 k_2 + d_2^2} - d_2\right) k_2^{-1}, \\
\qquad k_2 \neq 0,\ \xi \in [\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_2), \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_3)] \\
\qquad\qquad \vdots \\
\left(\theta_N d_N + \xi\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}} - \widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_N)\right) d_N^{-1}, \\
\qquad k_N = 0,\ \xi \in [\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_N), \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_M)] \\
\left(\sqrt{k_N(\theta_N^2 k_N + 2\xi\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}} - 2\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_N)) + 2\theta_N d_N k_N + d_N^2} - d_N\right) k_N^{-1}, \\
\qquad k_N \neq 0,\ \xi \in [\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_N), \mathrm{cdf}_{f_{\mathrm{uv}}}(\theta_M)]
\end{cases}
, \tag{4.25}
$$

where a random value in the interval $[0, 1)$ creates an angle θ along the arc with a distribution according to $\mathrm{pdf}_{f_{\mathrm{uv}}}(\theta)$. We currently have only used a sampling methodology that is proportional to the product of the phase functions. In the next step we also include the inverse squared distance.

Creation of the final pdf

To create the final pdf, the pdf of the previous section has to be combined with the isotropic approach. As was explained in that chapter, the equi-angular approach is used there. The combination of both approaches is the not normalized pdf

$$\widetilde{\mathrm{pdf}}(\theta) = \mathrm{pdf}_{f_{\mathrm{uv}}}(\theta)\,\mathrm{pdf}_{w^{-2}}(\theta). \tag{4.26}$$

With the corresponding $\widetilde{\mathrm{cdf}}(\theta)$

$$\widetilde{\mathrm{cdf}}(\theta) = \frac{\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta)}{\widetilde{\mathrm{cdf}}_{f_{\mathrm{uv}}}(\theta_M - \theta_1)} = \frac{\mathrm{cdf}_{f_{\mathrm{uv}}}(\theta)}{\theta_M - \theta_1} \tag{4.27}$$

that is used to divide it. The final pdf can therefore be written as

$$\mathrm{pdf}(\theta) = \frac{\widetilde{\mathrm{pdf}}_{f_{\mathrm{uv}}}(\theta)h}{h^2 + t^2}. \tag{4.28}$$

Fortunately, in the normalization of this pdf, various terms drop out to make the end step relatively easy. The resulting distribution of the samples can be seen in Figure 4.6 (right), compared to the distribution in the isotropic case Figure 4.6 (left).

CHAPTER 5

New Approaches

In our evaluation of VRL we found that in the anisotropic case, the construction of the piecewise linear function is not only a complication to understanding the sampling methodology, but also a performance problem. Therefore, we evaluated different techniques of replacing this piecewise linear function.

As the piecewise linear function approximates the product of two phase functions, the obvious solution would have been to just use the product of the phase functions for sampling. Unfortunately, this product cannot be integrated in a closed form and can therefore not be used in the Monte Carlo sampling approach.

Furthermore, the same complications arose when trying to use the product of two general probability functions. We evaluated this for a multitude of probability functions, but for none of them the product of them was integrable.

Therefore, we went on from the multiplication, and evaluated that the sum of two probability distributions can result in a close approximation of our target functions if the probability functions are chosen and adjusted accordingly. The sum of two probability functions can be used as an approximation of the product of the phase functions in this case as both, the sum and the product, need to be normalized. Only due to the normalization are the results close enough for a successful approximation. An interesting case is the Cauchy Distribution, as we will explain in Section 5.1, as it can be used in the Monte Carlo sampling approach.

For other probability functions, the sum of two of these sampling functions cannot be used in the Monte Carlo approach. Luckily, we can use the Mixture Model approach to solve this issue. In this approach, we provide multiple probability distributions and select one of them randomly in each evaluation. The results of the Mixture Model depend on how well the probability functions are fitted to our target, but overall, the results are fast and reliable. This approach is discussed in Section 5.2.

In Appendix B, we will briefly describe all other variations that we tried, even if they did not provide the desired results or our mathematical analysis failed.

5.1 Sum of two Cauchy Distributions

5.1.1 Description

In this first solution we try to approximate the product of the two phase functions via the sum of two Cauchy distributions. Although, there is obviously a difference between the sum of two distributions and the product of them, we find that with a correctional value, the result is very close in all cases.

We first fit a Cauchy distribution to the phase function along the camera ray $f_s(\theta_u)$ and the phase function along the light ray $f_s(\theta_v)$, respectively. Afterwards we create the sum of these two distributions and invert the corresponding cdf to be able to sample it. The sampling then takes place in the angular range of θ_o and θ_e with the distribution of Kulla et al. Therefore, some normalization terms have to be accounted for.

5.1.2 Theory

The Cauchy distribution is defined as

$$f(x) = \frac{1}{\pi s(1 + (\frac{x-m}{s})^2)}, \tag{5.1}$$

where m is the location and s is the scale. To set the location for one of the phase functions, we simply set it to the position of its peak. The scale is set to match the value at the peak. This can be done with the derived formula

$$s = \frac{1}{\pi \max}, \tag{5.2}$$

where max is the value at the peak.

With this setup we can already match the Cauchy distribution to the two phase functions. As can be seen in Figure 5.1 the matching of the Cauchy distributions (dashed lines) to the target distributions (solid lines) works well. As the Cauchy distribution does not work in the angular domain, some differences can be seen, especially for lower anisotropy values.

We now make a change in parameters, so that we get the two pdfs that approximate the phase functions on the camera ray $f_s(\theta_u)$ and the light ray $f_s(\theta_v)$.

$$\mathrm{pdf}_u(x) = \frac{1}{\pi s_u(1 + (\frac{x-m_u}{s_u})^2)}$$
$$\mathrm{pdf}_v(x) = \frac{1}{\pi s_v(1 + (\frac{x-m_v}{s_v})^2)}$$

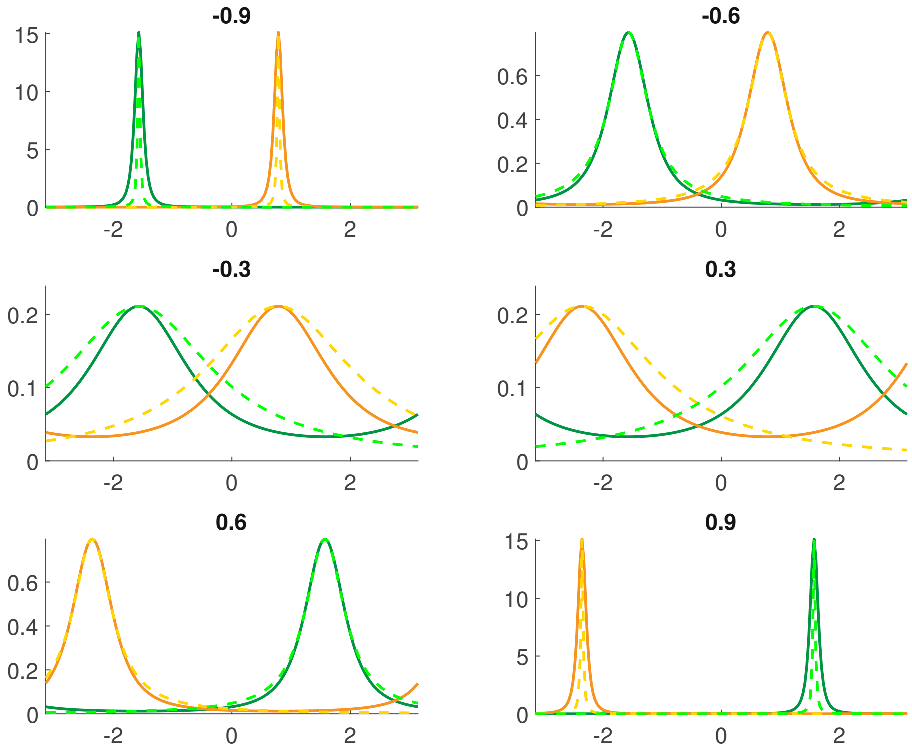


Figure 5.1: The matched Cauchy distributions for the light (dashed yellow) and camera (dashed green) compared to the Henyey-Greenstein phase function for the light (solid yellow) and camera (solid green) for different anisotropy values of the Henyey-Greenstein phase function.

Which have the corresponding cdfs

$$\begin{aligned}\mathrm{cdf}_u(x) &= \frac{1}{\pi}\arctan\left(\frac{x-m_u}{s_u}\right)+\frac{1}{2}\\ \mathrm{cdf}_v(x) &= \frac{1}{\pi}\arctan\left(\frac{x-m_v}{s_v}\right)+\frac{1}{2}\end{aligned}$$

.

The pdf of the sum with equal weight is therefore simply

$$\mathrm{pdf}(x) = \left(\frac{1}{2}\frac{1}{\pi s_u(1+(\frac{x-m_u}{s_u})^2)}+\frac{1}{2}\frac{1}{\pi s_v(1+(\frac{x-m_v}{s_v})^2)}\right) \tag{5.3}$$

where as the cdf is

$$\mathrm{cdf}(x) = \frac{1}{2}\left(\frac{1}{\pi}\arctan\left(\frac{x-m_u}{s_u}\right)+\frac{1}{2}\right)+\frac{1}{2}\left(\frac{1}{\pi}\arctan\left(\frac{x-m_v}{s_v}\right)+\frac{1}{2}\right) \tag{5.4}$$

.

The pdf is currently only normalized to the target pdf over the interval $(-\infty, \infty)$, but we only need it in the interval $[\theta_o, \theta_e]$. For the pdf this problem can be solved by dividing the result through $\text{cdf}(\theta_e) - \text{cdf}(\theta_o)$.

The normalized pdf compared to the normalized target functions can be seen in Figure 5.2. The dashed lines represent our approximation whereas the solid lines represent the target. As can be seen from this example, our approach works very well in cases of high anisotropy. For lower anisotropy, our approach is not as close anymore, but as the target functions are more evenly partitioned, the values are still close.

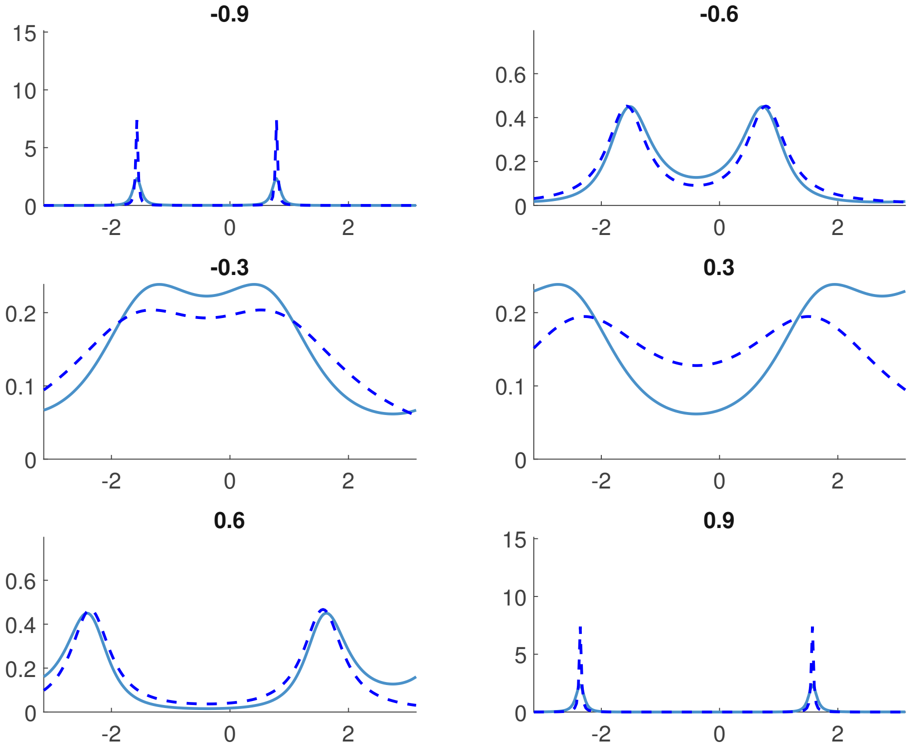


Figure 5.2: The pdf of our reconstruction (dashed blue) compared to the actual normalized target pdf (solid blue) for different anisotropy values of the Henyey-Greenstein phase function.

For the inversion of the cdf we apply the normalization to the random value $\epsilon \in [0, 1)$ beforehand via

$$u = \epsilon * (\text{cdf}(\theta_e) - \text{cdf}(\theta_o)) + \text{cdf}(\theta_o). \tag{5.5}$$

The addition of $\text{cdf}(\theta_o)$ is necessary to not start at 0 automatically.

With this we can start the inversion process.

$$
\begin{aligned}
\frac{1}{2}\left(\frac{1}{\pi}\arctan\left(\frac{x-m_u}{s_u}\right)+\frac{1}{2}\right)+\frac{1}{2}\left(\frac{1}{\pi}\arctan\left(\frac{x-m_v}{s_v}\right)+\frac{1}{2}\right) &= u\\
\arctan\left(\frac{x-m_u}{s_u}\right)+\arctan\left(\frac{x-m_v}{s_v}\right) &= \pi(2u-1)\\
\arctan\left(\frac{\frac{x-m_u}{s_u}+\frac{x-m_v}{s_v}}{1-\frac{x-m_u}{s_u}\frac{x-m_v}{s_v}}\right) &= \pi(2u-1)
\end{aligned}
$$

Note that in the last equation, the combination of the two arctan can actually result in the by-product of $\pm\pi$. As in the next step, this part falls under a tan it was left out, as tan is π-cyclic and the result is therefore the same in all cases.

$$
\frac{\frac{(x-m_u)s_v+(x-m_v)s_u}{s_us_v}}{1-\frac{(x-m_u)(x-m_v)}{s_us_v}} = \tan(\pi(2u-1))
$$

To keep things readable we substitute $z=\tan(\pi(2u-1))$.

$$
\begin{aligned}
\frac{x(s_u+s_v)-s_um_v-s_vm_u}{s_us_v-(x-m_u)(x-m_v)} &= z\\
x(s_u+s_v)-s_um_v-s_vm_u-z(s_us_v-(x-m_u)(x-m_v)) &= 0\\
x(s_u+s_v)-s_um_v-s_vm_u-z(s_us_v-x^2+x(m_u+m_v)-m_um_v) &= 0\\
x^2+x\frac{s_u+s_v-m_uz-m_vz}{z}+\frac{-s_um_v-s_vm_u-s_us_vz+m_um_vz}{z} &= 0
\end{aligned}
$$

As we can now plug this into the reduced quadratic equation formula to get a result for x, we get

$$
\begin{aligned}
p &= \frac{s_u+s_v-m_uz-m_vz}{z}\\
q &= \frac{-s_um_v-s_vm_u-s_us_vz+m_um_vz}{z}
\end{aligned}
$$

for the coefficients. We evaluated that for $u<\frac{1}{2}$ the square-root term has to be subtracted and for $u>\frac{1}{2}$ the term has to be added to get the correct results. In the special case where $u=\frac{1}{2}$ the solution simplifies to $x=\frac{s_um_v+s_vm_u}{s_u+s_v}$ as in this case z is 0.

5.2 Mixture Model

5.2.1 Description

For this solution, we use the well-known Mixture Model approach to get an approximation of the product of the two phase functions. The mixture model chooses one of the probability functions randomly for each sampling step. as we already used the Cauchy distribution in the previous approach, we will also use it here. Other distributions can be used as well, as can be seen in the Appendix.

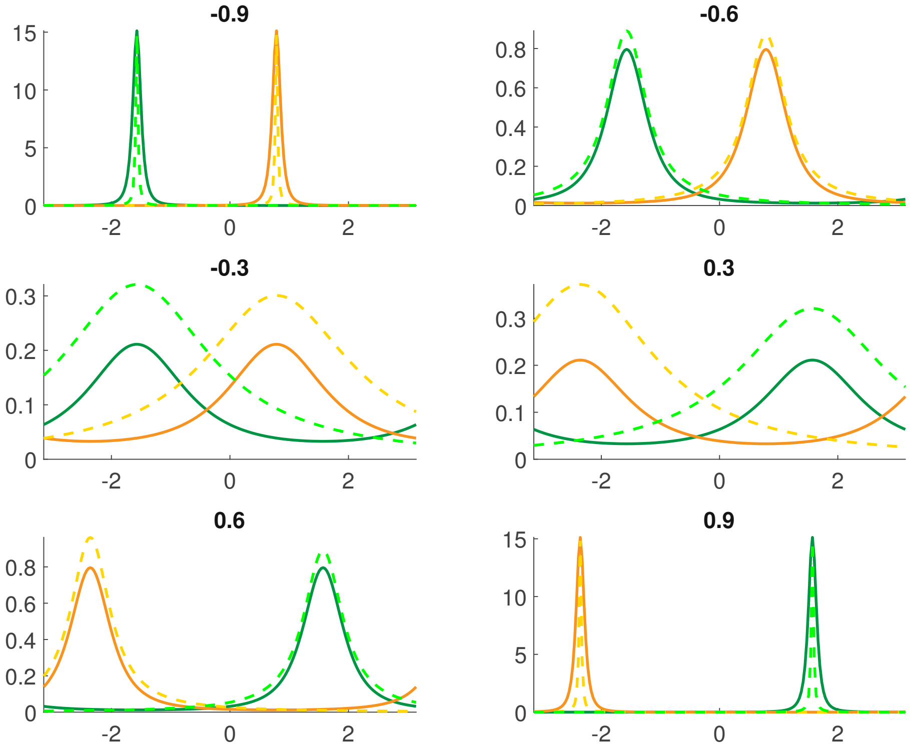


Figure 5.3: The matched and normalized Cauchy distributions for the light (dashed yellow) and camera (dashed green) compared to the Henyey-Greenstein phase function for the light (solid yellow) and camera (solid green) for different anisotropy values of the Henyey-Greenstein phase function. Compared to Figure 5.1 the Cauchy distributions are already normalized as this is a requirement for their use in the mixture model.

5.2.2 Theory

The matching of the Cauchy distribution to our target is done exactly in the same way as the previous approach.Equivalently to the previous approach, the pdfs and cdfs are created, this time they have to be normalized beforehand to work in the mixture model. Therefore Figure 5.3 shows our normalized pdfs (dashed lines) compared to the normalized target (solid lines). The general pdf and cdf are therefore already normalized. When it comes to the actual sampling process, we take a random value $\epsilon \in [0, 1)$ and select one of the two distributions depending on it. Compared to the previous solution of Section 5.1, the difference comes with the time of normalization. In the mixture model approach, the two Cauchy distributions have to be normalized when using them, in the solution of Section 5.1, they do not have to be, as the sum of the two Cauchy distributions is normalized in a later step. This leads to slightly different results.

As the pdf and cdf of the single pdfs are only normalized over the interval $(-\pi, \pi)$, we have to make some adjustments. In this case that means adjusting the random value with which we sample beforehand again, but in this case tailored to the single distributions. For the camera ray that means

$$u_u = \epsilon * (\mathrm{cdf}_u(\theta_e) - \mathrm{cdf}_u(\theta_o)) + \mathrm{cdf}_u(\theta_o) \tag{5.6}$$

as well as for the light ray

$$u_v = \epsilon * (\mathrm{cdf}_v(\theta_e) - \mathrm{cdf}_v(\theta_o)) + \mathrm{cdf}_v(\theta_o) \tag{5.7}$$

.

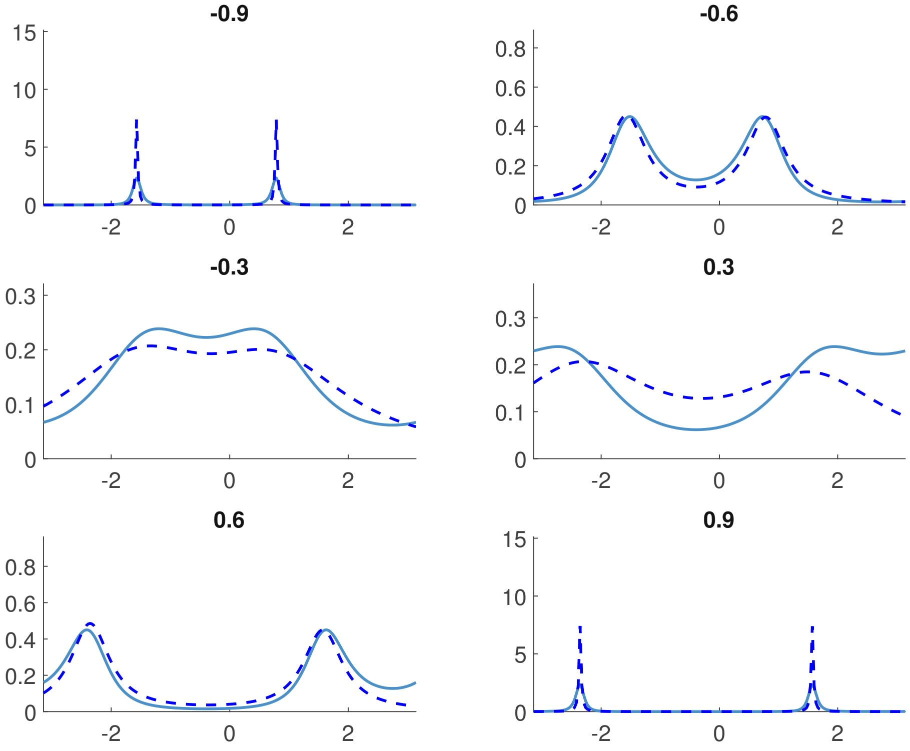


Figure 5.4: The pdf of our reconstruction (dashed blue) compared to the actual normalized target pdf (solid blue) for different anisotropy values of the Henyey-Greenstein phase function.

With this we can sample the inverse Cauchy distribution and evaluate the result with the general pdf of the mixture model. The pdf of this approach can be seen in Figure 5.4.

CHAPTER 6

Implementation

In this chapter, we will first show implementation details on the isotropic and anisotropic sampling functions as they were described in VRL (Section 6.1 and 6.2).

In Section 6.3 we will describe the implementation of the sampling with the sum of two Cauchy distributions, whereas in Section 6.4 we will explain our mixture model approach for this problem.

6.1 Original Approach - Isotropic Sampling

Algorithm 6.1: ISOTROPICSAMPLING($\bar{s}, p, \mathrm{cp}$)

```
$\mathrm{dist}_s = (\bar{s}.o - \mathrm{cp}).\mathrm{length}$
$\mathrm{dist}_{\mathrm{cp}} = (\mathrm{cp} - p).\mathrm{length}$

$\mathrm{start} = \tan^{-1}(-\mathrm{dist}_s \,/\, \mathrm{dist}_{\mathrm{cp}})$
$\mathrm{end} = \tan^{-1}((\bar{s}.\mathrm{dist} - \mathrm{dist}_s) / \mathrm{dist}_{\mathrm{cp}})$
$\mathrm{dist} = \mathrm{dist}_{\mathrm{cp}} \cdot \tan((1 - \xi) \cdot \mathrm{start} + \xi \cdot \mathrm{end})$

$\mathrm{sample} = \mathrm{cp} + \bar{s}.\vec{d} \cdot \mathrm{dist}$
$\mathrm{pdf} = \mathrm{dist}_{\mathrm{cp}} / ((\mathrm{end} - \mathrm{start}) \cdot (\mathrm{dist}_{\mathrm{cp}}^2 + \mathrm{dist}^2))$
return [sample, pdf]
```

In the isotropic case, we use the sampling function described by Kulla et al. [KF11]. This function is used in VRL as the simple version of the algorithm that is then expanded for anisotropic cases. We call this distribution ISOTROPICSAMPLING and show it in Algorithm 6.1. As an input to this function, a segment (either camera or light segment, depending on the use case) $\bar{s}$ has to be provided. A sample will be chosen on this segment. Additionally, the point of interest p, and the closest point cp to the point of interest have to be provided.

First, a change in variables takes place. We calculate $dist_s$ and $dist_{cp}$, the distance from the origin $\bar{s}.o$ to the closest point cp and the distance from the closest point cp to the point of interest p, respectively. The start and end variables that are computed next are the angles from the point of interest to the start and end of the segment. In Line 5 of Algorithm 6.1 the cdf^{-1} (Equation 4.8) is already used to create the distance $dist$ from the closest point cp to our newly created sample sample. With this distance, we can compute the sample and the pdf.

6.2 Original Approach - Anisotropic Sampling

The approach to solve anisotropic sampling expands the previously explained isotropic approach. It still works within the angular domain, but as the result of the product of the two phase functions is now not a constant anymore, a piecewise linear function is constructed for sampling.

Additionally to the segment $\bar{s}$, the point of interest p and the closest point cp, the direction $\vec{d}$ of the point of interest has to be provided. Again, we create the variables $dist_s$ and $dist_{cp}$ as in the isotropic case. Furthermore, we calculate the direction $\vec{a}$ from the point of interest to the start of the segment, the direction $\vec{b}$ from the point of interest to the end of the segment and the direction $\vec{c}$ which is the cross product of the two directions. The variables θ_1 and θ_M represent the angles to the start of the segment and the end of it.

With the created variables we can start to distribute the samples along the arc. The methodology to create the piecewise linear function is explained later on in the CreateSamplesOnArc-function. Furthermore, the next function ConstructPdfCdf, which calculates the non-normalized pdf and the corresponding cdf, will also be explained later on.

The data structure for the piecewise linear function can be seen in Table 6.1. We now evaluate in a loop for which piece of the function we sample from. Therefore, we adjust our random value $\xi \in [0,1)$ by our non-normalized cdf. For each segment, we evaluate if the adjusted random value is part of the linear function area.

If the linear segment is found, there are two cases to evaluate. In the first case, the linear function is constant. This will always be the case for isotropic media, but can also happen for anisotropic media. In that case, the calculation of the angle to the sample θ_s and the pdf of the phase functions $\mathrm{pdf}_{f_{\mathrm{uv}}}$ is straightforward. If the linear segment is not constant, the complexity increases. We follow the formula of Algorithm 6.3 in that case.

After we have found the angle to the sample θ_s and the pdf of the phase functions $pdf_{f_{\mathrm{uv}}}$ we have to calculate the actual sample and the final pdf pdf in the last steps.

Algorithm 6.2: ANISOTROPICSAMPLING($\bar{s}, p, \vec{d}$, cp)

```
$\text{dist}_s = (\bar{s}_c.o - \text{cp}).\text{length}$
$\text{dist}_{\text{cp}} = (\text{cp} - p).\text{length}$
$\vec{a} = \bar{s}.o - p$
$\vec{b} = \bar{s}.e - p$
$\vec{c} = \vec{a} \times \vec{b}$
$\theta_1 = \tan^{-1}(-\text{dist}_s / \text{dist}_{\text{cp}})$
$\theta_M = \tan^{-1}((\bar{s}.\text{dist} - \text{dist}_s) / \text{dist}_{\text{cp}})$
CreateSamplesOnArc($\theta_1, \theta_M, \vec{a}, \vec{b}, \vec{c}, \vec{d}, \bar{s}.\vec{d}$)
$\text{cdf} =$ ConstructPdfCdf()
$\hat{\xi} = \xi \cdot \text{cdf}$
for all piecewise segments do
    linear = the current piecewise linear function
    if $\text{cdf}_{\text{comb}} < \hat{\xi}$ and $\hat{\xi} < \text{cdf}_{\text{comb}} + \text{linear.cdf}$ then
        if $\text{linear}.k = 0$ then
            $\theta_s = (\text{linear}.\theta_o \cdot \text{linear}.d + \hat{\xi} - \text{cdf}_{\text{comb}}) / \text{linear}.d$
            $\text{pdf}_{f_{\text{uv}}} = \text{linear}.d / \text{cdf}$
        else
            $\text{temp} = \text{linear}.k \cdot (\text{linear}.\theta_o^2 \cdot \text{linear}.k + 2 \cdot \hat{\xi} - 2 \cdot \text{cdf}_{\text{comb}}) +$
                $2 \cdot \text{linear}.\theta_o \cdot \text{linear}.d \cdot \text{linear}.k + \text{linear}.d^2$
            $\theta_s = (\text{sqrt}(\text{temp}) - \text{linear}.d) / \text{linear}.k$
            $\text{pdf}_{f_{\text{uv}}} = (\text{linear}.k \cdot \theta_s + \text{linear}.d) / \text{cdf}$
        end
    end
    $\text{cdf}_{\text{comb}} = \text{cdf}_{\text{comb}} + \text{linear.cdf}$
end
$\text{dist} = \text{dist}_{\text{cp}} \cdot \tan(\theta_s)$
$\text{sample} = \text{cp} + \bar{s}.\vec{d} \cdot \text{dist}$
$\text{pdf} = \text{pdf}_{f_{\text{uv}}} \cdot (\text{dist}_{\text{cp}} / (\text{dist}_{\text{cp}}^2 + \text{dist}^2))$
return [sample, pdf]
```

Name	Description
θ_o, θ_e	the interval boundaries
f_{s_o}, f_{s_e}	the product of the phase functions at the interval boundaries
k	the slope of the function
d	the y-intercept of the function
cdf	the computed definite integral of this function

Table 6.1: The structure of the LINEARFUNCTION data type.

Algorithm 6.3: CREATESAMPLESONARC($\theta_1, \theta_M, \vec{a}, \vec{b}, \vec{c}, \vec{d_1}, \vec{d_2}$)

```
1  $\vec{e} = (\vec{c} \times \vec{d_2}) \times \vec{c}$
2  $\theta_{\text{peak}} = \cos^{-1}(\vec{a} \cdot \vec{e})$
3  if $\theta_{\text{peak}} \notin [\theta_1, \theta_M]$ then
4      $\vec{e} = -\vec{e}$
5      $\theta_{\text{peak}} = \cos^{-1}(\vec{a} \cdot \vec{e})$
6  end
7  if $\theta_{\text{peak}} \in [\theta_1, \theta_M]$ then
8      $j_{\text{peak}} = \text{floor}((\theta_{\text{peak}} - \theta_1)/(\theta_M - \theta_1) \cdot M - 0.5)$
9      for $j < j_{\text{peak}}$ do
10         $\theta_j = (\theta_{\text{peak}} - \theta_1)/2 \cdot (1 - \cos((\pi \cdot (j-1))/(j_{\text{peak}} - 1)))$
11         $f_{\text{uv}_j} = \text{PHASEFUNCTIONHG}(\vec{d_1}, \theta_j) \cdot \text{PHASEFUNCTIONHG}(\vec{d_2}, \theta_j)$
12         SAVEARCSAMPLE($\theta_j, f_{\text{uv}_j}$)
13     end
14     for $j > j_{\text{peak}}$ and $j < M$ do
15         $\theta_j = (\theta_M - \theta_{\text{peak}})/2 \cdot (1 - \cos((\pi \cdot (j - j_{\text{peak}}))/(M - j_{\text{peak}})))$
16         $f_{\text{uv}_j} = \text{PHASEFUNCTIONHG}(\vec{d_1}, \theta_j) \cdot \text{PHASEFUNCTIONHG}(\vec{d_2}, \theta_j)$
17         SAVEARCSAMPLE($\theta_j, f_{\text{uv}_j}$)
18     end
19 else
20     for $j < M$ do
21         $\theta_j = (\theta_M - \theta_1)/2 \cdot (1 - \cos((\pi \cdot (j-1))/M))$
22         $f_{\text{uv}_j} = \text{PHASEFUNCTIONHG}(\vec{d_1}, \theta_j) \cdot \text{PHASEFUNCTIONHG}(\vec{d_2}, \theta_j)$
23         SAVEARCSAMPLE($\theta_j, f_{\text{uv}_j}$)
24     end
25 end
```

6.2.1 The CreateSamplesOnArc-Function

The CREATESAMPLESONARC-function describes how to distribute the piecewise linear function segments along the arc. There are two cases in this function, either the peak θ_{peak} lies within the interval $[\theta_1, \theta_M]$ or it does not. The peak θ_{peak} is the maximum of the product of the two phase functions. In addition to evaluating if the peak is in the interval, if it is not, we also have to evaluate if the negative peak is within the interval. This happens in the first six lines.

In our first case, the peak θ_{peak} or its negative lies within the interval $[\theta_1, \theta_M]$. Therefore, we distribute the samples in a cosine-warped distribution in the intervals $[\theta_1, \theta_{\text{peak}}]$ and $[\theta_{\text{peak}}, \theta_M]$. The number of samples in each interval is determined by the position of the peak, therefore the index j_{peak} is used to separate the samples in two sets. The second case distributes all samples within the interval $[\theta_1, \theta_M]$.

The construction of the piecewise linear segments follows a relatively simple process. First, we create the angle θ_j per segment, which marks the start of the segment. Then, we create the product of the phase functions at that point.

6.2.2 The ConstructPdfCdf-Function

Algorithm 6.4: ConstructPdfCdf()

```
for $j < M$ do
    $k = (f_{\mathrm{uv}_{j+1}} - f_{\mathrm{uv}_j})/(\theta_{j+1} - \theta_j)$
    $d = f_{\mathrm{uv}_{j+1}} - k \cdot \theta_{j+1}$
    $\mathrm{end} = (k \cdot \theta_{j+1}^2)/2 + d \cdot \theta_{j+1}$
    $\mathrm{start} = (k \cdot \theta_j^2)/2 + d \cdot \theta_j$
    $\mathrm{cdf} = \mathrm{end} - \mathrm{start}$
    SaveLinearFunction($\theta_j, \theta_{j+1}, f_{\mathrm{uv}_j}, f_{\mathrm{uv}_{j+1}}, k, d, \mathrm{cdf}$)
    $\mathrm{cdf}_{\mathrm{comb}} = \mathrm{cdf}_{\mathrm{comb}} + \mathrm{cdf}$
end
return $\mathrm{cdf}_{\mathrm{comb}}$
```

To create the actual linear functions, ConstructPdfCdf of Algorithm 6.4 is used. Here, the sampled data is transformed into linear functions. Additionally, the definite integral per segment is computed and the sum for all segments is returned.

6.3 Sum of Cauchy Distributions

For the sum of two Cauchy distributions, the creation of our initial variables is identical to the anisotropic approach that was described by Novák et al [NNDJ12b]. We create the variables $dist_s$ and $dist_{cp}$, the distance from the origin of the ray $\bar{s}$ to the closest point cp and the distance from closest point cp to the point of interest p, respectively. The vectors $\vec{a}$, $\vec{c}$ and $\vec{e}$ point from the point of interest p to the start of $\bar{s}$, the end of $\bar{s}$ and depict the plane direction spanned by these vectors. The values of θ_o and θ_e are once again the angles to the origin and to the end of the ray.

From this point on, the algorithm has only very little to do with its predecessor. As the next step we compute the values for the maxima, the scale and the location for both Cauchy distributions that approximate the Henyey-Greenstein phase functions. The maximum for the light max_{light} uses the Henyey-Greenstein phase function with the dot product of the direction of the light $\vec{d}$ and the plane normal $\vec{e}$. This maximum is used to compute the light scale s_{light}. The location m_{light} depends on the anisotropy and the dot product of the plane normal $\vec{e}$ and the direction between the closest point cp and the point of interest p. For the camera, the maximum max_{camera} is always the maximum of the Henyey-Greenstein phase function with the current anisotropy value, which is constant in a homogeneous medium, therefore the scale s_{camera} is constant as

Algorithm 6.5: SUMOFCAUCHYDISTRIBUTION($\bar{s}, p, \vec{d}, \text{cp}$)

```
$\text{dist}_s = (\bar{s}_c.o - \text{cp}).\text{length}$
$\text{dist}_{\text{cp}} = (\text{cp} - p).\text{length}$
$\vec{a} = \bar{s}.o - p$
$\vec{c} = \vec{a} \times (\bar{s}.e - p)$
$\vec{e} = (\vec{c} \times \bar{s}.d) \times \vec{c}$
$\theta_o = \tan^{-1}(-\,\text{dist}_s \,/\, \text{dist}_{\text{cp}})$
$\theta_e = \tan^{-1}((\bar{s}.\text{dist} - \text{dist}_s) /\, \text{dist}_{\text{cp}})$
$max_{light} = \text{PHASEFUNCTIONHG}(sign(g)(\vec{d} \cdot \vec{e}))$
$s_{light} = 1/(\pi max_{light})$
$m_{light} = (g > 0?0 : \pi) - \arccos(\vec{e} \cdot (\text{cp} - p))$
$max_{camera} = \text{PHASEFUNCTIONHG}(sign(g)1)$
$s_{camera} = 1/(\pi\, \text{max}_{camera})$
$m_{camera} = sign(g)\pi/2$
$\text{cdf}_{\theta_o} = 0.5 * \text{CAUCHYCDF}(\theta_o, s_{light}, m_{light}) + 0.5 * \text{CAUCHYCDF}(\theta_o, s_{camera}, m_{camera})$
$\text{cdf}_{\theta_e} = 0.5 * \text{CAUCHYCDF}(\theta_e, s_{light}, m_{light}) + 0.5 * \text{CAUCHYCDF}(\theta_e, s_{camera}, m_{camera})$
$u = \epsilon * (\text{cdf}_{\theta_e} - \text{cdf}_{\theta_o}) + \text{cdf}_{\theta_o}$
$\theta_s = \text{CAUCHYCDFMIXTUREINV}(u, s_{camera}, s_{light}, m_{camera}, m_{light})$
$dist = \text{dist}_{\text{cp}} * tan(\theta_s)$
$\text{sample} = \text{cp} + \bar{s}.\vec{d} \cdot \text{dist}$
$\text{pdf}_{f_{\text{uv}}} = 0.5 * \text{CAUCHYPDF}(\theta_s, s_{camera}, m_{camera}, (\text{cdf}_{\theta_e} - \text{cdf}_{\theta_o})) + 0.5 * \text{CAUCHYPDF}(\theta_s, s_{light}, m_{light}, (\text{cdf}_{\theta_e} - \text{cdf}_{\theta_o}))$
$\text{pdf} = \text{pdf}_{f_{\text{uv}}} \cdot (\text{dist}_{\text{cp}} / (\text{dist}_{\text{cp}}^2 + \text{dist}^2))$
return [sample, pdf]
```

well. As the location is simply the sign of the g-value times $\pi/2$, this term is constant in homogeneous media as well.

The next step is to calculate the cdf at θ_o and at θ_e to get the normalization term. Here we calculate the cdf of the Cauchy distribution with CAUCHYCDF for the camera and the light at the mentioned positions. We then sum their halves. We need these cdfs to adjust our random value $\epsilon \in [0, 1)$ to our specified area to get our adjusted sampling value u. With this value, we can now use the CAUCHYCDFMIXTUREINV to get our sampled θ_s.

The steps that follow convert the sampled angle θ_s into a distance *dist* and furthermore create our sample and the pdf of the sum of the two Cauchy distributions $\text{pdf}_{f_{uv}}$. As we work in the angular domain, we finally need to adjust the pdf to get our final pdf *pdf*.

6.4 Mixture Model

Algorithm 6.6: $\textsc{MixtureModelDistribution}(\bar{s}, p, \vec{d}, \text{cp})$

```
1  $\text{dist}_s = (\bar{s}_c.o - \text{cp}).\text{length}$
2  $\text{dist}_{\text{cp}} = (\text{cp} - p).\text{length}$
3  $\vec{a} = \bar{s}.o - p$
4  $\vec{c} = \vec{a} \times (\bar{s}.e - p)$
5  $\vec{e} = (\vec{c} \times \bar{s}.d) \times \vec{c}$
6  $\theta_o = \tan^{-1}(-\text{dist}_s \,/\, \text{dist}_{\text{cp}})$
7  $\theta_e = \tan^{-1}((\bar{s}.\text{dist} - \text{dist}_s)/\,\text{dist}_{\text{cp}})$
8  $max_{light} = \textsc{PhaseFunctionHG}(sign(g)(\vec{d} \cdot \vec{e}))$
9  $s_{light} = 1/(\pi max_{light})$
10 $m_{light} = (g > 0?0 : \pi) - \arccos(\vec{e} \cdot (\text{cp} - p))$
11 $max_{camera} = \textsc{PhaseFunctionHG}(sign(g)1)$
12 $s_{camera} = 1/(\pi\, \text{max}_{camera})$
13 $m_{camera} = sign(g)\pi/2$
14 $\text{cdf}_{light_{\theta_o}} = \textsc{CauchyCDF}(\theta_o, s_{light}, m_{light})$
15 $\text{cdf}_{camera_{\theta_o}} = \textsc{CauchyCDF}(\theta_o, s_{camera}, m_{camera})$
16 $\text{cdf}_{light_{\theta_e}} = \textsc{CauchyCDF}(\theta_e, s_{light}, m_{light})$
17 $\text{cdf}_{camera_{\theta_e}} = \textsc{CauchyCDF}(\theta_e, s_{camera}, m_{camera})$
18 if $\epsilon > 0.5$ then
19     $u = \epsilon * (\text{cdf}_{light_{\theta_e}} - \text{cdf}_{light_{\theta_o}}) + \text{cdf}_{light_{\theta_o}}$
20     $\theta_s = \textsc{CauchyCDFInv}(u, s_{light}, m_{light})$
21 else
22     $u = \epsilon * (\text{cdf}_{camera_{\theta_e}} - \text{cdf}_{camera_{\theta_o}}) + \text{cdf}_{camera_{\theta_o}}$
23     $\theta_s = \textsc{CauchyCDFInv}(u, s_{camera}, m_{camera})$
24 end
25 $dist = \text{dist}_{\text{cp}} * tan(\theta_s)$
26 $\text{sample} = \text{cp} + \bar{s}.\vec{d} \cdot \text{dist}$
27 $\text{pdf}_{f_{\text{uv}}} = 0.5 * \textsc{CauchyPDF}(\theta_s, s_{camera}, m_{camera}, (\text{cdf}_{camera_{\theta_e}} - \text{cdf}_{camera_{\theta_o}})) +$
       $0.5 * \textsc{CauchyPDF}(\theta_s, s_{light}, m_{light}, (\text{cdf}_{light_{\theta_e}} - \text{cdf}_{light_{\theta_o}}))$
28 $\text{pdf} = \text{pdf}_{f_{\text{uv}}} \cdot (\text{dist}_{\text{cp}} / (\text{dist}_{\text{cp}}^2 + \text{dist}^2))$
29 return $[\text{sample}, \text{pdf}]$
```

In our mixture model approach, the majority of the code is identical to the approach with the sum of two Cauchy distributions. Therefore, we will omit the explanation of these parts.

The differences start in Line 14. Here we calculate the cdf of the light and the camera at the angles θ_o and θ_e. Therefore we get four variables $\text{cdf}_{light_{\theta_o}}$, $\text{cdf}_{camera_{\theta_o}}$, $\text{cdf}_{light_{\theta_e}}$

and $\mathrm{cdf}_{camera_{\theta_e}}$.

In the next step we use a random value $\epsilon \in [0, 1)$ to choose which distribution to sample from. In our first case, we sample from the light phase function, in the second one from the camera phase function. From line 25 on, the lines are again almost identical to the previous approach. Only in Line 27 the normalization terms are for the light or camera, and not the combined segment.

CHAPTER 7

Results & Comparison

In this chapter we are evaluating the results for both of our approaches and compare them to the results from Kulla et al. [KF11] and Novák et al. [NNDJ12b]. Our comparison is done with three different scenes:

1. **Cornell Box** A Cornell Box scene which was inspired by a similar variation by Novák et al. [NNDJ12b].
2. **Unidirectional Light** A unidirectional light in an empty space.
3. **Bidirectional Light** A bidirectional light in an empty space.

For each of these scenes, we evaluate four different cases:

1. **Equal Error** We render the scene until an RMSE of 1 was reached. We then compare how long it took to get this result for each approach.
2. **Equal Time** We render the scene for one minute. We then calculate the RMSE and number of samples that were reached after one minute.
3. **Equal Samples 1000** We render the scene with 1000 samples. We then calculate the RMSE to compare the results. Additionally, we also compare the time it took to reach that number of samples.
4. **Equal Samples 10000** We render the scene with 10,000 samples. We then calculate the RMSE to compare the results. Additionally, we also compare the time it took to reach that number of samples.

For each case we render the scene with different anisotropy values. We use the g-value of the Henyey-Greenstein function to get a wide range of results of forward, backward and isotropic scattering ($g \in [-0.9, -0.6, -0.3, 0.0, 0.3, 0.6, 0.9]$). Our tests were made on an Intel Core i7-4770 CPU @ 3.4GHz with 32GB RAM running parallelized over all cores, with all results having a resolution of 512x512. The reference images were created by using the original VRL algorithm and using a high sample count (1,000,000 samples). The results of the PSNR and RMSE tables are always comparisons from the rendered tests to these references. All times in this chapter are provided in the format hh:mm:ss:ms. While we will only highlight a few representative cases in this chapter, all results can be seen in Appendix A.

Our expectation is that on a single sample basis, the original approach by Novák et al. will outperform our approaches, as the time it takes per sample is longer. Additionally, we expect the approach by Kulla et al. to outperform all other approaches for isotropic media. As all other approaches are built on top of the approach by Kulla et al. to account for anisotropic media, for isotropic media, these approaches will therefore be slower. For the equal time and equal error comparisons we expect our approaches to have at least a similar, if not better performance than Novák et al. Even a similar performance would be excellent as both approaches are easier to understand and implement.

7.1 Equal Error

For this test case all scenes were rendered until an RMSE of 1 was reached. We stopped this test if 100,000 samples were exceeded and the RMSE was still above one. This threshold of 100,000 samples was reached in the unidirectional light scene with all approaches for high backward and forward scattering ($g = -0.9$ and $g = 0.9$, respectively).

Out of the 21 test cases (3 scenes, 7 anisotropy values) only in two cases was the original approach by Novák et al. the fastest approach. Both of these cases were with very high forward scattering ($g = 0.9$). A plot of the RMSE progression in this case can be seen in Figure 7.1 where we can see the progression of Novák et al. in red, Kulla et al. in blue, the Sum of Cauchy distributions in green and the Mixture Model approach in orange. As can be seen in this figure, although Novák et al. provide the fastest progression to an RMSE of 1, our approaches are reasonably fast, especially when compared to Kulla et al.

The approach by Kulla et al. was the fastest in all isotropic cases and four out of the six cases with low anisotropy ($g \in [-0.3, 0.3]$). As Kulla et al. take significantly less time per sample, more samples can be taken. For low anisotropic media, this therefore results in faster convergence.

For all other cases our approaches were faster, where the sum of Cauchy distributions was faster for backward scattering ($g < 0$) and the mixture model of Cauchy distributions was faster for forward scattering ($g > 0$). Representative results can be seen in Figure 7.2 where we can see the progression of Novák et al. in red, Kulla et al. in blue, the Sum of Cauchy distributions in green and the Mixture Model approach in orange. Both

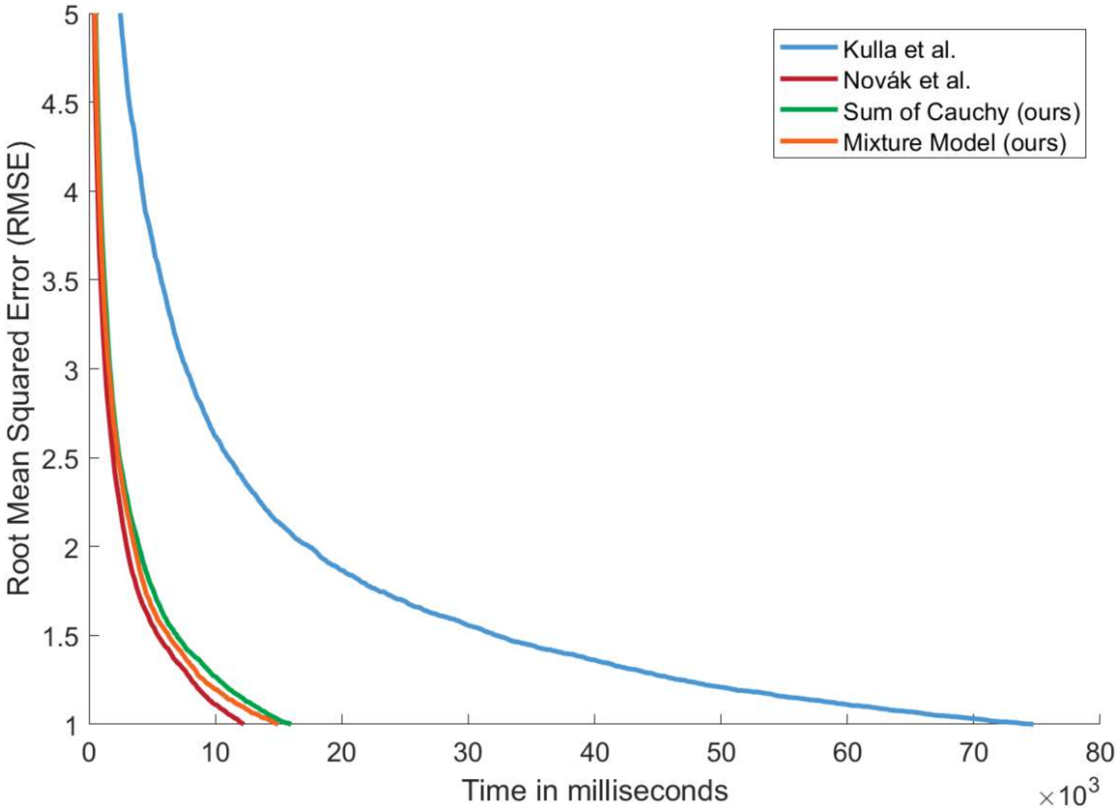


Figure 7.1: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.9 for the Henyey-Greenstein phase function.

or approaches outperform Novák et al. and Kulla et al. where the Sum of Cauchy distributions is especially fast to converge.

7.2 Equal Time

For this test, all scenes were rendered for 60 seconds. Then the RMSE and the PSNR were calculated by comparing the results to the reference solutions. The insights that we get from these tests are similar to the ones from the equal error tests. We will again highlight representative results for the different approaches.

As for the equal error tests, in these tests, the original approach by Novák et al. outperformed all other approaches for high forward scattering media ($g = 0.9$). An example of this can be seen in Figure 7.3 (top row). The approach of Novák et al. reached a RMSE of 1.04, whereas Kulla et al. achieved 2.46, the sum of Cauchy distributions 1.42 and the mixture model with Cauchy distributions 1.25.

As expected, the approach by Kulla et al. was the fastest in all isotropic cases. An

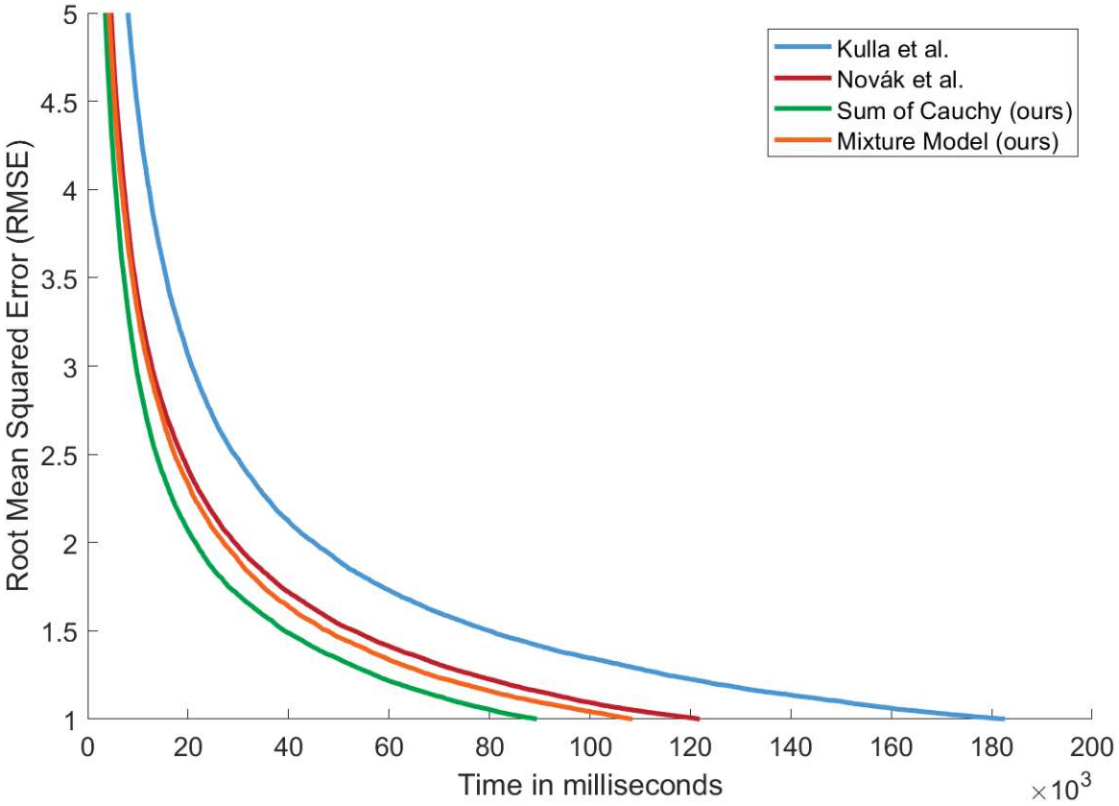


Figure 7.2: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a *g*-value of −0.9 for the Henyey-Greenstein phase function.

example can be seen in Figure 7.3 (bottom row). For this scene, the approach by Kulla et al. achieved a RMSE of 0.94 in 60 seconds, whereas Novák et al. achieved 0.99, the sum of Cauchy distributions 0.95 and the mixture model approach 1.04 in the same time frame.

For most other test cases, our approaches either outperformed Kulla et al. and Novák et al. or were at least competitive. Especially for backward scattering media ($g < 0$), our approaches outperformed Novák et al. in all test cases. Only in one out of six cases Novák et al. reached a better RMSE for forward scattering media ($g > 0$).

In Figure 7.4 (top row) a representative example of the approach with the sum of Cauchy distributions can be seen. In the 60 seconds that this image was rendered, a RMSE of 2.09 was reached for this scene with high backward scattering ($g = -0.9$). The approach by Kulla et al. reached 2.98, Novák et al. 2.44 and the mixture model approach 2.36 in the 60 seconds they were rendered.

In the bottom row of Figure 7.4 a representative example of the mixture model can be seen, where it outperforms all other approaches. For this scene with higher forward

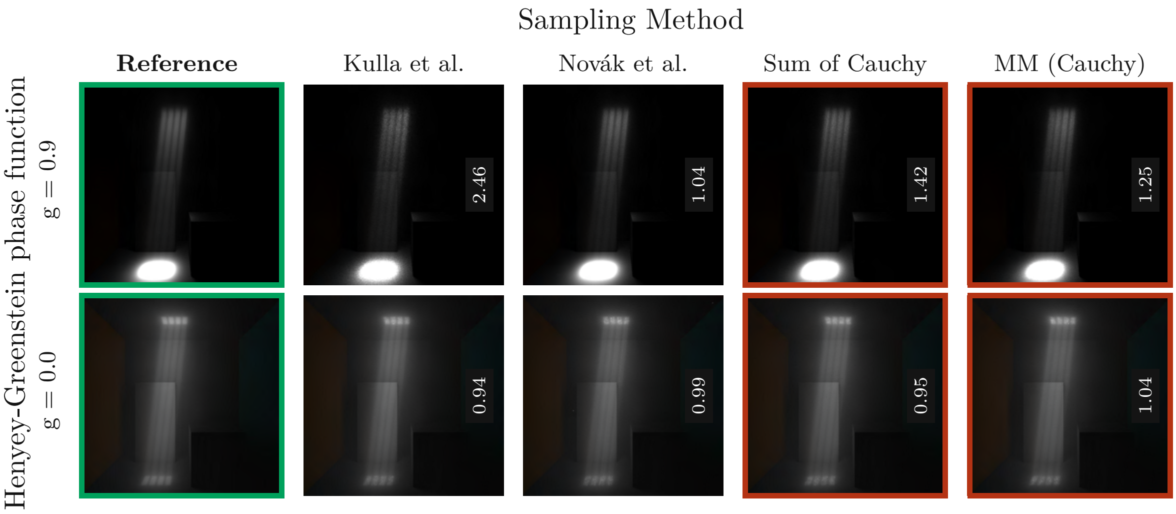


Figure 7.3: Representative examples of the equal time case where other approaches (Novák et al., Kulla et al.) were faster than our approaches. On the top row, the results of the Cornell box scene that were rendered for 60 seconds with high forward scattering media ($g = 0.9$) can be seen. The approach by Novák et al. achieved a RMSE of 1.04 which was the best result, compared to all other approaches. In the bottom row, the results of the Cornell box scene that were rendered for 60 seconds with isotropic media ($g = 0.0$) can be seen. The approach by Kulla et al. achieved a RMSE of 0.94 which was the best result, compared to all other approaches. The achieved RMSE for each approach is displayed within the corresponding image.

scattering ($g = 0.6$) a RMSE of 2.25 was reached. In this test scenario, Kulla et al. achieved a RMSE of 2.63, Novák et al. achieved a RMSE of 2.44 and the sum of Cauchy distributions achieved a RMSE of 2.26.

7.3 Equal Sample

The equal sample test case will not be discussed in detail, but can be viewed in full in the appendix A. As the approach by Novák et al. can be adapted to have more or less samples within the importance sampling function, a higher number of samples can be chosen to match the target function better. As this would result in a slow result per sample, but an accurate result, it is not a fair comparison, as all other approaches do not have this option.

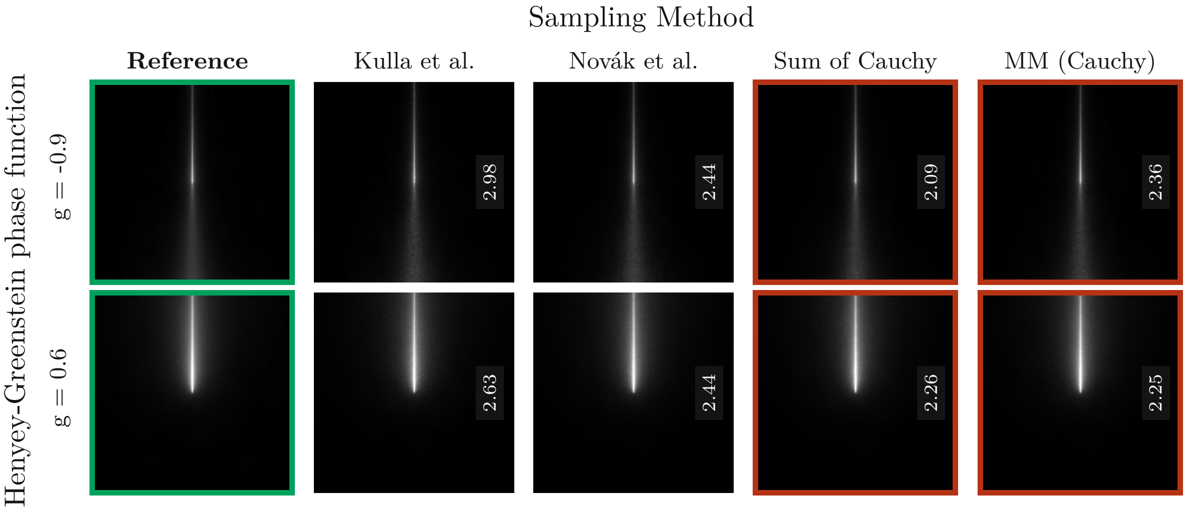


Figure 7.4: Representative examples of the equal time case where our approaches were faster than the other approaches (Novák et al., Kulla et al.). On the top row, the results of the bidirectional light scene that were rendered for 60 seconds with high backward scattering media ($g = -0.9$) can be seen. The Sum of Cauchy distributions approach achieved a RMSE of 2.09 which was the best result, compared to all other approaches. In the bottom row, the results of the bidirectional light scene that were rendered for 60 seconds with higher forward scattering media ($g = 0.6$) can be seen. The mixture model approach with Cauchy distributions achieved a RMSE of 2.25 which was the best result compared to all other approaches. The achieved RMSE for each approach is displayed within the corresponding image.

CHAPTER 8

Conclusion & Future Work

8.1 Conclusion

In this thesis we have presented two new importance sampling techniques that solve the anisotropic case of VRL. First, we discussed light transport in general and different ray-based approaches for rendering participating media. Then we analyzed the work by Novák et al. in detail and provided two new approaches for solving the anisotropic case of VRL. We provide them as mathematical formulations, pseudo code and actual code via a GitHub project. This repository additionally contains further implementations with other attempted solutions as discussed in Appendix B.

In our first approach we use the sum of two Cauchy distributions to approximate the product of the phase functions, which is our target function. The sum of two Cauchy distributions can be integrated and then inverted, which is necessary when being used in Monte Carlo importance sampling approach. Therefore, this approach provides a solution that can be directly sampled from.

In our second approach we use a mixture model with two Cauchy distributions. Although similar, this approach differs from the first approach as the two Cauchy distributions have to be normalized when used in the mixture model. This therefore leads to different results and accuracy when comparing it to the target function. The advantage of this approach is its mathematical simplicity as the Cauchy distributions only need to be integrated and inverted individually.

Both of our approaches outperform the original approach by Novák et al. in most of our test cases. Only for high forward scattering media ($g = 0.9$), Novák et al. still maintain a better performance. Not only do our approaches outperform Novák et al. in most of our tests, the mathematical formulation is less complex and can therefore be evaluated easier for future adaptations. We provide detailed test scenarios in Appendix A which were analyzed in Section 7.

8.2 Future Work

There are many different areas in which the ideas that are presented in this thesis can be further developed. In Appendix B we discuss other approaches that we have already evaluated that could be of use for future work.

Other distributions in the mixture model - In our approaches we use the Cauchy distribution in the mixture model as we found this distribution to fit well to the Henyey-Greenstein phase function in our application. We tried other distributions with varying results, but settled on the Cauchy distribution as it seemed the most promising. As there are many other distributions, there could be one that fits even better to this use case.

Product of distributions - As the target function is a product of two Henyey-Greenstein functions, the product of two other functions might be a good fit to solve this problem. We did not find a solution for which the product of two functions that was a good fit and analytically solvable, but there might be one.

Improved matching for the distributions - In our approaches we kept the matching of the functions, choosing the location and scale, simple and took the obvious choice. By adapting those values, the final result could fit even better without much overhead. There could even be a clear way of finding these values for different medium properties.

APPENDIX A

Full Results

In this appendix we present the results of the approaches that were introduced and compare them to the original algorithm. We are evaluating our results in three different scenes.

1. a variant of the Cornell box that was used in the original paper
2. a single light ray
3. a unidirectional light

For all of these scenes, the following three sets of tests are conducted.

1. equal error
2. equal time
3. equal sample

With these test sets we try to cover the most cases that we can. Our tests were made on an Intel Core i7-4770 CPU @ 3.4GHz with 32GB RAM running parallelized over all cores, with all results having a resolution of 512x512.

In our tests we will always use a the same table structure. The columns of the table represent different anisotropy values of the Henyey-Greenstein phase function, whereas the rows show the results of different algorithms. In the first row we always show the reference image (with a green border). The second and the third row show the original approach, where the second row only uses the equi-angular sampling by Kulla et al. [KF11] and the third row uses the full implementation of Novák et al. [NNDJ12b]. The fourth and fifth row show our own implementations with a red border. In the fourth row, the results of the sum of two Cauchy distributions is shown, whereas in the fifth row, the Mixture Model approach is depicted.

The reference images were created by using the original VRL algorithm and using a high sample count (1,000,000 samples). The results of the PSNR and RMSE tables are always comparisons from the rendered tests to these references.

A.1 Cornell Box

Scene, Light & Medium Settings	
Light Position	(0, 1.9, -3)
Light Intensity	50,000
Medium (σ_a, σ_s)	0.1, 0.01

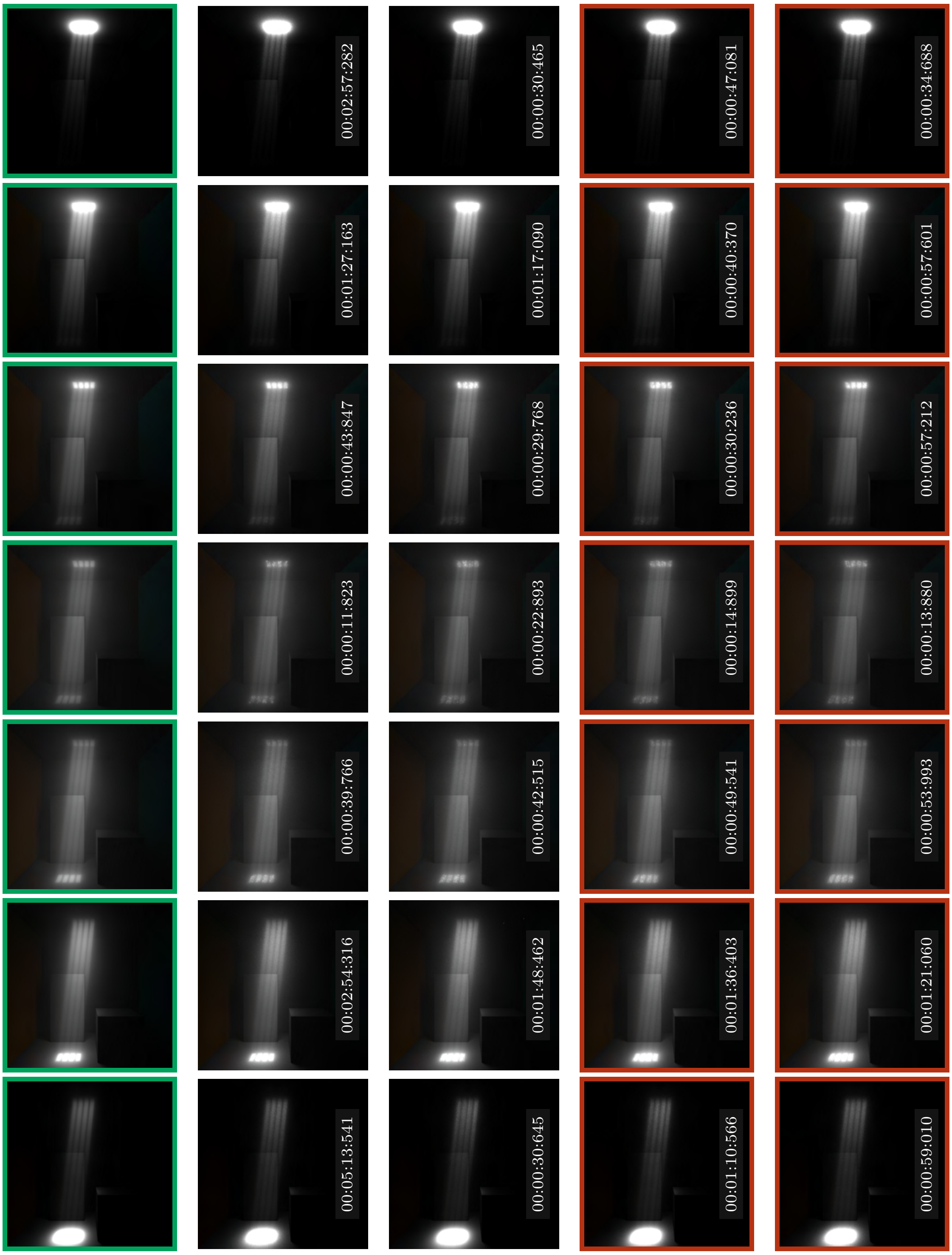
Table A.1: The light and media properties of the Cornell box scene.

This test case is a recreation of the well know Cornell box scene. The media and light properties can be seen in Table A.1. This scene was inspired by a similar scene in the original paper. The starting position of the light rays is chosen randomly within the four rectangles on the ceiling, the direction is always the same. The surface illumination is solely calculated by the contribution of the radiance from the media on to the surface.

A.1.1 Equal Error

The equal error comparisons evaluate how long the rendering took until a certain RMSE threshold was reached.

Equal Error - RMSE 1 Table A.2 shows the results of a Cornell box scene when rendered until the RMSE is lower than 1.



Henyey-Greenstein phase function	Reference	Kulla et al.	Novák et al.	sum of Cauchy	MM (Cauchy)
g = -0.9		00:02:57:282	00:00:30:465	00:00:47:081	00:00:34:688
g = -0.6		00:01:27:163	00:01:17:090	00:00:40:370	00:00:57:601
g = -0.3		00:00:43:847	00:00:29:768	00:00:30:236	00:00:57:212
g = 0.0		00:00:11:823	00:00:22:893	00:00:14:899	00:00:13:880
g = 0.3		00:00:39:766	00:00:42:515	00:00:49:541	00:00:53:993
g = 0.6		00:02:54:316	00:01:48:462	00:01:36:403	00:01:21:060
g = 0.9		00:05:13:541	00:00:30:645	00:01:10:566	00:00:59:010

Sampling Method

Table A.2: Comparison of the Cornell box scene until a RMSE of 1 is reached. (Time in hh:mm:ss:ms)

A.1.2 Equal Time

The equal time comparison renders until a time limit is reached. Afterwards, RMSE and PSNR can be calculated from these results to see how far they have converged in that time.

Equal Time - 60 Seconds Table A.4 shows the results of a Cornell box scene when rendered for 60 seconds. The corresponding RMSE and PSNR can be seen in Table A.3.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	2.74	39.36	1.68	43.61	1.68	43.60	1.66	43.69
-0.6	2.11	41.63	2.19	41.29	1.72	43.38	2.00	42.08
-0.3	1.37	45.36	1.78	43.08	1.31	45.78	1.67	43.64
0.0	0.93	48.70	1.26	46.08	1.10	47.30	1.19	46.56
0.3	1.60	44.03	1.57	44.17	1.68	43.59	1.47	44.72
0.6	2.76	39.28	2.33	40.76	2.36	40.64	2.07	41.77
0.9	4.01	36.05	1.24	46.25	1.89	42.55	1.74	43.27

Table A.3: Comparison of the RMSE and PSNR of Table A.4.

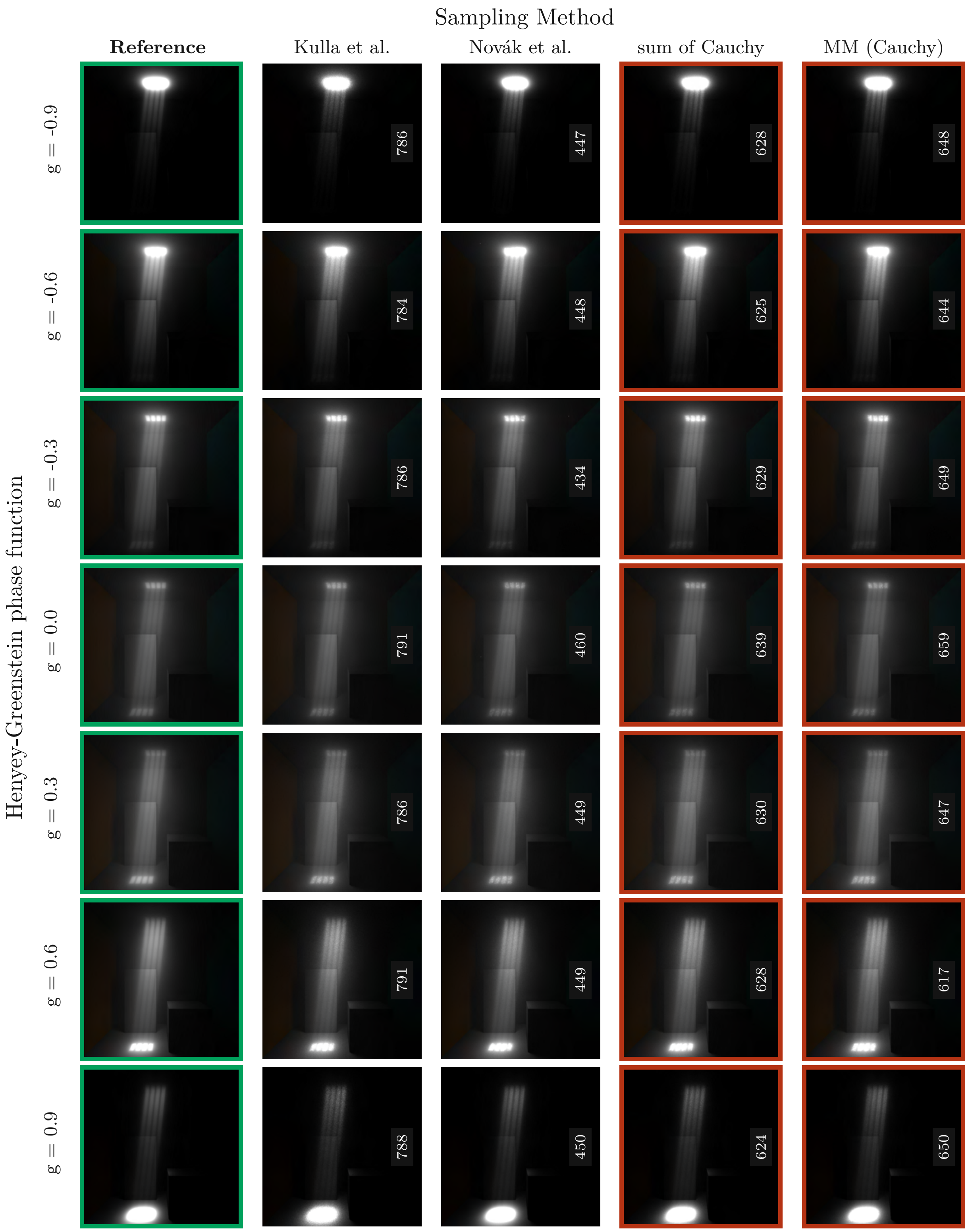


Table A.4: Comparison of the Cornell box scene that was rendered for 60 seconds. Numbers of samples is provided in the pictures.

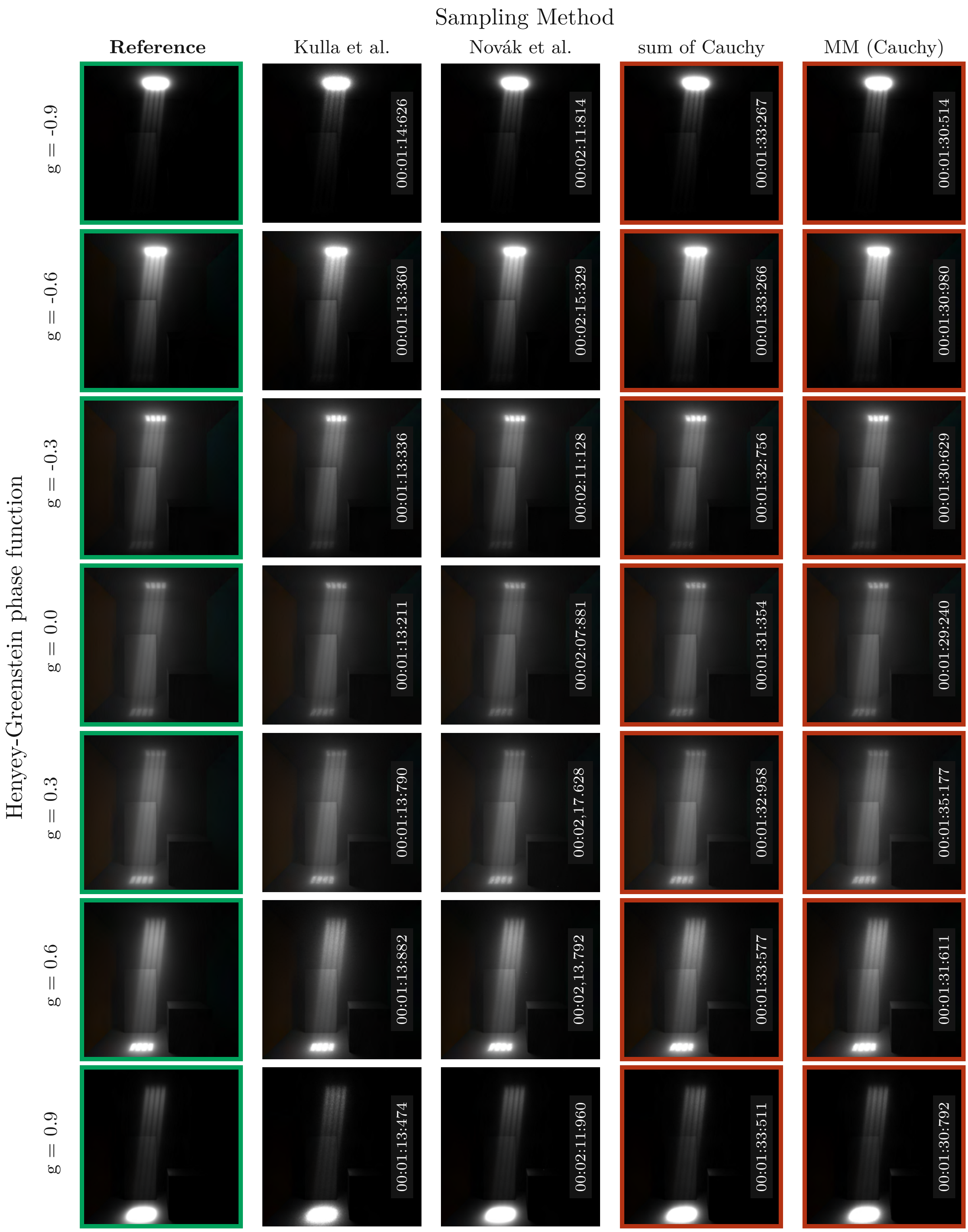
A.1.3 Equal Sample

The equal sample approach evaluates the efficiency of the algorithm on a per sample basis. Therefore it is expected that the original algorithm will outperform our approaches as the time taken per sample is longer. But this test still provides valid insight on how close the results are.

Equal Sample - 1000 Samples Table A.6 shows the results of a Cornell box scene when rendered for 60 seconds. The corresponding RMSE and PSNR can be seen in Table A.5.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	2.46	40.29	1.04	47.78	1.42	45.04	1.25	46.16
-0.6	2.12	41.58	1.10	47.26	1.13	47.01	1.18	46.68
-0.3	1.36	45.41	1.27	45.99	1.22	46.35	1.09	47.30
0.0	0.94	48.60	0.99	48.18	0.95	48.53	1.04	47.76
0.3	1.44	44.95	1.32	45.67	1.23	46.26	1.23	46.29
0.6	2.50	40.14	1.50	44.58	1.85	42.74	1.66	43.69
0.9	3.55	37.11	1.06	47.61	1.74	43.28	1.39	45.22

Table A.5: Comparison of the RMSE and PSNR of Table A.6.



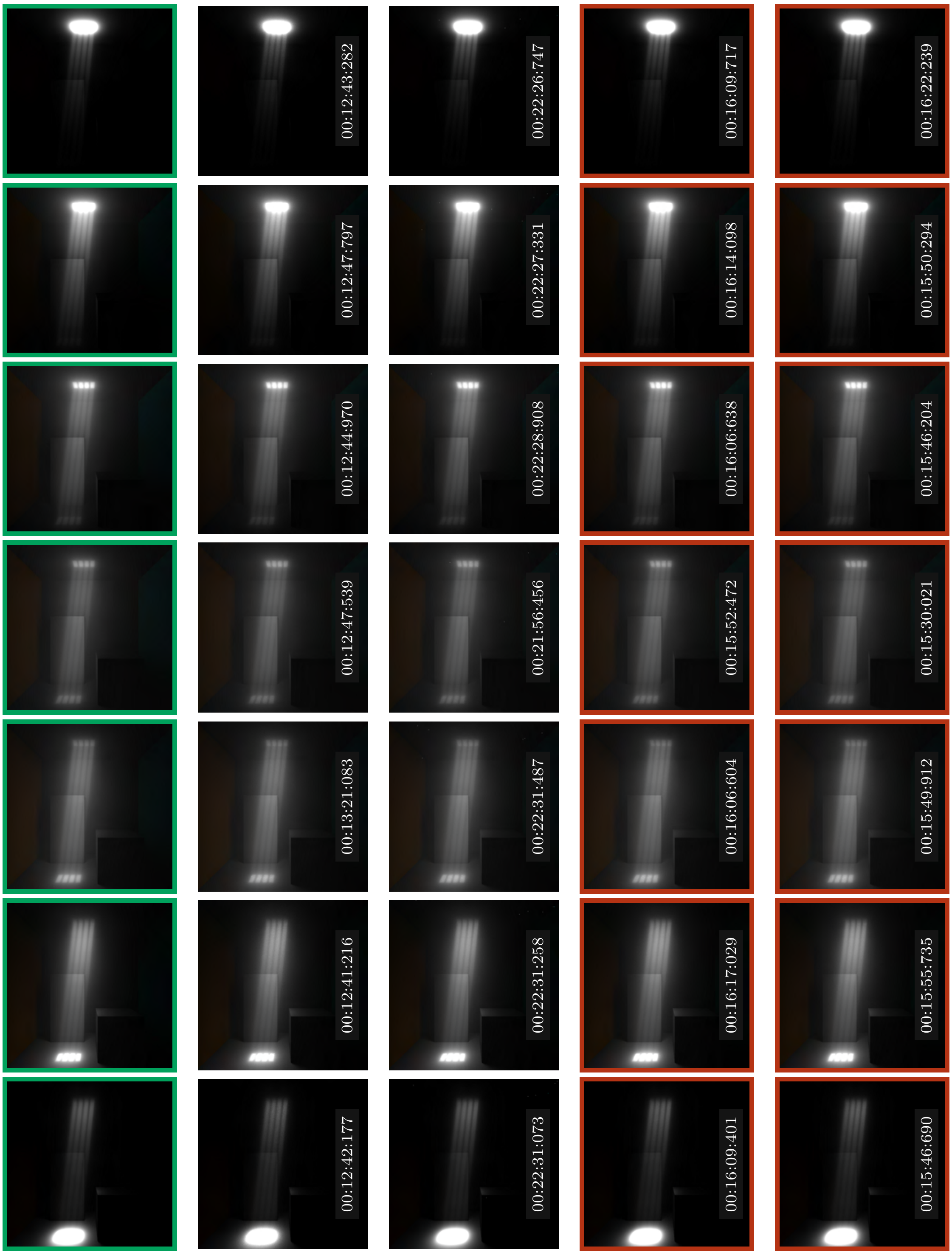
	Reference	Kulla et al.	Novák et al.	sum of Cauchy	MM (Cauchy)
g = -0.9		00:01:14:626	00:02:11:814	00:01:33:267	00:01:30:514
g = -0.6		00:01:13:360	00:02:15:329	00:01:33:266	00:01:30:980
g = -0.3		00:01:13:336	00:02:11:128	00:01:32:756	00:01:30:629
g = 0.0		00:01:13:211	00:02:07:881	00:01:31:354	00:01:29:240
g = 0.3		00:01:13:790	00:02,17.628	00:01:32:958	00:01:35:177
g = 0.6		00:01:13:882	00:02,13.792	00:01:33:577	00:01:31:611
g = 0.9		00:01:13:474	00:02:11:960	00:01:33:511	00:01:30:792

Table A.6: Comparison of the Cornell box scene until 1000 samples are reached. (Time in hh:mm:ss:ms)

Equal Sample - 10,000 Samples Table A.8 show the results and times of the scene being rendered with 10,000 samples. As can be seen from Table A.7, all results have an extremely low RMSE.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	0.80	49.97	1.00	48.07	0.53	53.51	0.44	55.24
-0.6	0.66	51.72	1.24	46.25	0.48	54.45	0.49	54.30
-0.3	0.49	54.29	0.73	50.79	0.48	54.33	0.62	52.26
0.0	0.36	56.87	0.93	48.73	0.39	56.27	0.43	55.42
0.3	0.63	52.00	1.24	46.23	0.48	54.37	0.60	52.43
0.6	0.89	49.07	1.28	45.96	0.65	51.86	0.60	52.47
0.9	1.15	46.85	1.33	45.63	0.55	53.24	0.52	53.65

Table A.7: Comparison of the RMSE and PSNR of Table A.8.



Henyey-Greenstein phase function	**Reference**	Kulla et al.	Novák et al.	sum of Cauchy	MM (Cauchy)
		Sampling Method			
g = -0.9		00:12:43:282	00:22:26:747	00:16:09:717	00:16:22:239
g = -0.6		00:12:47:797	00:22:27:331	00:16:14:098	00:15:50:294
g = -0.3		00:12:44:970	00:22:28:908	00:16:06:638	00:15:46:204
g = 0.0		00:12:47:539	00:21:56:456	00:15:52:472	00:15:30:021
g = 0.3		00:13:21:083	00:22:31:487	00:16:06:604	00:15:49:912
g = 0.6		00:12:41:216	00:22:31:258	00:16:17:029	00:15:55:735
g = 0.9		00:12:42:177	00:22:31:073	00:16:09:401	00:15:46:690

Table A.8: Comparison of the Cornell box scene until 10,000 samples are reached. (Time in hh:mm:ss:ms)

A.2 Unidirectional

Scene, Light & Medium Settings	
Light Position	(0, 0, -2)
Light Intensity	100
Medium (σ_a, σ_s)	0.1, 0.25

Table A.9: The light and media properties of the Unidirectional scene.

This test case is a recreation of a standard unidirectional light scene. The media and light properties can be seen in Table A.9. Although there are other approaches that render unidirectional lights faster, we show in this section that it is still possible with VRL, although the convergence is relatively slow compared to other test cases.

A.2.1 Equal Error

The equal error comparisons evaluate how long the rendering took until a certain RMSE threshold was reached.

Equal Error - RMSE 1 Table A.10 shows the results of a unidirectional light scene when rendered until the RMSE is lower than 1. If this RMSE goal was not reached within 100,000 samples, a red cross is displayed instead.

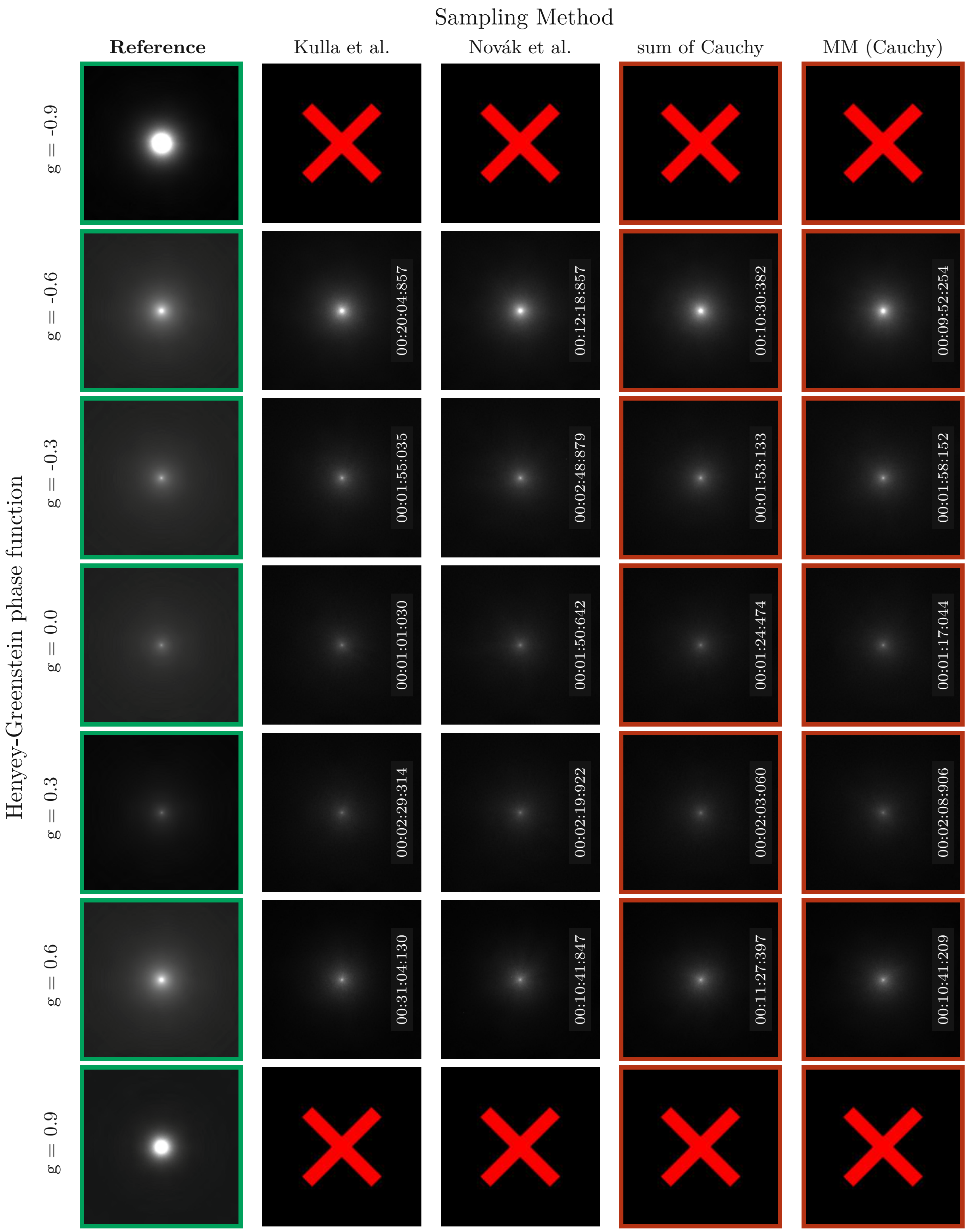


Table A.10: Comparison of the unidirectional light scene until a RMSE of 1 is reached. If this RMSE was not reached in 100,000 samples, a red cross is displayed. (Time in hh:mm:ss:ms)

A.2.2 Equal Time

The equal time comparison renders until a time limit is reached. Afterwards, RMSE and PSNR can be calculated from these results to see how far they have converged in that time.

Equal Time - 60 Seconds Table A.12 shows the results of a Cornell box scene when rendered for 60 seconds. The corresponding RMSE and PSNR can be seen in Table A.11.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	24.73	20.26	21.46	21.49	21.77	21.37	21.73	21.38
-0.6	8.18	29.86	5.62	33.12	5.52	33.28	4.89	34.34
-0.3	2.40	40.51	3.34	37.65	2.38	40.59	2.37	40.60
0.0	1.83	42.85	2.43	40.38	1.97	42.23	1.93	42.39
0.3	2.92	38.80	2.49	40.18	2.48	40.23	2.52	40.09
0.6	9.48	28.58	4.53	35.00	5.11	33.94	5.33	33.59
0.9	24.15	20.47	18.05	22.99	18.53	22.77	17.08	23.48

Table A.11: Comparison of the RMSE and PSNR of Table A.12.

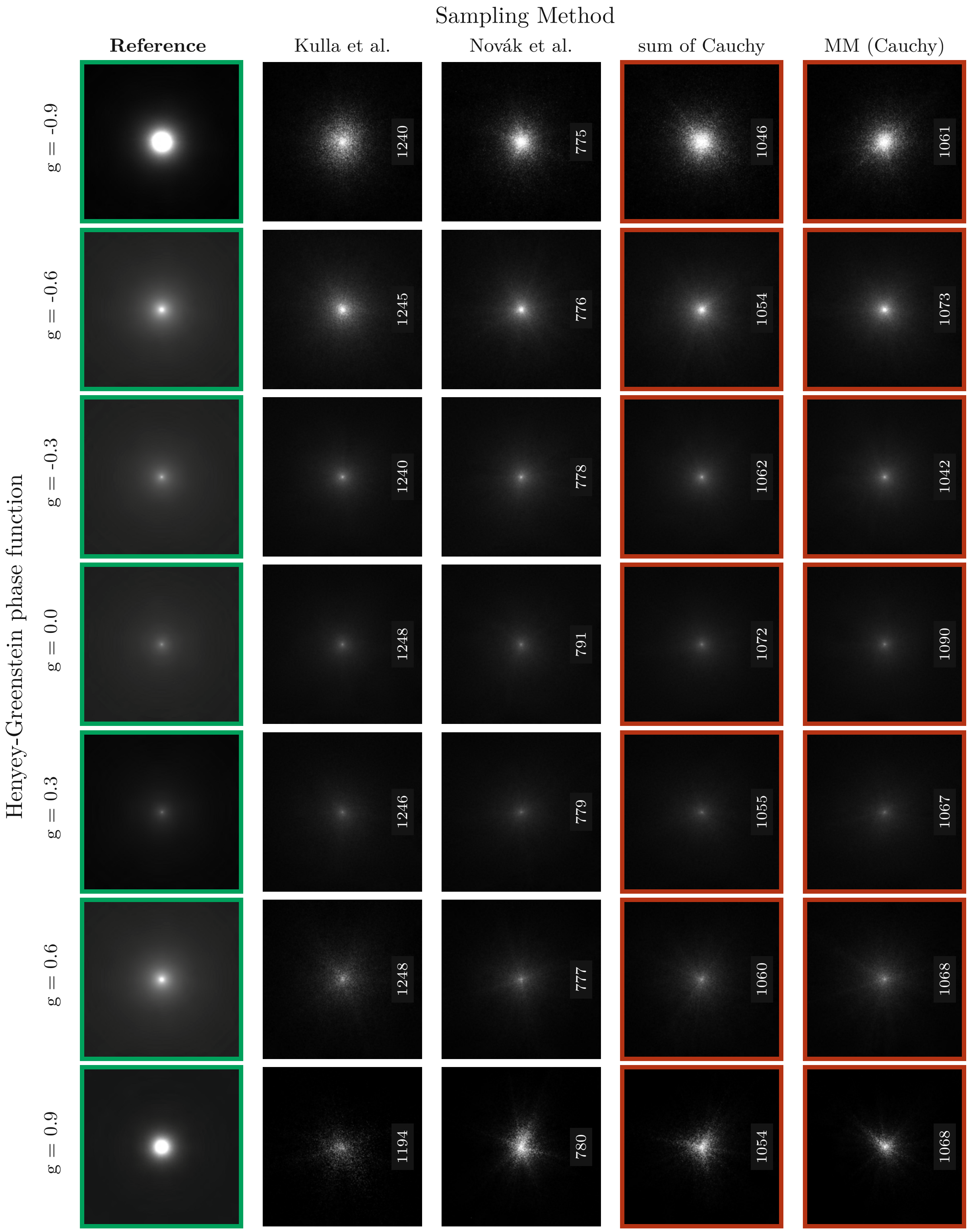


Table A.12: Comparison of the unidirectional light scene that was rendered for 60 seconds. Numbers of samples is provided in the pictures.

A.2.3 Equal Sample

The equal sample approach evaluates the efficiency of the algorithm on a per sample basis. Therefore it is expected that the original algorithm will outperform our approaches as the time taken per sample is longer. But this test still provides valid insight on how close the results are.

Equal Sample - 1000 Samples Table A.14 shows the results of a Cornell box scene when rendered for 60 seconds. The corresponding RMSE and PSNR can be seen in Table A.13.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	29.52	18.72	21.05	21.66	20.97	21.69	22.66	21.02
-0.6	7.30	30.85	5.22	33.76	5.40	33.47	5.65	33.08
-0.3	2.90	38.87	2.61	39.79	2.51	40.12	2.43	40.41
0.0	2.00	42.07	2.04	41.90	1.99	42.11	2.03	41.95
0.3	3.05	38.41	2.28	40.94	2.68	39.55	2.62	39.73
0.6	11.84	26.66	5.19	33.82	5.56	33.22	5.70	33.00
0.9	25.46	20.01	16.65	23.70	19.60	22.28	19.28	22.42

Table A.13: Comparison of the RMSE and PSNR of Table A.14.

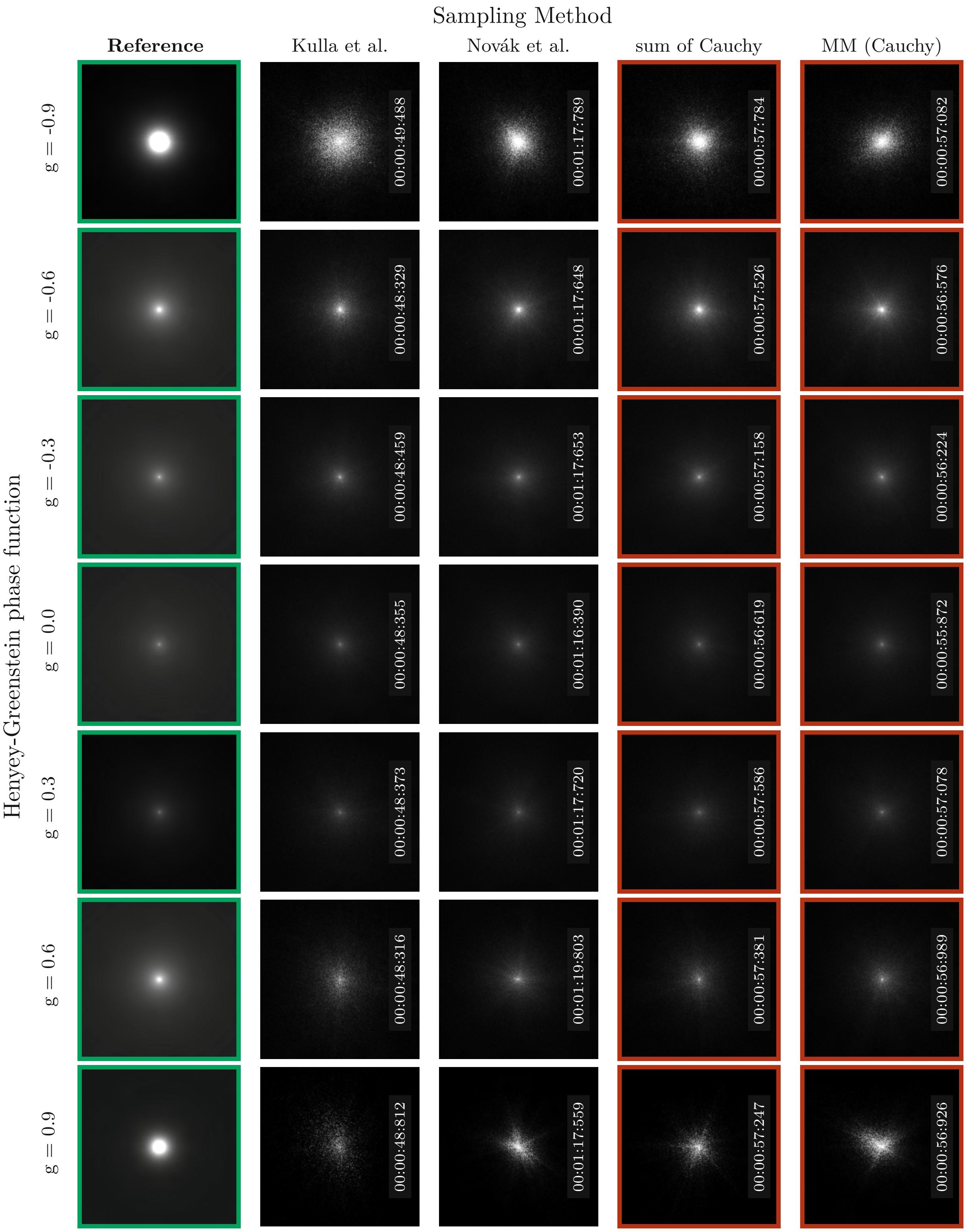


Table A.14: Comparison of the Cornell box scene until 1000 samples are reached. (Time in hh:mm:ss:ms)

Equal Sample - 10,000 Samples Table A.16 show the results and times of the scene being rendered with 10,000 samples. As can be seen from Table A.15, all results have an extremely low RMSE.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	12.93	25.89	8.77	29.26	10.20	27.95	10.35	27.82
-0.6	2.75	39.33	1.80	43.01	1.86	42.70	1.80	43.01
-0.3	0.93	48.69	0.88	49.15	0.86	49.37	0.91	48.94
0.0	0.75	50.59	0.74	50.69	0.75	50.52	0.77	50.31
0.3	1.08	47.45	0.85	49.53	0.91	48.90	0.92	48.84
0.6	3.36	37.58	1.66	43.68	2.01	42.02	1.77	43.15
0.9	18.37	22.84	7.83	30.24	9.37	28.69	8.51	29.52

Table A.15: Comparison of the RMSE and PSNR of Table A.16.

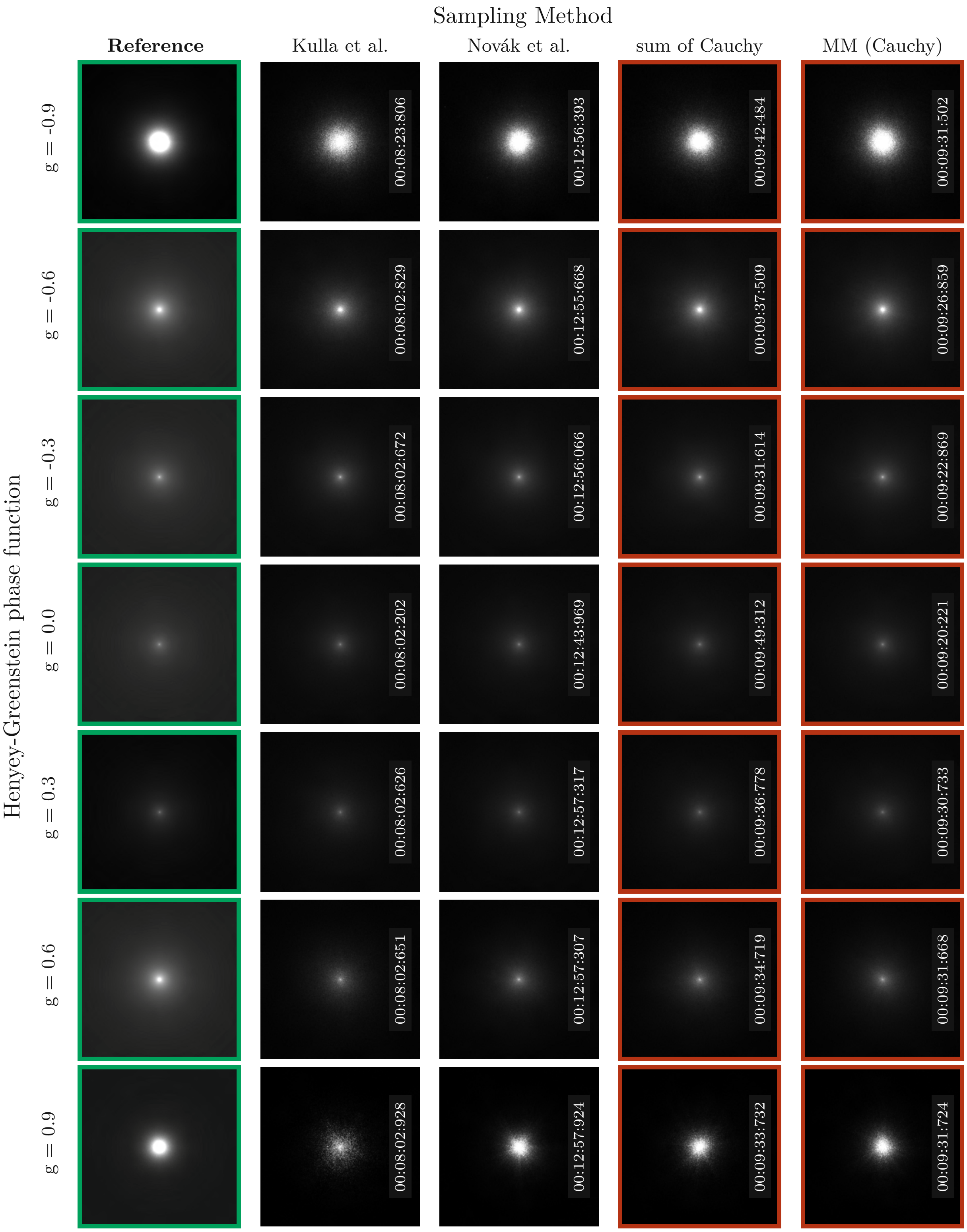


Table A.16: Comparison of the unidirectional light scene until 10,000 samples are reached. (Time in hh:mm:ss:ms)

A.3 Bidirectional1

Scene, Light & Medium Settings	
Light Position	(0, 0, -2)
Light Intensity	250
Medium (σ_a, σ_s)	0.1, 0.25

Table A.17: The light and media properties of the bidirectional light scene.

This test case is a recreation of the bidirectional light scene. The media and light properties can be seen in Table A.17. This scene was inspired by the general behavior of the VRL. As there is always a ray to ray light calculation (light and camera ray), it is a good test scene to just visualize a single ray.

A.3.1 Equal Error

The equal error comparisons evaluate how long the rendering took until a certain RMSE threshold was reached.

Equal Error - RMSE 1 Table A.18 shows the results of a bidirectional light scene when rendered until the RMSE is lower than 1. The corresponding data for the progression of the RMSE per sample can be seen in Figures A.1, A.2, A.3, A.4, A.5, A.6 and A.7.

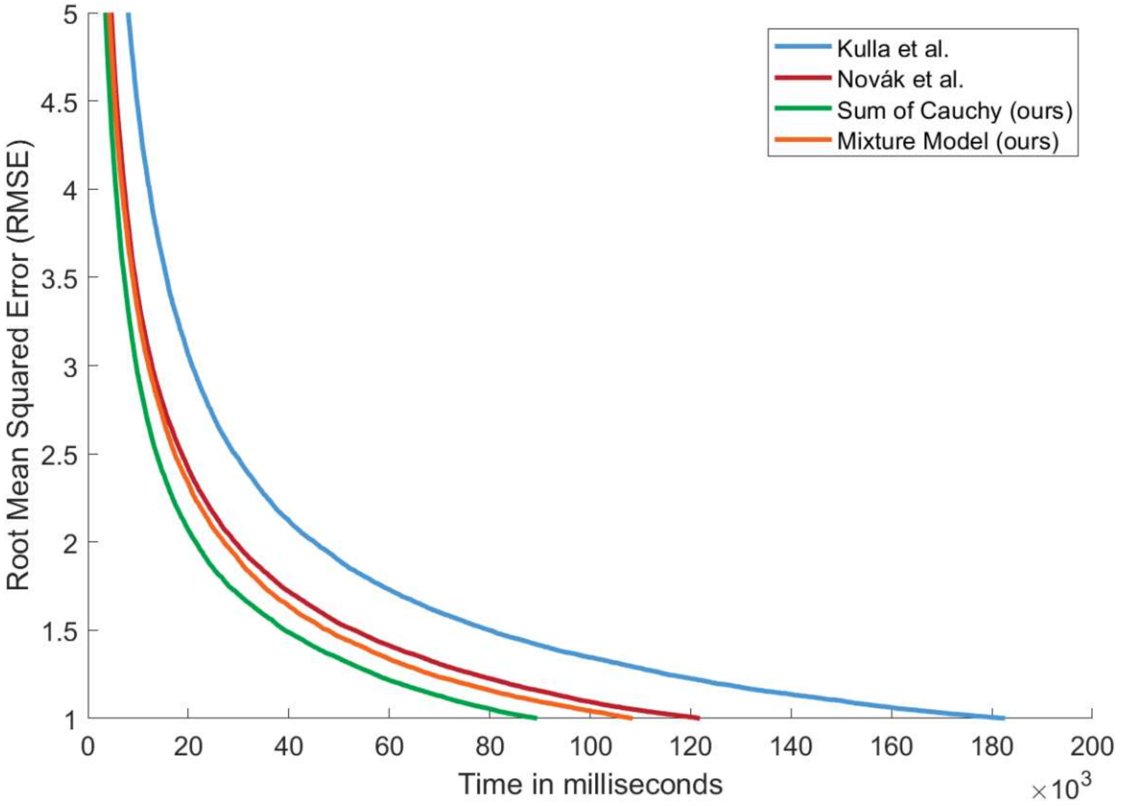


Figure A.1: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of -0.9 for the Henyey-Greenstein phase function.

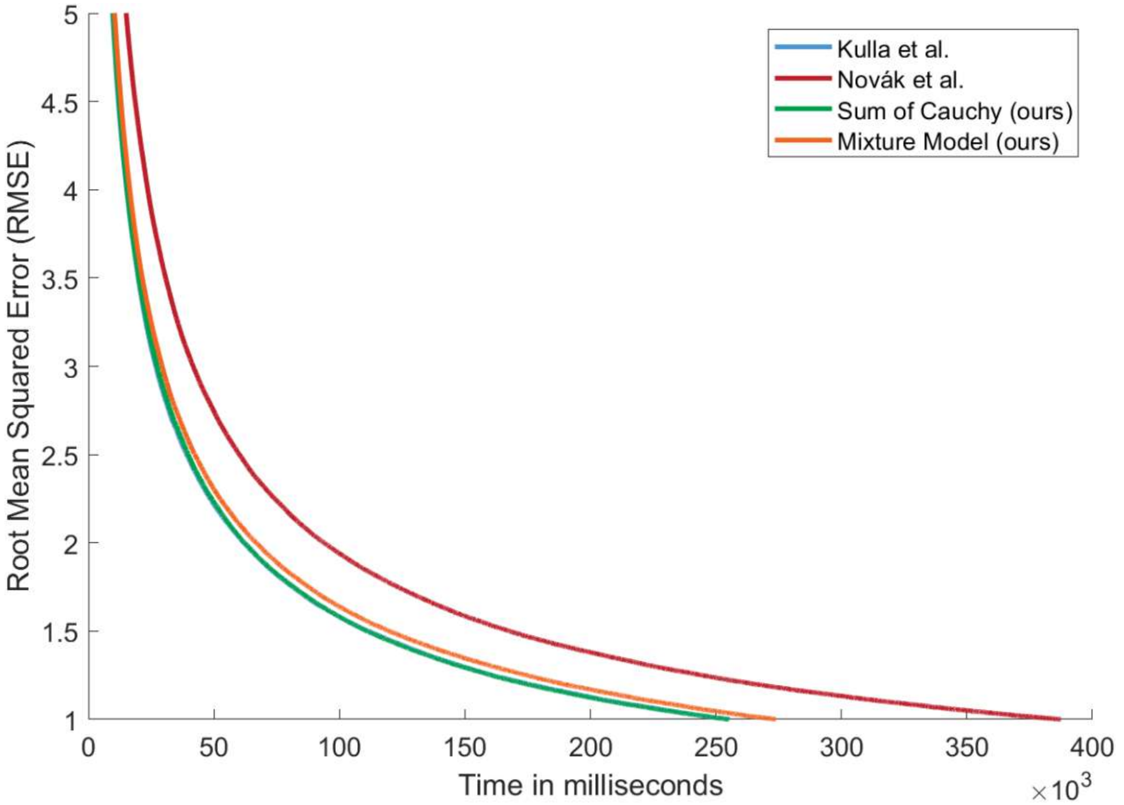


Figure A.2: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of -0.6 for the Henyey-Greenstein phase function.

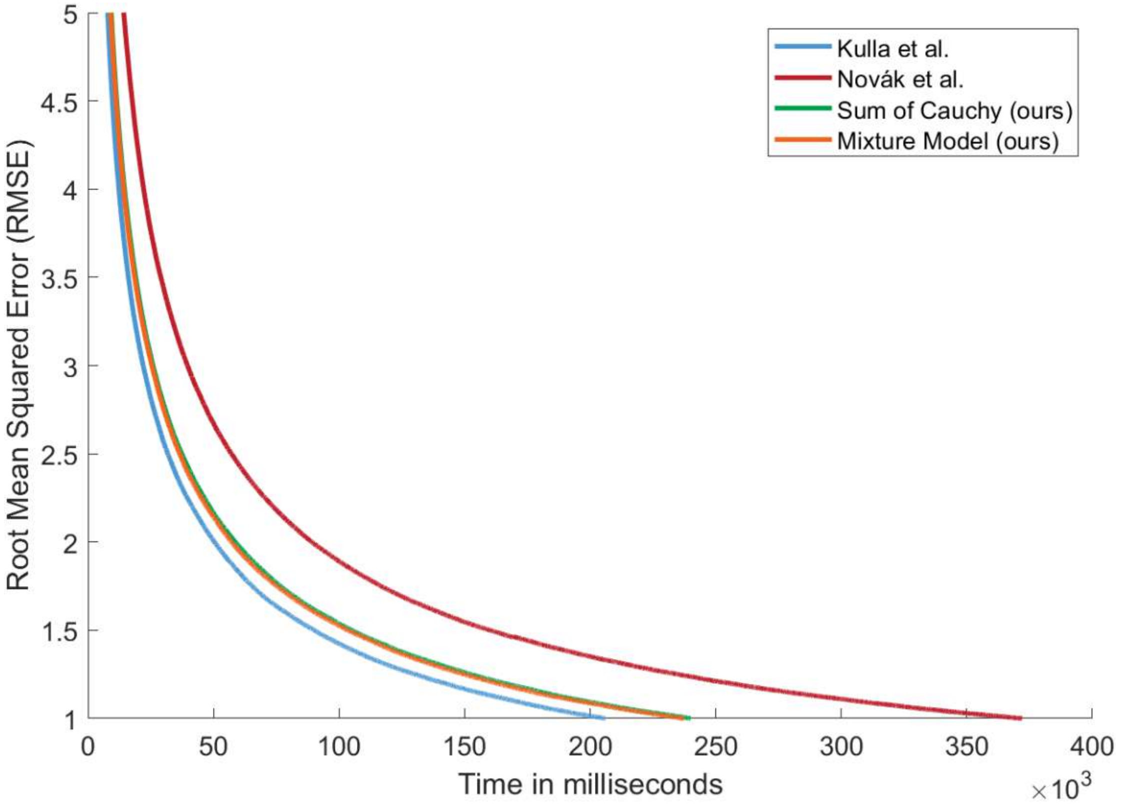


Figure A.3: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of -0.3 for the Henyey-Greenstein phase function.

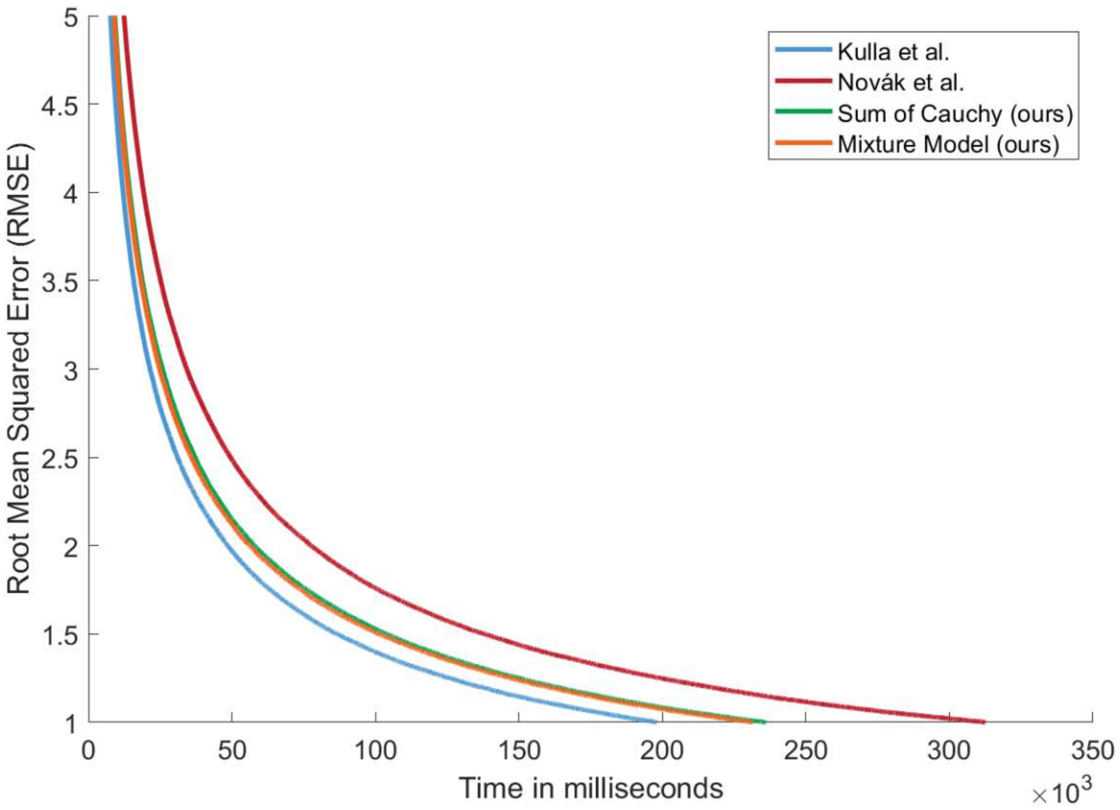


Figure A.4: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.0 for the Henyey-Greenstein phase function.

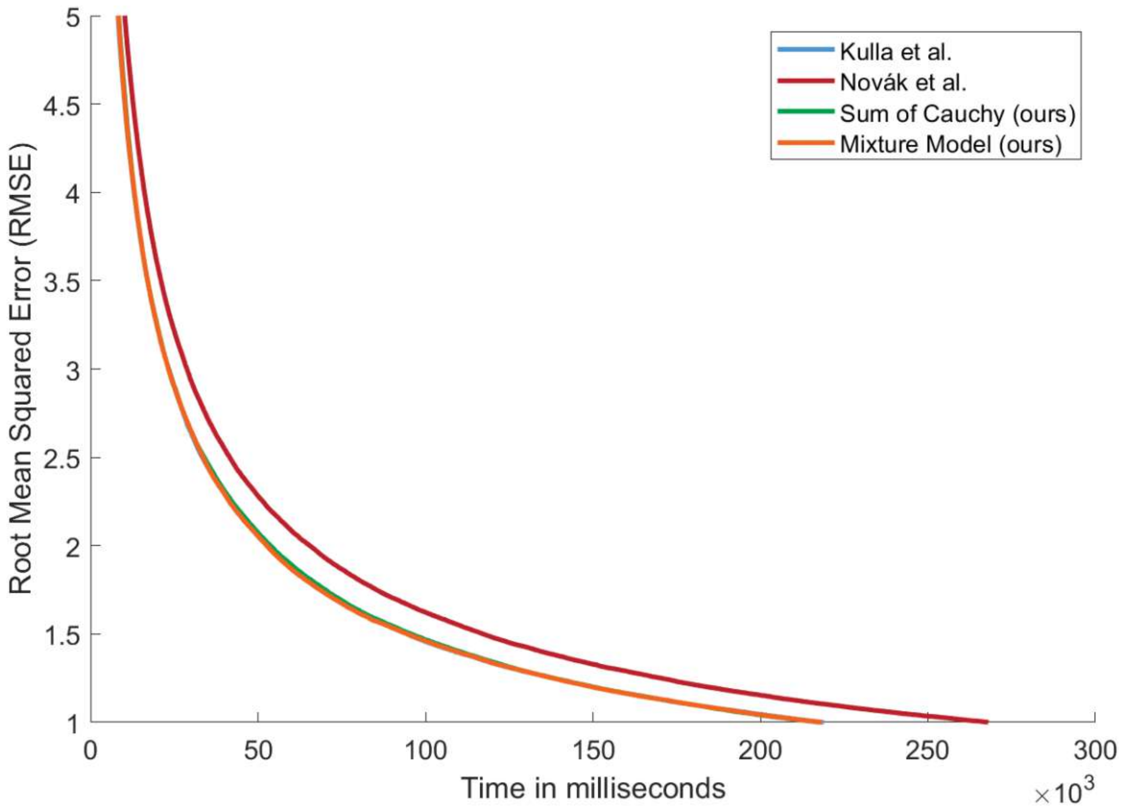


Figure A.5: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a *g*-value of 0.3 for the Henyey-Greenstein phase function.

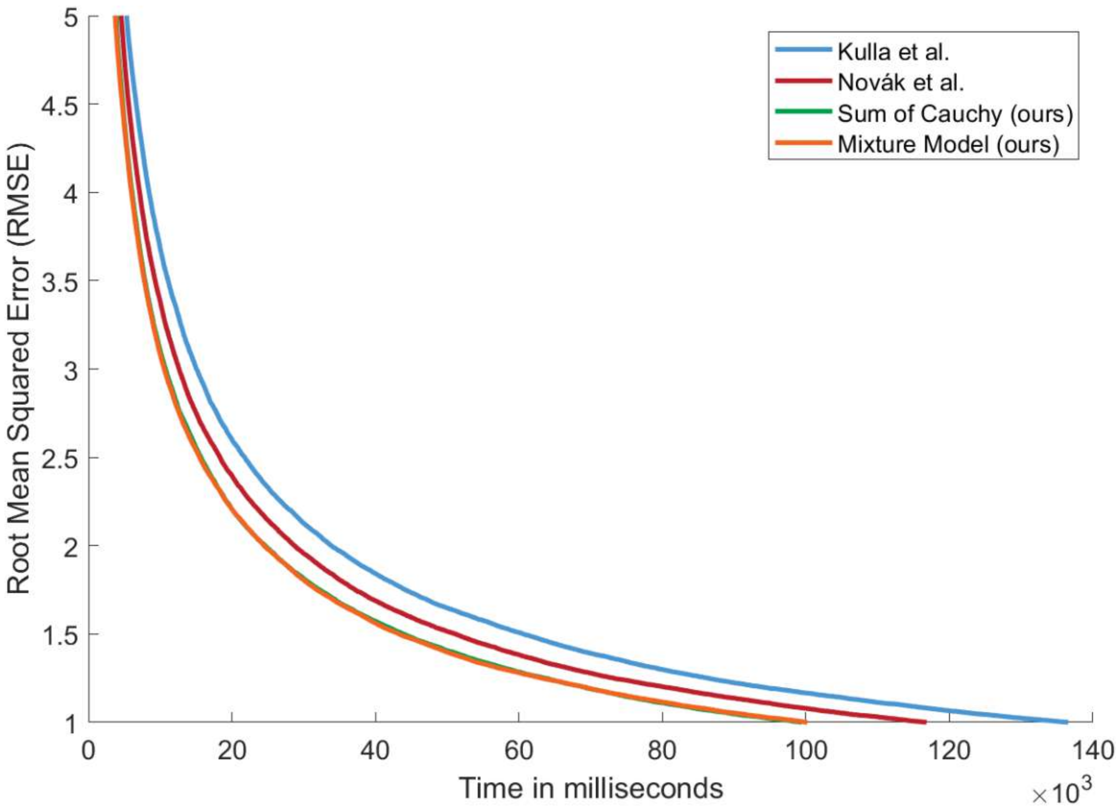


Figure A.6: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.6 for the Henyey-Greenstein phase function.

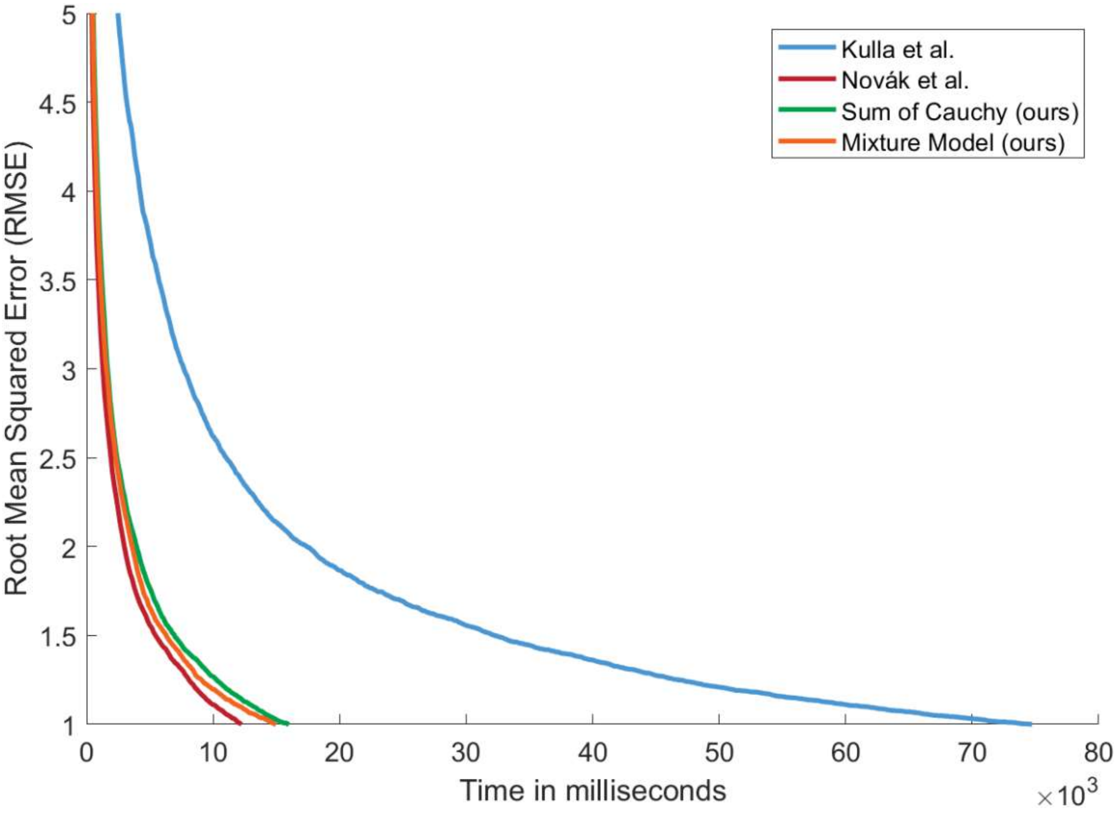


Figure A.7: The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.9 for the Henyey-Greenstein phase function.

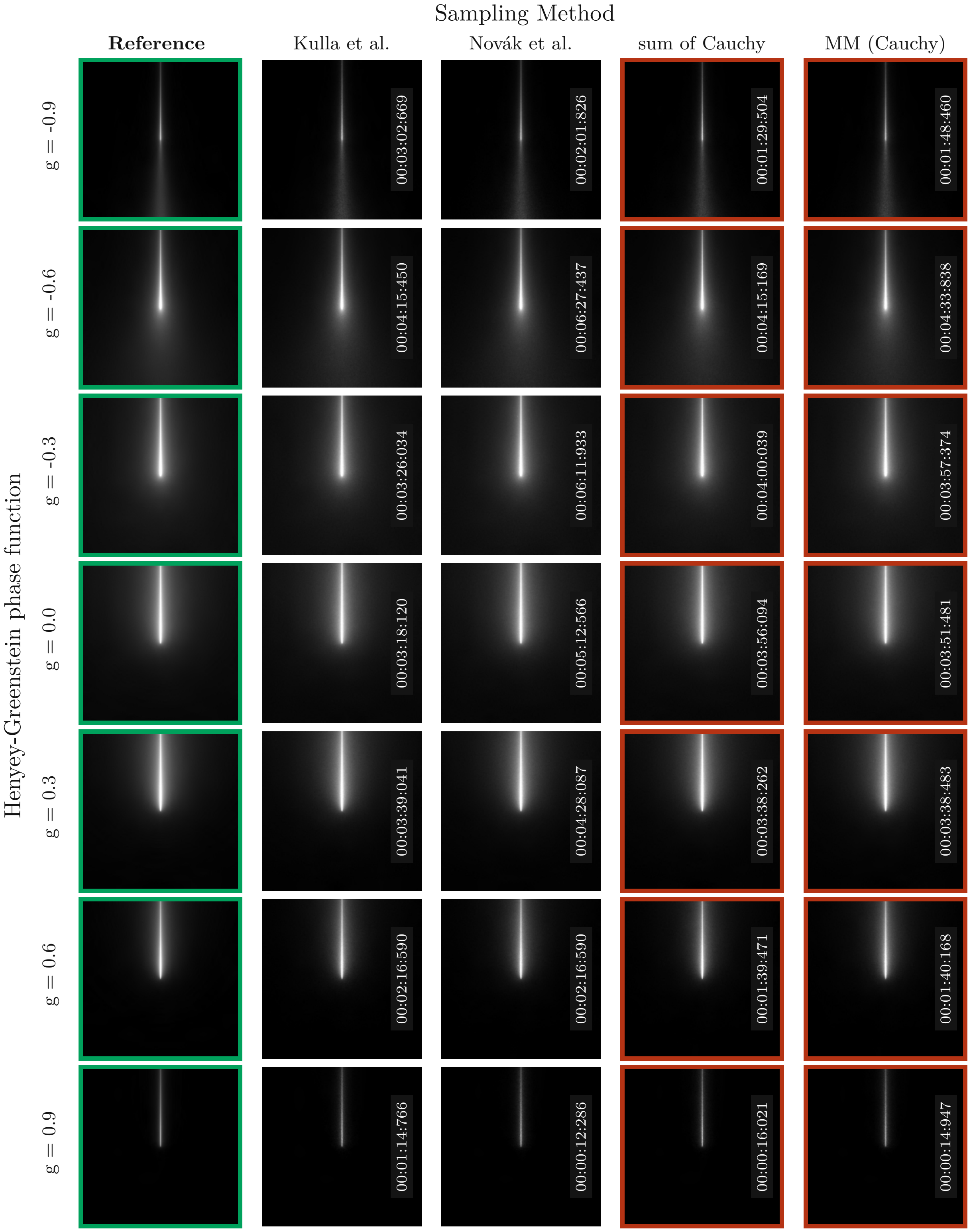


Table A.18: Comparison of the bidirectional light scene until a RMSE of 1 is reached. (Time in hh:mm:ss:ms)

A.3.2 Equal Time

The equal time comparison renders until a time limit is reached. Afterwards, RMSE and PSNR can be calculated from these results to see how far they have converged in that time.

Equal Time - 60 Seconds Table A.20 shows the results of a Cornell box scene when rendered for 60 seconds. The corresponding RMSE and PSNR can be seen in Table A.19.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	2.98	38.65	2.44	40.38	2.09	41.71	2.36	40.65
-0.6	3.52	37.20	4.36	35.34	3.52	37.19	3.67	36.82
-0.3	3.19	38.05	4.25	35.57	3.43	37.41	3.41	37.48
0.0	3.15	38.17	3.93	36.24	3.39	37.52	3.38	37.54
0.3	3.28	37.80	3.60	36.99	3.28	37.82	3.27	37.84
0.6	2.63	39.73	2.44	40.39	2.26	41.05	2.25	41.08
0.9	1.97	42.25	0.93	48.80	0.90	49.02	0.87	49.33

Table A.19: Comparison of the RMSE and PSNR of Table A.20.

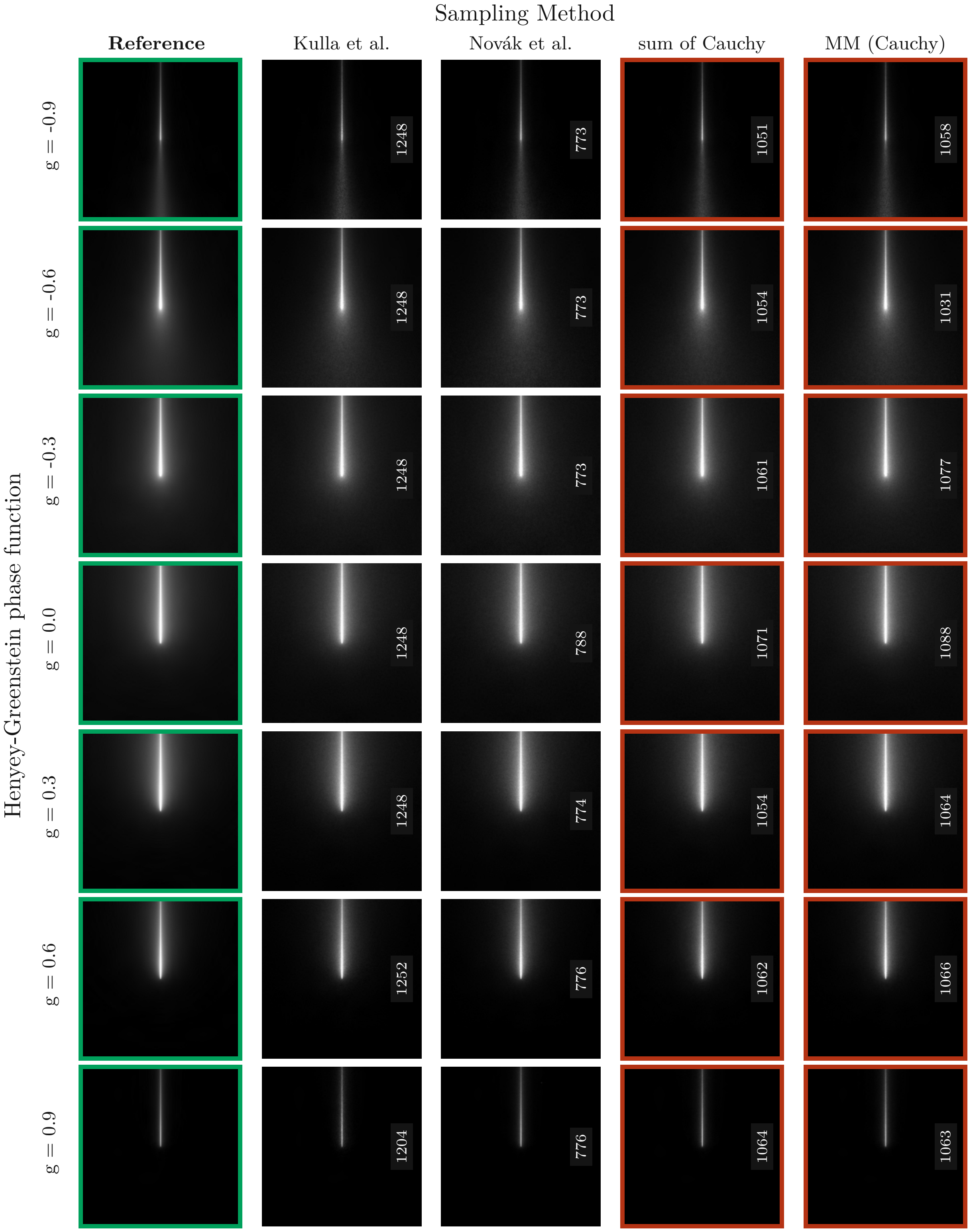


Table A.20: Comparison of the bidirectional light scene that was rendered for 60 seconds. Numbers of samples is provided in the pictures.

A.3.3 Equal Sample

The equal sample approach evaluates the efficiency of the algorithm on a per sample basis. Therefore it is expected that the original algorithm will outperform our approaches as the time taken per sample is longer. But this test still provides valid insight on how close the results are.

Equal Sample - 1000 Samples Table A.22 shows the results of a Cornell box scene when rendered for 60 seconds. The corresponding RMSE and PSNR can be seen in Table A.21.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	3.30	37.76	2.15	41.48	2.16	41.44	2.41	40.50
-0.6	3.95	36.19	3.86	36.39	3.61	36.98	3.72	36.72
-0.3	3.57	37.08	3.74	36.67	3.52	37.19	3.54	37.14
0.0	3.51	37.21	3.48	37.30	3.52	37.20	3.51	37.22
0.3	3.65	36.87	3.21	37.99	3.37	37.59	3.36	37.60
0.6	2.94	38.75	2.15	41.49	2.34	40.72	2.33	40.76
0.9	2.16	41.46	0.72	51.02	0.93	48.75	0.91	48.92

Table A.21: Comparison of the RMSE and PSNR of Table A.22.

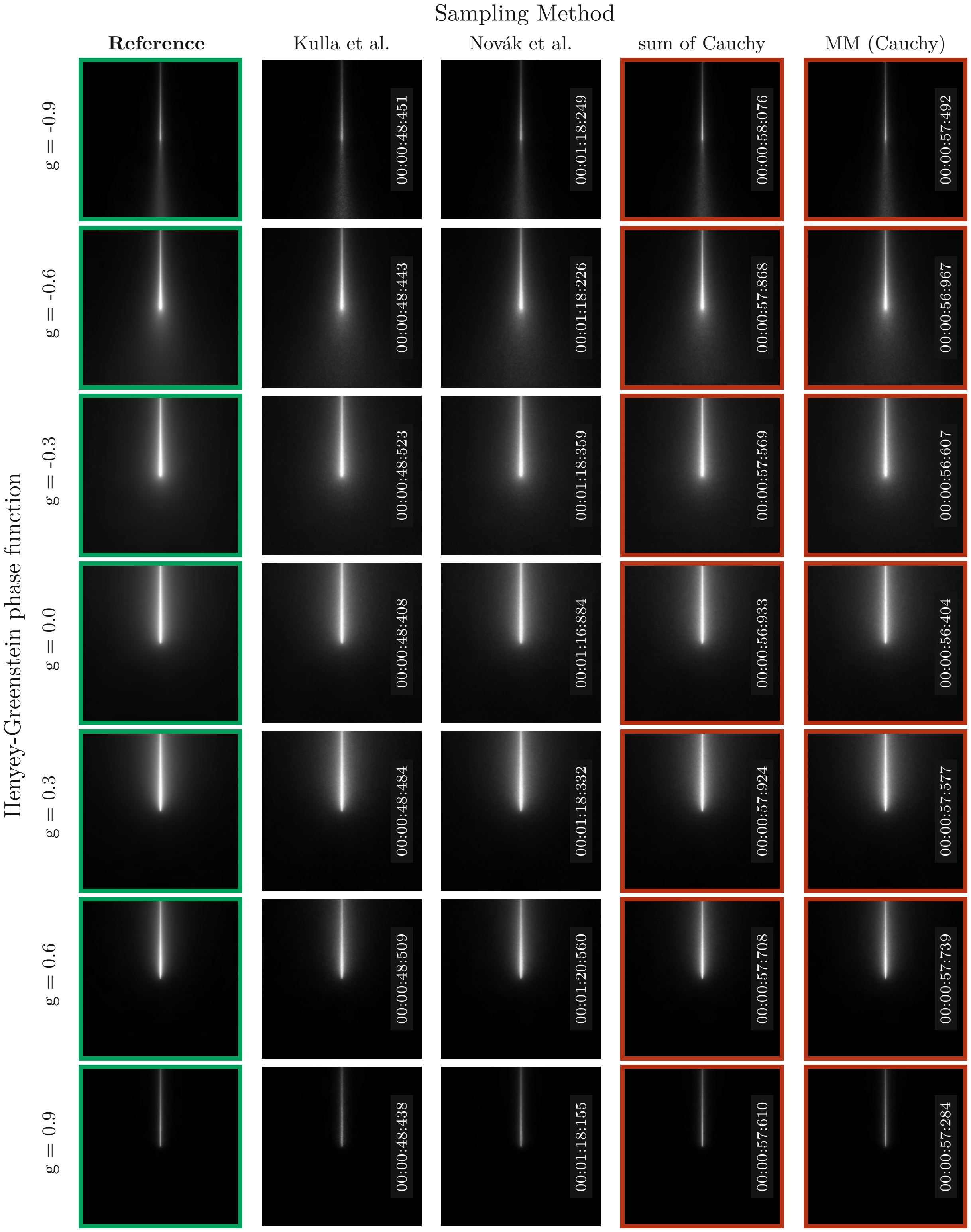


Table A.22: Comparison of the bidirectional light scene until 1000 samples are reached. (Time in hh:mm:ss:ms)

Equal Sample - 10,000 Samples Table A.24 show the results and times of the scene being rendered with 10,000 samples. As can be seen from Table A.23, all results have an extremely low RMSE.

	Kulla et al.		**Novák et al.**		**sum of Cauchy**		**MM (Cauchy)**	
HG	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
-0.9	1.11	47.25	0.77	50.42	0.76	50.51	0.83	49.76
-0.6	1.35	45.49	1.33	45.67	1.26	46.15	1.29	45.89
-0.3	1.24	46.23	1.30	45.88	1.23	46.34	1.24	46.27
0.0	1.22	46.37	1.31	45.79	1.23	46.34	1.23	46.35
0.3	1.25	46.18	1.13	47.05	1.17	46.77	1.18	46.71
0.6	1.02	47.94	0.78	50.34	0.83	49.73	0.82	49.72
0.9	0.71	51.06	0.26	59.67	0.33	57.63	0.32	57.96

Table A.23: Comparison of the RMSE and PSNR of Table A.24.

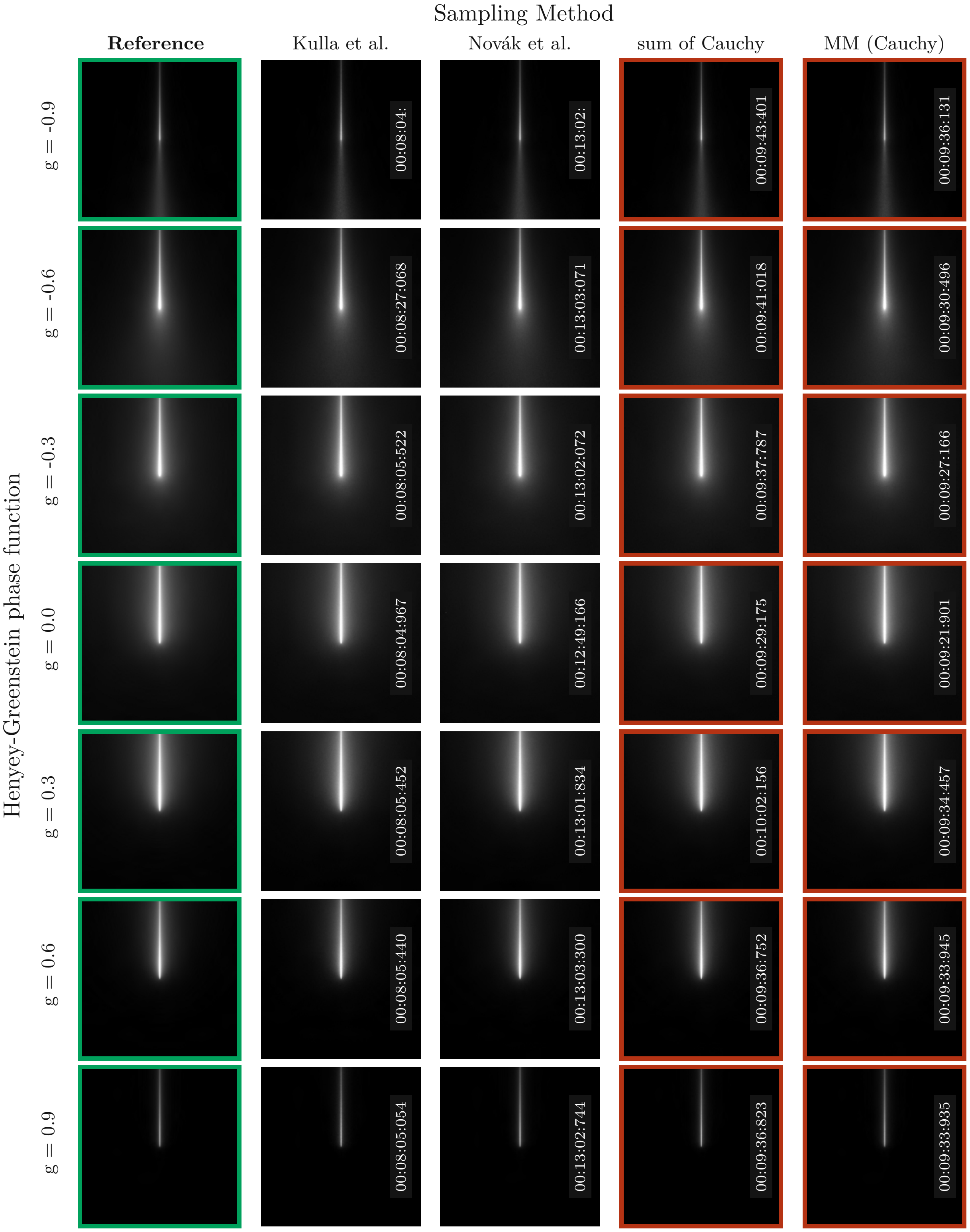


Table A.24: Comparison of the Cornell box scene until 10,000 samples are reached. (Time in hh:mm:ss:ms)

APPENDIX B

Other Approaches

In general, it should be noted that although there we only mention probability distributions being used in the approaches, other functions can be used as well. One of the main advantages of the probability distributions, being normalized, falls out, as most of the distributions are matched to their underlying targets in other intervals. Still, the location and scale parameters of most distributions make the matching easier and that is why they are used in the approaches.

B.1 Product of two distributions

For a long time during the work on this thesis, it was the main goal to find a solution where two functions can be multiplied, and the result can be used as a pdf to approximate the product of the two phase functions. It was believed that the product of two functions will have the closest fit to the target distribution. After some time it was found out that two Gaussian distributions can be multiplied, the result can be integrated and further inverted. These are all steps that are necessary to sample from a distribution. Unfortunately, the results were very lackluster, as the discovery was that the functions could not be fitted close enough to the target and therefore the product of them looked very different. In the case of the Gaussian distribution, there is mainly only one peak at a completely different position, as the Gaussian approaches zero much faster. In some tests with higher anisotropy values, convergence could not be reached, as the results were even worse than a uniform importance sampling function. The implementation of this case can be seen in the GitHub project.

B.2 Product of two Cauchy distributions

Although the product of two Cauchy distributions is much closer to the target result than the product of two Gaussians, this approach had to be discarded early on as it seems to be impossible to sample from this distribution. It is just mentioned here for completeness.

B.3 Other Distributions for the Mixture Model

Besides the Cauchy distribution, it is also possible to use other distributions in the mixture model approach. In this case it was also tested with the Gaussian, Logistic and a variation of the hyperbolic secant distribution with a scale and location factor, which are all available in the GitHub project. Although convergence can be reached with all of these distributions, the Cauchy distribution provides the best results. Still, it is good to see that this approach is very adaptable.

List of Figures

1.1 Different variations of participating media. Source: [NSJ14] (left), [KGH+14] (right) 2

2.1 The four properties of participating media. 8
2.2 The possibilities of scattering in a participating media. The light ray (yellow) is scattered into various directions depending on the media properties. . . 10
2.3 Different types of participating media. Source: [Jar08] 11
2.4 A visual comparison of single and multiple scattering. 11

3.1 To search for a specific number of photons (in this case four) around a point. The sphere is expanded until enough photons are enclosed by it. In the first step (blue, inner sphere) no photons are found within the area. The sphere is therefore expanded (green, outer sphere) to find four photons inside of it. 14
3.2 A comparison of VPM, the BRE and a reference solution that was rendered with VPM and a small step size. The render time is provided in hh:mm:ss. Source: [JZJ08] 15
3.3 A visual comparison of the ideas behind VPM and PB. 16
3.4 A comparison of BRE and the PB, to a reference solution. The render time is provided in hh:mm:ss. Source: [JNSJ11] 16
3.5 A comparison of UPBP, VPM, BRE, PB and BPT. The images for VPM, BRE, PB and BPT are weighted by their contribution to the overall scene. Source: [KGH+14] 17
3.6 A comparison of photon beams, photon planes and photon volumes. All of these images were created within the same render time. Source: [BJ17] . 18
3.7 A scene rendered with VPL. The singularities in the medium can be seen as there are not enough point lights to illuminate the scene evenly. Source: [NNDJ12b] 19
3.8 A visual comparison of the ideas behind VPL and VSL. 20
3.9 A visual comparison of the ideas behind VPL and VRL. 21
3.10 A comparison of VPL, PPB and VRL. All of these images were created within the same render time (600 seconds). Source: [NNDJ12b] 21
3.11 A comparison of VRL and VBL. Both of these images were created within the same render time (186 seconds for volume to volume events, 363 seconds for volume to surface events). Source: [NNDJ12a] 22

4.1 A visual comparison of the ideas behind VPL and VRL. 23
4.2 A visualization of the terms of Equation 4.1. The incoming radiance at the point x in a direction $\vec{\omega}$ is the sum of the radiance from the nearest surface $L_s(x_s, \vec{\omega})$ when going back along the ray and the radiance from the medium $L_m(x, \vec{\omega})$. The radiance from the surface is attenuated due to the transmittance $T_r(s)$. For the radiance from the medium, only the gathered radiance from one light path segment is visualized in this Figure, although all light path segments need to be evaluated. 25
4.3 A visualization of the terms of Equation 4.3. The green ray is the camera ray that has the length s. The orange ray is the light ray with the length t. In this case, the contribution of the point light at the position v on the light ray is evaluated for the position u on the camera ray. 26
4.4 A visualization of the importance sampling by Kulla et al. [KF11]. The pdf of this technique is proportional to the inverse squared distance. 27
4.5 A comparison between choosing a uniformly distributed sample on the light ray and an equi-angular sample on the camera ray (left), to choosing these samples with a joint distribution (right). Source: [NNDJ12b] 28
4.6 A comparison between the isotropic approach (left), to the anisotropic approach (right). Source: [NNDJ12b] 30
4.7 An illustration of the spherical domain around v_i as well as the arc that is generated by the projection of the camera ray (green) onto the sphere (top). Furthermore, the projection of the light ray into the angular domain can be seen in the same figure. The figures on the bottom show the isotropic (left) and anisotropic (right) cases. In the isotropic case, the product of the phase functions at θ_{peak} is equal to the result at all other angles. 31
4.8 Two examples of the product of the phase functions (blue). The green line is the phase function along the camera ray and the orange line the phase function along the light ray. 32

5.1 The matched Cauchy distributions for the light (dashed yellow) and camera (dashed green) compared to the Henyey-Greenstein phase function for the light (solid yellow) and camera (solid green) for different anisotropy values of the Henyey-Greenstein phase function. 39
5.2 The pdf of our reconstruction (dashed blue) compared to the actual normalized target pdf (solid blue) for different anisotropy values of the Henyey-Greenstein phase function. 40
5.3 The matched and normalized Cauchy distributions for the light (dashed yellow) and camera (dashed green) compared to the Henyey-Greenstein phase function for the light (solid yellow) and camera (solid green) for different anisotropy values of the Henyey-Greenstein phase function. Compared to Figure 5.1 the Cauchy distributions are already normalized as this is a requirement for their use in the mixture model. 42

5.4 The pdf of our reconstruction (dashed blue) compared to the actual normalized target pdf (solid blue) for different anisotropy values of the Henyey-Greenstein phase function. 43

7.1 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.9 for the Henyey-Greenstein phase function. 55

7.2 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of -0.9 for the Henyey-Greenstein phase function. 56

7.3 Representative examples of the equal time case where other approaches (Novák et al., Kulla et al.) were faster than our approaches. On the top row, the results of the Cornell box scene that were rendered for 60 seconds with high forward scattering media ($g = 0.9$) can be seen. The approach by Novák et al. achieved a RMSE of 1.04 which was the best result, compared to all other approaches. In the bottom row, the results of the Cornell box scene that were rendered for 60 seconds with isotropic media ($g = 0.0$) can be seen. The approach by Kulla et al. achieved a RMSE of 0.94 which was the best result, compared to all other approaches. The achieved RMSE for each approach is displayed within the corresponding image. 57

7.4 Representative examples of the equal time case where our approaches were faster than the other approaches (Novák et al., Kulla et al.). On the top row, the results of the bidirectional light scene that were rendered for 60 seconds with high backward scattering media ($g = -0.9$) can be seen. The Sum of Cauchy distributions approach achieved a RMSE of 2.09 which was the best result, compared to all other approaches. In the bottom row, the results of the bidirectional light scene that were rendered for 60 seconds with higher forward scattering media ($g = 0.6$) can be seen. The mixture model approach with Cauchy distributions achieved a RMSE of 2.25 which was the best result compared to all other approaches. The achieved RMSE for each approach is displayed within the corresponding image. 58

A.1 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of -0.9 for the Henyey-Greenstein phase function. 80

A.2 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of -0.6 for the Henyey-Greenstein phase function. 81

A.3 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of -0.3 for the Henyey-Greenstein phase function. 82
A.4 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.0 for the Henyey-Greenstein phase function. 83
A.5 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.3 for the Henyey-Greenstein phase function. 84
A.6 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.6 for the Henyey-Greenstein phase function. 85
A.7 The progression of the RMSE values plotted on the y-axis for the corresponding time in milliseconds on the x-axis. This plot contains the data from the bidirectional light scene with a g-value of 0.9 for the Henyey-Greenstein phase function. 86

List of Tables

4.1 The terms of the equation for the surface radiance. 24
4.2 The terms of the equation for the radiance from the medium. 25

6.1 The structure of the LinearFunction data type. 47

A.1 The light and media properties of the Cornell box scene. 63
A.2 Comparison of the Cornell box scene until a RMSE of 1 is reached. (Time in hh:mm:ss:ms) 64
A.3 Comparison of the RMSE and PSNR of Table A.4. 65
A.4 Comparison of the Cornell box scene that was rendered for 60 seconds. Numbers of samples is provided in the pictures. 66
A.5 Comparison of the RMSE and PSNR of Table A.6. 67
A.6 Comparison of the Cornell box scene until 1000 samples are reached. (Time in hh:mm:ss:ms) 68
A.7 Comparison of the RMSE and PSNR of Table A.8. 69
A.8 Comparison of the Cornell box scene until 10,000 samples are reached. (Time in hh:mm:ss:ms) 70
A.9 The light and media properties of the Unidirectional scene. 71
A.10 Comparison of the unidirectional light scene until a RMSE of 1 is reached. If this RMSE was not reached in 100,000 samples, a red cross is displayed. (Time in hh:mm:ss:ms) 72
A.11 Comparison of the RMSE and PSNR of Table A.12. 73
A.12 Comparison of the unidirectional light scene that was rendered for 60 seconds. Numbers of samples is provided in the pictures. 74
A.13 Comparison of the RMSE and PSNR of Table A.14. 75
A.14 Comparison of the Cornell box scene until 1000 samples are reached. (Time in hh:mm:ss:ms) 76
A.15 Comparison of the RMSE and PSNR of Table A.16. 77
A.16 Comparison of the unidirectional light scene until 10,000 samples are reached. (Time in hh:mm:ss:ms) 78
A.17 The light and media properties of the bidirectional light scene. 79
A.18 Comparison of the bidirectional light scene until a RMSE of 1 is reached. (Time in hh:mm:ss:ms) 87
A.19 Comparison of the RMSE and PSNR of Table A.20. 88

A.20 Comparison of the bidirectional light scene that was rendered for 60 seconds. Numbers of samples is provided in the pictures. 89
A.21 Comparison of the RMSE and PSNR of Table A.22. 90
A.22 Comparison of the bidirectional light scene until 1000 samples are reached. (Time in hh:mm:ss:ms) 91
A.23 Comparison of the RMSE and PSNR of Table A.24. 92
A.24 Comparison of the Cornell box scene until 10,000 samples are reached. (Time in hh:mm:ss:ms) 93

List of Algorithms

6.1 IsotropicSampling($\bar{s}, p, \mathrm{cp}$) 45
6.2 AnisotropicSampling($\bar{s}, p, \vec{d}, \mathrm{cp}$) 47
6.3 CreateSamplesOnArc($\theta_1, \theta_M, \vec{a}, \vec{b}, \vec{c}, \vec{d_1}, \vec{d_2}$) 48
6.4 ConstructPdfCdf() 49
6.5 SumOfCauchyDistribution($\bar{s}, p, \vec{d}, \mathrm{cp}$) 50
6.6 MixtureModelDistribution($\bar{s}, p, \vec{d}, \mathrm{cp}$) 51

Acronyms

BPT Bidirectional Path Tracing. 17, 97

BRDF bidirectional reflectance distribution function. 7, 20, 24

BRE Beam Radiance Estimate. 2, 14–17, 20, 97

cdf cumulative density function. 29, 33, 34, 38–40, 42, 43, 46, 50, 51

HG Henyey-Greenstein phase function. 65, 67, 69, 73, 75, 77, 88, 90, 92

PB Photon Beams. 15–18, 20, 97

pdf probability density function. 3, 8, 27–34, 36, 38–40, 42, 43, 46, 50, 95, 98, 99

PM Photon Mapping. 8, 13, 14, 16, 18, 20

PP Photon Planes. 18

PPB Progressive Photon Beams. 16, 21, 97

PPM Progressive Photon Mapping. 13, 16

PSNR peak signal-to-noise ratio. 54, 55, 62, 65, 67, 69, 73, 75, 77, 88, 90, 92, 101, 102

PV Photon Volumes. 18

RMSE root-mean-square error. 53–58, 62–65, 67, 69, 71–73, 75, 77, 79–88, 90, 92, 99–102

RRT Recursive Ray Tracing. 8

RT Ray Tracing. 13

SPPM Stochastic Progressive Photon Mapping. 13

UPBP Unifying Points, Beams and Paths. 16–18, 97

VBL Virtual Beam Lights. 21, 22, 97

VL Virtual Lights. 18–20, 23

VPL Virtual Point Lights. 18–21, 23, 29, 97, 98

VPM Volumetric Photon Mapping. 2, 14–17, 97

VPT Volumetric Path Tracing. 2

VRL Virtual Ray Lights. 2–5, 8, 13, 20–23, 37, 45, 54, 59, 62, 71, 79, 97, 98

VSL Virtual Spherical Lights. 19–21, 97

Bibliography

[BJ17] Benedikt Bitterli and Wojciech Jarosz. Beyond points and beams: Higher-dimensional photon samples for volumetric light transport. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[Buc95] Anthony Bucholtz. Rayleigh-scattering calculations for the terrestrial atmosphere. *Applied Optics*, 34(15):2765–2773, 1995.

[FW59] Kenneth W Ford and John A Wheeler. Semiclassical description of scattering. *Annals of Physics*, 7(3):259–286, 1959.

[Gla89] Andrew S Glassner. *An introduction to ray tracing*. Elsevier, 1989.

[HG41] Louis G Henyey and Jesse L Greenstein. Diffuse radiation in the galaxy. *The Astrophysical Journal*, 93:70–83, 1941.

[HJ09] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Trans. Graph.*, 28(5):141:1–141:8, December 2009.

[HKWB09] Miloš Hašan, Jaroslav Křivánek, Bruce Walter, and Kavita Bala. Virtual spherical lights for many-light rendering of glossy scenes. In *ACM Transactions on Graphics (TOG)*, volume 28, page 143. ACM, 2009.

[HOJ08] Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Transactions on Graphics (TOG)*, 27(5):130, 2008.

[ICG86] David S Immel, Michael F Cohen, and Donald P Greenberg. A radiosity method for non-diffuse environments. In *ACM SIGGRAPH Computer Graphics*, volume 20, pages 133–142. ACM, 1986.

[Jar08] Wojciech Jarosz. *Efficient Monte Carlo methods for light transport in scattering media*. University Of California, San Diego, 2008.

[JCKS02] Henrik Wann Jensen, Per H Christensen, Toshiaki Kato, and Frank Suykens. A practical guide to global illumination using photon mapping. *SIGGRAPH 2002 Course Notes CD-ROM*, 2002.

[Jen96a] Henrik Wann Jensen. Global illumination using photon maps. *Rendering techniques*, 96:21–30, 1996.

[Jen96b] Henrik Wann Jensen. The photon map in global illumination. *Technical University of Denmark: Department of Planning: Graphical Communication*, 1996.

[Jen01] Henrik Wann Jensen. *Realistic image synthesis using photon mapping*. AK Peters, Ltd., 2001.

[JNSJ11] Wojciech Jarosz, Derek Nowrouzezahrai, Iman Sadeghi, and Henrik Wann Jensen. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics (TOG)*, 30(1):5, 2011.

[JZJ08] Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. The beam radiance estimate for volumetric photon mapping. In *ACM SIGGRAPH 2008 classes*, page 3. ACM, 2008.

[Kaj86] James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.

[Kel97] Alexander Keller. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 49–56. ACM Press/Addison-Wesley Publishing Co., 1997.

[KF11] Christopher D Kulla and Marcos Fajardo. Importance sampling of area lights in participating media. In *SIGGRAPH Talks*, page 55, 2011.

[KGH$^+$14] Jaroslav Křivánek, Iliyan Georgiev, Toshiya Hachisuka, Petr Vévoda, Martin Šik, Derek Nowrouzezahrai, and Wojciech Jarosz. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics (TOG)*, 33(4):103, 2014.

[LW93] Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, pages 145–153, Alvor, Portugal, December 1993.

[Mar15] Steve Marschner. Realistic image synthesis, 2015. http://www.cs.cornell.edu/courses/cs6630/2015fa/index.shtml.

[NNDJ12a] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Progressive virtual beam lights. In *Computer Graphics Forum*, volume 31, pages 1407–1413. Wiley Online Library, 2012.

[NNDJ12b] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics (TOG)*, 31(4):60, 2012.

[Nov14] Jan Novák. *Efficient Many-Light Rendering of Scenes with Participating Media*. PhD thesis, Karlsruhe Institute of Technology", May 2014.

[NSJ14] Jan Novák, Andrew Selle, and Wojciech Jarosz. Residual ratio tracking for estimating attenuation in participating media. *ACM Trans. Graph.*, 33(6):179–1, 2014.

[Opp18] Michael Oppitz. Implementing virtual ray lights for rendering scenes with participating media. *Institute of Computer Graphics and Algorithms, Vienna University of Technology*, 2018.

[PJH16] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[Vév15] Petr Vévoda. Robust light transport simulation in participating media. *Univerzita Karlova, Matematicko-fyzikální fakulta*, 2015.

[VG97] Eric Veach and Leonidas J Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76. ACM Press/Addison-Wesley Publishing Co., 1997.

[Whi79] Turner Whitted. An improved illumination model for shaded display. In *ACM SIGGRAPH Computer Graphics*, volume 13, page 14. ACM, 1979.

[ZF18] Károly Zsolnai-Fehér. Tu wien rendering course, 2018. https://users.cg.tuwien.ac.at/zsolnai/gfx/rendering-course/.