Charles University in Prague

Faculty of Mathematics and Physics

**DOCTORAL THESIS**

RNDr. Jiří Vorba

# Adjoint-Driven Importance Sampling in Light Transport Simulation

Department of Software and Computer Science Education

| | |
|---|---|
| Supervisor: | doc. Ing. Jaroslav Křivánek, Ph.D. |
| Study programme: | Informatics |
| Study branch: | Computer Graphics and Image Analysis |

Prague 2017

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ........ date ............ signature of the author

**Title:** Adjoint-Driven Importance Sampling in Light Transport Simulation
**Author:** RNDr. Jiří Vorba
**Department:** Department of Software and Computer Science Education
**Supervisor:** doc. Ing. Jaroslav Křivánek, Ph.D., Department of Software and Computer Science Education

**Abstract:** Monte Carlo light transport simulation has recently been adopted by the movie industry as a standard tool for producing photo realistic imagery. As the industry pushes current technologies to the very edge of their possibilities, the unprecedented complexity of rendered scenes has underlined a fundamental weakness of MC light transport simulation: slow convergence in the presence of indirect illumination. The culprit of this poor behaviour is that the sampling schemes used in the state-of-the-art MC transport algorithms usually do not adapt to the conditions of rendered scenes. We base our work on the observation that the vast amount of samples needed by these algorithms forms an abundant source of information that can be used to derive superior sampling strategies, tailored for a given scene. In the first part of this thesis, we adapt general machine learning techniques to train directional distributions for biasing scattering directions of camera paths towards incident illumination (radiance). Our approach allows progressive training from a stream of particles while maintaining bounded memory footprint. This progressive nature makes the method robust even in scenarios where we have little information in the early stages of the training due to difficult visibility. The proposed method is not restricted only to path tracing, where paths start at the camera, but can be employed also in light tracing or photon mapping, where paths are emitted from light sources, as well as in combined bidirectional methods.

In the second part of this thesis we revisit Russian roulette and splitting, two variance reduction techniques that have been used in computer graphics for more than 25 years. So far, however, the path termination (Russian roulette) and splitting rates have been based only on local material properties in the scene which can result in inefficient simulation in the presence of indirect illumination. In contrast, we base the termination and splitting rates on a pre-computed approximation of the adjoint quantity (i.e. radiance in the case of path tracing) which yields superior results to previous approaches. To increase robustness of our method, we adopt the so called weight window, a standard technique in neutron transport simulations. Both methods, that is the biasing of scattering directions introduced in the first part of the thesis and the adjoint-driven Russian roulette and splitting, are based on the prior estimate of the adjoint quantity. Nevertheless, they constitute two complementary importance sampling strategies of transported light and as we show, their combination yields superior results to each strategy alone. As one of our contributions, we present a theoretical analysis that provides insights into the importance sampling properties of our adjoint-driven Russian roulette and splitting, and also explains the synergic behaviour of the two strategies.

**Keywords:** computer graphics, light transport simulation, rendering, Monte Carlo, zero-variance sampling

**Název:** Vzorkování důležitosti v simulaci transportu světla založené na adjungovaném řešení
**Autor:** RNDr. Jiří Vorba
**Katedra:** Katedra softwaru a výuky informatiky
**Vedoucí:** doc. Ing. Jaroslav Křivánek, Ph.D., Katedra softwaru a výuky informatiky

**Abstrakt:** Monte Carlo (MC) simulace transportu světla byla nedávno přijata filmovým průmyslem jako standardní nástroj pro tvorbu fotorealistických efektů. Jelikož filmový průmysl posunul současné technologie na samou hranici jejich možností, nevídaná složitost zobrazovaných scén odhalila zásadní nedostatek MC simulace: pomalou konvergenci transportu nepřímého osvětlení. Hlavním viníkem této pomalé konvergence jsou vzorkovací schémata, která se obvykle nepřizpůsobují zobrazované scéně. My zakládáme naší práci na pozorovaní, že velké množství vzorků, které současné algoritmy transportu světla vyžadují, představují bohatý zdroj informací. Tento zdroj využíváme k vytvoření vzorkovacích schémat pro zobrazovanou scénu. V první části práce využíváme metod strojového učení, které jsme uzpůsobily pro trénování směrových distribucí. Tyto distribuce využíváme během rozptylu k ovlivnění vzorkování směru cest od kamery tak, aby se vzorkovaly směry úměrně příchozímu osvětlení (radiance). Náš přístup umožňuje postupné trénování z proudu částic a zachovává tak omezené paměťové nároky. Tento on-line přístup zaručuje, že naše metoda funguje i v situacích, kdy máme k dispozici pouze malé množství informace v raných fázích trénovacího procesu kvůli složité viditelnosti ve scéně. Navržená metoda se neomezuje pouze na trasování cest od kamery (path tracing), ale jak ukazujeme, lze ji aplikovat také na trasování cest od světel (light tracing) nebo v kombinovaných dvousměrových algoritmech.

Ve druhé části této práce se vracíme k ruské ruletě a větvení cest, dvěma technikám snižování variance, které se používají v počítačové grafice již více než 25 let. Nicméně dosud se míra ukončování (ruská ruleta) a větvení zakládala pouze na lokálních vlastnostech materiálů, což může vést k neefektivní simulaci transportu nepřímého osvětlení. My naproti tomu zakládáme míru ukončování a větvení cest na předpočítané aproximaci adjungované veličiny (t.j. radiance v případě trasování cest od kamery), což vede k lepším výsledkům ve srovnání s předchozími přístupy. Abychom zvýšili robustnost naší metody, převzali jsme takzvané váhové okno, standardní techniku v simulacích transportu neutronů. Obě metody, jak ovlivnění vzorkování směru během rozptylu popsané v první části práce tak ruská ruleta a větvení založené na adjungovaném řešení, jsou založeny na předchozím odhadu adjungované veličiny. Nicméně tvoří doplňující se strategie vzorkování důležitosti přenášeného světla a jak ukazujeme, jejich kombinace vede k lepším výsledkům oproti použití každé z těchto strategií samostatně. Jako jeden z našich příspěvků prezentujeme teoretickou analýzu, která vysvětluje vlastnosti vzorkování důležitosti naší ruské rulety a větvení cest založené na adjungovaném řešení a také synergii obou výše zmíněných strategií.

**Klíčová slova:** počítačová grafika, simulace transportu světla, rendering, Monte Carlo, vzorkování s nulovou variancí

# Acknowledgements

I would like to express my sincere gratitude to my advisor, Jaroslav Křivánek, who guided me through my doctoral studies patiently and encouraged me to accept great challenges that I would not have accepted if it had not been for him. I also thank him for the substantial amount of time he has invested in me, for teaching me the way of writing scientific publications, and for considerable help with all the published manuscripts. I do believe that I will benefit from the experience I have gained over my doctoral studies for the rest of my professional life.

I also would like to thank all my colleagues I had the pleasure to work with. Namely, I thank Ondřej Karlík, Martin Šik and Tobias Ritschel for their work on the research published in this thesis. Further, I thank all my colleagues from the computer graphics group at the Faculty of Mathematics and Physics and the faculty staff for creating very friendly and pleasant working environment.

This research has been supported by Charles University Grant Agency, projects GA UK no. 580612 and no. 340915, by the grants SVV-2014-260103 and SVV-2016-260332, by the Czech Science Foundation grants 13-26189S and 16-18964S, and partially by Weta Digital.

Last but not least, I would like to thank my parents, Jiří and Mária, for their support, love, and care over the years. I dedicate this work to my wife Lenka to express my deepest appreciation for her support, patience, and for making me happy.

*Leničce*

# Contents

# Chapter 1

# Introduction

The goal of this dissertation is to develop a light transport algorithm that can efficiently compute transport in scenes with *complex and difficult visibility* (see Fig. 1.1). Despite recent advances, robust and efficient light transport simulation is still a challenging open issue. Current state-of-the-art algorithms are able to cope with complex materials, various geometric settings, lighting effects such as caustics, subsurface scattering and transport in participating media. However, they are all highly inefficient in scenes where light reaches the camera only after several bounces.

Most existing unidirectional and bidirectional methods rely on an incremental, local construction of transport sub-paths, which is oblivious to the global distribution of radiance or importance. As a result, the probability of obtaining a non-zero contribution upon sub-path connection in highly occluded scenes is low and the rendering times can reach as much as thousands of CPU hours which is unacceptable in practice.

While Metropolis light transport and related methods [Veach and Guibas, 1997, Kelemen et al., 2002, Cline et al., 2005] strive for importance sampling on the entire path space, they suffer from sample correlation and are often outperformed by the classic Monte Carlo approaches. We aim for an *unbiased solution* that retains the qualities of current MC algorithms without resorting to a compromise between the quality of computed images and efficiency.

In the rest of this chapter, we introduce light transport applications and delimit the impact of our research, we outline our approach to addressing the problem of inefficient simulations due to occlusions while presenting the structure of the dissertation, and finally, we list our contributions and our original publications.

## 1.1 Light Transport

Simulation of light transport in virtual scenes that comprise description of geometry, materials, lights and cameras, allows us to achieve highly photo-realistic images. Such level of realism is especially required in architectural visualisation, product design and in the movie industry where accurate simulation is essential for seamless and believable composition of real footage with artificial scene models and CG characters.

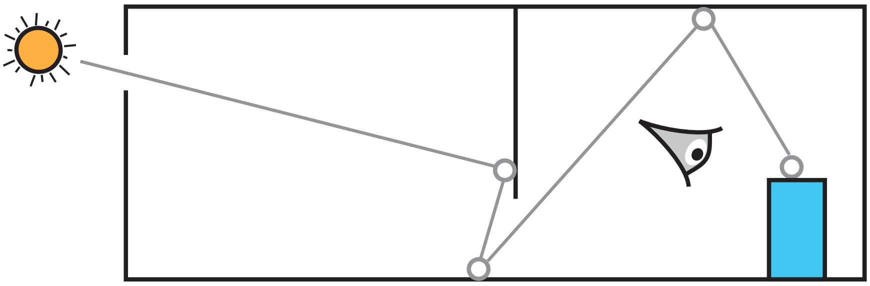


Figure 1.1: In scenes with complex and difficult visibility, light reaches the camera only after several bounces (complex) and it usually flows through relatively small openings (difficult). This type of transport is especially difficult to sample which makes state-of-the-art algorithms highly inefficient.

Recently, Monte Carlo light transport simulation has been adopted by the movie industry as a standard tool for producing photo-realistic imagery [Seymour, 2014a]. Until then, computing *global illumination* (i.e. full light transport) in movies was traditionally achieved by hybrid approaches that split the calculation of direct and indirect illumination. While direct illumination was rendered through the Reyes [Cook et al., 1987] scan-line algorithm with ray-traced reflections, soft indirect illumination was approximated using various approaches (distribution ray-tracing and irradiance caching, point-based global illumination) [Christensen and Jarosz, 2016]. Nowadays, *Monte Carlo simulation* computing the full light transport has superseded these hybrid rasterization approaches [Keller et al., 2015, Christensen and Jarosz, 2016]. This "revolution" [Keller et al., 2015] was possible thanks to improvements in sampling and transport algorithms in the last two decades and also thanks to a steady increase in computational power and memory capacity. Remarkably, such a simulation has been adopted even for the production of fully animated films as it notably simplifies the production pipelines and saves artists' time previously spent on lighting setup [Christensen and Jarosz, 2016].

Because of the widespread acceptance of Monte Carlo transport simulation by the industry, we believe that the results of this dissertation will find use also outside of academia.

## 1.2 Our Approach

Current MC transport algorithms exhibit a poor convergence in scenes with complex visibility for two main reasons. First, they do not utilize any global information about the given scene to adapt their sampling schemes so that highly contributing paths are discovered at a higher rate. Instead, they repeatedly sample paths that do not connect the camera to light sources. This is due to the fact that all these methods (both unidirectional and bidirectional) sample paths from the camera (or sources) incrementally while importance sampling only material properties described by bidirectional scattering distribution function (BSDF). Second, Russian roulette (RR) that is a common part of all MC path-sampling based algorithms, aims to keep the rate of sampled paths with a potentially low contribution low. Unfortunately, the contribution is estimated based on the local material properties and RR typically terminates paths that would contribute significantly to the solution if they survived over a few more bounces.

In Chapter 3, we tackle the former problem by learning distributions of incident illumination (in case of e.g. path tracing) that we use to improve importance sampling at each scattering event. We represent the distributions by mixtures of Gaussians that we train from a continuous stream of particles that represent an adjoint transport solution. We train the distributions in a pre-processing step prior to rendering and we keep them cached over the scene surfaces. The key feature of our approach is an *on-line learning procedure* ensuring that our method works also in difficult scenarios where we have only little information about the illumination at the early stages of training.

In Chapter 4, we address the latter problem by introducing an adjoint-driven method based on the so called *weight window* technique that we adopt from neutron transport. The method combines Russian roulette with splitting (branching) of paths to achieve optimal sampling rates in the space of paths. By leveraging an approximation of the adjoint transport solution, we control termination and splitting rates according to the expected contribution of the path to the estimated pixel. As a consequence, we are able to save computational resources by terminating paths with low expected contribution while reducing variance by keeping or even splitting paths whose expected contribution is high.

Further in Chapter 4, we provide a theoretical analysis of adjoint-driven Russian roulette that explains its importance sampling properties by direct juxtaposition to zero-variance sampling schemes. The same analysis also implies that the direction sampling method that we describe in Chapter 3 and adjoint-driven Russian roulette and splitting (ADRRS) that we develop in Chapter 4 are orthogonal and in fact work in synergy. As a direct consequence, we design an algorithm that combines both ADRRS and path guiding (based on the on-line learning of Gaussian mixtures) and in complex scenes, it achieves results that are superior to the results of each of the individual methods.

**Discussion.** Our work is based on leveraging approximation of the adjoint solution of the computed transport equation for importance sampling. Remarkably, we are able to use a relatively coarse approximation to make path sampling more efficient by orders of magnitude in scenes with complex and difficult visibility even if we account for the pre-processing time.

It is indeed remarkable, because computing the precise adjoint solution is a far more difficult problem than computing the image itself. The reason is that to compute the image we only need to know a projection of the light field into the image while for perfect importance sampling, we would need to know the adjoint solution in the whole scene. If we knew the adjoint solution precisely, we could compute a noiseless image with only one sample per pixel. This suggests that, given a desired error (noise level) of the computed image, there is a sweet spot in the precision of the pre-computed approximation for which the total computation time (preprocessing and rendering) is optimal.

We limit our focus to light transport only between surfaces and suppose that radiance is constant along lines between any two surface points. We believe that our approach can be extended to participating media, however, this extension is left for future work. At the moment, the efficient rendering of participating media is still a subject of active research and the extension of our approach might significantly contribute to the solution of this problem.

## 1.3 List of Original Contributions

Our main contribution is a light transport simulation algorithm that, unlike previous works, can efficiently handle scenes with *complex and difficult visibility* without compromising the image quality. Here, we list our partial contributions in more detail.

**Introduction of machine learning to image synthesis.** To our best knowledge, our on-line learning of parametric mixture models is the first attempt to tackle importance sampling in the light transport simulation by a machine learning technique. Furthermore, we formulate the on-line learning procedure in such a way that we can handle particles with *highly varying weights*.

**Compact and flexible model of importance function.** There are many prior works on steering sampling of light paths (guiding) towards directions of strong incident illumination. However, we show that parametric mixtures (namely mixtures of Gaussians) are a superior model in terms of the ability to represent both high and low-frequency signals while using a reasonable amount of memory. Our model is even capable of targeting refracted sun light and results in better variance reduction than previous methods.

**Demonstration of the potential of guided unidirectional methods.** We experimentally show that guided unidirectional algorithms (path tracing, light tracing) have a potential to match their guided bidirectional counterparts (bidirectional path tracing, vertex connection and merging/unified path space) which are substantially more difficult to implement and maintain in a full-feature production renderer.

**Revisiting Russian roulette and splitting.** Since Russian roulette and splitting have been used in a more or less unchanged form for the last 25 years or so, we believe that revisiting these variance reduction techniques is our valuable contribution. We propose adjoint-driven Russian roulette and splitting where paths are terminated or split according to their expected contribution to the image.

**Theoretical analysis of adjoint-driven Russian roulette and splitting.** We study relation of our ADRRS to zero-variance schemes and rigorously explain its importance sampling properties.

**Combined algorithm.** Our analysis of ADRRS shows that this method is orthogonal to path guiding. We verify this result by combining path guiding with ADRRS to show that the combined unidirectional algorithm surpasses efficiency of both methods alone.

## 1.4 Publications

The content of this dissertation is based on the following publications:

- Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (SIGGRAPH 2014)*, 33(4), July 2014.

- Jiří Vorba and Jaroslav Křivánek. Adjoint-driven russian roulette and splitting in light transport simulation. *ACM Transactions on Graphics (SIGGRAPH 2016)*, 35(4), July 2016.

The research presented in the first one is the subject of Chapter 3 while the second one is the subject of Chapter 4.

# Chapter 2

# Light Transport

In this chapter, we present a mathematical framework that allows us to formulate our adjoint-driven importance sampling method. Namely, we review *rendering and visual importance transport equations* and point out their *adjoint* relationship. Further, we describe commonly used MC transport algorithms and discuss the way they benefit from our path-guiding method developed in Chapter 3 and adjoint-driven Russian roulette described in Chapter 4. We also present *Russian roulette and splitting* that are common variance reduction techniques used along MC transport algorithms. Finally, we review a *zero-variance random walk*, a highly useful tool for studying properties of importance sampling methods that we use in Chapter 4 to analyze a combined algorithm comprising our adjoint-driven Russian roulette and splitting, and path guiding.

## 2.1 Particle Transport

The origins of simulating the transport of neutral particles (e.g. neutrons or photons) by Monte Carlo (MC) processes goes back to the 1940s. The idea of using MC stems from the probabilistic nature of the particles' behavior [Spanier and Gelbard, 1969]. In the real world, a particle has a certain probability of being emitted in a given time interval and its further fate is also governed by probabilistic events: collisions, absorption and scattering. If a simulated particle follows the events according to the precise physical probabilities, the MC process is said to be *analog*. In such a simulation, all particles have equal, unit statistical *weight*. In order to improve computation efficiency, *non-analog simulations* can be designed by altering the probabilities of various events in the simulation. The particle weights are then modified upon each event so that the simulation remains unbiased [Lux and Koblinger, 1991].

The close relation between MC light transport simulation in computer graphics and MC processes simulating the transport of neutral particles has been pointed out by several authors [Křivánek and d'Eon, 2014, Christensen, 2003, Veach, 1997, Arvo and Kirk, 1990]. While light transport is described by the *rendering equation* [Kajiya, 1986], particle transport in physics is governed by the linear Boltzmann equation. The similarities between the two were first pointed out by Arvo and Kirk [1990], which allowed them to adopt useful techniques, such as *Russian roulette (RR) and splitting* [Kahn, 1956, Kahn and Harris, 1951] in computer graphics.

## 2.2 Transport Equations

Light transport in a scene without participating media is described by the *rendering equation* [Kajiya, 1986, Dutré et al., 2006]:

$$L_\mathrm{o}(\mathbf{y},\omega_\mathrm{o}) = L_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{o}) + \underbrace{\int_\Omega L_\mathrm{i}(\mathbf{y},\omega_\mathrm{i}) f_\mathrm{s}(\mathbf{y},\omega_\mathrm{i}{\to}\omega_\mathrm{o}) |\cos\theta_\mathrm{i}| \,\mathrm{d}\omega_\mathrm{i}}_{L_\mathrm{o}^\mathrm{r}(\mathbf{y},\omega_\mathrm{o})} . \tag{2.1}$$

Here $L_\mathrm{o}(\mathbf{y},\omega_\mathrm{o})$ and $L_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{o})$ are, respectively, the total and the self-emitted outgoing radiance from a surface point $\mathbf{y}$ in a direction $\omega_\mathrm{o}$, $f_\mathrm{s}$ denotes the bidirectional scattering distribution function (BSDF), $\theta_\mathrm{i}$ is the angle between the surface normal at $\mathbf{y}$ and an incident direction $\omega_\mathrm{i}$, and $\Omega$ is the unit sphere. We use the arrow notation in $f_\mathrm{s}$ to mark the direction of light flow. The incident radiance $L_\mathrm{i}(\mathbf{y},\omega_\mathrm{i})$ at the point $\mathbf{y}$ visible from a point $\mathbf{x}$ in the direction $\omega_\mathrm{i}$ is equal to the outgoing radiance $L_\mathrm{o}(\mathbf{x},-\omega_\mathrm{i})$. $L_\mathrm{o}^\mathrm{r}$ denotes the part of the outgoing radiance that is only due to surface reflection at $\mathbf{y}$.

In a similar way, we can describe transport of *visual importance* $W$ that is governed by the following transport equation [Spanier and Gelbard, 1969, Veach, 1997, Sec. 3.7]:

$$W_\mathrm{o}(\mathbf{y},\omega_\mathrm{o}) = W_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{o}) + \underbrace{\int_\Omega W_\mathrm{i}(\mathbf{y},\omega_\mathrm{i}) f_\mathrm{s}(\mathbf{y},\omega_\mathrm{o}{\to}\omega_\mathrm{i}) |\cos\theta_\mathrm{i}| \,\mathrm{d}\omega_\mathrm{i}}_{W_\mathrm{o}^\mathrm{r}(\mathbf{y},\omega_\mathrm{o})} . \tag{2.2}$$

Here, $W_\mathrm{o}^\mathrm{e}$ is self-emitted visual importance, $W_\mathrm{i}$ is incident visual importance and $W_\mathrm{o}^\mathrm{r}$ is reflected visual importance.

The visual importance is only a mathematical construct which is not based on any physical concept. However, it is *adjoint* (see Sec. 2.4) to the rendering equation and its solution indicates how important are various parts of the scene with respect to the rendered image [Christensen, 2003].

Let us emphasize that, throughout this thesis, we use a convention, depicted in Fig. 2.1, that $\omega_\mathrm{o}$ always points in the direction of the transported quantity. This allows us to formulate our method in a unified framework that holds for both *path tracing* and *light tracing* with only a minor modification that the arguments of the BSDF need to be swaped (see $f_\mathrm{s}$ terms in Eq. (2.1) and Eq. (2.2)).

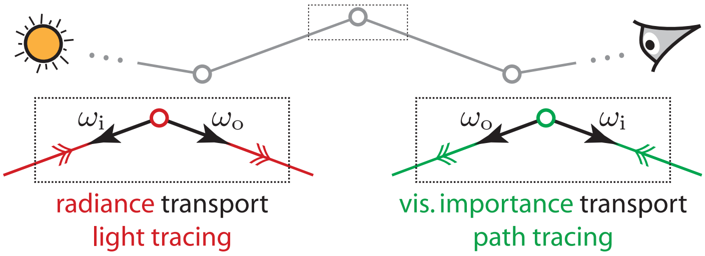


Figure 2.1: By convention, the direction $\omega_\mathrm{o}$ is always aligned with the direction of the transported quantity.

## 2.3 Path Tracing and Light Tracing

*Path tracing* [Kajiya, 1986] is perhaps the most popular light transport simulation method used in computer graphics nowadays. It can be thought of as following particles of visual importance starting from the camera. It is especially good at rendering view-dependent effects (reflections) and large scenes because important surfaces are immediately visible through the camera.

The pixel value $I$ is given by the visual importance *measurement equation* [Veach, 1997]

$$I = \int_{\mathcal{M}} \int_{\Omega} W_{\mathrm{i}}(\mathbf{y}, \omega)\, L_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}, \omega)\, |\cos\theta|\, \mathrm{d}\omega\, \mathrm{d}\mathbf{y}, \tag{2.3}$$

where $\mathcal{M}$ is the scene surface. By tracing $M$ *visual importance* particles from the camera, we can estimate $I$ using the following MC estimator [Veach, 1997]:

$$\langle I \rangle = \frac{1}{M} \sum_{k} \nu_{\mathrm{i}}(\mathbf{y}_k, \omega_k)\, L_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}_k, \omega_k), \tag{2.4}$$

where $\nu_{\mathrm{i}}$ is particle weight (see below).

Alternatively, one can use the *light tracing* algorithm [Dutré and Willems, 1994] which estimates the same pixel value $I$ given by radiance measurement equation

$$I = \int_{\mathcal{M}} \int_{\Omega} L_{\mathrm{i}}(\mathbf{y}, \omega)\, W_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}, \omega)\, |\cos\theta|\, \mathrm{d}\omega\, \mathrm{d}\mathbf{y}. \tag{2.5}$$

This equation is *adjoint* (see Sec. 2.4) to Eq. (2.3) and can be estimated using the following MC estimator [Veach, 1997, 4.A]:

$$\langle I \rangle = \frac{1}{N} \sum_{k} \nu_{\mathrm{i}}(\mathbf{y}_k, \omega_k)\, W_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}_k, \omega_k). \tag{2.6}$$

The crucial difference to path tracing is that evaluating this estimator involves following random paths of $N$ particles emitted *from light sources.* Therefore light tracing is very good at computing caustics, however, it is not used in practice because it fails at transport over reflective/refractive surfaces. The reason is that probability of hitting such a material from a direction that would yield a significant contribution decreases quickly with decreasing material roughness (see Fig. 2.2). Light tracing efficiency also deteriorates drastically with increased scene extent because light particles struggle to find the camera. Interestingly, our importance sampling method developed throughout this thesis and described in Section 4.5 addresses both of these issues.

Both sums (Eqs. (2.4) and (2.6)) are updated when a particle $k$ with its *weight* $\nu_{\mathrm{i}}(\mathbf{y}_k, \omega_k)$, coming from a direction $\omega_k$, collides at a location $\mathbf{y}_k$. In fact, the particle contributes to the sum only when the self emitted outgoing radiance $L_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}_k, \omega_k)$ or importance $W_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}_k, \omega_k)$, respectively, is non-zero. Note that such an estimator corresponds to unidirectional path tracing or light tracing without explicit connections to light sources or a camera respectively, i.e. without next event estimation [Dutré et al., 2006].

We deliberately use the formulation without next event estimation, because our work aims at improved importance sampling within unidirectional path-sampling techniques. In practice, however, it is efficient to sample light sources

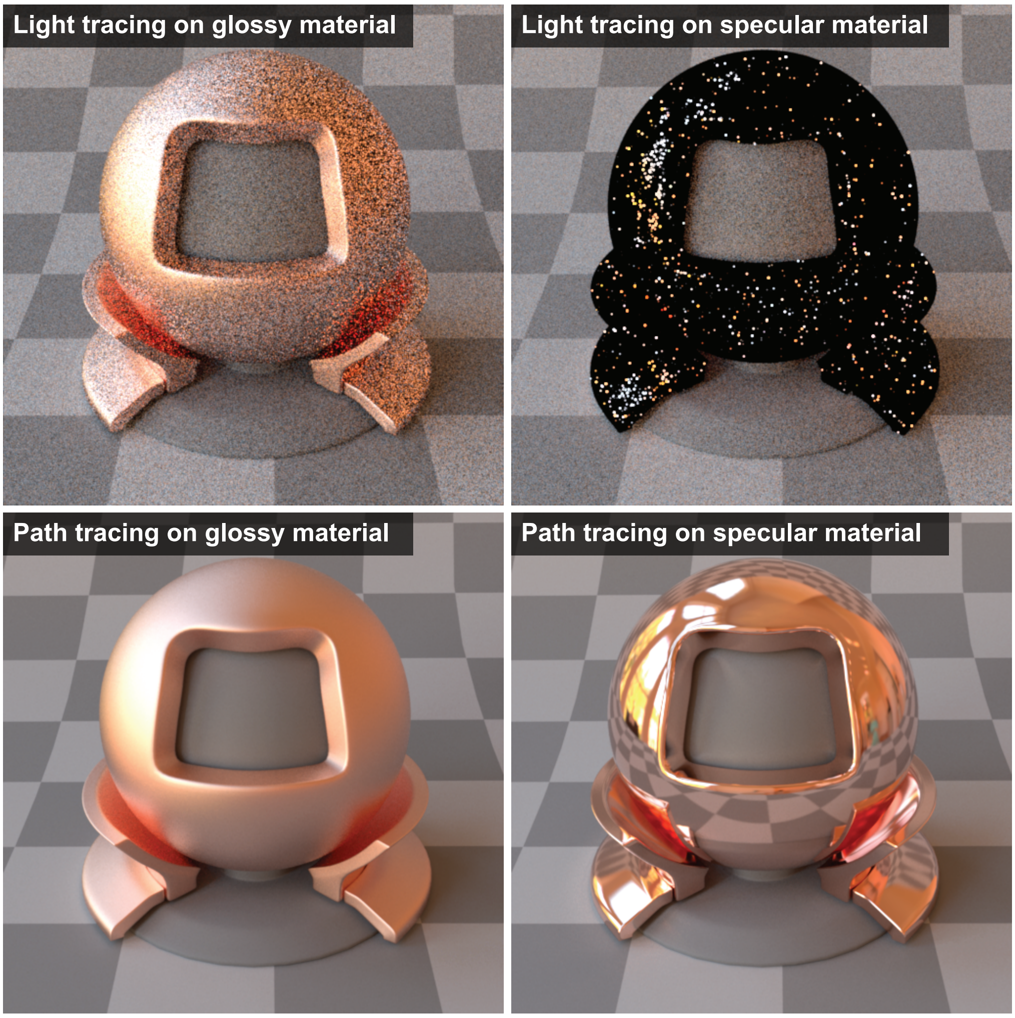

Light tracing on glossy material
Light tracing on specular material
Path tracing on glossy material
Path tracing on specular material

Figure 2.2: Light tracing is not favored in practice due to its slow convergence on glossy and specular materials. Left column: glossy material (Beckmann distribution with roughness 0.3), right column: specular material (Beckmann distribution with roughness 0.01), top row: light tracing, bottom row: path tracing. Scene courtesy of Jonas Pilo and Bernhard Vogl.

or cameras explicitly and in our implementation, we use next event estimation along our proposed importance sampling method. Actually in light tracing, this is the only viable option because camera sensor is usually too small in comparison to the scene extent and the probability of hitting the sensor by a particle is very small.

**Particle weight.** In both path and light tracing, the particle weight is given by the product of self-emitted outgoing visual importance $W_\mathrm{o}^\mathrm{e}$ or radiance $L_\mathrm{o}^\mathrm{e}$ [Veach, 1997, p. 91] respectively, and the bidirectional scattering distribution functions (BSDFs) and geometric factors along the particle path, divided by the probability density (pdf) of generating the path [Pharr and Humphreys, 2010]. To describe the way in which the weight is updated during a collision at some surface point $\mathbf{y}$, we distinguish between the *incident* weight $\nu_\mathrm{i}(\mathbf{y}, \omega_\mathrm{i})$ just before the collision, the weight after application of RR/splitting $\hat{\nu}(\mathbf{y}, \omega_\mathrm{i})$ (see Sec. 2.6), and the *outgoing* weight $\nu_\mathrm{o}(\mathbf{y}, \omega_\mathrm{o})$, just after the scattering. In path tracing, the last two are related by the weight update formula

$$\nu_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{o}}) = \hat{\nu}(\mathbf{y}, \omega_{\mathrm{i}}) \frac{f_{\mathrm{s}}(\mathbf{y}, \omega_{\mathrm{o}} \to \omega_{\mathrm{i}}) |\cos \theta_{\mathrm{o}}|}{p(\omega_{\mathrm{o}} | \, \mathbf{y})}, \tag{2.7}$$

where $p(\omega_{\mathrm{o}}| \, \mathbf{y})$ is a directional pdf for sampling the scattering direction $\omega_{\mathrm{o}}$. In light tracing, the same equation holds when directional arguments of BSDF $f_{\mathrm{s}}$ are swapped. The outgoing weight $\nu_{\mathrm{o}}$ after one collision then enters the next collision as its incoming weight $\nu_{\mathrm{i}}$. Extension for participating media would include attenuation between $\mathbf{y}$ and the next collision, which is the topic of our ongoing work.

## 2.4 Adjoints

We stated in the previous section that the pixel value $I$ can be expressed by both the visual importance measurement equation (Eq. (2.3)) and the radiance measurement equation (Eq. (2.5)):

$$I = \int_{\mathcal{M}} \int_{\Omega} W_{\mathrm{i}}(\mathbf{y}, \omega) \, L_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}, \omega) \, |\cos \theta| \; = \int_{\mathcal{M}} \int_{\Omega} L_{\mathrm{i}}(\mathbf{y}, \omega) \, W_{\mathrm{o}}^{\mathrm{e}}(\mathbf{y}, \omega) \, |\cos \theta|. \tag{2.8}$$

We refer to this special relationship when we say that *visual importance is adjoint to radiance* or vice versa. This terminology comes from mathematical theories describing spaces with inner products and the fact that light transport can be formulated in terms of linear operators on function spaces. We restrain from formal definition of these operators and instead, we refer interested reader to work of Veach [1997] or to standard textbooks [Dutré et al., 2006].

The adjoint relationship is very important and many transport algorithms in both neutron transport and computer graphics are based on it [Christensen, 2003]. In computer graphics, adjoint nature of light transport allows formulation of bidirectional algorithms (see Sec. 2.7) that are capable of handling difficult lighting conditions under presence of glossy and specular materials. The adjoint relationship also justifies approaches where visual importance has been used to optimize transport simulation so that the computation effort is focused on visually important places. For example, in photon mapping (see Sec. 2.7), one can use visual importance to stochastically store photons in places near the camera [Keller and Wald, 2000, Suykens and Willems, 2000]. This, in turn, results in high particle density and thus high quality of reconstructed radiance in places that contributes to the image the most.

Our methods introduced in this thesis are also based on adjoints. We precompute approximation of adjoint quantities and we employ them in path guiding and Russian roulette and splitting to increase efficiency of path sampling.

To keep our text uncluttered, in some parts of this thesis, we use $\Psi$ to denote the adjoint. Its meaning then depends on the actual context. For a path traced from a light source, it stands for the visual importance $W$, while for a path traced from the camera, it stands for radiance $L$.

## 2.5 Importance Sampling of Scattering Directions

In the case of path tracing, we could reduce the Monte Carlo estimator's (Eq. (2.4)) variance, if we were able to employ an importance sampling strategy that, at every collision, samples directions from a distribution closely proportional to the integrand of the rendering (2.1) transport equation:

$$p(\omega_\mathrm{o}) \propto L_\mathrm{i}(\mathbf{y}, \omega_\mathrm{o}) f_\mathrm{s}(\mathbf{y}, \omega_\mathrm{o} {\to} \omega_\mathrm{i}) |\cos\theta_\mathrm{o}|. \tag{2.9}$$

The traditional BSDF importance sampling can be ineffective when the incoming radiance term $L_\mathrm{i}(\mathbf{x}, \omega_\mathrm{i})$ is the primary source of the integrand's variation. Common examples include caustics, indirect highlights, and difficult visibility. Analogously for light tracing, to minimize the variance of the estimator given by Eq. (2.6), we would need to sample from a distribution closely proportional to an integrand of visual importance (2.2) transport equation:

$$p(\omega_\mathrm{o}) \propto W_\mathrm{i}(\mathbf{y}, \omega_\mathrm{o}) f_\mathrm{s}(\mathbf{y}, \omega_\mathrm{i} {\to} \omega_\mathrm{o}) |\cos\theta_\mathrm{o}|. \tag{2.10}$$

It is therefore beneficial to enhance the importance sampling with an estimate of the directional distribution of *incoming adjoint* $\Psi$ (i.e. radiance for path tracing; visual importance for light tracing). One way to achieve this is to obtain the distribution from a 'photon map' generated by particle tracing before the rendering starts [Jensen, 1995]. In Chapter 3, we follow this general strategy and note that the problem can be viewed as density estimation in the directional domain. We advocate the use of a parametric mixture model to represent the distributions, which enables on-line learning from a potentially infinite stream of particles.

In a follow-up work, Herholz and colleagues [2016] exploit mathematical properties of Gaussians and compute product of incoming adjoint and BSDF to further improve importance sampling.

## 2.6 Russian Roulette and Splitting

Arvo and Kirk [1990] suggest to use *Russian roulette* in non-analog (see Sec. 2.1) simulations for unbiased termination of particles with low *weights*. At any collision, if the weight drops below a threshold, the particle path is terminated with a certain probability $1 - q > 0$. On the other hand, if the particle survives, its weight is divided by the survival probability $q$ to keep the estimator unbiased.

In general, Russian roulette increases variance per sampled particle path $\sigma^2$ but at the same time it allows to sample more particles in the given time frame. This property allows Russian roulette to reduce the total estimator's error $\frac{\sigma}{\sqrt{N}}$, where $N$ is the total number of sampled particles. However, it can also increase the total variance if the sampling rate is not increased sufficiently.

In computer graphics, RR decisions have usually been based on local surface reflectivity [Jensen, 2001, Dutré et al., 2006, Jakob, 2010] or on the accumulated path weight (a.k.a. throughput) [Arvo and Kirk, 1990, Jensen, 1996, Veach, 1997]. We show in Chapter 4 that this approach is often sub-optimal and that our adjoint-driven approach yields superior results.

*Splitting* is a variance reduction technique that strives to reduce variance in an exactly opposite manner to Russian roulette. Unlike Russian roulette, the total number of sampled particles in a given time decreases, because the particle path can be split into $n$ independent paths at any collision, tracing of which costs the additional time. However, since splitting reduces the one-sample variance $\sigma^2$, the total estimator error can be significantly reduced. Note, that the resulting estimator stays unbiased because the particle weight is divided by $n$ after every collision.

A general example of splitting in computer graphics, as it has been noted by Arvo and Kirk [1990], is distributed ray tracing [Cook et al., 1984]. The main issue with splitting is the optimal choice of the splitting rate $n$. For instance, in next event estimation, which is another example of splitting, this rate is usually preset to higher values for early collisions when tracing particles from the camera. The reasoning behind this heuristic is the fact that direct illumination usually contributes the most to the resulting image and thus the chance to decrease $\sigma^2$ is high. In general case, the number of splits has often relied on heuristics based on local BSDF roughness [Szirmay-Kalos and Antal, 2005] or has been simply exposed as a simulation parameter.

## 2.7 Bidirectional Algorithms

Both path tracing and light tracing can be inefficient for some kinds of light transport. As we have already mentioned in previous sections, path tracing struggles under caustic lighting while light tracing cannot handle well reflections and large scenes.

Advantages of both approaches are combined in *bidirectional path tracing* [Lafortune and Willems, 1993, Veach and Guibas, 1994, Veach, 1997]. It traces particles from the camera and light sources and accounts for contributions from both light tracing and path tracing (Eqs. (2.4) and (2.6)). Moreover, it connects intermediate collision points between light and camera particles to form new paths comprising camera and light sub-paths. This results into many path-sampling strategies that are combined by multiple importance sampling (MIS) [Veach, 1997]. MIS simply assigns higher weight to strategies with higher sampling probability which often results in lower variance as opposed to having only one sampling strategy.

However, bidirectional path tracing is not suitable for rendering notoriously difficult transport through specular-diffuse-specular (SDS) interactions. For this purpose, one can choose *photon mapping* [Jensen, 2001, 1996]. It is a two-pass algorithm mostly recognized for its ability to simulate SDS transport and caustics in general.

In the first pass, light particles (photons) are emitted and traced in the scene just as they are in light tracing. Collisions of all photons are recorded and stored in KD-tree (photon map). Finally, the image is computed in the second, so called final gather pass, by tracing paths from the camera and radiance values are reconstructed from nearby photons in the photon map. During final gather the photons are used for radiance reconstruction only after a few bounces of camera paths, length of which depends on roughness of encountered materials.

Thus, unlike light tracing, photon mapping handles better glossy and specular materials. However it remains inefficient in rendering of scenes with large extent (see Sec. 2.3) and in the presence of glossy-to-glossy transport. Further advancements in photon mapping include *progressive photon mapping* (PPM) [Hachisuka et al., 2008] that overcomes memory limitations of the original algorithm.

*Vertex connection and merging* (VCM)/*unified path space* (UPS) [Georgiev et al., 2012b, Hachisuka et al., 2012] is bidirectional algorithm that combines advantages of bidirectional photon mapping [Vorba and Křivánek, 2011] and bidirectional path tracing (BDPT). Bidirectional photon mapping does not rely on heuristics in final gather. Instead, it introduces multiple path sampling strategies that differ by the number of collisions after which the radiance is reconstructed.

VCM/UPS then applies MIS to combine all the path sampling strategies coming from BDPT and bidirectional photon mapping. Its practical advantage is robustness in the scenes with SDS transport while it retains high asymptotic performance of BDPT for most of the other light transport types [Georgiev et al., 2012b].

However, even though BDPT and VCM/UPS are abundant in the number of sampling strategies, they both struggle in scenes with complex and difficult visibility. Also in large scenes, when light particles (photons) cannot find the camera, all the strategies based on light sub-paths make almost no contribution and bidirectional algorithms degrade to path tracing with significant overhead of the extra sampling strategies.

We implemented our path guiding method developed in Chapter 3 in PPM, BDPT and VCM to show that it can dramatically improve efficiency of these methods. Interestingly, our results in Chapter 3 suggest that guiding in unidirectional algorithms performs comparably to the guided bidirectional algorithms, which makes complex combined sampling strategies almost redundant. This might be useful especially for industry as it turns out that maintaining bidirectional methods combining many estimators is challenging in the production renderers that are ever changing under new requirements.

## 2.8 Zero-Variance Random Walk Rules

It has been known for a long time in neutron transport [Kalos, 1963], and recently pointed out in computer graphics [Křivánek and d'Eon, 2014, Xu et al., 2001], that particle paths can be constructed such that the estimator in Eq. (2.4) or Eq. (2.6) has zero variance (ZV). In other words, the solution can be found with only one particle path. While this cannot be achieved without knowing the computed solution in advance, zero-variance schemes are an invaluable tool for studying and designing variance reduction techniques.

To construct a zero-variance estimator of, for example, radiance measurement equation (Eq. (2.5)), we must obey the following rules[1]:

- Emit light particles according to the pdf proportional to the product of incident importance distribution at the light sources and cosine-weighted

[1] Note that it is not the only way to achieve zero variance [Booth, 2012a].

outgoing self-emitted radiance:

$$p^{\mathrm{e}}_{\mathrm{zv}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}}) = \frac{W_{\mathrm{i}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}})\, L^{\mathrm{e}}_{\mathrm{o}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}})\, |\cos\theta_{\mathrm{o}}|}{I}, \tag{2.11}$$

where the division by the estimated solution $I$ makes the pdf integrate to 1 (see Eq. (2.3), the visual importance measurement equation).

- At each scattering event, sample the new direction from the pdf proportional to the product of the cosine-weighted BSDF $f_{\mathrm{s}}$ and the incident visual importance distribution:

  $$p_{\mathrm{zv}}(\omega_{\mathrm{o}}|\mathbf{y}) = \frac{W_{\mathrm{i}}(\mathbf{y}, \omega_{\mathrm{o}}) f_{\mathrm{s}}(\mathbf{y}, \omega_{\mathrm{i}} \to \omega_{\mathrm{o}}) |\cos\theta_{\mathrm{o}}|}{W^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})}. \tag{2.12}$$

  The division by reflected importance $W^{\mathrm{r}}_{\mathrm{o}}$ at $\mathbf{y}$ makes the pdf integrate to 1 as can be seen from Eq. (2.2) by replacing the role of $\omega_{\mathrm{i}}$ and $\omega_{\mathrm{o}}$ (we trace from light sources here while Eq. (2.2) is defined with respect to tracing from the camera).

- Use the following survival probability

  $$q_{\mathrm{zv}}(\mathbf{y}, \omega_{\mathrm{i}}) = 1 - \frac{W^{\mathrm{e}}_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})}{W_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})} = \frac{W^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})}{W_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})}, \tag{2.13}$$

  that allows the walk to terminate only on the camera sensor.

- Contribute to the estimator only upon termination (i.e. with the probability $1 - q_{\mathrm{zv}}$).

These rules imply that we need to know the solution $I$ that we seek to compute. Furthermore, it also requires knowledge of visual importance $W$ everywhere in the scene, which, in general, is a more difficult problem than computing $I$. While this makes the zero-variance estimator unusable in practice, we use it in Chapter 4 to conduct an analysis which provides insights into properties of our adjoint-driven Russian roulette and splitting and path guiding.

## Proof of Zero-Variance

Here, we show that the rules stated in this section indeed result in a zero-variance walk. We will go through the life cycle of a particle that starts with its emission and follows a chain of collisions. At each collision it first faces stochastic termination and then — if it survives — it scatters.

**Emission.** After emission, the particle weight becomes

$$\begin{aligned} \nu^{\mathrm{e}}_{\mathrm{o}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}}) &= \frac{L^{\mathrm{e}}_{\mathrm{o}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}}) |\cos\theta_{\mathrm{o}}|}{p^{\mathrm{e}}_{\mathrm{zv}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}})} \\ &= \frac{L^{\mathrm{e}}_{\mathrm{o}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}}) |\cos\theta_{\mathrm{o}}| I}{W_{\mathrm{i}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}})\, L^{\mathrm{e}}_{\mathrm{o}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}})\, |\cos\theta_{\mathrm{o}}|} \\ &= \frac{I}{W_{\mathrm{i}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}})}. \end{aligned} \tag{2.14}$$

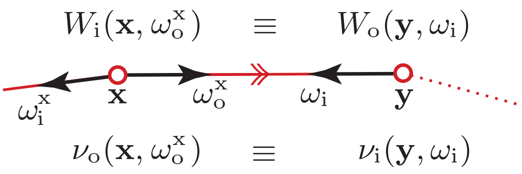

$W_\mathrm{i}(\mathbf{x},\omega_\mathrm{o}^\mathrm{x})$
$\equiv$
$W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})$
$\omega_\mathrm{i}^\mathrm{x}$
$\mathbf{x}$
$\omega_\mathrm{o}^\mathrm{x}$
$\omega_\mathrm{i}$
$\mathbf{y}$
$\nu_\mathrm{o}(\mathbf{x},\omega_\mathrm{o}^\mathrm{x})$
$\equiv$
$\nu_\mathrm{i}(\mathbf{y},\omega_\mathrm{i})$

Figure 2.3: A light tracing collisions where the collision at $\mathbf{x}$ precedes the collision at $\mathbf{y}$. The figure depicts important identities between the two events.

When this particle enters the next scattering event $\mathbf{y}$, due to the relationship of incident and outgoing visual importance $W_\mathrm{i}$ and $W_\mathrm{o}$ respectively (see Fig. 2.3), its incident weight (Eq. (2.14)) becomes

$$\frac{I}{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}. \tag{2.15}$$

**Termination.** Now, before the particle scatters at $\mathbf{y}$, it is stochastically terminated with probability $1-q_\mathrm{zv}$. If it is *terminated*, its weight is multiplied by the emitted visual importance $W_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{i})$, divided by the probability $1-q_\mathrm{zv}$ and the resulting weight $I$ contributes to the estimator:

$$\begin{aligned}&\frac{I}{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}W_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{i})\frac{1}{1-q_\mathrm{zv}(\mathbf{y},\omega_\mathrm{i})}\\ =&\frac{I}{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}W_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{i})\frac{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}{W_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{i})}\\ =&\boxed{I}\end{aligned} \tag{2.16}$$

We can see that upon termination, the particle weight is equal to the solution $I$ and thus it contributes with zero variance.

If the particle *survives*, its weight is divided by $q_\mathrm{zv}$:

$$\begin{aligned}\hat{\nu}(\mathbf{y},\omega_\mathrm{i}) &= \frac{I}{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}\frac{1}{q_\mathrm{zv}(\mathbf{y},\omega_\mathrm{i})}\\ &= \frac{I}{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}\frac{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}{W_\mathrm{o}^\mathrm{r}(\mathbf{y},\omega_\mathrm{i})}\\ &= \frac{I}{W_\mathrm{o}^\mathrm{r}(\mathbf{y},\omega_\mathrm{i})}.\end{aligned} \tag{2.17}$$

**Scattering.** Particle which has survived the termination step at the collision $\mathbf{y}$ is scattered. Its weight (Eq. (2.17)) is updated according to Eq. (2.7) using the zero-variance scattering pdf $p_\mathrm{zv}$ (Eq. (2.12)):

$$\begin{aligned}\nu_\mathrm{o}(\mathbf{y},\omega_\mathrm{o}) &= \hat{\nu}(\mathbf{y},\omega_\mathrm{i})\frac{f_\mathrm{s}(\mathbf{y},\omega_\mathrm{i}{\to}\omega_\mathrm{o})|\cos\theta_\mathrm{o}|}{p_\mathrm{zv}(\omega_\mathrm{o}|\mathbf{y})}\\ &= \frac{I}{W_\mathrm{o}^\mathrm{r}(\mathbf{y},\omega_\mathrm{i})}\frac{W_\mathrm{o}^\mathrm{r}(\mathbf{y},\omega_\mathrm{i})f_\mathrm{s}(\mathbf{y},\omega_\mathrm{i}{\to}\omega_\mathrm{o})|\cos\theta_\mathrm{o}|}{W_\mathrm{i}(\mathbf{y},\omega_\mathrm{o})f_\mathrm{s}(\mathbf{y},\omega_\mathrm{i}{\to}\omega_\mathrm{o})|\cos\theta_\mathrm{o}|}\\ &= \frac{I}{W_\mathrm{i}(\mathbf{y},\omega_\mathrm{o})}.\end{aligned} \tag{2.18}$$

Note that, after scattering, the particle is at the same state as after emission (compare Eq. (2.14) and Eq. (2.18)) and at a next collision it will take form of equation (2.15). The particle follows further termination/scattering events until it is finally terminated while contributing by its weight equal to the solution $I$ (Eq. (2.16)).

# Chapter 3

# Path Guiding through On-line Learning of Parametric Mixture Models

## 3.1 Introduction

In this chapter, to address the problem of rendering highly occluded scenes, we augment sampling of local scattering directions and light emission with global information. When constructing transport sub-paths, we sample the scattering and emission proportionally to an approximation of the equilibrium adjoint quantity which is radiance for camera sub-paths or importance for light sub-paths. As a result, the sub-paths are guided to each other – camera sub-paths toward light sources and light sub-paths toward the camera – which increases the probability of constructing full paths with non-zero contributions. This, in turn, significantly reduces variance without introducing bias.

Our work adopts the idea of reconstructing the sampling distributions from particles [Jensen, 1995]. While a number of works have taken this route, they often use inflexible representations of the distributions [Jensen, 1995] and incur significant overhead [Hey and Purgathofer, 2002]. Most importantly, these methods rely on a limited number of particles, which is usually insufficient to recover useful sampling distributions in highly occluded scenes.

We propose to represent the sampling distributions with the Gaussian mixture model (GMM), extensively used in machine learning [Bishop, 2006]. The GMM is efficient to learn, easy to evaluate and sample, and compact to store. The core of our method is an on-line (progressive) learning step: Instead of learning the distributions only once from a limited set of particles, we continuously train them using a potentially infinite stream of particles while keeping a bounded memory footprint. We use importance sampling based on this model in a number of light transport algorithms, including the state-of-the-art bidirectional ones [Georgiev et al., 2012b, Hachisuka et al., 2012, Veach, 1997].

Our key contributions in this chapter, which is based on our previously published work [Vorba et al., 2014], are:

- introduction of on-line learning of parametric mixture models to image synthesis,

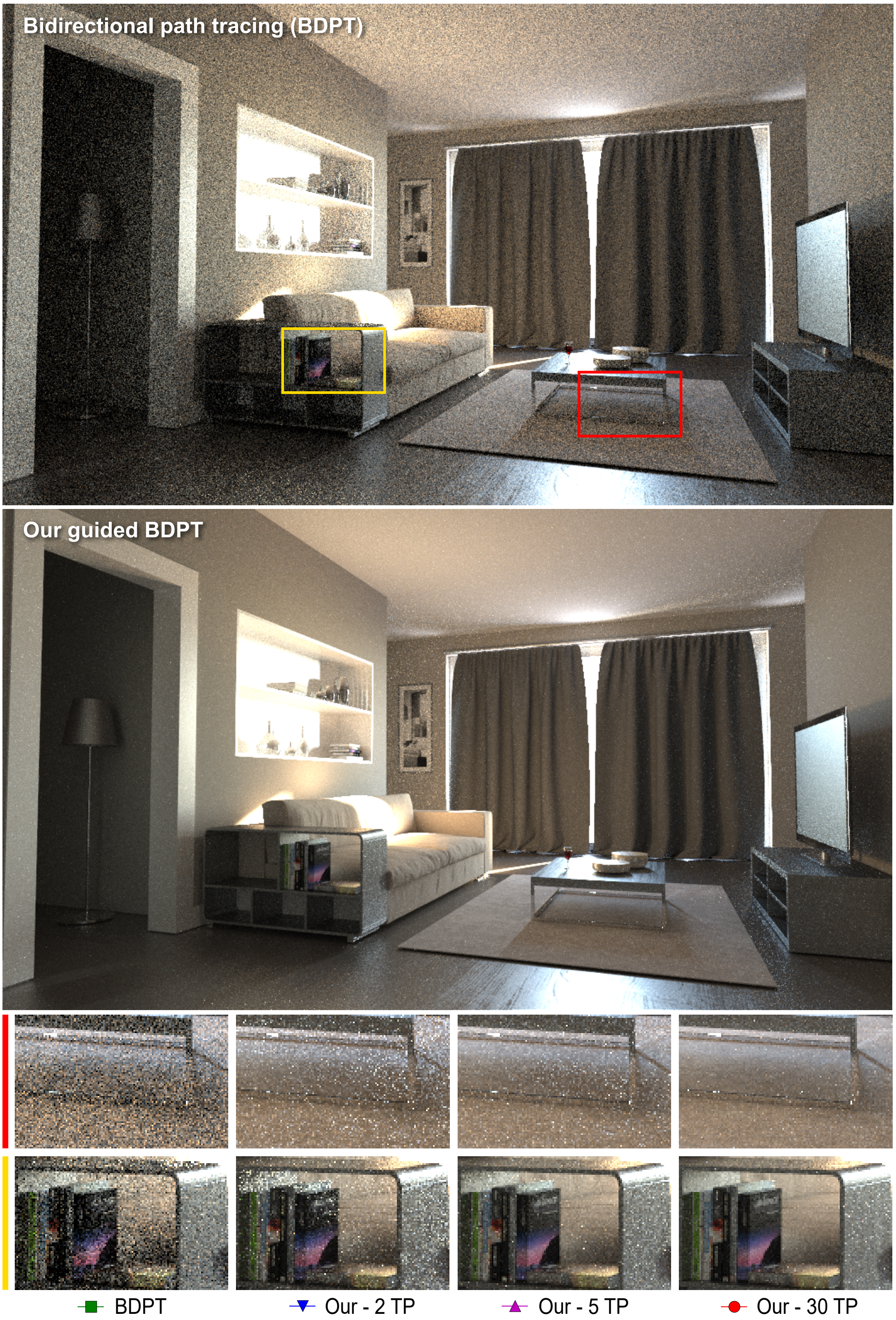


Figure 3.1: We render a scene featuring difficult visibility with bidirectional path tracing (BDPT) guided by our parametric distributions learned on-line in a number of training passes (TP). The insets show equal-time (1h) comparisons of images obtained with different numbers of training passes. The results reveal that the time spent on additional training passes is quickly amortized by the superior performance of the subsequent guided rendering. Convergence plot corresponding to these four methods is shown in Fig. 3.2.

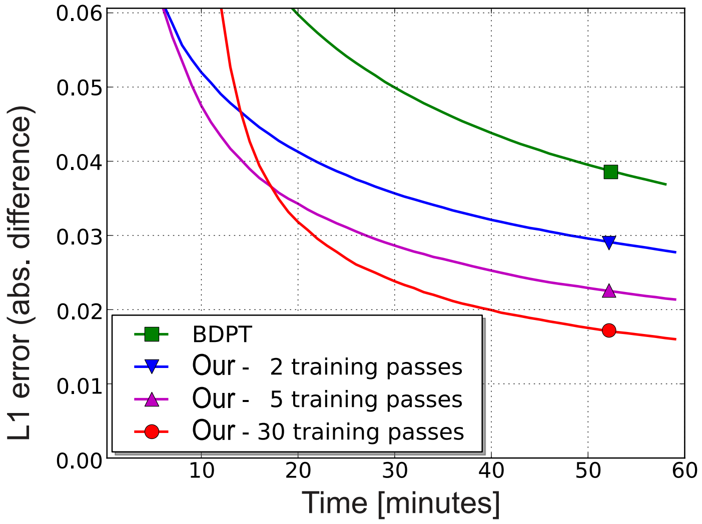


Figure 3.2: The plot related to Fig. 3.1 shows convergence of four simulations running with different numbers of training passes over 1h. The results reveal that the time spent on additional training passes is quickly amortized by the superior performance of the subsequent guided rendering.

- a learning procedure that can handle particles with highly varying weights,
- employing parametric mixtures for importance-driven particle emission from environment light sources,
- an efficiency improvement of path sampling-based light transport algorithms in complex, highly occluded scenes.

In the rest of this chapter, we position our work on importance sampling of scattering directions and emission with respect to existing methods, we review learning of parametric mixtures, we expose our on-line learning algorithm and present a caching scheme for learned distributions, and finally, we present achieved results.

## 3.2 Related Work

**Sampling distributions from particles.** Jensen [1995] proposed the use of light-carrying particles, or photons, to guide direction sampling in a path tracer. He reconstructs the directional PDFs by counting the number of photons whose directions fall into constant-sized bins. This corresponds to simple histogram density estimation, which is known to be a poor density estimation method prone to under- or over-fitting [Bishop, 2006]. The method could be made progressive by accumulating an infinite stream of particles to the bins. However, this progressivity is rather deceiving because the regular histogram grid does not adapt to details, thereby producing poor PDF reconstructions no matter how many particles are used. A similar approach was taken by Steinhurst and Lastra [2006] and Budge et al. [2008]. Peter and Pietrek [1998] extended this idea to using importance particles for guiding photons towards the camera. Hey and Purgathofer [2002] represent the directional PDFs with cones of adaptive width centered on gathered photons' directions. This method yields better results than previous work, but incurs substantial overhead. Pharr and Humphreys [2010] implement a simplified version of Hey and Purgathofer's method with cones of constant width. Its reduced overhead, however, comes at the expense of quality. Our paper follows this general line of work, pointing out that 'reconstructing sampling PDFs from photons' is a general density estimation problem. We adopt parametric mixture model estimation to learn the sampling PDFs in an on-line (progressive) manner.

**Adaptive sampling.** A number of works propose adaptive construction of sampling distributions during rendering. Lafortune and Willems [1995] store radiance samples in a 5D tree and use this information for importance sampling and as a control variate. Pegoraro et al. [2008a, 2008b] replace the 5D tree with per-pixel directional distributions. Dutré and Willems [1994, 1995] use adaptive sampling similar to the VEGAS algorithm [Lepage, 1978] to emit paths from light sources. Cline et al. [2008] use fixed-size adaptive importance sampling tables in a path tracer. A common problem shared by these works is a fixed-resolution or hierarchical representation of the sampling function which makes it difficult to capture high-frequency features. The Gaussian mixture model that we adopt does not suffer from this issue.

**Caching.** Bashford-Rogers et al. [2012] employ a parametric model for importance sampling where cosine lobes are used to model directional distributions, cached and re-used across pixels. The idea of caching sampling distributions was also used by Georgiev et al. [2012a], who precompute and cache discrete distributions of the contributions of virtual point lights (VPLs) to scene points. As opposed to these techniques, we store the learned directional distributions in a spatial cache not only to amortize the overhead, but mainly to maintain a persistent representation of the distributions, thereby enabling their progressive refinement through on-line learning. Other solutions such as irradiance and radiance caching [Ward et al., 1988, Křivánek et al., 2005] can be used when systematic error is acceptable; however, we pursue unbiased results.

**Sampling emission from environment maps.** Tsai et al. [2008] employ spherical Gaussians to sample from the product of environment illumination and BSDFs. Bashford-Rogers et al. [2013] guide emission from environment maps using importance. We address a more general problem of sampling both indirect illumination and environment emission.

**Progressive GMM learning in rendering.** Jakob et al. [2011] use a Gaussian mixture model to represent spatial distribution of scalar irradiance in participating media. The accelerated expectation maximization (EM) algorithm [Verbeek et al., 2006] used for this purpose allows progressive model updates, but it fundamentally relies on the ability to produce a good fit from the initial batch of particles. Thanks to the maximum a posteriori (MAP) formulation of the model estimation, our on-line technique robustly handles situations where particles are extremely scarce.

## 3.3 Parametric Mixture Models and the EM Algorithm

We review learning of parametric mixture models that forms basis of our approach. We start by presenting the classic *batch EM* (Expectation Maximization) algorithm, which is well known to the graphics community. We then describe the *off-line stepwise EM* algorithm, a generalization of batch EM with better convergence properties [Liang and Klein, 2009]. The stepwise EM formulation allows deriving the *on-line stepwise EM* algorithm for learning from a potentially infinite stream of particles. While batch EM is reviewed only for reference, both stepwise EM variants are essential components of our method. Bishop [2006] and Cappé [2011] provide more details on EM.

**Parametric mixture models.** A parametric finite mixture model is a convex combination of simpler parametric distributions. We use the Gaussian mixture model (GMM) with $K$ components:

$$\mathrm{GMM}(\mathbf{s}|\theta) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{s}|\mu_j, \mathbf{\Sigma}_j), \tag{3.1}$$

where $\mathcal{N}(\mathbf{s}|\mu_j, \mathbf{\Sigma}_j)$ is a Gaussian distribution over $\mathbf{s} \in \mathbb{R}^d$ with a mean $\mu_j$ and a covariance matrix $\mathbf{\Sigma}_j$. The *mixing coefficients* $\pi_j$ satisfy $\pi_j \geq 0$ and $\sum_{j=1}^{K} \pi_j = 1$. The mixture is defined by a parameter vector $\theta = \{\pi_1, \mu_1, \mathbf{\Sigma}_1, \ldots, \pi_K, \mu_K, \mathbf{\Sigma}_K\}$.

**Maximum likelihood (ML) estimation.** The density estimation problem for a mixture model $p(\mathbf{s}|\theta)$ (e.g. the GMM), consists in finding parameters $\theta$ so that $p(\mathbf{s}|\theta)$ is a good approximation of the unknown distribution that generated a given finite set of $N$ observed samples $\mathbf{S} = \{\mathbf{s}_0, \ldots, \mathbf{s}_{N-1} \in \mathbb{R}^d\}$. A common approach is to use the parameter vector $\theta^{\mathrm{ML}}$ that maximizes the *log-likelihood* $\mathcal{L}(\mathbf{S}, \theta) = \ln p(\mathbf{S}|\theta) = \sum_{q=0}^{N-1} \ln p(\mathbf{s}_q|\theta)$.

**Maximum a-posteriori (MAP) estimation.** A fundamental problem with ML estimation is over-fitting, i.e. introducing patterns not present in the original distribution [Bishop, 2006]. This issue is particularly pressing in our approach, where we may have only a few observed samples available to construct initial estimates. To deal with this issue, we adopt the *maximum a posteriori* solution $\theta^{\mathrm{MAP}}$, which seeks the mode of the *posterior* distribution $p(\theta|\mathbf{S})$ over model parameters $\theta$, given by Bayes' theorem: $p(\theta|\mathbf{S}) \propto p(\mathbf{S}|\theta)p(\theta)$ (i.e. posterior $\propto$ likelihood $\times$ prior). With only a few samples, the solution is mostly determined by our *prior* beliefs (e.g. that PDFs with extreme values are unlikely), modeled by the prior distribution $p(\theta)$, which is overridden as more samples are observed. Appendix 3.A and especially Gauvain and Lee [1994] provide more details.

**Batch expectation maximization (Batch EM).** Expectation maximization (EM) [Dempster et al., 1977] is an iterative procedure to find the ML or MAP estimates for mixture models. The classic, or *batch* EM algorithm [Liang and Klein, 2009] for a finite set of observed samples starts with an initial guess of parameters and proceeds in iterations over the sample set. In each iteration, which consists of the *expectation* (E) and the *maximization* (M) steps, it obtains a new estimate $\theta^{\mathrm{new}}$ based on the current estimate $\theta^{\mathrm{old}}$ (Algorithm 1).

Since the log-likelihood $\mathcal{L}(\mathbf{S}, \theta)$ over the EM iterations is a non-decreasing function of $\theta$, the iterative solution $\theta^{\mathrm{new}}$ converges to a local maximum. The

**Algorithm 1** Pseudocode of batch expectation maximization (EM).

```
$\theta^{\text{old}}$ := INITIALIZE( )
repeat
    $\mathbf{u}^j_{N-1}$ := COMPUTESUFFICIENTSTATS($\mathbf{S}, \theta^{\text{old}}$)    ▷ E-step: Eq. (3.4)
    $\theta^{\text{new}}$ := $\overline{\theta}(\mathbf{u}^1_{N-1}, \dots, \mathbf{u}^K_{N-1})$    ▷ M-step
until CONVERGED( )    ▷ Eq. (3.2)
```

following condition is often used as a convergence criterion:

$$\left|\mathcal{L}(\mathbf{S}, \theta^{\text{old}}) - \mathcal{L}(\mathbf{S}, \theta^{\text{new}})\right| < \epsilon \left|\mathcal{L}(\mathbf{S}, \theta^{\text{new}})\right|. \tag{3.2}$$

In the **E-step**, the *responsibilities* $\gamma_{qj}$ of every component $j$ for each sample $\mathbf{s}_q$ are evaluated. Informally, they give the probability that the sample $\mathbf{s}_q$ would be drawn from the component $j$ if we sampled from the mixture $\theta^{\text{old}}$. For the GMM, the responsibilities are computed as

$$\gamma_{qj} = \frac{\pi_j \mathcal{N}(\mathbf{s}_q \mid \theta^{\text{old}}_j)}{\sum_{h=1}^{K} \pi_h \mathcal{N}(\mathbf{s}_q \mid \theta^{\text{old}}_h)}. \tag{3.3}$$

With these responsibilities, we can compute the *sufficient statistics* $\mathbf{u}^j_{N-1}$ for every mixture component $j$ as the weighted average

$$\mathbf{u}^j_{N-1} = \frac{1}{N} \sum_{q=0}^{N-1} \gamma_{qj} \mathbf{u}(\mathbf{s}_q), \tag{3.4}$$

where $\mathbf{u}(\mathbf{s}_q) = (1, \mathbf{s}_q, \mathbf{s}_q \mathbf{s}_q^T)$ is a triplet consisting of the number 1, the vector $\mathbf{s}_q$, and the matrix $\mathbf{s}_q \mathbf{s}_q^T$. The subscript $N-1$ suggests that the sufficient statistics are based on $N$ observed samples $\mathbf{s}_0, \dots, \mathbf{s}_{N-1}$. The ML and MAP estimates depend on the observed samples in $\mathbf{S}$ only through these sufficient statistics.

In the **M-Step**, the sufficient statistics are used to obtain a new GMM estimate $\theta^{\text{new}}$ using a closed form update formula $\theta^{\text{new}} = \overline{\theta}(\mathbf{u}^1_{N-1}, \dots, \mathbf{u}^K_{N-1})$. We do not provide details of the update formula $\overline{\theta}$ here as we do not use batch EM in our solution.

**Off-line stepwise EM.** We now describe the stepwise EM formulation [Liang and Klein, 2009], whose on-line variant forms the basis of our approach. In batch EM, the sufficient statistics are recomputed from all $N$ samples (E-step) and only then, the distribution parameters can be updated (M-step). The stepwise formulation, on the other hand, continuously updates the statistics with every observed sample, which enables more frequent parameter updates, and therefore faster convergence.

*Off-line stepwise EM*, a generalization of batch EM [1] , also iterates over the sample set until convergence, as shown in Alg. 2. In the **E-step**, the sufficient statistics for each mixture component $j$ are updated using the formula:

$$\mathbf{u}^j_i = (1 - \eta_i)\mathbf{u}^j_{i-1} + \eta_i \, \gamma_{qj} \mathbf{u}(\mathbf{s}_q), \tag{3.5}$$

[1] In stepwise EM, if we set the stepsize parameter $\alpha$ to be equal to 1 and the update rate $m$ to be equal to the dataset size $N$ (i.e. $m = N$), we obtain the batch EM algorithm.

where $i \geq 1$ is an index that increments with each processed sample and $q = (i-1) \bmod N$ is the index of the sample $\mathbf{s}_q$ in the sample set $\mathbf{S}$. In other words, the samples from $\mathbf{S}$ are processed over and over, while the index $i$ keeps growing. The updated statistics $\mathbf{u}_i^j$ are given by a weighted average of the statistics $\gamma_{qj}\mathbf{u}(\mathbf{s}_q)$ for the currently observed sample $\mathbf{s}_q$ and the statistics $\mathbf{u}_{i-1}^j$ for all the previously observed samples. The weight in this average is given by the decreasing sequence $\{\eta_i\}_{i\geq 1}$ of *stepsizes* that must obey $\sum_i \eta_i = \infty$ and $\sum_i \eta_i^2 < \infty$. A sequence which satisfies these conditions is $\eta_i = i^{-\alpha}$ with the effective values of the stepsize parameter $\alpha \in [0.6, 0.9]$ [Cappé, 2011].

The distribution parameters $\theta^{\text{old}}$ are updated in the **M-step** after processing every $m$-th sample ($1 \leq m \leq N$). The formula is the same as in batch EM, with the current sufficient statistics $\mathbf{u}_i$ as inputs:

$$\theta^{\text{new}} = \overline{\theta}(\mathbf{u}_i^1, \ldots, \mathbf{u}_i^K). \tag{3.6}$$

We provide the update formula $\overline{\theta}$ for MAP solution from stream of weighted particles in Sec. 3.4.2.

**Algorithm 2** Off-line stepwise EM

```
 1: $i := 0$                                              ▷ Index of sufficient statistics
 2: repeat
 3:     for $q := 0$ to $N - 1$ do                         ▷ Iterate over a batch of $N$ samples
 4:         $i := i + 1$
 5:         $\mathbf{u}_i^j :=$ UPDATESUFFICIENTSTATS($\mathbf{s}_q, \theta^{\text{old}}$)      ▷ E-step: Eq. (3.5)
 6:         if $i \bmod m = 0$ then                       ▷ M-step: every $m$-th observed sample
 7:             $\theta^{\text{new}} := \overline{\theta}(\mathbf{u}_i^1, \ldots, \mathbf{u}_i^K)$                  ▷ Eq. (3.6)
 8:         end if
 9:     end for
10: until CONVERGED( )                                     ▷ Eq. (3.2)
```

**On-line stepwise EM.** The batch EM and the off-line stepwise EM algorithms base inference on a finite set of $N$ samples stored in memory. Our method, however, targets scenarios where the number of samples (e.g. photons) necessary for reliable inference would be impractical or even impossible to store. The off-line stepwise EM formulation, unlike batch EM, can be easily modified for this purpose [Sato and Ishii, 2000, Liang and Klein, 2009, Cappé, 2011]. The key is Equation (3.5) that enables progressive embedding of the information from any number of particles into a small set of statistics. If we consider the input set $\mathbf{S} = \{\mathbf{s}_0, \mathbf{s}_1, \ldots \in \mathbb{R}^d\}$ to be an infinite stream of samples, then the *on-line stepwise EM* algorithm is obtained from Alg. 2 by removing the outer cycle (lines 2 and 10) that iterates over the finite batch of samples. As such, the on-line algorithm continues learning as long as the samples are streamed. Details with respect to our application are given in Sec. 3.4.2.

## 3.4 Our Unbiased Guiding Method

We now present an overview of our unbiased guiding method (Sec. 3.4.1) followed by details of our technical contribution: MAP density estimation from weighted particles (Sec. 3.4.2), caching of distributions (Sec. 3.4.3), and emission from environment light sources (Sec. 3.4.4). In Sec. 3.4.5, we discuss the form of Russian roulette that we used to obtain our results presented in this chapter.

### 3.4.1 Overview

Our method is split into two strictly separated phases: a) training of hemispherical distributions representing the incoming radiance or importance from particles (training phase), and b) using the trained distributions for importance sampling in rendering (rendering phase). During the training phase, the directional distributions are placed and cached at scene surfaces and progressively updated. The distributions stay fixed during the entire rendering phase.

**Training phase.** The training phase, depicted in Fig. 3.3, consists of several *training passes*. Each training pass comprises tracing a batch of importons from the camera followed by a batch of photons from the light sources. We start by tracing a batch of importons without our guiding (Fig. 3.3a). Every succeeding particle tracing step is then guided by our distributions, which are created and progressively refined throughout the entire training phase. Radiance distributions are trained by photons and used to guide paths from the camera. Conversely, importance distributions are trained by importons and used to guide paths from the light sources. *Guiding* refers to importance sampling of local scattering directions and emission from environment light sources based on our distributions, as described below.

New distributions are created on-the-fly during the guided particle tracing steps (Fig. 3.3c, f). They are stored in a *spatial cache* so that they can be reused at nearby locations and refined in the subsequent training passes. We keep two separate caches, one for importance and the other for radiance distributions. If

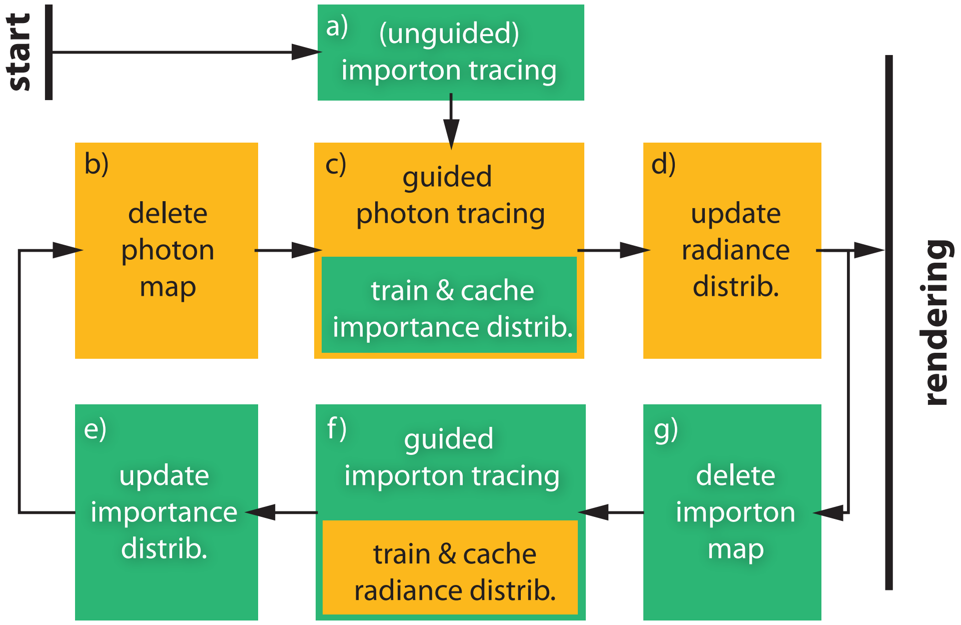

start
a) (unguided) importon tracing
b) delete photon map
c) guided photon tracing
train & cache importance distrib.
d) update radiance distrib.
rendering
e) update importance distrib.
f) guided importon tracing
train & cache radiance distrib.
g) delete importon map

Figure 3.3: The training phase preceding the rendering phase. Processes related to importons or importance distributions use green background while processes related to photons or radiance distributions are in orange.

the particle tracing process requests a distribution at a certain point and none is available at any nearby location, a new distribution is created and cached. We train the new distribution from the particle map constructed during the preceding particle tracing step (e.g. the importon map if we currently trace a photon) by collecting $N$ nearest particles and using them as the input for *off-line stepwise EM*. The new distribution is then cached together with meta-information necessary for its refinement in the subsequent training passes (see Sec. 3.4.2).

After a batch of particles has been traced, we use it to update all distributions in the respective cache (Fig. 3.3d, e). For each cached distribution, we find the $N$ nearest particles and use *on-line stepwise EM* to update it. At any given time, only the last two particle maps (one for radiance and the other for importance) are kept in memory. We delete the maps before constructing new ones (Fig. 3.3b, g), and thus our method keeps a *bounded memory footprint* while allowing for an arbitrary number of training passes. Unlike the particle maps, the two distribution caches are persistent and are continuously updated throughout the whole training phase.

The motivation behind our use of interleaved, mutually guided importon and photon tracing steps is that with every training pass, the distributions become more accurate and provide improved importance sampling for the subsequent training passes. This approach significantly improves the efficiency of the training phase.

**Rendering phase.** With the two caches of distributions obtained in the training phase, we can guide the construction of both camera and light sub-paths in virtually any path sampling-based light transport algorithm (including bidirectional ones). To guide a unidirectional algorithm, such as path tracing, we simply discard the unnecessary distribution cache. As in the training phase, we use the distributions for importance sampling of the local scattering directions and emission from environment light sources. If no guiding distribution is available at a nearby position, a new one is trained from the latest batch of particles.

**Distribution representation.** After investigating a number of alternatives, we decided to model the directional distribution $p(\omega|\mathbf{y})$ at a spatial location $\mathbf{y}$ with a mixture of bi-variate Gaussians GMM($\mathbf{s}$) on a 2D plane, i.e. $\mathbf{s} \in \mathbb{R}^2$. To do this, we project the hemisphere $\mathcal{H}^+$ onto a unit square using the area preserving mapping $\mathcal{S}$ of Shirley and Chiu [1997].

**Sampling from the distribution model.** To generate a new direction $\omega$ after a particle has collided with a surface at a position $\mathbf{y}$, we randomly choose between BSDF sampling and sampling from our *guiding distribution* $p(\omega|\mathbf{y})$. Both strategies are then combined using multiple importance sampling [Veach, 1997]. To sample a direction $\omega'$ from our guiding distribution, we first draw a 2D position $\mathbf{s}'$ from the GMM($\mathbf{s}$) and then we apply the inverse mapping so that $\omega' = \mathcal{S}^{-1}(\mathbf{s}')$. To compensate for the change of variables, we multiply the PDF value by the Jacobian of $\mathcal{S}^{-1}$, which is a constant $\frac{1}{2\pi}$ for Shirley and Chiu's mapping. Should the sample $\mathbf{s}'$ lie outside of the unit square, the particle path is terminated.

**Direct vs. indirect illumination.** Reflected light ($L_o^r$ in Eq. (2.1)) can be split into integration of direct light that scatters only once and indirect light due to multiple scattering. In our implementation, to achieve optimal importance sampling of unidirectional path tracing, our radiance distributions contain both direct and indirect illumination. However, we exclude direct visual importance from distributions used to guide photons because they cannot hit the camera implicitly.[2]

Note that, in practice (and also in our implementation), it is common to use multiple importance sampling to combine next event estimation (explicit sampling of light sources) and unidirectional sampling. In this case, our decision to train distributions to include both direct and indirect illumination might become sub-optimal. For example, if next event estimation forms an almost perfect sampling strategy then we would achieve better importance sampling if we excluded direct illumination from our distributions. This has two reasons. First, we can allocate all unidirectional samples for indirect illumination because direct illumination is already sampled well by next event estimation. Second, all GMM components would be used to capture finer details in indirect illumination which, in turn, would yield a better importance sampling of indirect illumination.

Nevertheless, our decision to include both direct and indirect illumination in the guiding distribution is also supported by the fact that, due to occlusion, next event estimation is often not optimal sampling strategy. For example, if an environment light source contains also sun light then most of next even estimation samples will be allocated to sample the sun. However, this is not optimal for shadows where the sun contribution is zero but other light source features are important.

### 3.4.2 Learning Distributions from Weighted Particles

As described in Sec. 2.3, a particle tracing algorithm generates particles $\mathbf{p} = (\mathbf{y}, \omega_i, \nu_i)$, defined by their position $\mathbf{y}$, incoming direction $\omega_i$, and incident *weight* $\nu_i$ (also referred to as 'flux' when tracing photons). The particle weight is a product of the particle emission function, BSDFs, and geometry factors divided by the probability density of generating the particle path (see Sec. 2.3). The density of the photons or importons together with their weights form an unbiased representation of the equilibrium incoming radiance $L_{\mathrm{i}}$ or importance $W_{\mathrm{i}}$, respectively [Veach, 1997]. Therefore, an approximation of a sampling PDF $p(\omega|\mathbf{y}) \propto L_{\mathrm{i}}(\mathbf{y}, \omega)|\cos\theta_{\mathrm{i}}|$ can be reconstructed from the *directions* and *weights* of photons $\mathbf{p}$ in the vicinity of $\mathbf{y}$. Similarly, directions and weights of importons in the vicinity of $\mathbf{y}$ can be used to reconstruct a sampling PDF $p(\omega|\mathbf{y}) \propto W_{\mathrm{i}}(\mathbf{y}, \omega)|\cos\theta_{\mathrm{i}}|$. Note that the cosine term is implicitly present in the particles density that hit a (supposedly) flat surface.

Here, we present a new generalization of the stepwise EM algorithms (both off-line and on-line) that supports density estimation from a set of *weighted particles.* Before delving into its description, let us clarify, on an example of a single distribution, the use of off-line and on-line stepwise EM in our method (see Fig. 3.4). A distribution is initialized by *off-line* stepwise EM from the set of particles available in the first training pass. We then store its sufficient statistics $\mathbf{u}_i$ and its counter $i$ so that the learning can be resumed in the subsequent training passes by *on-line* stepwise EM. Note that, in the rest of this section, we override

[2] Similarly, for point light sources, direct illumination should be excluded as such source cannot be hit implicitly.

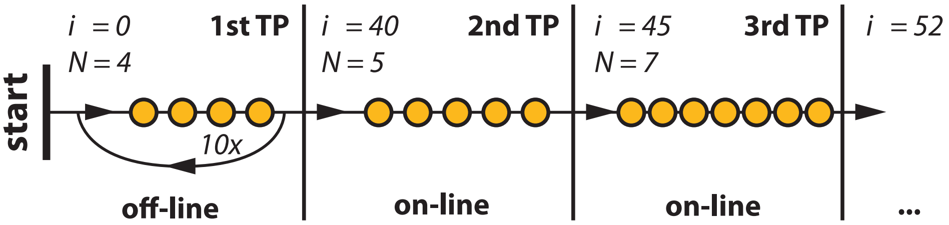


Figure 3.4: An example of learning of a single distribution from a stream of particles processed first by the off-line and then by the on-line stepwise EM algorithms. In the first training pass (TP), we use the off-line algorithm that iterates until convergence (10 times in our example) over the $N = 4$ particles available in the first TP. In the subsequent training passes, the on-line algorithm is used, which processes each particle only once. The index $i$ of the sufficient statistics associated with the trained distribution is incremented with each processed particle.

meaning of index $i$ to stand for a data counter instead of depicting something 'incident'.[3]

Save for some subtle differences, our new weight-aware formulation affects both off-line and on-line stepwise EM in the same manner. Thus, in the following statements, we refer to both algorithms as stepwise EM unless the algorithm variant is explicitly stated.

**Weighted data log-likelihood.** The log-likelihood $\mathcal{L}(\mathbf{S}, \theta)$, as given in Sec. 3.3, does not allow density estimation from a weighted set of samples. Thus, we introduce the following *weighted data log-likelihood*

$$\mathcal{L}(\mathbf{S}, \mathbf{w}, \theta) = \sum_{q=0}^{N-1} \nu_q \ln p(\mathbf{s}_q | \theta), \tag{3.7}$$

where $\mathbf{S} = \{\mathbf{s}_0, \ldots, \mathbf{s}_{N-1}\}$ is a set of $N$ observed samples with their corresponding weights $\mathbf{w} = \{\nu_0, \ldots, \nu_{N-1}\}$. This definition is in line with the intuition that a weighted sample $(\mathbf{s}_q, \nu_q)$ corresponds to an unweighted sample $\mathbf{s}_q$ observed $\nu_q$ times.

Taking measures against over-fitting is extremely important when using weighted particles, because the particle weights might differ by orders of magnitude. To this end, we employ a MAP solution (see Sec. 3.3) based on *conjugate priors*. Conjugate priors have the same functional form as the resulting posterior distribution and therefore lead to a greatly simplified Bayesian analysis [Bishop, 2006]. Our approach based on the *weighted data log-likelihood* requires a careful treatment of the MAP approach. This is reflected in both the E-step and the M-step of stepwise EM. In the following paragraphs, we extend the sufficient statistics (E-step) and provide formulae for updating the model parameters. Please refer to Appendix 3.A for their derivation.

**Our weighted stepwise EM: E-step.** To account for the weight $\nu_q$ of an observed sample $\mathbf{s}_q$, we modify the sufficient statistics update formula (3.5) to

$$\mathbf{u}_i^j = (1 - \eta_i)\mathbf{u}_{i-1}^j + \eta_i \nu_q \gamma_{qj} \mathbf{u}(\mathbf{s}_q), \tag{3.8}$$

[3] In this section, we treat particles rather as a general dataset for learning and we do not need their light transport semantic.

where $\mathbf{u}(\mathbf{s}_q) = (1, \mathbf{s}_q, \mathbf{s}_q\mathbf{s}_q^T)$. The only difference from Equation (3.5) is the multiplication of the second summand by $\nu_q$. This corresponds to the interpretation of weight $\nu_q$ as a multiplicity of the new observed sample $\mathbf{s}_q$. Additionally, we keep track of the averaged total particle weight required for normalization of mixture weights in the M-step:

$$\overline{\nu}_i = (1 - \eta_i)\overline{\nu}_{i-1} + \eta_i \nu_q, \tag{3.9}$$

**Our weighted stepwise EM: M-step.** We have derived an update function for model parameters, $\theta^{\text{new}} = \overline{\theta}(\mathbf{u}_i^1, \dots, \mathbf{u}_i^K, \overline{\nu}_i)$, that takes the current modified sufficient statistics (3.8) and the averaged total particle weight $\overline{\nu}_i$. By letting $\mathbf{u}_i^j = ((u_\gamma)_i^j, (\mathbf{s})_i^j, (\mathbf{s}\mathbf{s}^T)_i^j)$, we decompose the sufficient statistics $\mathbf{u}_i^j$ into a real number, a vector and a matrix that are computed from Equation (3.8). Then the specific formulae defining the vector function $\overline{\theta}$ read as follows:

$$\pi_j^{\text{new}} = \frac{\dfrac{(u_\gamma)_i^j}{\overline{\nu}_i} + \dfrac{\delta - 1}{n}}{1 + \dfrac{K(\delta - 1)}{n}}, \tag{3.10}$$

$$\mu_j^{\text{new}} = \frac{(\mathbf{s})_i^j}{(u_\gamma)_i^j}, \tag{3.11}$$

$$\mathbf{\Sigma}_j^{\text{new}} = \frac{\dfrac{b}{n}\mathbf{I} + \dfrac{(\mathbf{s}\mathbf{s}^T)_i^j - \mathbf{A} + (u_\gamma)_i^j \mathbf{B}}{\overline{\nu}_i}}{\dfrac{a-2}{n} + \dfrac{(u_\gamma)_i^j}{\overline{\nu}_i}} \tag{3.12}$$

where

$$\mathbf{A} = (\mathbf{s})_i^j (\mu_j^{\text{new}})^T + \mu_j^{\text{new}} (\mathbf{s}^T)_i^j \ , \qquad \mathbf{B} = \mu_j^{\text{new}} (\mu_j^{\text{new}})^T,$$

$\mathbf{I}$ is the identity matrix, $K$ is the number of mixture components and $n$ is the total number of observed samples (see the details below). Scalars $a$, $b$ and $\delta$ are parameters of conjugate priors induced by the MAP solution (see Appendix 3.A for more details).

We ran extensive experiments with both synthetic and real data from light transport simulation and by comparing our algorithm results to the reference solutions, we have concluded that the most suitable values for our application are $a = 2.01$, $b = 5 \times 10^{-4}$, $\delta = 1.01$. Likewise, we found that both the on-line and the off-line stepwise EM algorithms achieve the best results when the M-step is executed every $m = 10$ samples (see Alg. 2) and with the stepsize parameter $\alpha = 0.7$ (see Sec. 3.3). We decided to use $K = 8$ components in the mixture as it proved sufficient in all tested scenes.

**Differences between the off-line and the on-line versions.** The number of observed samples $n$ in Equations (3.10) and (3.12) governs the effect of our prior beliefs. The more samples we have observed the weaker the effect of priors. In *on-line* stepwise EM, we simply set $n$ to the current value of the index of sufficient statistics $i$. However, to fully exploit the MAP approach and thus to prevent over-fitting in our *off-line* stepwise EM, it is necessary to set $n = \min(i, N)$. This is necessary because the algorithm iterates over the same batch of $N$ samples multiple times (see Fig. 3.4) before it converges and the index $i$ could be much higher than the actual number of unique observed samples.

### 3.4.3 Caching of Distributions

Once the EM algorithm creates a hemispherical distribution $p(\omega|\mathbf{y})$ at $\mathbf{y}$, we cache it for reuse at nearby locations. The main reason for using a persistent cache of distributions is to enable their on-line refinement. Our cache is inspired by the traditional lazy evaluation scheme from (ir-)radiance caching [Křivánek et al., 2005, Ward et al., 1988]. It maintains a set of distributions, and for each query point it either returns an existing distribution or creates (i.e. trains) and stores a new one. Our caching scheme, however, exhibits an important difference from (ir-)radiance caching. While (ir-)radiance caching *blurs* the stored values to obtain a biased, yet perceptually plausible result, we strive for an unbiased result.

**Spacing of cached distributions.** To achieve a good performance, we space the distributions so that they adapt to the angular frequency of the radiance (or importance) function, as illustrated in Fig. 3.5 (e.g. for radiance, more distributions should be created in sharp caustics). To space the distributions, we assign to each of them a *validity radius* that determines the maximum spatial distance where the distribution can be reused. The validity radius for a distribution is computed as a harmonic mean of the validity radii of its individual mixture lobes (i.e. GMM components), weighted by the mixing coefficients $\pi_j$. To determine the validity radius of a lobe, we first predict how the lobe would change if we observed the environment from a slightly different position (see Fig. 3.6). We then set the validity radius such that the Kullback-Leiber divergence [Bishop, 2006] between the original and the changed lobe stays below a certain threshold for any location within the validity radius. We additionally improve the spacing of distributions by the neighbour clamping heuristic [Křivánek et al., 2006]. We also clamp the validity radii to be between 0.5 and 1 times the distance between the distribution position $\mathbf{y}$ and the furthest particle used for its training. We provide details of validity radius computation in Appendix 3.B.

**Distribution reuse.** We have experimented with different interpolation strategies and concluded that simply re-using the nearest suitable distribution is the most robust solution. When we query the cache at the position $\mathbf{y}$, we search for $M$ nearby distributions which include the query point $\mathbf{y}$ within their validity radius. From among those we select one with a suitable position and normal orientation. Specifically, we choose the distribution that minimizes

$$\frac{\|\mathbf{y} - \mathbf{y}_i\|^2}{h} + 2\sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i}, \tag{3.13}$$

where $\mathbf{y}_i$ and $\mathbf{n}_i$ are the position and the normal of $i$-th candidate respectively, $\mathbf{n}$ is the normal at $\mathbf{y}$ and $h$ is the distance to the furthest of the $M$ candidates. If there is no suitable distribution that could be reused, we create a new one.

### 3.4.4 Environment Emission Sampling

We have observed that guiding only the scattering directions is insufficient in complex scenes with environment lighting. This is because most particles emitted

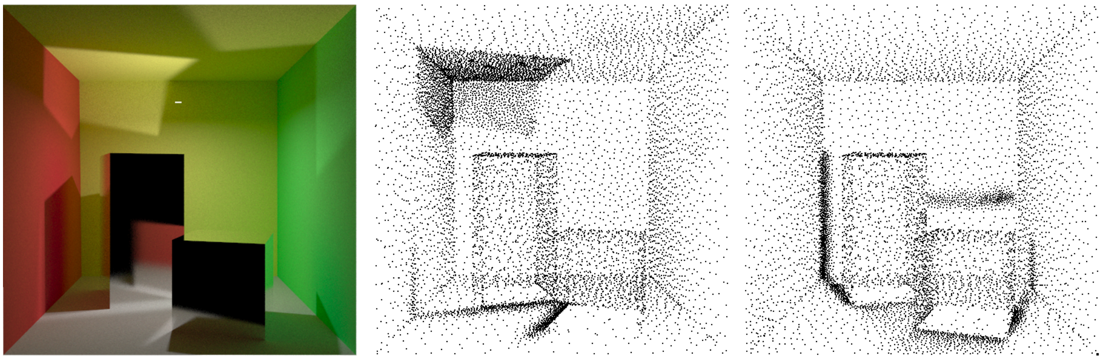


Figure 3.5: An example of radiance and importance caches in a Cornell box scene (left). The black dots represent positions of distributions used for guiding camera (center) and light paths (right). The cache adapts to the scene by placing more records where the radiance/importance function contains high angular frequencies.

from the environment fail to enter important parts of the scene through small openings (see Fig. 3.1). Here, we present our approach for driving the particle emission from an environment light source by visual importance (see Sec. 2.2). Emission from other light source types is left for future work.

An environment light defines the emitted radiance $L_e(\omega)$ over directions $\omega$. Emitting a particle from an environment light requires sampling a joint PDF $p(\mathbf{x}, \omega) = p(\omega)p(\mathbf{x}|\omega)$ that we factor into the PDF $p(\omega)$ to sample the particle direction $\omega$, and $p(\mathbf{x}|\omega)$ to sample the starting position $\mathbf{x}$ on a disk outside the scene that is perpendicular to $\omega$ [Georgiev, 2012, Pharr and Humphreys, 2010]. These distributions are trained by the importons that have left the scene (see Fig. 3.7a) during the *importance distributions update* step in Fig. 3.3.

**Directional distribution.** We compute $p(\omega)$ as a product of $p_{\mathrm{L}}(\omega) \propto L_e(\omega)$ given by the environment map and of $p_{\mathrm{W}}(\omega)$ that is the directional distribution of importance reconstructed from the importons that left the scene. We represent $p_{\mathrm{W}}(\omega)$ with a fixed-resolution bitmap. In each pixel, the PDF is computed from importon directions using a progressive kernel density estimate [Hachisuka et al., 2008]. To avoid introducing bias due to zero probability of sampling directions that did not receive any importons, we combine sampling from the product $p_{\mathrm{L}}(\omega) \times p_{\mathrm{W}}(\omega)$ and from $p_{\mathrm{L}}(\omega)$ via multiple importance sampling [Veach, 1997].

**Position distributions.** The distributions $p(\mathbf{x}|\omega)$ defined on a disk perpendicular to $\omega$ (see Fig. 3.7b) are stored in a unit sphere cache indexed by $\omega$. This allows to reuse the stored distributions for nearby directions and enables their on-line training. To represent $p(\mathbf{x}|\omega)$, we train our GMM (see Sec. 3.3) from importons that left the scene in a direction less than 1° from $\omega$. Prior to the training, these importons are projected onto the perpendicular disk (see Fig. 3.7a).

### 3.4.5 Russian Roulette

Path termination via Russian Roulette (RR) [Arvo and Kirk, 1990] affects the distribution of particles together with their weights, without changing the expected value of the quantity they represent (see Sec. 2.6). This is commonly called *biasing*

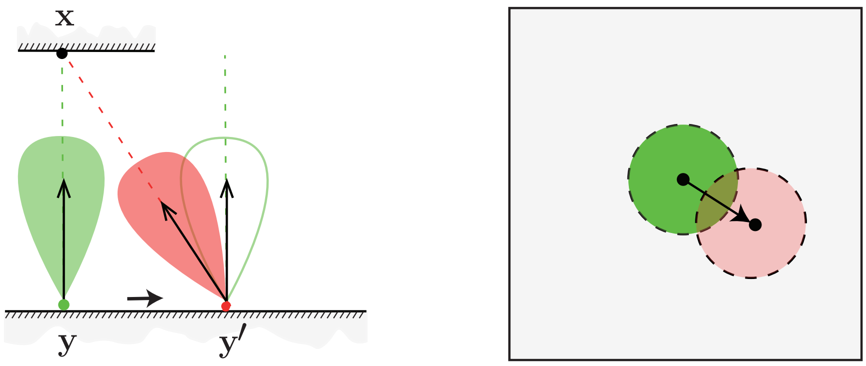


Figure 3.6: Calculation of the lobe validity radius. Left: We assume that the green lobe of a distribution at the position $\mathbf{y}$ corresponds to an importance/radiance highlight at the position $\mathbf{x}$ seen from $\mathbf{y}$ along the lobe axis. If we move the lobe to a new position $\mathbf{y}'$, the direction of the lobe (red) pointing towards the highlight changes. The validity radius is set such that the resulting change of the lobe, measured by the KL-divergence, does not exceed a given threshold. Right: The same situation in the unit square domain where the corresponding Gaussian distribution is defined and where the KL-divergence is actually measured.

in the neutron transport literature [Booth, 1985a, 2012b]. Our guiding method is also a biasing technique: we strive to reduce variance by guiding particles toward important areas (e.g. in front of the camera). However, combining different biasing techniques could be counterproductive. When we use our guiding method, we must not base the random walk termination by RR on a local decision (derived from e.g. the surface albedo) as is commonly done in light transport simulation [Pharr and Humphreys, 2010]. This approach might result in a termination of particles that could eventually yield an important contribution.

Ideally, we should adapt Russian roulette to the distribution of radiance (or visual importance for photons) in the scene [Haghighat and Wagner, 2003]. We devote to this problem Chapter 4 where we suggest to terminate (or split) particle paths according to their expected contribution to the computed solution. We also investigate the impact of such Russian roulette and splitting to path guiding (Sec. 4.5).

Nonetheless, in this chapter, *to demonstrate and isolate benefits* of our guiding method, we base our Russian roulette on particle weights. Note that due to guiding, important areas will contain many particles with very small weights[4] and we want to minimize the chance that these particles are killed. Thus we set the RR survival probability to

$$\min\{\frac{\nu_\mathrm{i}}{\nu_\mathrm{o}^\mathrm{e}}\frac{1}{\nu_\mathrm{thr}}, 1\}, \tag{3.14}$$

where $\nu_\mathrm{i}$ is the particle weight at the current scattering location, $\nu_\mathrm{o}^\mathrm{e}$ is the particle weight just after emission before any scattering and $\nu_\mathrm{thr}$ is a fixed, empirically determined threshold that we set to $10^{-6}$ for all our renderings. In other words,

[4] Because particles form an unbiased representation of radiance/visual importance, an increase in particle density must result in a decrease in their respective weights.

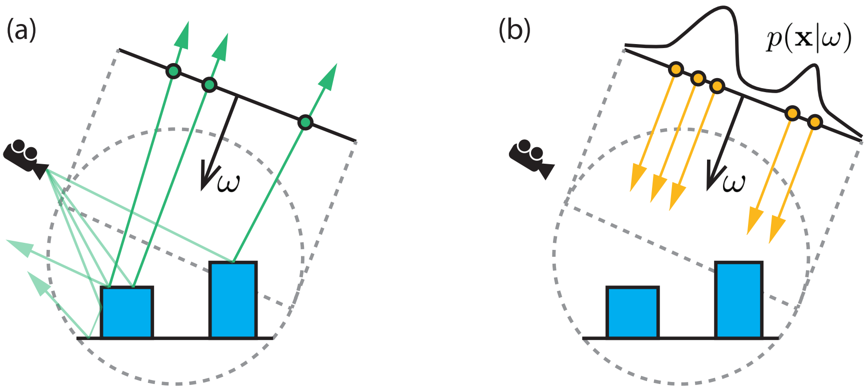


Figure 3.7: To learn the importance distribution $p(\mathbf{x}|\omega)$ for sampling starting positions of light paths from the environment, conditional on the direction $\omega$, we construct a disk perpendicular to $\omega$ outside the scene. A GMM model of the distribution is trained from importons that leave the scene in a direction less than 1° from $\omega$ (a). The position $\mathbf{x}$ of the light paths emitted in a direction close to $\omega$ is sampled using the learned distribution $p(\mathbf{x}|\omega)$ (b).

we do not attempt to terminate a particle unless its weight drops at least $\nu_{\mathrm{thr}}$ times from its initial value $\nu_{\mathrm{o}}^{\mathrm{e}}$. Note that in practice, for camera particles, $\nu_{\mathrm{o}}^{\mathrm{e}}$ equals to 1 while for photons that are emitted according to light source power profile it is equal to the total power of all sources in the scene.

Additionally, to avoid excessive particle weights that could accumulate over multiple scattering events, we split the particle path whose original weight $\nu_{\mathrm{o}}^{\mathrm{e}}$ was increased more than twice. This is relatively conservative splitting rate that results into a very small number of splits. The reason is that, in our scheme, at maximum, the particle weight can increase twice per scattering event (under the assumption that we achieve perfect importance sampling of BSDF). We show this claim in Appendix 3.C.

## 3.5 Applications and Results

We first demonstrate the flexibility of the Gaussian mixture model in rendering. Then we show that guiding various path-sampling algorithms using our progressively trained distributions provides superior rendering results in complex, highly occluded scenes.

### 3.5.1 Flexibility of the Gaussian Mixture Model

We compare our GMM to Jensen's histograms [1995] and Hey and Purgathofer's hemispherical footprints [2002] in terms of their ability to model distributions encountered in rendering. Fig. 3.8 shows a simple Cornell box scene with two glossy blocks. It is rendered with path tracing using the histogram model with two different histogram resolutions, the hemispherical footprints model, and finally our GMM. Note that the on-line learning, which is discussed in the following section, is not applied so that the comparison is fair. Instead, during the training

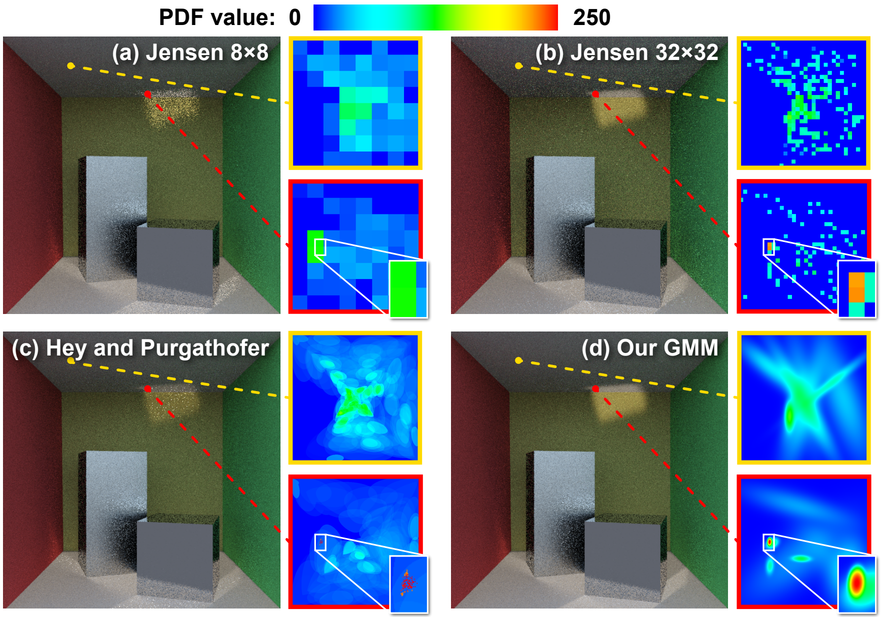


Figure 3.8: Demonstration of the superior flexibility of the parametric Gaussian mixture model (GMM) over previously used models. Four renderings of a Cornell box scene with diffuse walls and two glossy blocks lit by the sun are rendered by guided path tracing using Jensen's [1995] method with the histogram resolution of 8×8 (a) and 32×32 (b), Hey and Purgathofer's [2002] hemispherical footprints (c), and with our GMM (d). The distributions trained at two selected locations in the scene are also visualized. One distribution contains low-frequency illumination while the other contains a sharp peak caused by a reflection of the sun.

phase, we emit 5M photons in a single batch and all the models are trained from 250 nearest photons.

The figure shows that higher histogram resolution captures high frequencies in the incoming radiance distribution inside caustics, while the ability to represent low frequencies deteriorates at the same time due to over-fitting. Thus, while the quality of caustics at the histogram resolution of $32 \times 32$ pixels is superior to the quality at the $8 \times 8$ resolution, the noise is increased on the walls.

Hemispherical footprints are more flexible than the fixed histogram grid as they take into account the directional density of particles. However, the method is closely related to kernel density estimation and as such it suffers from the optimal bandwidth selection problem [Silverman, 1986]. Visualization of a distribution inside the sun's reflection suggests insufficient generalization – the distribution is discontinuous between individual observed samples in its peak.

Finally, the GMM with only 8 components in the mixture exhibits sufficient flexibility to represent both low frequencies and sharp peaks in the distributions. Note that when compared to the other methods, the GMM excels at generalizing from the observed data (i.e. it avoids discontinuities between the individual samples). This is further supported by Fig. 3.9, which shows that the GMM provides superior rendering results in a scene with difficult visibility.

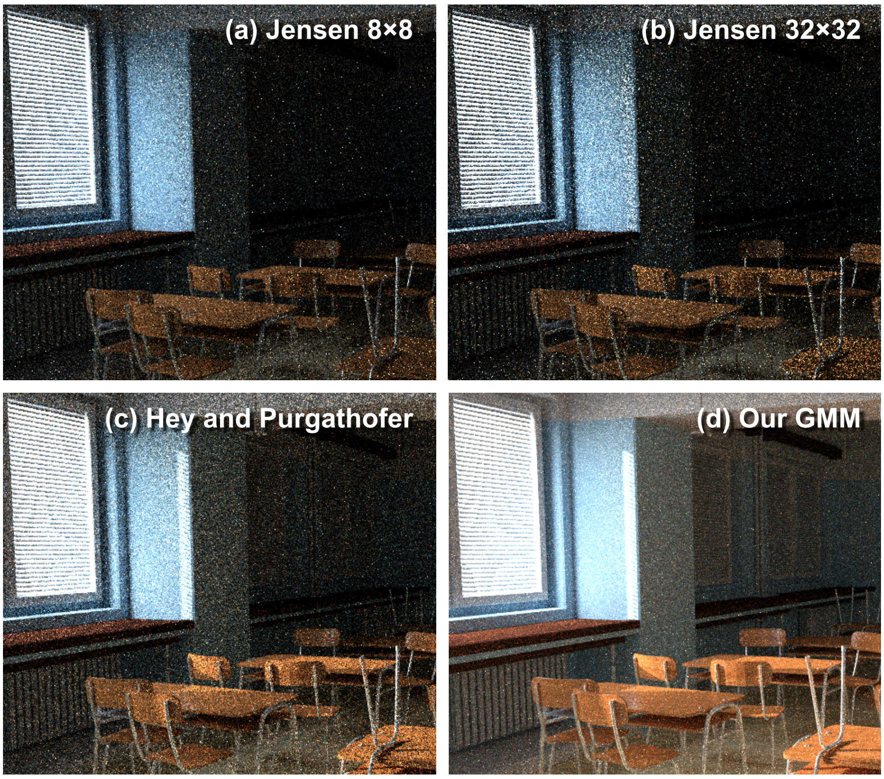


Figure 3.9: Equal-time (1h) comparison of different distribution models in a scene with difficult visibility. Light enters the classroom from the sun and the sky through small gaps between the window blinds. The scene was rendered by path tracing guided by 8×8 (a) and 32×32 (b) histogram model, hemispherical footprints model (c), and our Gaussian mixture model (d). Each method was given a fixed number of emitted photons (500M) in one unguided tracing step. Each distribution was then trained from a single batch of 500 nearest photons without additional on-line learning.

### 3.5.2 On-line Learning Results

In Figs. 3.12–3.14, we compare classical implementations of (1) path tracing (PT), (2) bidirectional path tracing (BDPT) and (3) vertex connection and merging (VCM) [Georgiev et al., 2012b] to the same implementations guided by our method with the distributions trained in the on-line (progressive) manner. We also show images rendered by (4) Veach's Metropolis light transport with manifold exploration (Veach MLT) [Jakob and Marschner, 2012]. We also present corresponding $L_1$ error plots in Fig. 3.10. The supplemental material contains additional results for progressive photon mapping (PPM) and other flavours of MLT (Kelemen MLT [2002] and energy redistribution path tracing [Cline et al., 2005]) as well as RMSE plots.

**Setup.** Our method uses 30 training passes in all the presented scenes. We set the maximum path length to 40 bounces. All images, save the references, were rendered in 1 hour (including the training phase) on a single Intel Core i7-2600K CPU using 8 logical cores. The reference images were rendered with BDPT for 10–60 days. We used the Corona Renderer [Karlík, 2009] to produce all the results except the MLT images, which were rendered in Mitsuba [Jakob,

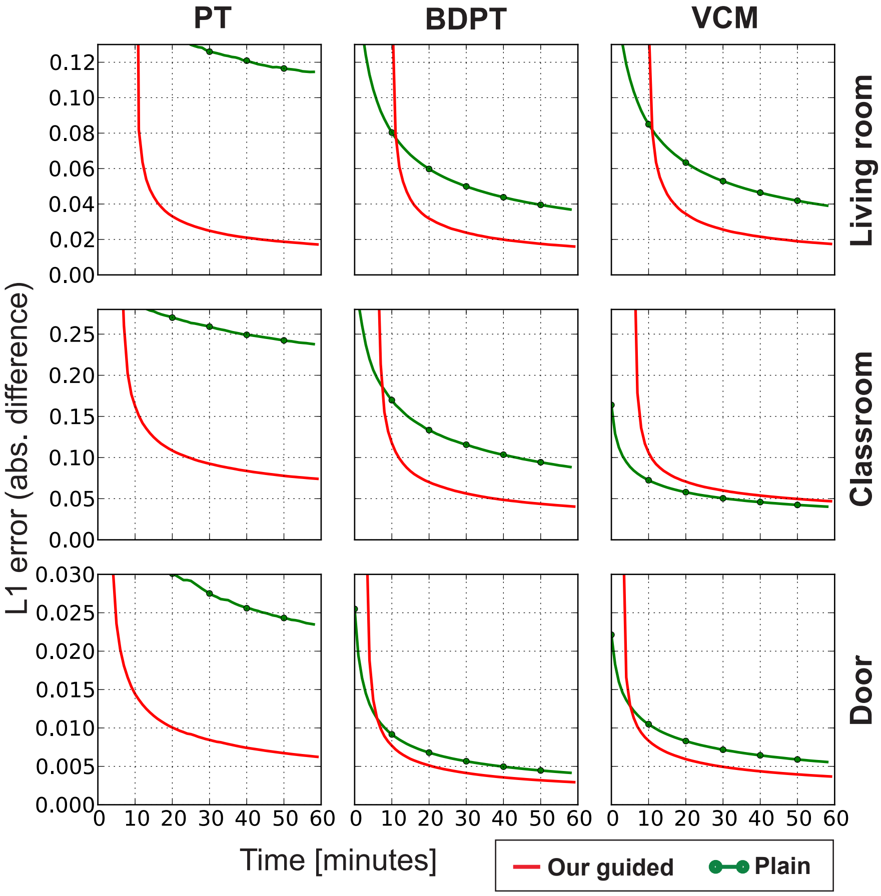


Figure 3.10: Time dependence of $L_1$ error for 60 minutes of rendering. The learning phase of our method is included in the comparison, so the graphs for the guided versions do not start at zero time.

2010] with default settings. We verified that our BDPT implementations in both renderers converge to the same result.

**Scenes.** In Figs. 3.12–3.14, we show renderings of three scenes that feature difficult visibility and contain many locations that are poorly sampled with regular path sampling-based methods. Two of the scenes (Living room and Classroom) feature environment lighting and use our method for importance-driven emission sampling. All the light in rendered images is due to indirect illumination.

The **Living room** scene allows only a small fraction of the light from the sun and the sky to enter the room through a glass window and a small gap between the curtains. The walls are diffuse and there are a few semi-glossy objects, including the floor. Note that even BDPT and VCM struggle to resolve the image in the dark closet on the left. The scene was rendered at a resolution of 1024 × 576 pixels. One million photons and importons were emitted in each training pass. The training phase took 10.3 minutes.

The **Classroom** scene is lit by the sun and the sky through window blinds. For regular MC algorithms, it is especially difficult to sample the dark half of the classroom. Windows and highly glossy chairs and table legs together with a semi-glossy floor create specular–glossy–glossy light transport paths that form many caustics/indirect highlights. While our method improves sampling of the

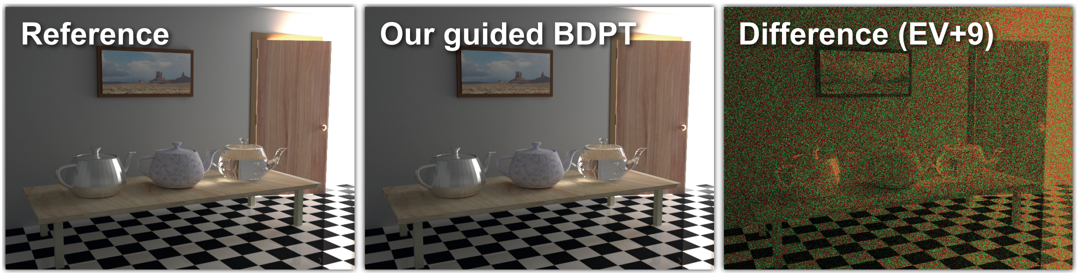


Figure 3.11: Our method converges to the reference image computed by plain BDPT (left). We rendered the scene for approximately two days with our guided BDPT that used 30 TP (middle). The reference image was rendered by BDPT for approximately 10 days. The uniform distribution of positive (green) and negative (red) differences (right) suggests that any residual error is only due to variance. Brightness of the difference image was multiplied by $2^9$.

caustics and the dark parts of the scene, sampling of the glossy-glossy highlights remains a challenge. The reason is that we do not sample from the product of the BSDF and the incoming radiance/importance, but rather combine the two via multiple importance sampling. The scene was rendered at a resolution of $960 \times 480$ pixels. 0.5 million photons and importons were emitted in each training pass. The training phase took 7 minutes.

In the **Door** scene, light enters the room through a small slit. This is a recreation of the well-known scene from Veach and Guibas' paper [1997] provided by Lehtinen et al. [2013]. To make the scene more realistic, we have made the area light source much smaller and used more light bounces. The scene was rendered at a resolution of $800 \times 600$ pixels. 300k photons and importons were emitted in each training pass. The training phase took 3.75 minutes.

**Error and convergence.** Fig. 3.10 shows the dependence of the $L_1$ error on time in all three scenes, comparing both the classical and our guided versions of the algorithms. Fig. 3.11 demonstrates that our method converges to the reference.

**On-line learning.** To demonstrate progressive improvement of distributions during the training phase, we rendered the *Living room* scene several times with guided BDPT. We used 2, 5 and 30 training passes, while all other settings were kept the same. In Fig. 3.1, we present an equal-time comparison (1 hour) including the time spent on the training phase. The results reveal that the time spent on additional training passes is quickly amortized by the superior performance of the subsequent guided rendering.

**Discussion.** Guided BDPT and PPM (see the supplemental material) yield superior results compared to their classical versions. However, although VCM is a combination of BDPT and PPM, the improvement of guided VCM is only subtle over guided BDPT. We suspect that the guiding might render some path sampling techniques – that would otherwise be an essential component of the combined algorithm – less important.

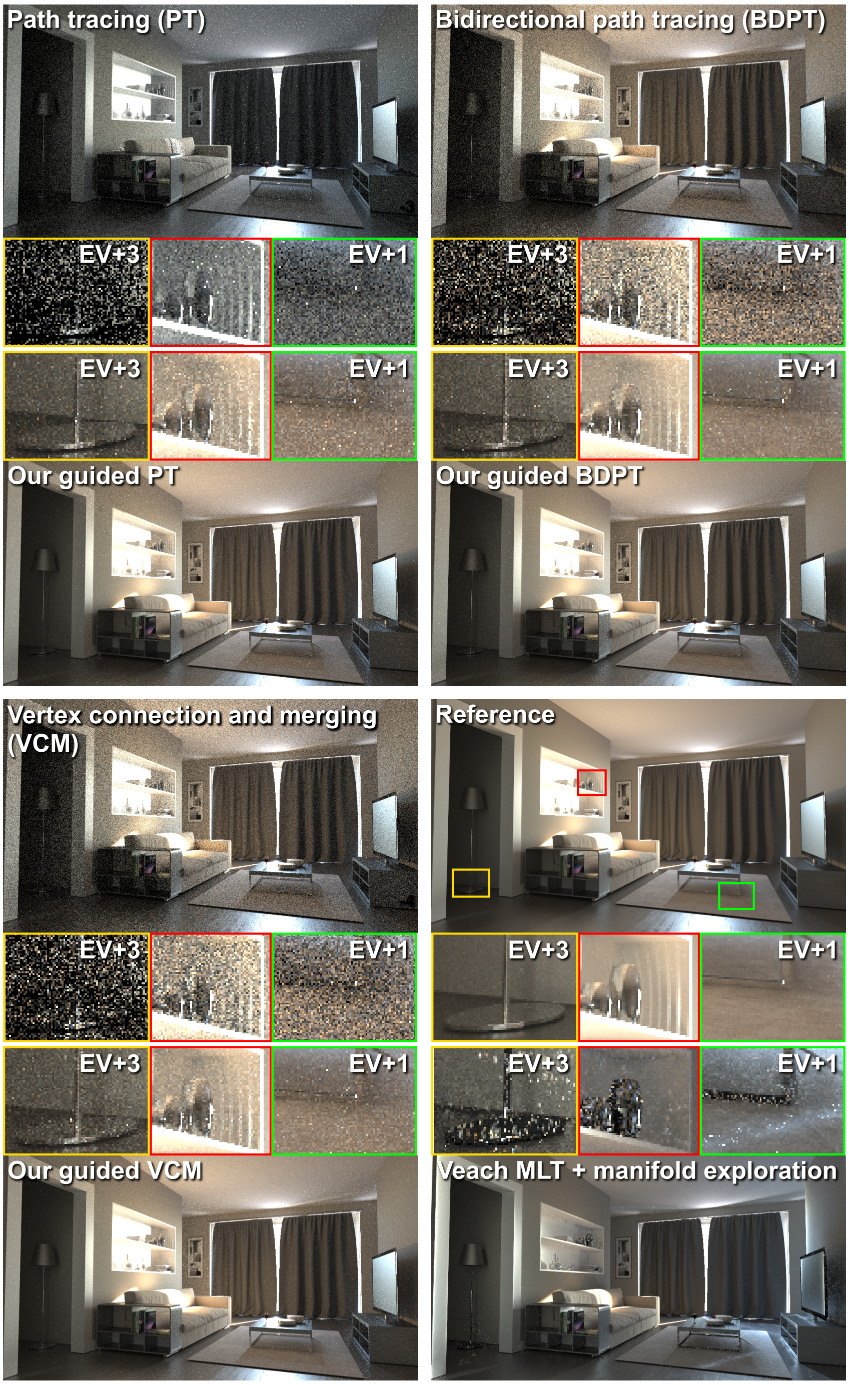


Figure 3.12: The living room scene with difficult visibility rendered with path tracing (PT), bidirectional path tracing (BDPT), vertex connection and merging (VCM), and their respective versions guided by our method. We also present reference image and the result of Veach's Metropolis light transport (MLT) with manifold exploration. All images, except the reference, were calculated in one hour, including the time spent on 30 training passes. The results show that the overhead of our method is amortized by the improved sampling, as the noise levels are reduced in all tested algorithms, especially in dark areas. ("EV+$v$" in the insets refers to a multiplication of the image brightness by $2^v$.)

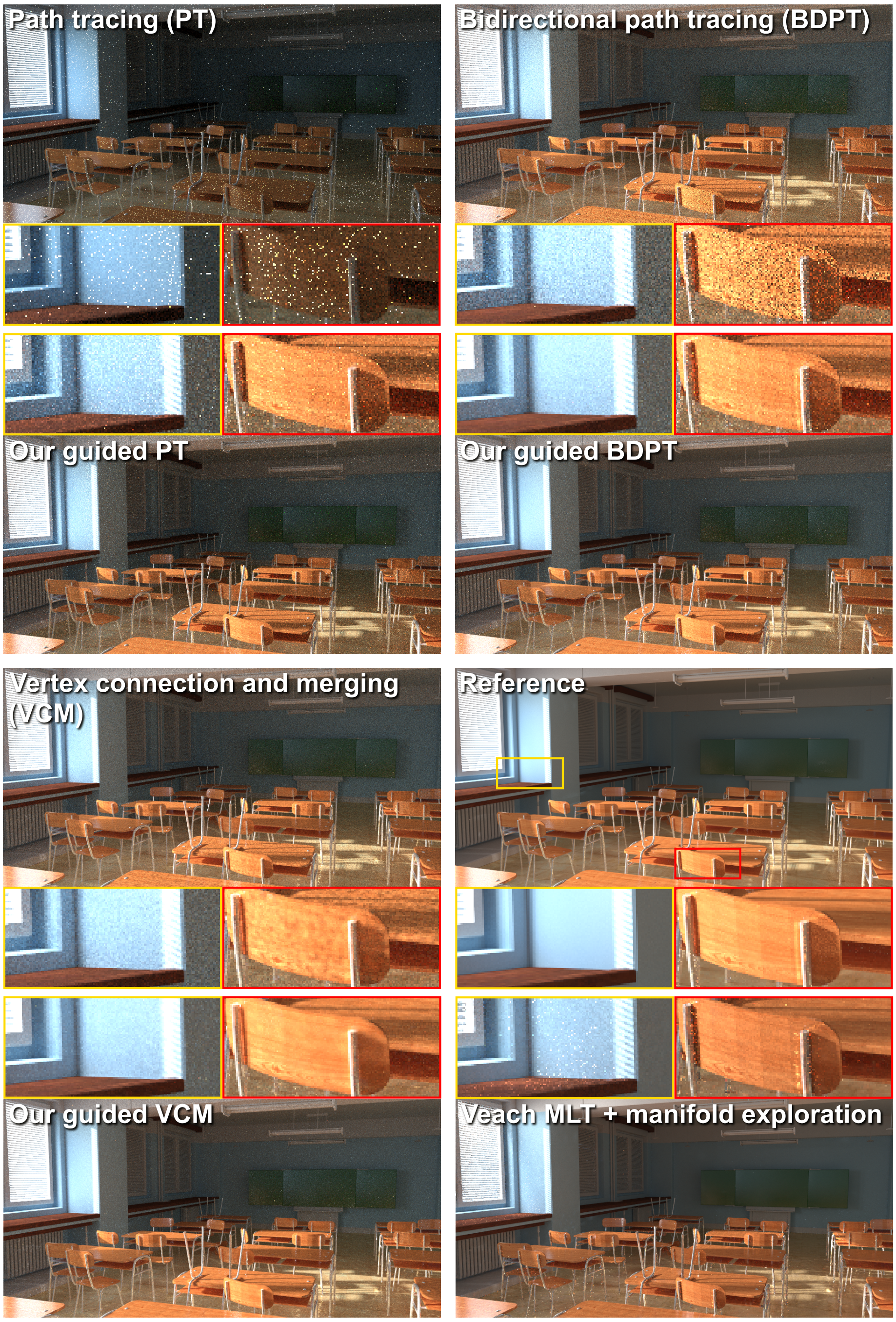

Path tracing (PT)
Bidirectional path tracing (BDPT)
Our guided PT
Our guided BDPT
Vertex connection and merging (VCM)
Reference
Our guided VCM
Veach MLT + manifold exploration

Figure 3.13: The classroom scene with difficult visibility rendered with path tracing (PT), bidirectional path tracing (BDPT), vertex connection and merging (VCM), and their respective versions guided by our method. We also present reference image and the result of Veach's Metropolis light transport (MLT) with manifold exploration. All images, except the reference, were calculated in one hour, including the time spent on 30 training passes. The results show that the overhead of our method is amortized by the improved sampling, as the noise levels are reduced in all tested algorithms, especially in dark areas.

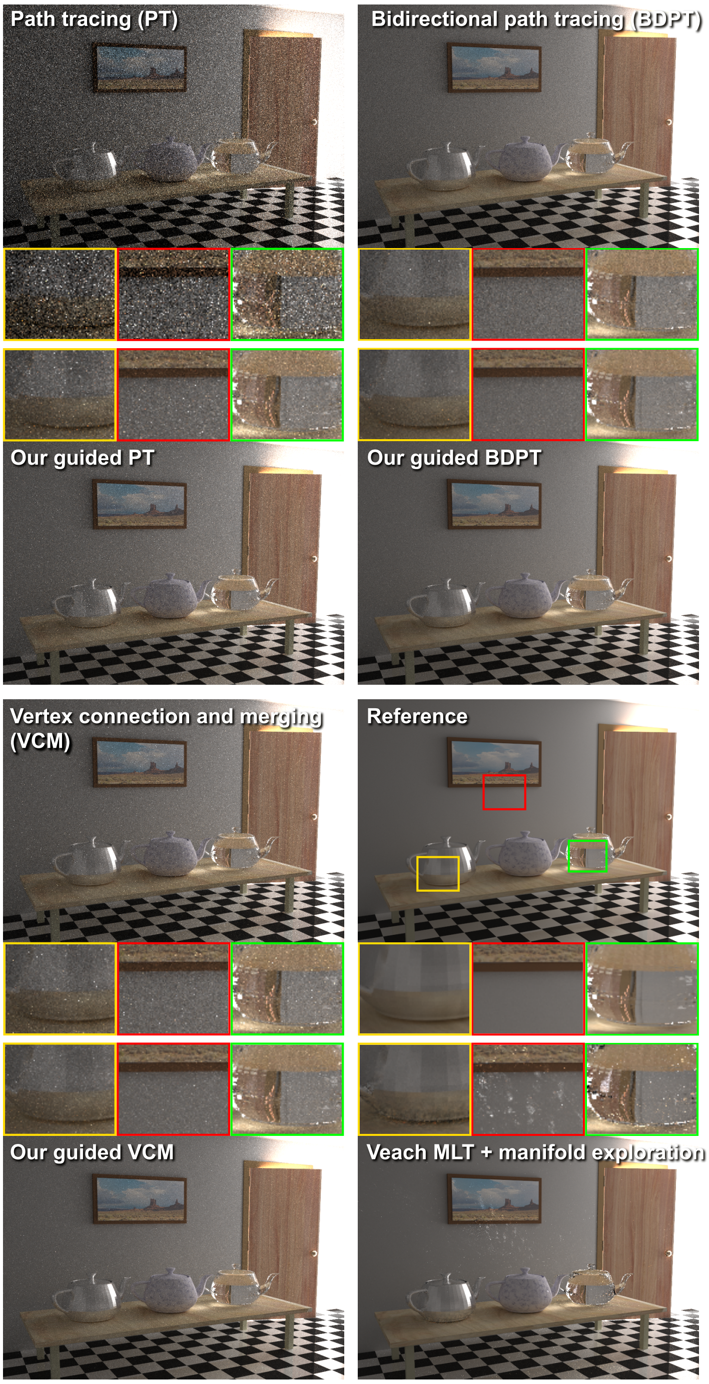

Path tracing (PT)
Bidirectional path tracing (BDPT)
Our guided PT
Our guided BDPT
Vertex connection and merging (VCM)
Reference
Our guided VCM
Veach MLT + manifold exploration

Figure 3.14: The door scene with difficult visibility rendered with path tracing (PT), bidirectional path tracing (BDPT), vertex connection and merging (VCM), and their respective versions guided by our method. We also present reference image and the result of Veach's Metropolis light transport (MLT) with manifold exploration. All images, except the reference, were calculated in one hour, including the time spent on 30 training passes. The results show that the overhead of our method is amortized by the improved sampling, as the noise levels are reduced in all tested algorithms, especially in dark areas.

## 3.6 Discussion, Limitations and Future Work

**Alternative distribution models.** We have experimented with a range of alternatives to Gaussian mixtures. The *von Mises-Fisher* (vMF) distribution [1953] is defined directly on the unit sphere, but its isotropic shape is not suitable for fitting highly anisotropic structures in the directional domain, often produced by caustics. The *Kent* distribution [1982] is a directional distribution that does support anisotropy, but becomes bi-modal for a range of parameters and is numerically unstable. An anisotropic spherical function [Xu et al., 2013] derived from the Bingham distribution [1974] was developed for interactive rendering. However, at the moment, there is no known analytical sampling procedure. We further experimented with an anisotropic distribution similar to the lobe of the *Ward BSDF* [Ward, 1992], but its learning was too expensive.

**Limitations and Future Work.** The overhead of our method in the rendering phase, that may offset its advantages in simple scenes, comes mostly from querying the cache. This takes 26% of the execution time for path tracing, and up to 45% for bidirectional methods, since they need to access both radiance and importance caches at each path vertex. Optimization should focus on this aspect, since sampling and evaluation of the GMMs each takes only about 5% of the total time.

Importance sampling based on our distributions can occasionally generate excessive particle weights (training phase) or path contributions (rendering phase) and produce spiky image noise. Splitting and Russian roulette based on our importance/radiance distributions and particle weights [Haghighat and Wagner, 2003] could mitigate this issue and improve overall performance (see Chapter 4).

While the fixed number of components in the mixture may be insufficient to capture some complex distributions, we have not encountered any problems due to this limitation in our tests. Nonetheless, adaptive determination of the number of components would be an interesting avenue for future work.

Because the distributions created in the later training passes have access to fewer particles, they may be less refined than the distributions created earlier. Adaptive refinement of the distributions during rendering could help resolve this issue.

Since our distributions only model incoming radiance or importance, we cannot provide good importance sampling of some complex glossy-to-glossy interreflections. This could be alleviated by sampling from the product of the incoming radiance or importance and the BSDF.

Our importance-driven emission is currently limited to environment light sources. Using our distributions to sample emission from other light source types, as well as for radiance-driven emission of importons from the camera, would improve the efficiency of our method.

Finally, our method shares the same overall goal with Metropolis light transport, that is globally optimized importance sampling of entire light transport paths. It would be interesting to see if the two approaches can complement each other to achieve further benefits.

## 3.7 Conclusion

In this chapter, we have proposed the use of a parametric mixture model to represent directional distributions for importance sampling in Monte Carlo light transport simulation. The core of our approach is an on-line learning procedure that allows one to train the distributions from a potentially infinite stream of particles. With this approach, we can recover good importance sampling distributions in difficult lighting configurations, where an excessively large number of particles would otherwise be necessary. This, in turn, enables rendering scenes with difficult visibility, where the existing state-of-the-art methods are inefficient.

# Appendix

## 3.A Derivation of MAP Update Formulae

In this appendix, we provide a derivation of the update formulae $\overline{\theta}$ for the model parameters in our stepwise expectation-maximization algorithm described in Sec. 3.4.2 that supports weighted particles. The same formulae apply to both the *off-line* and *on-line* versions of stepwise EM.

### 3.A.1 MAP and Conjugate Priors

To alleviate over-fitting that is associated with *maximum likelihood* estimation, we pursue a *maximum a posteriori* (MAP) solution. In other words, observing a set of samples $\mathbf{S}$, we seek the mode of the *posterior* distribution $p(\theta|\mathbf{S})$ over mixture model parameters $\theta$, given by the Bayes' theorem: $p(\theta|\mathbf{S}) \propto p(\mathbf{S}|\theta)p(\theta)$ (i.e. posterior $\propto$ likelihood $\times$ prior). To enable this Bayesian treatment, we have to express our prior beliefs about the source of our observed samples via the prior distribution $p(\theta)$. A good choice are *conjugate priors* that take the same functional form as the resulting posterior distribution and therefore lead to a greatly simplified Bayesian analysis [Bishop, 2006].

A particular choice of the conjugate prior $p(\theta)$ which expresses our prior beliefs about the covariance matrix $\mathbf{\Sigma}_j$ and the mixing coefficients $\pi_j$ is:

$$\mathrm{Dir}(\pi_1, \ldots, \pi_K|\delta_1, \ldots, \delta_K) \prod_{j=1}^{K} \mathrm{Wish}(\mathbf{\Sigma}_j|a_j, b_j\mathbf{I}). \tag{3.15}$$

This is a product of a conjugate Dirichlet prior on mixing coefficients $\mathrm{Dir}(\ldots)$ with hyper-parameters $\delta_j > 0$ and isotropic conjugate Wishart priors $\mathrm{Wish}(\mathbf{\Sigma}_j|a_j, b_j\mathbf{I})$ on the covariance matrix of every mixture component $j$. Here $\mathbf{I}$ is the identity matrix, $K$ is the number of components in the mixture, $a_j > d-1$, $b_j > 0$ are hyper-parameters, and $d = 2$ is the dimension. Bishop [2006] provides details on the use of Dirichlet and Wishart distributions as conjugate priors.

We base our MAP solution on the prior distribution in the form of Equation (3.15), that is recommended by Gauvain and Lee [1994] in the context of batch EM. Unlike Gauvain and Lee, we do not take any prior assumptions about the Gaussian means $\mu_j$, because there is no reason to a priori prefer one lobe direction over another. In our final solution, we use the same hyper-parameters $a$, $b$ and $\delta$ (denoted as $\nu$ in Guavain's text) for all components so that $\forall j \in \{1, \ldots, K\}$; $a_j = a$, $b_j = b$ and $\delta_j = \delta$. Nevertheless, for the sake of generality, we provide the derivation with possibly different hyper-parameters for each mixture component.

### 3.A.2 Derivation Overview

We provide a derivation of the update formulae for the covariance matrix $\mathbf{\Sigma}_j$ of each Gaussian $j$ in the mixture (Eq. (3.12)) and for their respective mixing coefficients $\pi_j$ (Eq. (3.10)). The formula for updating the mean $\mu_j$ of each Gaussian $j$ (Eq. (3.11)) is straightforward – we just normalize the weighted sum of observed samples $\mathbf{s}_q$.

The derivation of the update formulae for both $\mathbf{\Sigma}_j$ and $\pi_j$ follow the same steps. We start from the formulae given by Gauvain and Lee [1994] that describe a MAP update of Gaussian mixture model (GMM) parameters in the batch EM algorithm (see Sec. 3.2 in the paper). Their formulae do not account for weights of observed samples. We generalize these results to stepwise EM while taking sample weights into account. Our generalization proceeds in three steps:

a) We express the parameters of each mixture component $j$ in terms of the batch EM sufficient statistics $\mathbf{u}^j_{N-1}$.

b) We use the fact that stepwise EM is a generalization of batch EM and replace the use of the batch EM statistics $\mathbf{u}^j_{N-1}$ with the stepwise EM statistics $\mathbf{u}^j_i$.

c) Finally, we interpret the weight $\nu_q$ associated with every observed sample $\mathbf{s}_q$ as its multiplicity.

### 3.A.3 Covariance Matrices

**Step a.** The Gauvain's update formula for the matrix $\mathbf{\Sigma}^{'}_j$ of $j$-th Gaussian in the mixture reads:

$$\mathbf{\Sigma}^{'}_j = \frac{b_j\mathbf{I} + \sum_{q=0}^{N-1}\gamma_{qj}(\mathbf{s}_q - \mu_j)(\mathbf{s}_q - \mu_j)^T}{(a_j - 2) + \sum_{q=0}^{N-1}\gamma_{qj}}, \tag{3.16}$$

where $\mathbf{I}$ is the identity matrix, $a_j$ and $b_j$ are hyper-parameters of the Wishart's distribution priors and $\gamma_{qj}$ is the responsibility of a component $j$ for an observed sample $\mathbf{s}_q$ (see Eq. (3.3)).

By using simple algebra, we get:

$$(\mathbf{s}_q - \mu_j)(\mathbf{s}_q - \mu_j)^T = \mathbf{s}_q\mathbf{s}_q^T - \mathbf{s}_q(\mu_j)^T - \mu_j\mathbf{s}_q^T + \mu_j(\mu_j)^T. \tag{3.17}$$

Substituting (3.17) into (3.16) and multiplying both the nominator and the denominator by $\frac{1}{N}$, the formula for $\mathbf{\Sigma}^{'}_j$ becomes:

$$\mathbf{\Sigma}^{'}_j = \mathbf{\Sigma}^{'}_j\frac{\frac{1}{N}}{\frac{1}{N}} \tag{3.18}$$

$$= \frac{\frac{b_j\mathbf{I}}{N} + \frac{1}{N}\sum_{q=0}^{N-1}\gamma_{qj}\mathbf{s}_q\mathbf{s}_q^T - \mathbf{A} + \mathbf{B}}{\frac{(a_j-2)}{N} + \sum_{q=0}^{N-1}\frac{\gamma_{qj}}{N}}, \tag{3.19}$$

where

$$\mathbf{A} = \frac{1}{N}\sum_{q=0}^{N-1}\gamma_{qj}\mathbf{s}_q\mu_j^T + \frac{1}{N}\sum_{q=0}^{N-1}\gamma_{qj}\mu_j\mathbf{s}_q^T$$

and

$$\mathbf{B} = \frac{1}{N}\sum_{q=0}^{N-1}\gamma_{qj}\mu_j\mu_j^T.$$

The *batch* sufficient statistics for $N$ samples (Eq. (3.4)) reads

$$\mathbf{u}^j_{N-1} = \frac{1}{N}\sum_{q=0}^{N-1}\gamma_{qj}\mathbf{u}(\mathbf{s}_q), \tag{3.20}$$

where the statistic $\mathbf{u}(\mathbf{s}_q) = (1, \mathbf{s}_q, \mathbf{s}_q\mathbf{s}_q^T)$ is based on an observed sample $\mathbf{s}_q$. By inspection of Equation (3.19), it is apparent that we have expressed Equation (3.16) in terms of the batch EM sufficient statistics $\mathbf{u}_{N-1}^j$.

**Step b.** The sufficient statistics in the *stepwise* EM formulation (Eq. (3.5)) reads

$$\mathbf{u}_i^j = (1-\eta_i)\mathbf{u}_{i-1}^j + \eta_i\,\gamma_{qj}\mathbf{u}(\mathbf{s}_q). \tag{3.21}$$

The sufficient statistics are expressed as a weighted sum with weights $\eta_i = i^{-\alpha}$, where $\alpha$ is the stepsize parameter.

For $\alpha = 1$, the batch EM sufficient statistics, Eq. (3.20), for $N$ samples and the stepwise EM sufficient statistics, Eq. (3.21), for the $N$-th sample (i.e. $i = N-1$) would be equivalent:

$$\mathbf{u}_{N-1}^j \equiv^{[\alpha=1]} \mathbf{u}_i^j. \tag{3.22}$$

We use this fact to obtain the MAP update formula of the covariance matrix from Eq. (3.19). If we write the sufficient statistics $\mathbf{u}_i^j$ as a triplet $\mathbf{u}_i^j = ((u_\gamma)_i^j, (\mathbf{s})_i^j, (\mathbf{s}\mathbf{s}^T)_i^j)$ where the first component is the weighted average of all responsibilities $\gamma_{qj}$, similarly $(\mathbf{s})_i^j$ is a vector, and $(\mathbf{s}\mathbf{s}^T)_i^j$ is a matrix, the stepwise update formula reads:

$$\mathbf{\Sigma}_j' = \frac{\frac{b_j\mathbf{I}}{N} + (\mathbf{s}\mathbf{s}^T)_i^j - \mathbf{A} + (u_\gamma)_i^j\mathbf{B}}{\frac{a_j-2}{N} + (u_\gamma)_i^j} \tag{3.23}$$

where

$$\mathbf{A} = (\mathbf{s})_i^j\mu_j^T + \mu_j(\mathbf{s}^T)_i^j\;, \qquad \mathbf{B} = \mu_j\mu_j^T. \tag{3.24}$$

Note that if $\alpha < 1$, the equivalence in Equation (3.22) does not hold. Nonetheless, we take the liberty to generalize the result in Equations (3.23) and (3.24) to values of $\alpha$ other than 1. In our implementation we use $\alpha = 0.7$.

**Step c.** Finally, we interpret the weight $\nu_q$ associated with every observed sample $\mathbf{s}_q$ as its multiplicity. The stepwise EM sufficient statistics in our weights-aware algorithm are given by Eq. (3.8):

$$\mathbf{u}_i^j = (1-\eta_i)\mathbf{u}_{i-1}^j + \eta_i\nu_q\gamma_{qj}\mathbf{u}(\mathbf{s}_q). \tag{3.25}$$

To obtain a correct result that takes the weights into account, we normalize these weighted statistics by the total sample weight (Eq. (3.9)):

$$\overline{\nu}_i = (1-\eta_i)\overline{\nu}_{i-1} + \eta_i\nu_i. \tag{3.26}$$

This results in the desired update formula for $\mathbf{\Sigma}_j$ (Eq. (3.12)), that respects the observed sample weights and provides the MAP solution:

$$\mathbf{\Sigma}_j = \frac{\frac{b_j}{n}\mathbf{I} + \frac{(\mathbf{s}\mathbf{s}^T)_i^j - \mathbf{A} + (u_\gamma)_i^j\mathbf{B}}{\overline{\nu}_i}}{\frac{a_j-2}{n} + \frac{(u_\gamma)_i^j}{\overline{\nu}_i}}.$$

Here $\mathbf{A}$ and $\mathbf{B}$ are given in Equation (3.24) and $n$ is the total number of currently processed samples. (Details on $n$ are given in the last paragraph of Sec. 3.4.2.) Note, that we let the effect of priors diminish with the number of observed samples rather than with the total observed weight $\overline{\nu}_i$. This helps to avoid over-fitting in the early stages of training, when there may be only a few observed samples with potentially enormous weights.

### 3.A.4 Mixing Coefficients

To derive the weights-aware update formula (Eq. (3.10)) for the mixing coefficients $\pi_j$, we follow the same procedure as in the above derivation of the covariance matrices $\mathbf{\Sigma}_j$. We start with the update formula given by Gauvain and Lee,

$$\pi_j = \frac{(\delta_j - 1) + \sum_{q=0}^{N-1} \gamma_{qj}}{\sum_{j=1}^{K}(\delta_j - 1) + \sum_{j=1}^{K} \sum_{q=0}^{N-1} \gamma_{qj}}, \tag{3.27}$$

and after multiplying both the nominator and the denominator by $\frac{1}{N}$ (step a), using the equivalence (3.22) (step b), and normalizing for sample weights (step c) we finally arrive at Eq. (3.10):

$$\pi_j = \frac{\frac{(u_\gamma)_i^j}{\overline{\nu}_i} + \frac{\delta_j - 1}{n}}{1 + \frac{\sum_j^K (\delta_j - 1)}{n}}.$$

## 3.B Spacing of Cached Distributions

In this appendix, we present details on our distribution caching, described in Sec. 3.4.3. Specifically, we detail the computation of the validity radius that determines spacing of cached directional distributions. The *validity radius* $r$, assigned to each distribution, is a scalar that gives the maximum spatial distance from the distribution where it can be reused. We compute the validity radius as a weighted harmonic mean of validity radii $r_j$ of the individual mixture lobes (i.e. individual GMM components):

$$r = \frac{1}{\sum_j^K \frac{\pi_j}{r_j}}. \tag{3.28}$$

Here $\pi_j$ are the mixing coefficients that sum to one over all $K$ components.

When computing the validity radius, we assume that each single lobe of a distribution corresponds to a single highlight and that the lobes are isotropic. These assumptions are made only when computing the validity radius while the training and sampling from the distributions still uses the full anisotropic model.

### 3.B.1 Estimating and Limiting Distribution Change

To determine the validity radius $r_j$ of a single lobe $l_j$, we first predict how the lobe $l_j$ of a distribution at the position $\mathbf{y}$ would change if we observed the corresponding highlight from a slightly different position $\mathbf{y}'$ (see Fig. 3.15). Because of this change, using a distribution constructed at $\mathbf{y}$ at the different point $\mathbf{y}'$ decreases the importance sampling quality and thus increases variance of the result.

This effect will be small if we ensure that a significant mass of each pair of an original lobe $l_j$ and its predicted image $l'_j$ overlap. We measure this overlap by Kullback-Leiber (KL) divergence [Bishop, 2006], a tool for measuring difference between distributions. By imposing a limit on the KL divergence between $l_j$ and $l'_j$ we compute the maximum acceptable angle

$$\alpha_{\max} = \arccos(\omega_\mu \cdot \omega'_\mu) \tag{3.29}$$

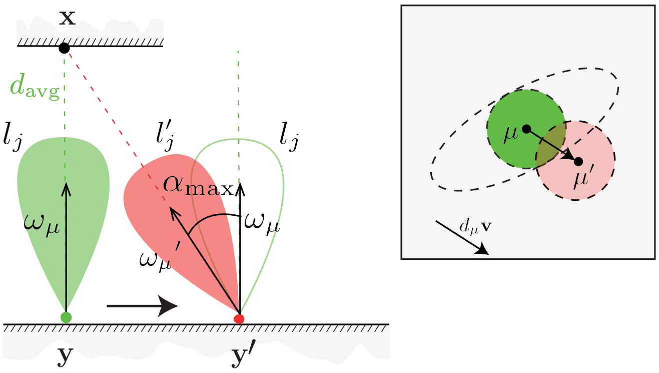


Figure 3.15: The assumed geometry used for the computation of the validity radius associated with the lobe of a distribution placed at the position $\mathbf{y}$. On the left, the lobe $l_j$ with its mean direction $\omega_\mu$ is shifted to a position $\mathbf{y}'$ and is compared to its predicted image $l'_j$ with its mean direction $\omega'_\mu$. On the right, the same situation is depicted in the unit square domain. The normal anisotropic distribution with the mean $\mu$ can be shifted at most about the distance $d_\mu$ in any direction $\mathbf{v}$. The ellipse depicts the real distribution shape while the circle centered in $\mu$ suggests its conservative isotropic approximation. The distance $d_\mu$ is determined by the imposed KL divergence limit between the two normal distributions.

between the lobe mean directions $\omega_\mu$ and $\omega'_\mu$. Details about the computation of $\omega'_\mu$ from KL divergence are given below. Using trigonometry between the original distribution position $\mathbf{y}$, the new position $\mathbf{y}'$ and the alleged highlight position $\mathbf{x}$ (see Fig. 3.15), the validity radius $r_j$ is then computed as

$$r_j = d_{\mathrm{avg}} \tan(\alpha_{\mathrm{max}}), \tag{3.30}$$

where $d_{\mathrm{avg}}$ is the distance between $\mathbf{y}$ and the position $\mathbf{x}$ of the alleged highlight. We estimate $d_{\mathrm{avg}}$ from all particles that were used to train all the lobes in the distribution. Specifically, we take the average of the distance that the particles traveled from their last bounce. We have decided to use the single common estimate $d_{\mathrm{avg}}$ for all lobes in a given mixture because it is, according to our experience, a more robust solution than having independent estimates for every lobe.

## 3.B.2 KL Divergence Limit

As we said, we determine $\omega'_\mu$ by imposing a limit on the KL divergence between the lobes $l_j$ and $l'_j$. This is computed in the unit square domain (see Fig. 3.15 right) where all the directions on the hemisphere $\mathcal{H}^+$ are projected through the area preserving mapping $\mathcal{S}$ of Shirley and Chiu [1997]. To keep the notation uncluttered, we omit writing the component index $j$ if there is no danger of confusion. In the unit square domain, a lobe $l$ corresponds to a Gaussian $\mathcal{N}(\mathbf{s}|\mu, \mathbf{\Sigma})$ and its shifted image $l'$ to a Gaussian $\mathcal{N}(\mathbf{s}|\mu', \mathbf{\Sigma})$, where $\mu = \mathcal{S}(\omega_\mu)$ and $\mu' = \mathcal{S}(\omega'_\mu)$. Note that these two normal distributions differ only in their means.

The KL divergence formula $\mathrm{KL}(\mu', \mathbf{\Sigma}' \,||\, \mu, \mathbf{\Sigma})$ for two bivariate normal distributions [Duchi, 2014] reads

$$\frac{1}{2}\left(\mathrm{tr}(\mathbf{\Sigma}^{-1}\mathbf{\Sigma}') + (\mu - \mu')^T \mathbf{\Sigma}^{-1}(\mu - \mu') - 2 - \ln \frac{|\mathbf{\Sigma}'|}{|\mathbf{\Sigma}^{-1}|}\right),$$

where $|\cdot|$ is the determinant and $\mathrm{tr}(\cdot)$ the trace of a matrix. Since in our case $\mathbf{\Sigma}$ and $\mathbf{\Sigma}'$ are identical, this reduces to one half of a square of Mahalanobis distance $\Delta$ between the two means:

$$\mathrm{KL}(\mu', \mathbf{\Sigma}' \,||\, \mu, \mathbf{\Sigma}) = \frac{1}{2}\Delta^2 = \frac{1}{2}(\mu - \mu')^T \mathbf{\Sigma}^{-1}(\mu - \mu').$$

Recall that we assume both normal distributions to be isotropic. This allows us to replace the covariance matrix inverse $\mathbf{\Sigma}^{-1}$ by its eigenvalue $\lambda$ and thus to write $\Delta^2 = \lambda d_\mu^2$, where $d_\mu$ is the distance between the two vectors $\mu$ and $\mu'$. It follows that

$$d_\mu = \sqrt{\frac{\Delta^2}{\lambda}}. \tag{3.31}$$

We impose a maximum threshold $\Delta^2_{\mathrm{thr}} = 5$, which then yields a maximum allowed value of $d_\mu$, denoted $d_{\mu,\mathrm{thr}}$, for a given lobe.

Because our normal distributions are actually anisotropic, we set $\lambda$ to be the higher from the two eigenvalues of $\mathbf{\Sigma}^{-1}$. This choice is conservative because it causes the resulting lobe validity radius $r_j$ to be smaller than if we chose the other eigenvalue.

Finally, the direction $\omega'_\mu$, that appears in Equation (3.29), is computed as

$$\omega'_\mu = \mathcal{S}^{-1}(\mu'), \tag{3.32}$$

where $\mu' = d_\mu \mathbf{v}$ and $\mathcal{S}^{-1}$ is the inverse mapping of Shirley and Chiu. The direction $\mathbf{v}$ in the unit square domain can be chosen arbitrarily because we assume isotropic normal distributions (see Fig. 3.15). So the computed validity radius does not depend on the selected direction $\mathbf{v}$.

### 3.B.3 Summary

To summarize, the steps in calculating the validity radius $r_j$ for a single lobe $l_j$ are:

a) Compute the maximum distance $d_{\mu,\mathrm{thr}}$, in which the lobe $l_j$ can be shifted without exceeding the user specified threshold $\Delta^2_{\mathrm{thr}}$:

$$d_{\mu,\mathrm{thr}} = \sqrt{\frac{\Delta^2_{\mathrm{thr}}}{\lambda}}. \tag{3.33}$$

The parameter $\lambda$ is the larger of the two eigenvalues of $\mathbf{\Sigma}_j^{-1}$, that is the inverse of the lobe covariance matrix $\mathbf{\Sigma}_j$.

b) Select an arbitrary direction $\mathbf{v}$ in the unit square domain and compute the 3D direction $\omega'_\mu$ using the inverse mapping of Shirley and Chiu:

$$\omega'_\mu = \mathcal{S}^{-1}(d_{\mu,\mathrm{thr}}\mathbf{v}). \tag{3.34}$$

c) Calculate the validity radius $r_j$ using Equations (3.29) and (3.30):

$$r_j = d_{\mathrm{avg}} \tan\left(\arccos\left(\omega_\mu \cdot \omega'_\mu\right)\right). \tag{3.35}$$

## 3.C Maximum Particle Weight Increment

We claim, in Sec. 3.4.5, that when we sample a scattering direction by importance sampling the BSDF $f_\mathrm{s}$ and our directional distribution $p(\omega_\mathrm{o})$, particle weight cannot increase more than twice per scattering event. This is true under the following assumptions:

1. we combine both strategies using the balance heuristic [Veach, 1997],
2. we take equal number of samples from both strategies, and
3. we can achieve perfect importance sampling of the BSDF.

By perfect importance sampling of BSDF we mean sampling a direction $\omega_\mathrm{o}$ from a distribution that is *exactly* proportional to the projected BSDF $f_\mathrm{s}^{\perp}(\omega_\mathrm{o}) = f_\mathrm{s}(\mathbf{y}, \omega_\mathrm{i} \to \omega_\mathrm{o})|\cos\theta_\mathrm{o}|$. To keep our notation uncluttered, we omit incident direction $\omega_\mathrm{i}$ and the scattering event location $\mathbf{y}$ in the rest of this section. Formally, under Veach's *one-sample model* [Veach, 1997], our effective sampling PDF

$$\hat{p}(\omega_\mathrm{o}) = \frac{1}{2} \frac{f_\mathrm{s}^{\perp}(\omega_\mathrm{o})}{\int_\Omega f_\mathrm{s}^{\perp}(\omega) \mathrm{d}\omega} + \frac{1}{2} p(\omega_\mathrm{o}) \tag{3.36}$$

is a convex mixture of two distributions with equal weights (Assumptions 1 and 2).

According to Eq. (2.7), the current path weight $\hat{\nu}$ after RR (and possible splitting) was applied, is multiplied by BSDF and divided by our direction sampling probability factor

$$\nu_\mathrm{o}(\omega_\mathrm{o}) = \hat{\nu} \frac{f_\mathrm{s}^{\perp}(\omega_\mathrm{o})}{\hat{p}(\omega_\mathrm{o})}. \tag{3.37}$$

Here, $\nu_\mathrm{o}$ is the outgoing particle weight resulting from the current scattering event. If our directional distribution for the direction $\omega_\mathrm{o}$ is equal to 0, i.e. $p(\omega_\mathrm{o}) = 0$, by plugging Eq. (3.36) into Eq. (3.37), we arrive at

$$\nu_\mathrm{o}(\omega_\mathrm{o}) = 2\hat{\nu}\kappa, \tag{3.38}$$

where $\kappa = \int_\Omega f_\mathrm{s}^{\perp}(\omega) \mathrm{d}\omega$ is the material reflectance which can take values $\kappa \in [0, 1]$ [Dutré et al., 2006].

With increasing values of $p(\omega_\mathrm{o})$ the value of our sampling distribution $\hat{p}(\omega_\mathrm{o})$ for the direction $\omega_\mathrm{o}$ also increases and thus value of $\nu_\mathrm{o}(\omega_\mathrm{o})$ decreases (see Eq. (3.37)). This means that Eq. (3.38) states that the path weight $\nu_\mathrm{o}(\omega_\mathrm{o})$ can increase at maximum by a factor of two.

# Chapter 4

# Adjoint-Driven Russian Roulette and Splitting

## 4.1 Introduction

In this chapter, we revisit two classic techniques to improve efficiency, *Russian roulette* (RR) and *splitting* (see Sec. 2.6) and, following the idea introduced in the previous chapter, we augment them with a global knowledge of adjoint solution approximation (see Sec. 2.2). This, in turn, results in much more efficient transport simulation as opposed to traditional local approaches.

In general, RR aims to save computation time by terminating transport paths with small contribution, while splitting (a.k.a. distributed ray tracing [Cook et al., 1984]) branches paths into several independent trajectories. As we have already noted in Sec. 2.6, RR decisions in light transport have usually been based on local surface reflectivity [Jensen, 2001, Dutré et al., 2006, Jakob, 2010] or on the accumulated path weight (a.k.a. throughput) [Arvo and Kirk, 1990, Jensen, 1996, Veach, 1997]. Splitting often relies on heuristics based on local BSDF roughness [Szirmay-Kalos and Antal, 2005]. While simple to implement, these *local* approaches do not work well in scenes with non-uniform light distribution, as illustrated in Fig. 4.1.

The sub-optimal performance of traditional RR and splitting is due to the respective decisions being oblivious to the actual distribution of light (when tracing paths from the camera) or visual importance (when tracing paths from the light sources). For example, when the surface reflectivity is used as the termination probability in RR, effort is often spent on sampling long paths on bright surfaces, while dark surfaces suffer from high variance due to early path termination. The latter is particularly acute in scenes with difficult visibility or dense participating media, where most contributions are due to long paths that may absorb lots of energy but originate at bright sources. Unfortunately, the problem cannot be solved simply by making RR less aggressive across the entire scene, because doing so would waste resources on sampling longer paths even where not necessary.

We present *adjoint-driven Russian roulette and splitting* (ADRRS), a new approach for RR and splitting decisions. To address the shortcomings of the current approaches, we terminate or split paths according to an estimate of their *total expected contribution* to the image, relative to a *reference solution.* Paths with an expected contribution much higher than the reference are split, while

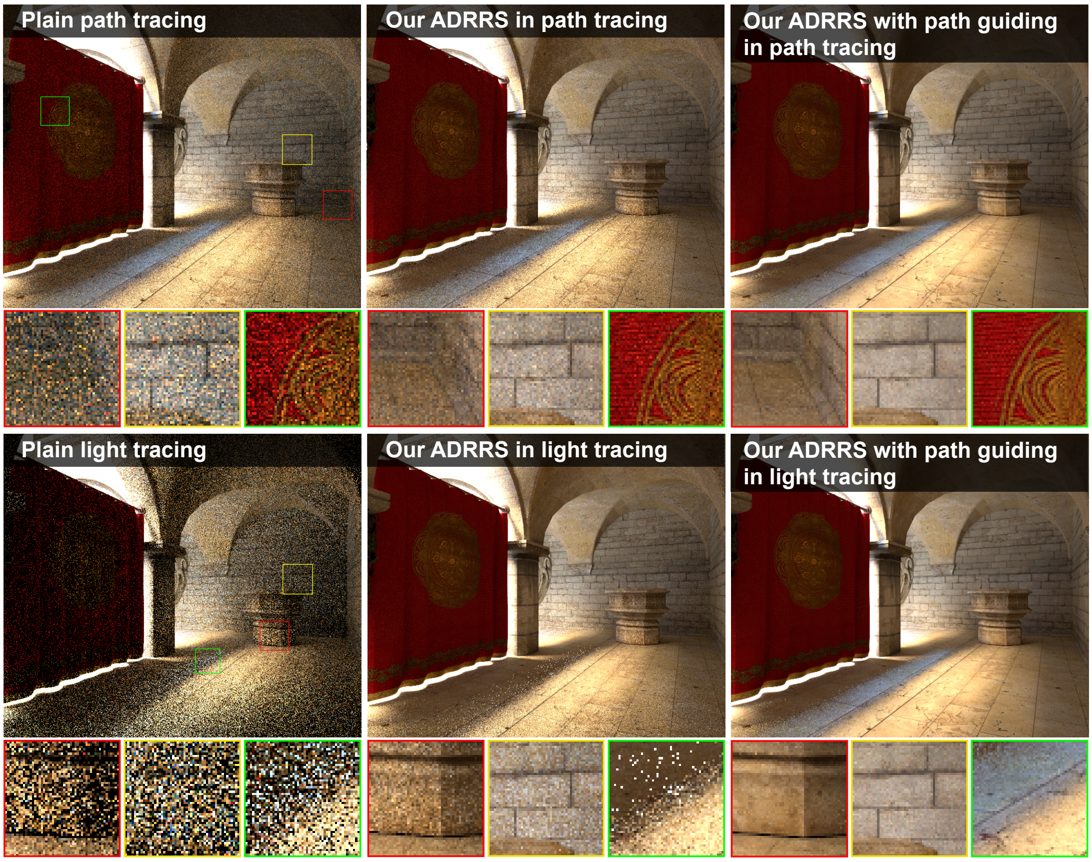

Plain path tracing
Our ADRRS in path tracing
Our ADRRS with path guiding in path tracing
Plain light tracing
Our ADRRS in light tracing
Our ADRRS with path guiding in light tracing

Figure 4.1: Russian roulette based on material reflectance, as traditionally applied in computer graphics, leads to suboptimal results in scenes with non-uniform light distribution or complex visibility (left). Our adjoint-driven Russian roulette and splitting (ADRRS) significantly increases the efficiency in such cases (middle). ADRRS complements the advantages of path guiding (PG) methods (Chapter 3) and in conjunction they provide superior results than either method alone (right). All images have been rendered in 1 hour by path tracing (top row) and light tracing (bottom row). Note on the middle-bottom image that while our ADRRS cannot improve direct-light sampling (the blue tint on the floor) in light tracing as splitting occurs only from the first bounce on, it greatly reduces variance of multi-bounce indirect illumination.

paths with a low expected contribution have higher chance of being terminated. The expected contribution is calculated as a product of the current path weight and an estimate of the adjoint transport solution (i.e. equilibrium radiance for camera sub-paths or equilibrium visual importance for light sub-paths), which we pre-compute and cache in the scene. We show that this approach leads to good importance sampling of the path space, and, in turn, it can significantly increase the overall efficiency of the simulation.

Our method works in synergy with path guiding (see Chapter 3) and similar methods [Bashford-Rogers et al., 2012, Hey and Purgathofer, 2002, Jensen, 1995], and in conjunction they provide superior results. This is an important advantage over the traditional local RR decisions, which counteract the guiding methods by terminating potentially important paths before they can even make a contribution. We provide a theoretical justification of this beneficial behavior based on the theory of zero-variance random walks [Křivánek and d'Eon, 2014, Xu et al., 2001, Kalos, 1963], which, as we show, is common basis of both the proposed method as well as the guiding schemes. We also demonstrate the advantages on implementation of ADRRS with our guiding method [Vorba et al., 2014] that we introduced in Chapter 3.

In summary, our contributions in this chapter, which is based on our previously published work [Vorba and Křivánek, 2016], are:

- We propose *adjoint-driven Russian roulette and splitting* (ADRRS) where paths are terminated or split according to their expected contribution to the image (Sec. 4.3).

- We provide a theoretical analysis of the close relation of ADRRS to the zero-variance random walk schemes, which explains its variance reduction properties (Sec. 4.6).

- We develop a solution for obtaining the paths' expected contribution necessary to use ADRRS in practice (Sec. 4.4).

- We show that ADRRS can improve the efficiency of path guiding methods (Sec. 4.5).

## 4.2 Related Work

**Russian roulette and splitting.** Some works in computer graphics derive the termination and splitting rates by direct optimization of *efficiency* (i.e. reciprocal of the product of variance and computation time). For example, Szirmay-Kalos and Antal [2005], using a series of simplifying assumptions, arrive at a RR/splitting heuristic based on local BSDF reflectivity and roughness, and user-specified global constants. Bolin and Meyer [1997] derive optimal termination and splitting rates through a variance analysis of nested estimators, but they do not describe a working method based on these results. In contrast, we provide a practical algorithm that relies on a well-founded theory, albeit not on the optimization of efficiency. While Veach [1997] proposes efficiency-optimized RR for light path connections in bidirectional path tracing, our work describes RR in a more general context.

**Importance sampling.** Unlike in graphics, in the neutron transport literature, RR and splitting are understood as *importance sampling* techniques [Veach, 1997, Hammersley and Handscomb, 1964, Spanier and Gelbard, 1969]. For example, to reliably estimate the radiation escaping through a nuclear reactor shield, it is impractical to use an analog (see Sec. 2.1) simulation since the probability of penetrating the thick shield by a particle is extremely low ($\leq 10^{-9}$). To solve similar problems in the nuclear engineering practice, users of the MC simulators define, usually semi-automatically, an *importance function* over the domain of interest [Wagner and Haghighat, 1998]. The simulation then terminates particles in the parts of the domain designated as unimportant, while splitting them in high-importance regions. This strategy effectively adapts the number of surviving particles to the user-specified importance.

As mentioned above, a common practice in computer graphics is to drive RR decisions by the particle weight [Veach, 1997, Jensen, 1996, Arvo and Kirk, 1990]. However, doing so results in poor importance sampling, because no information on the expected future behavior of the particle is taken into account. In our work, we show that – rather than relying solely on the particle weight – it is beneficial to drive RR and splitting also by the *adjoint* quantity (*radiance* when tracing a particle from the camera, and *visual importance* when starting from the light sources). This adjoint quantity value gives us an estimate of the path's expected future behavior, which – when multiplied by the path weight – provides the expected total contribution of the path to the solution.

Two closely related works, that also use adjoints, are that of Keller and Wald [2000] and Georgiev and Slusallek [2010]. They both use importance-driven RR to randomly decide about depositing a photon and a virtual point light, respectively, while they use classical RR based on local reflectance properties during path construction. Consequently, they may need to sample a tremendous number of paths to achieve low variance in visually important regions with low illumination. Our work, in contrast, address this issue by employing adjoint-driven RR during path construction itself to directly influence length of sampled paths. The works of Szécsi et al. [2003] and Szirmay-Kalos and Antal [2005] reduce variance due to RR by returning an irradiance estimate upon path termination (instead of the usual zero). In contrast, we exploit a similar estimate to compute optimal termination and splitting rate itself.

**Path guiding methods.** As described in Chapter 3, a direct approach to distributing paths according to visual importance is importance sampling of emission from light sources [Vorba et al., 2014, Bashford-Rogers et al., 2013, Dutré and Willems, 1994], and of scattering directions during incremental path construction [Vorba et al., 2014, Bashford-Rogers et al., 2012, Hey and Purgathofer, 2002, Jensen, 1995]. In this way, paths are directly guided towards regions with high contribution to the computed image. Such guided path sampling is a non-analog (see Sec. 2.1) simulation that typically leads to high local variation of particle weights [Vorba et al., 2014, Keller and Wald, 2000, Suykens and Willems, 2000]. In contrast, our adaptive RR and splitting achieves more balanced weights locally than the guiding methods.

**Weight window.** In neutron transport simulations, RR and splitting are combined in one variance reduction tool called the *weight window* [Hoogenboom and Légrády, 2005, X-5 Monte Carlo team, 2003, Booth and Hendricks, 1984].

This technique is designed to keep the particle weight within a certain interval that may vary over the simulation domain. This interval can be user-specified or based on an automatic importance computation [Wagner and Haghighat, 1998, Wagner, 1997, Booth and Hendricks, 1984]. If a particle weight is below or above the interval bound, RR or splitting is applied, respectively, so that the particle weight stays within the interval. Our path guiding described in the previous Chapter [Vorba et al., 2014] actually employs the weight window facility to avoid contradicting early path termination and thus, in turn, to demonstrate advantages of path guiding. However, the bounds are set manually, do not adapt to a given scene and are constant everywhere in the scene. We achieve a significant performance gain by setting the weight window bounds according to a spatially and directionally-varying radiance or visual importance solution in any given scene.

Booth and Hendricks [1984] set the interval bounds so that the weight window center is inversely proportional to an importance function. This keeps more particles in important regions and less in unimportant ones. The normalization constant for computing the weight window bounds from the importance is set heuristically so that particles are within the weight window immediately upon their emission. Based on an analysis of zero-variance random walk schemes, Wagner and Haghighat [1998, 1997] suggest a theoretically founded approach where an estimate of the final solution serves as the said normalization constant. Adopting this approach allows us to keep particle/path weights around the optimal levels.

While Booth and Hendricks [1984] compute the importance function solely by *forward* particle tracing, we follow the same idea introduced in Chapter 3 and use *interleaved* particle tracing from both the camera and light sources. This yields more reliable radiance or visual importance estimates even in scenes with difficult visibility.

**Go-with-the-winners.** Szirmay-Kalos and Antal [2005] base RR and splitting rate on efficiency analysis and use a heuristic variance approximation based on local material properties and scene-dependent parameters. They introduced the term "go-with-the-winners" to computer graphics that is often used in a general MC context to refer to RR and splitting. Note that Grassberger [2002] points out there is no fundamental difference between RR and splitting as described by Kahn [1951, 1956] and the "go-with-the-winners" strategy. The term was coined by Aldous and Vazirani [1994] as a mean for population control in randomized optimization algorithms.

## 4.3 Adjoint-Driven RR and Splitting

In this section, we describe our *adjoint-driven Russian roulette and splitting* (ADRRS) approach to termination and splitting along incrementally sampled paths. Practical rendering algorithms based on this approach are described in Sec. 4.4 and 4.5. While in Sec. 4.4, we describe utilization of ADRRS, in Sec. 4.5, we describe its combination with path guiding. In Sec. 4.6, we show variance reduction properties of our approach and also that it is motivated by zero-variance sampling schemes (see Sec. 2.8).

### 4.3.1 Unified Russian Roulette and Splitting

In our approach, we follow previous work [Szirmay-Kalos and Antal, 2005, Wagner and Haghighat, 1998, Booth and Hendricks, 1984] and we base the respective RR and splitting decisions on a single real value $q > 0$. We consider a particle that has just left a collision (or emitting) event at $\mathbf{x}$, as shown in Fig. 4.2. After sampling its outgoing direction $\omega_\mathrm{o}^\mathrm{x}$ and determining the position $\mathbf{y}$ of the next collision, we contribute the source radiance/importance from $\mathbf{y}$ to the solution (Eq. (2.4) and Eq. (2.6) for path tracing and light tracing, respectively). Then, we determine $q(\mathbf{y}, \omega_\mathrm{i})$, as described in the next section, and if $q < 1$ we play RR to randomly terminate the path with probability $1 - q$. Conversely, if $q > 1$, we split the path into $q$ new paths. (Details on dealing with non-integer $q$ are given in Sec. 4.4.1.)

To compensate for termination or splitting at $\mathbf{y}$, the incoming particle weight $\nu_\mathrm{i}$ (see Sec. 2.3) is divided by $q$ to obtain the weight $\hat{\nu}$ of each survived or split particle:

$$\hat{\nu}(\mathbf{y}, \omega_\mathrm{i}) = \frac{\nu_\mathrm{i}(\mathbf{y}, \omega_\mathrm{i})}{q(\mathbf{y}, \omega_\mathrm{i})}. \tag{4.1}$$

Each particle resulting from the collision at $\mathbf{y}$ is then traced using the same procedure independently.

Let us remind the convention illustrated in Fig. 2.1, that the direction $\omega_\mathrm{o}$ always points in the direction of the transported quantity.

### 4.3.2 Determining the RR/Splitting Factor $q$

In our ADRRS, the RR/splitting factor $q$ at $\mathbf{y}$ is directly proportional to the *total expected contribution* $E[c(\mathbf{y}, \omega_\mathrm{i})]$ of the particle that collided at $\mathbf{y}$ to the computed measurement $I$ (e.g. a pixel value) [Wagner and Haghighat, 1998, Booth and Hendricks, 1984]:

$$\boxed{q(\mathbf{y}, \omega_\mathrm{i}) = \frac{E[c(\mathbf{y}, \omega_\mathrm{i})]}{I} = \frac{\nu_\mathrm{i}(\mathbf{y}, \omega_\mathrm{i}) \Psi_\mathrm{o}^\mathrm{r}(\mathbf{y}, \omega_\mathrm{i})}{I}.} \tag{4.2}$$

For a path traced from a light source, the adjoint $\Psi$ stands for the visual importance $W$, while for a path traced from the camera, it stands for radiance $L$. Here, we use only its reflected part $\Psi^\mathrm{r} = \Psi - \Psi^\mathrm{e}$ without the source term $\Psi^\mathrm{e}$, because we are interested in the expected contribution of the particle scattered at $\mathbf{y}$.

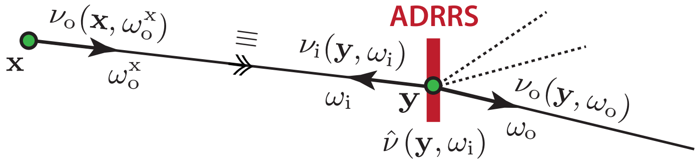

$\nu_\mathrm{o}(\mathbf{x}, \omega_\mathrm{o}^\mathrm{x})$
$\mathbf{x}$
$\omega_\mathrm{o}^\mathrm{x}$
$\equiv$
$\nu_\mathrm{i}(\mathbf{y}, \omega_\mathrm{i})$
ADRRS
$\omega_\mathrm{i}$
$\mathbf{y}$
$\nu_\mathrm{o}(\mathbf{y}, \omega_\mathrm{o})$
$\omega_\mathrm{o}$
$\hat{\nu}(\mathbf{y}, \omega_\mathrm{i})$

Figure 4.2: After we have accounted for a particle's contribution from a collision at $\mathbf{y}$ (see Eq. (2.4) and Eq. (2.6)), we apply our ADRRS to decide about the particle's termination/splitting. All potentially spawned particles at $\mathbf{y}$ have weight $\hat{\nu}(\mathbf{y}, \omega_\mathrm{i})$ and are scattered and traced independently. The relation between weight $\hat{\nu}$ of survived particles and outgoing weight $\nu_\mathrm{o}$ of each scattered particle is given by Eq. (2.7).

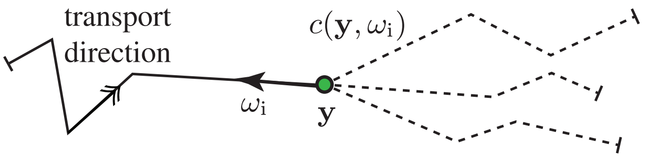


Figure 4.3: Realizations of the path contribution variable $c(\mathbf{y}, \omega_\mathrm{i})$ correspond to the different possible particle paths beyond $\mathbf{y}$.

The idea behind Eq. (4.2) is to compare $E[c]$ to the true value of the measurement $I$. We are likely to terminate particles that are not expected to make a contribution larger than $I$. This, in turn, saves resources for sampling particles with a more significant expected contribution. In contrast, when $E[c]$ exceeds the true measurement value $I$, we split the particle path. This results in a better exploration of the relevant regions of the path space, albeit at the expense of some additional computational resources.

**Particle contribution and its expected value.** The particle contribution $c(\mathbf{y}, \omega_\mathrm{i})$ is a random variable associated with a particle that has reached the point $\mathbf{y}$ from the direction $\omega_\mathrm{i}$ and has the weight $\nu_\mathrm{i}(\mathbf{y}, \omega_\mathrm{i})$. The variable is distributed over all possible realizations of the particle path beyond $\mathbf{y}$, as shown in Fig. 4.3. For example in path tracing, the outcome of $c$ for one such specific realization is given by the particle's contribution to the sum in the measurement estimator in Eq. (2.4). Note that each particle sampled beyond $\mathbf{y}$ is an unbiased estimate of $\Psi_\mathrm{o}^\mathrm{r}(\mathbf{y}, \omega_\mathrm{i})$. Thus the expected contribution $E[c(\mathbf{y}, \omega_\mathrm{i})]$ is given by the product of the path weight $\nu_\mathrm{i}$ and the outgoing reflected adjoint $\Psi_\mathrm{o}^\mathrm{r}$.

### 4.3.3 Weight Invariant in ADRRS

We design our ADRRS so that it maintains the following invariant:

$$\hat{\nu}(\mathbf{y}, \omega_\mathrm{i}) = \frac{I}{\Psi_\mathrm{o}^\mathrm{r}(\mathbf{y}, \omega_\mathrm{i})}, \tag{4.3}$$

which holds for particle weight in a zero-variance (ZV) scheme [Wagner and Haghighat, 1998] (see Sec. 2.8). While this invariant arises naturally under the ZV scheme (see Eq. (2.17)), our ADRRS keeps it through termination and splitting. Our termination/splitting rate $q$ (Eq. (4.2)) follows directly from Eq. (4.3) and the weight update formula after termination/splitting (Eq. (4.1)). Note that ADRRS keeps this invariant with arbitrary emission and scattering probabilities. This principle allows us to justify the importance sampling properties of ADRRS in Sec. 4.6 through inspection of $q$.

## 4.4 Algorithm

In this section we develop a practical solution for incremental path sampling (either from the camera or from the light sources) based on our *adjoint-driven Russian roulette and splitting* (ADRRS), described earlier. Evaluating the RR/splitting factor $q$ according to Eq. (4.2) requires knowing the final measurement $I$ as well as the value of the adjoint transport quantity $\Psi$ everywhere in the scene, none of which are readily available up front. Our solution builds on an approximate estimate of those quantities obtained in a preprocessing step, as described in Sec. 4.4.2 and 4.4.3. To make the resulting algorithm robust to the inaccuracies of these estimates, we apply the *weight window* facility as described next.

We conduct the following exposition in a general tenor that applies to tracing paths in either direction. We recall that the adjoint $\Psi$ stands for the radiance $L$ for a camera path (as in path tracing), and for the visual importance $W$ in the case of a path traced from the light sources (as in light or photon tracing). Differences between the two cases are pointed out when necessary.

### 4.4.1 Weight Window

The weight window is a classic technique from neutron transport used to control particle weights through RR and splitting [Hoogenboom and Légrády, 2005, Booth and Hendricks, 1984]. In our algorithm, it amends the ADRRS step shown in Fig. 4.2.

The weight window defines an interval of acceptable particle weights $\langle \delta^-, \delta^+ \rangle$ (see Fig. 4.4). A particle with a weight $\nu_\mathrm{i}$ that enters a weight window may be terminated, pass unchanged, or be split. In any case, the weight window ensures that the weight $\hat{\nu}$ of each leaving particle stays within the window bounds, i.e. $\hat{\nu} \in \langle \delta^-, \delta^+ \rangle$. If the weight of a particle entering the window is below the lower bound, $\nu_\mathrm{i} < \delta^-$, we play Russian roulette with the survival probability $q = \nu_\mathrm{i}/\delta^-$. If the particle survives, its new weight $\hat{\nu}$ is set to $\nu_\mathrm{i}/q = \delta^-$. Particles entering the window with a weight inside the bounds, $\nu_\mathrm{i} \in \langle \delta^-, \delta^+ \rangle$, pass the window intact (i.e. $\hat{\nu} = \nu_\mathrm{i}$). Finally, if $\nu_\mathrm{i} > \delta^+$, the particle is split into $q = \nu_\mathrm{i}/\delta^+$ new particles, each with a weight of $\hat{\nu} = \nu_\mathrm{i}/q = \delta^+$.

**Non-integer splitting.** If the particle is to be split into a non-integer number of new particles, we use an *expected-value split* approach [Booth, 1985b]. We split the particle into $n = \lfloor q \rfloor$ new particles with the probability $n + 1 - q$, and into $n + 1$ particles otherwise. Irrespective of that decision, each new particle is assigned a weight of $\hat{\nu} = \nu_\mathrm{i}/q$. Although this splitting strategy does not preserve the original weight exactly, the total weight is still preserved in an expected-value sense and thus the estimator stays unbiased.

For completeness, we state another two options. We could apply so called *sampled splitting* [Booth, 1985b] which uses the same probabilities as an expected-value splitting but sets the particle weight to $\hat{\nu} = \nu_\mathrm{i}/n$ in case of splitting into $n$ particles or $\hat{\nu} = \nu_\mathrm{i}/(n+1)$ otherwise. The disadvantage of this approach is that, in an unfortunate chain of decisions to split to $n$ particles, hight particle

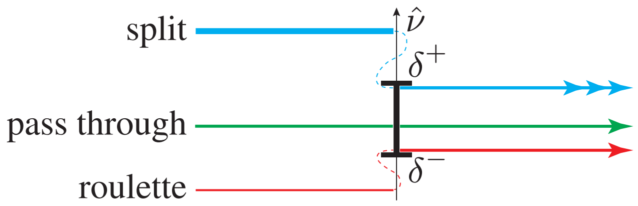


Figure 4.4: The weight window ensures – through selective RR or splitting – that the weight of all particles that pass the window is within the interval of acceptable weights $\langle \delta^-, \delta^+ \rangle$.

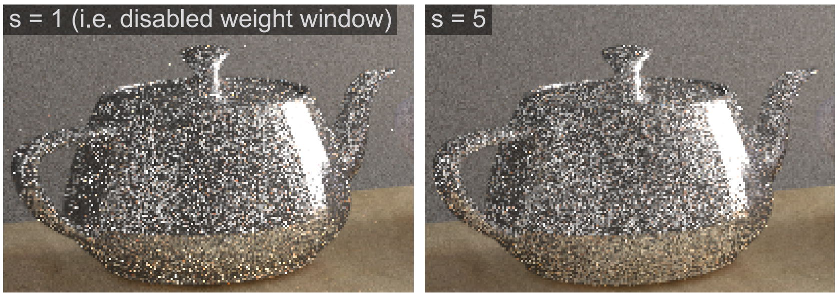


Figure 4.5: Weight window increases robustness of our ADRRS. It relaxes termination and splitting when the path weight is close to the weight window center not to introduce additional noise. It also compensates for imprecisions in the adjoint estimate (glossy teapot). Left: Without weight window. Right: Our weight window size. Images are rendered by light tracing in 30 minutes.

weights can be accumulated [Booth, 1985b]. Both expected-value split and sampled splitting introduce an extra variance. However, this seems to be marginal in comparison to variance present in for example plain path tracing with standard BSDF importance sampling and RR based on the local properties.

Finally, without introducing any variance, we could use an *integer splitting* where particles are always split into either $n$ particles or always to $n+1$ particles. The variant choosing the smaller integer, in fact, has been used in the implementation of MCNP [X-5 Monte Carlo team, 2003] which is the MC particle transport simulator from Los Alamos National Laboratory[1]. We have favoured the expected-value splitting because it allows the particle weights to obey the weight window bounds precisely.

**Weight window bounds.** To calculate the weight window bounds for a particle incident from the direction $\omega_\mathrm{i}$ at $\mathbf{y}$, we start by setting the window center $C_\mathrm{ww} = (\delta^- + \delta^+)/2$ to the desired particle weight given by Eq. (4.3),

$$C_\mathrm{ww} = \frac{\tilde{I}}{\tilde{\Psi}^\mathrm{r}_\mathrm{o}(\mathbf{y}, \omega_\mathrm{i})}. \tag{4.4}$$

The measurement estimate $\tilde{I}$ is invariant along the entire particle path while the adjoint quantity estimate $\tilde{\Psi}^\mathrm{r}_\mathrm{o}(\mathbf{y}, \omega_\mathrm{i})$ depends on the scattering location $\mathbf{y}$ and the incoming direction $\omega_\mathrm{i}$. Computation of these two estimates is explained in Sec. 4.4.2 and 4.4.3.

Booth and Hendricks [1984] as well as Wagner and Haghighat [1997] all suggest to set the window width as $\delta^+ = s\delta^-$, with the ratio parameter $s = 5$. The formula for computing the weight window lower bound then reads

$$\delta^- = \frac{2C_\mathrm{ww}}{1+s}. \tag{4.5}$$

We have experimented with the parameter $s$ and verified that performance is not particularly sensitive to its value [Booth, 2006].

A practical consequence of using weight window are more relaxed RR and splitting decisions than those given by Eq. (4.2). As a result, the algorithm is

[1] From e-mail communication with Thomas Booth, the author of the weight window technique.

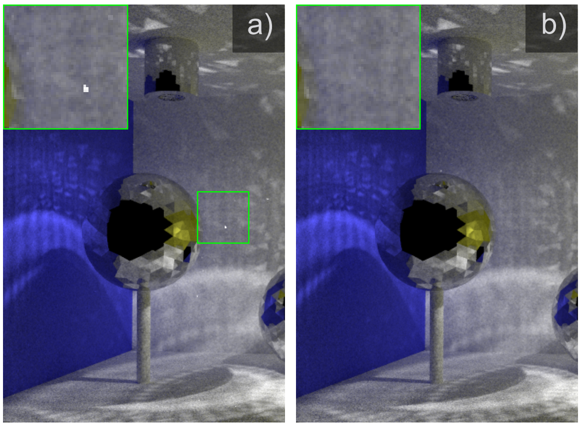


Figure 4.6: Extremely small survival probabilities could result in high variance (a). We force the survival probability to be above 0.1 (b).

more robust to the inaccuracies of our measurement and adjoint estimates. The weight window achieves this by allowing some leeway on the particle weight before any RR/splitting action is taken (Fig. 4.5).

Let us emphasize that weight window is different from clamping of the termination and splitting factor $q$ to a finite interval, which we do apply on top of the weight window output. When RR is to be played, we additionally force the survival probability $q$ to be above 0.1 (Fig. 4.6b). Very small survival probabilities could otherwise result in high variance in some cases due to the inaccuracies in the measurement and adjoint estimates (Fig. 4.6a).

### 4.4.2 Adjoint Solution Estimate $\tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})$

To setup the weight window bounds at $\mathbf{y}$, we need an estimate of the outgoing reflected adjoint quantity $\tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})$. One could use a photon/importon density estimate but that would be neither accurate nor fast enough. Alternatively, a solution similar to radiance caching [Křivánek et al., 2005, Gassenbauer et al., 2009] could be used that stores the spatial-directional distribution of the adjoint. However, we have found that a simpler approach outlined below and illustrated in Fig. 4.7 (a) provides fairly robust estimates without having to store any directional information.

We obtain the outgoing adjoint at $\mathbf{y}$ from a *pre-computed cache*. Instead of the full spatial-directional distribution of $\Psi_{\mathrm{i}}(\mathbf{y}, \omega)$, we pre-compute and cache an estimate $\tilde{G}(\mathbf{y})$ of the quantity

$$G(\mathbf{y}) = \int \Psi_{\mathrm{i}}(\mathbf{y}, \omega) \cos\theta_{\mathbf{y}} \, \mathrm{d}\omega, \tag{4.6}$$

that corresponds to *irradiance* or *diffuse visual importance*. Here $\theta_{\mathbf{y}}$ is the angle between the direction $\omega$ and the surface normal at $\mathbf{y}$. An estimate of the reflected outgoing adjoint is then calculated as

$$\tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y}) = \frac{\kappa(\mathbf{y})}{\pi} \tilde{G}(\mathbf{y}), \tag{4.7}$$

where $\kappa(\mathbf{y})$ is the total material reflectivity at $\mathbf{y}$. We obtain $\tilde{G}(\mathbf{y})$ by querying a spatial cache at $\mathbf{y}$.

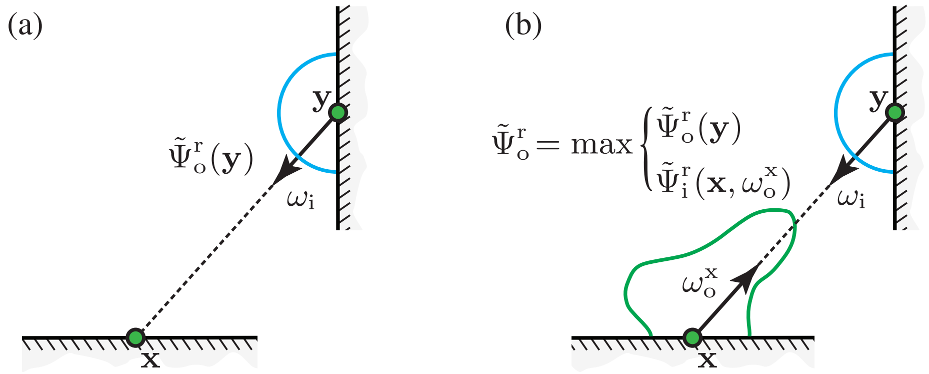

(a)
$\tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y})$
$\mathbf{y}$
$\omega_{\mathrm{i}}$
$\mathbf{x}$
(b)
$\tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}} = \max \begin{cases} \tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y}) \\ \tilde{\Psi}^{\mathrm{r}}_{\mathrm{i}}(\mathbf{x}, \omega^{\mathrm{x}}_{\mathrm{o}}) \end{cases}$
$\mathbf{y}$
$\omega_{\mathrm{i}}$
$\omega^{\mathrm{x}}_{\mathrm{o}}$
$\mathbf{x}$

Figure 4.7: Adjoint solution estimate in our basic ADRRS implementation (a) and when ADRRS is combined with path guiding (b).

**Pre-processing and caching.** We base our pre-processing stage on the iterative scheme proposed in Chapter 3, where we interleave shooting particles from the camera and from light sources for faster convergence. Our basic implementation differs from the algorithm in Sec. 3.4 in that the traced particles are used to compute an approximation to *irradiance* and *diffuse visual importance*, respectively (i.e. $\tilde{G}$), rather than for fitting directional distributions. (A more advanced implementation that uses directional distributions (i.e. "path guiding") will be described in Sec. 4.5.)

We also adopt the same caching scheme introduced in Sec. 3.4. When an estimate of $\tilde{G}$ is not available we use *kernel density estimation* to compute a new estimate from nearby particles (e.g. from photons when we currently trace from the camera). Each cached estimate has a validity radius where the record can be reused. The radius is never allowed to be larger than the furthest particle used for the estimation and the caching scheme makes the cache denser in places of strong light field changes.

We refine $\tilde{G}$ in every iteration using progressive kernel density estimation and at the same time, we estimate its relative error (i.e. $\mathrm{stddev}(\tilde{G})/\tilde{G}$). Our ADRRS is applied at a collision only if the relative error of the associated cache record is below a threshold value of 30%. Otherwise we use a large weight window with the globally-fixed size as described by Vorba et al. [2014] and in Sec. 4.5. Note that unlike bidirectional path tracing [Veach, 1997] or vertex connection and merging [Georgiev et al., 2012b, Hachisuka et al., 2012] we do not construct a combined estimator to compute $\tilde{G}$ and rather keep our implementation simple. To achieve smoother estimates of $\tilde{G}$ we average it over nearby cached estimates.

### 4.4.3 Measurement Estimate $\tilde{I}$

When calculating the weight window center using Eq. (4.4), Booth [1984] recommends, instead of $\tilde{I}$, using a normalization constant so that the particle weight is exactly in the weight window center after the first collision. We adopt a more principled approach due to Wagner and Haghighat [1997], where they use an approximation $\tilde{I}$ of the measurement $I$ that we eventually strive to calculate. In this way, the particle weights $\hat{\nu}(\mathbf{y}, \omega_{\mathrm{i}})$ oscillate around the ideal value $I/\Psi^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})$, which is motivated by the zero-variance theory, as discussed in Sec. 4.6.

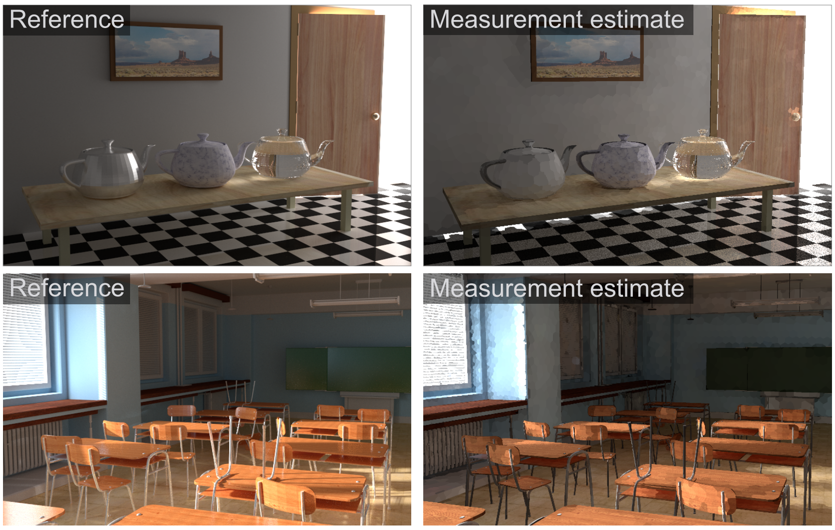


Figure 4.8: Measurement estimates $\tilde{I}$ for two example scenes.

In our implementation, the actual meaning of $\tilde{I}$ depends on the direction of path sampling. For paths from the camera (path tracing), we set $\tilde{I}$ to be an estimate of the pixel value that the respective path passes through. We compute the pixel value from the pre-computed irradiance cache (Sec. 4.4.2) using four jittered primary rays. We query the cache immediately on diffuse and glossy surfaces while continuing the path on purely specular surfaces. If no non-specular surface is hit within ten bounces, we treat the surface of the 10th bounce as diffuse and query the irradiance cache. Fig. 4.8 shows the resulting estimates for two example scenes.

For paths from light sources (light or photon tracing), we set $\tilde{I}$ to the average pixel value estimated as above, over the entire image. We use this approach because we do not know up front which pixel the path would contribute to. While this heuristic has worked well in our tests, a thorough analysis of ADRRS when calculating several measurements (i.e. pixel values) simultaneously would be an interesting avenue for future work.

### 4.4.4 Path Sampling Algorithm

Algorithm 3 shows simplified pseudocode for processing a collision in our path sampling algorithm. The procedure receives the previous and the current collision locations $\mathbf{x}$ and $\mathbf{y}$ respectively, the incident direction $\omega_{\mathrm{i}}$ at $\mathbf{y}$, and the incident particle weight $\nu_{\mathrm{i}}$ (see Fig. 4.2). Its steps should be self-explanatory. Below we discuss some additional details of the full path sampling algorithm.

**First collision.** When tracing a particle from the camera, we initialize the RR/splitting factor $q$ to 1 at the first collision so no RR or splitting is carried out there. In Fig. 4.2, $\mathbf{x}$ is the camera vertex and $\mathbf{y}$ corresponds to the first collision. Applying our ADRRS in this case would only serve to compensate the variations of the light contribution within one pixel, which are usually small. On the other hand, when tracing particles form light sources, we initialize $q$ according to our algorithm right at the first collision, so RR/splitting can take place there. (In this case, in Fig. 4.2, $\mathbf{x}$ is a light source vertex.)

**Algorithm 3** Pseudocode showing the ADRRS-related steps in processing a collision event along an incrementally constructed path.

```
procedure HANDLECOLLISION($\mathbf{x}, \mathbf{y}, \omega_\mathrm{i}, \nu_\mathrm{i}$)
    // $\mathbf{x}$ ... previous collision location,  $\mathbf{y}$ ... current collision location
    // $\omega_\mathrm{i}$ ... incident direction at $\mathbf{y}$,  $\nu_\mathrm{i}$ ... particle weight
    CONTRIBUTE($\mathbf{y}$, $\omega_\mathrm{i}$, $\nu_\mathrm{i}$)                          ▷ Eq. (2.4) or (2.6)
    $\tilde{G}$ := LOOKUPCACHE($\mathbf{y}$)                                  ▷ Sec. 4.4.2
    $\tilde{\Psi}^\mathrm{r}_\mathrm{o}$ := CALCADJOINT($\tilde{G}$, $\mathbf{x}$, $\mathbf{y}$, $\omega_\mathrm{i}$)                 ▷ Eq. (4.7) or (4.9)
    $\langle\delta^-, \delta^+\rangle$ := CALCWWBOUNDS($\tilde{\Psi}^\mathrm{r}_\mathrm{o}$, $\tilde{I}$)              ▷ Eqns. (4.4) and (4.5)
    $[n, \hat{\nu}]$ := APPLYWW($\nu_\mathrm{i}$, $\langle\delta^-, \delta^+\rangle$)                      ▷ Sec. 4.4.1
    for $j = 1 \dots n$ do
        $\omega_\mathrm{o}$ := SAMPLEDIR($\mathbf{y}$)
        $\nu_\mathrm{o}$ := UPDATEWEIGHT($\hat{\nu}$, $\omega_\mathrm{i}$, $\omega_\mathrm{o}$)                    ▷ Eq. (2.7)
        $\mathbf{z}$ := INTERSECT($\mathbf{y}$, $\omega_\mathrm{o}$)
        HANDLECOLLISION($\mathbf{y}$, $\mathbf{z}$, $-\omega_\mathrm{o}$, $\nu_\mathrm{o}$)              ▷ Recurse to next event
    end for
end procedure
```

**Tree pruning.** To avoid overly bushy ray trees due to splitting, we impose a maximum splitting factor of 100 at each collision. In addition, we limit the size of the entire ray tree by the following heuristic. We associate a number $s_\mathrm{count}$ with every event along a path which conservatively estimates the total number of rays in the tree. We initialize $s_\mathrm{count}$ to 1 and we multiply it at every collision by the splitting factor determined at that event. We disable splitting once $s_\mathrm{count} > 1000$.

**Next-event estimation.** So far, we have only discussed unidirectional path or light tracing algorithms, but our ADRRS naturally extends to using next-event estimation (i.e. explicit connections to light sources or the camera). A theoretical justification is based on the idea of replacing self-emission, $L^\mathrm{e}_\mathrm{o}$ or $W^\mathrm{e}_\mathrm{o}$, by the sources of first-scattering events (e.g. direct illumination on surfaces serves as the new emission term). All the derivations can then be carried out with those re-defined source terms as before [Hoogenboom, 2008]. In practice, we do the following. Suppose we have reached a scattering event at $\mathbf{y}$ and determined the integer splitting factor of $n$. At this point, we in fact draw $n$ *pairs* of sample rays, one ray in each pair by sampling a scattering direction $\omega_\mathrm{o}$ and another by explicit light source or sensor sampling. The direct illumination contributions of these rays are then combined using multiple importance sampling [Veach, 1997].

## 4.5 Combining ADRRS with Path Guiding

In this section we combine our ADRRS with *path guiding* methods based on adjoint-driven importance sampling of scattering directions [Vorba et al., 2014, Bashford-Rogers et al., 2012, Hey and Purgathofer, 2002, Jensen, 1995]. Specifically, we show on our guiding method introduced in Chapter 3 that ADRRS works in synergy with path guiding and the combination leads to superior results than either method alone.

**Motivation.** In Chapter 3, we observed that the classic RR, based only on local material properties, is adverse to path guiding in scenes with difficult visibility (see Sec. 3.4.5). The reason is that a particle guided towards a high-contribution

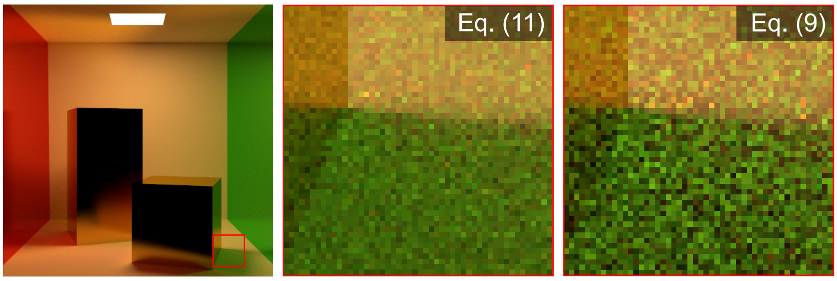


Figure 4.9: A glossy surface with (middle) and without (right) the conservative adjoint estimate given by Eq. (4.9). Both insets were rendered at 64 samples per pixel.

region may be terminated before it is able to reach it. In fact, aggressive RR may offset the advantages of path guiding entirely, and just add overhead.

To mitigate this problem and, in turn, to show benefits of path guiding, in Sec. 3.4.5, we employ RR with very low weight window threshold $\delta^- = \nu_\mathrm{o}^\mathrm{e} 10^{-6}$ leading to vastly increased average path length. Such a conservative, globally fixed threshold results in killing only particles which weight after emission $\nu_\mathrm{o}^\mathrm{e}$ decreases more than $10^{-6}\times$. Such an approach is not satisfactory because effort is wasted on sampling long paths in unimportant regions (see Fig. 4.11), while other paths may still be terminated prematurely. In addition to RR, in Sec. 3.4.5, we already applied splitting if the particle weight exceeded two times its original value $\nu_\mathrm{o}^\mathrm{e}$. Effectively, this approach is equivalent to using a globally fixed weight window with an extensive size of $s = 2 \times 10^6$.

**Our approach.** We address the above problem by using our ADRRS, which allows us to sample close-to-optimal path lengths without any adverse effects on path guiding. Moreover, ADRRS effectively improves the quality of path space importance sampling over baseline path guiding, as discussed in Sec. 4.6.1 and shown in Figs. 4.11, 4.13 and 4.14. We base our implementation on the on-line learning algorithm that we introduced in Chapter 3. We pre-compute and cache the diffuse quantity $\tilde{G}$ (Sec. 4.4.2) together with directional sampling distributions represented by GMM (see Sec. 3.3). ADRRS is applied both in the training and the rendering stages. As opposed to the basic approach from Sec. 4.4, the scattering directions $\omega_\mathrm{o}$ are sampled from the pre-computed directional distributions as in Sec. 3.4. Furthermore, we exploit the cached directional distributions to obtain more accurate estimates of the adjoint, which substantially improves the robustness of our method, especially in scenes with *glossy materials* (Fig. 4.9).

**Adjoint from directional distribution.** We use the cached directional distribution $\tilde{p}(\omega|\,\mathbf{x})$ at $\mathbf{x}$ to estimate the incoming adjoint at $\mathbf{x}$ reflected from $\mathbf{y}$, that does not involve the source illumination from $\mathbf{y}$, as

$$\tilde{\Psi}_\mathrm{i}^\mathrm{r}(\mathbf{x}, \omega_\mathrm{o}^\mathrm{x}) = \frac{\tilde{p}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x})\tilde{G}(\mathbf{x})}{\cos\theta_\mathbf{x}} - \Psi_\mathrm{o}^\mathrm{e}(\mathbf{y}, \omega_\mathrm{i}). \tag{4.8}$$

This equation follows from the fact that the directional distribution is designed such that $\tilde{p}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x}) \propto \Psi_\mathrm{i}(\mathbf{x}, \omega_\mathrm{o}^\mathrm{x}) \cos\theta_\mathbf{x}$ with $\tilde{G}(\mathbf{x})$ (Eq. (4.6)) being the normalization factor (see Sec. 3.4.2).

In practice, we obtain the final adjoint estimate as shown in Fig. 4.7 (b):

$$\tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y},\omega_{\mathrm{i}}) = \max\left\{\underbrace{\tilde{\Psi}^{\mathrm{r}}_{\mathrm{o}}(\mathbf{y})}_{\text{Eq. (4.7)}},\ \underbrace{\tilde{\Psi}^{\mathrm{r}}_{\mathrm{i}}(\mathbf{x},\omega^{\mathrm{x}}_{\mathrm{o}})}_{\text{Eq. (4.8)}}\right\}. \tag{4.9}$$

This conservative estimate produces a lower weight window center and thus lower termination rates, which eliminates some high-frequency noise otherwise appearing in certain scenarios especially in the presence of glossy materials (Fig. 4.9).

## 4.6 Adjoint-Driven Russian Roulette and Splitting and Zero-Variance Schemes

To understand the mechanism how ADRRS achieves importance sampling we first revise the relationship between particle weights and their density in path guiding methods. When particles are guided towards an important region where they make a significant contribution their density in the region increases while their weight decreases implicitly. Analogously, when the density of particles is decreased, their weight must increase. In other words the density of particles and their respective weights are inversely proportional. This is natural for MC simulations because the density of the particles (samples) together with their weights form an unbiased representation of radiance/importance in the scene in any given time[2] [Veach, 1997].

The holy grail of importance sampling that can be achieved by guiding methods is following the zero-variance (ZV) sampling scheme described in Sec. 2.8. Because, under this scheme, we would obtain the expected value of our estimator with only one sample, it is desirable to design any importance sampling technique, including our ADRRS, to achieve similar particle density as we have under the ZV scheme.[3] However, while path guiding changes the particle density explicitly by placing more samples into important directions, our ADRRS rather adjusts the particle weights. To achieve the desirable particle density we have to look how the particle weight behaves under the ZV scheme (Sec. 2.8).

Thus, as described in Sec. 4.3.3, we set the termination/splitting rate $q$ in our ADRRS so that the particle weights follow the same principle (Eq. (4.3)) as the particle weights under the ZV scheme. In this section, we study the importance sampling properties of ADRRS formally through the inspection of the rate $q$.

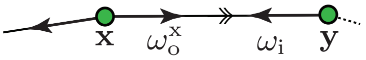
### 4.6.1 Zero Variance, Importance Sampling, and ADRRS

To study importance sampling properties of ADRRS and its relation to ZV schemes, we derive, in Appendix 4.A, the following equation for the RR/splitting rate $q$ at a collision $\mathbf{y}$:

$$\boxed{q(\mathbf{y},\omega_{\mathrm{i}}) = \frac{p_{\mathrm{ZV}}(\omega^{\mathrm{x}}_{\mathrm{o}}|\,\mathbf{x})}{p(\omega^{\mathrm{x}}_{\mathrm{o}}|\,\mathbf{x})}.} \tag{4.10}$$

[2] This is not entirely true for the expected-value split (see Sec. 4.4.1), however, the statement still holds in the limit.

[3] Even under the ZV scheme there is a density that the sample comes from. To approximate this density visually, imagine taking more samples from the ZV scheme.



That is to say, the RR/splitting rate at $\mathbf{y}$ is given by the ratio of the zero-variance pdf to the actually used pdf for sampling the scattering direction $\omega_\mathrm{o}^\mathrm{x}$ at the *preceding* vertex $\mathbf{x}$. The pdf $p_\mathrm{zv}(\omega|\,\mathbf{x})$ that ensures ZV estimation (see Sec. 2.8) is proportional to the product of the cosine-weighted BSDF lobe and the directional distribution of the adjoint quantity incident at $\mathbf{x}$: $p_\mathrm{zv}(\omega|\,\mathbf{x}) \propto \Psi_\mathrm{i}(\mathbf{x},\omega) f_\mathrm{s}^+(\mathbf{x},\omega_\mathrm{i}^\mathrm{x}{\to}\omega)|\cos\theta_\mathbf{x}|$ (see Eq. (2.12)). Here, we use $\omega_\mathrm{i}^\mathrm{x}$ to denote the incident direction at $\mathbf{x}$. Note that to describe both light tracing and path tracing, we define $f_\mathrm{s}^+$ so that $f_\mathrm{s}^+(\cdot,\omega_\mathrm{i}^\mathrm{x}{\to}\omega_\mathrm{o}^\mathrm{x}) = f_\mathrm{s}(\cdot,\omega_\mathrm{i}^\mathrm{x}{\to}\omega_\mathrm{o}^\mathrm{x})$ for light tracing while for path tracing, we swap the directions: $f_\mathrm{s}^+(\cdot,\omega_\mathrm{i}^\mathrm{x}{\to}\omega_\mathrm{o}^\mathrm{x}) = f_\mathrm{s}(\cdot,\omega_\mathrm{o}^\mathrm{x}{\to}\omega_\mathrm{i}^\mathrm{x})$. To simplify the following discussion, we assume, with no bearing on our results, that $\mathbf{y}$ is not on a source or sensor; general form of Eq. (4.10) is presented in Appendix 4.A.

Interestingly, Eq. (4.10), which is derived for infinitesimal weight window (i.e. $s = 1$ in Eq. (4.5)), shows that $q$ takes the same form at any collision irrespective of any previous collisions before $\mathbf{x}$ (i.e. there is no weight term $\nu_\mathrm{i}$ of a particle incident at $\mathbf{x}$). We can thus limit ourselves to discussing the effect of ADRRS on the variance of a local estimator of the (hemi)spherical integral at $\mathbf{x}$.

**RR and splitting as importance sampling.** Suppose we have drawn the scattering direction $\omega_\mathrm{o}^\mathrm{x}$ from a general pdf $p(\omega|\,\mathbf{x})$. Eq. (4.10) states that the factor $q$ at $\mathbf{y}$ is determined by comparing the pdf value for the sampled direction to the pdf value $p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x})$ dictated by the ZV scheme. We keep samples untouched in those parts of the (hemi)sphere where $p(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x}) = p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x})$, we split where our sampling rate is too low, i.e. $p(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x}) < p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x})$, while we randomly terminate where we place too many samples, i.e. $p(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x}) > p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x})$. We illustrate these situations in Fig. 4.10.

**Residual variance.** In the case of RR, the above procedure is equivalent to *rejection sampling*, where $p(\omega|\,\mathbf{x})$ serves as the proposal density and $p_\mathrm{zv}(\omega|\,\mathbf{x})$ is the target density. However, splitting cannot reduce any variance introduced due to sampling from $p(\omega|\,\mathbf{x})$ — as opposed to the ideal pdf $p_\mathrm{zv}(\omega|\,\mathbf{x})$ — because splitting only occurs at the next collision event $\mathbf{y}$. Only variance from subsequent bounces can be reduced. It is therefore not effective when most of the variance is gained through the use of an inappropriate scattering pdf at $\mathbf{x}$. An example is shown and discussed in Sec. 4.7. Booth [2012a] provides further discussion of this limitation.

**Zero-variance sampling.** By using ADRRS on top of a ZV scheme (i.e. when $p(\omega|\,\mathbf{x}) = p_\mathrm{zv}(\omega|\,\mathbf{x})$ for all $\omega$), we still obtain a ZV estimator because $q$ becomes

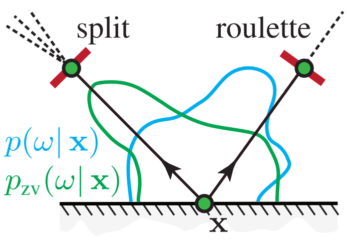


Figure 4.10: We split in those parts of the (hemi)sphere where our sampling rate is too low, i.e. $p(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x}) < p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x})$, while we randomly terminate where we place too many samples, i.e. $p(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x}) > p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x}|\,\mathbf{x})$.

$q_{\text{zv}}$, the ZV termination rate (Eq. (2.13) for light tracing). In other words, it follows from Eq. (4.10) that $q = 1$ when $\mathbf{y}$ is not on a light source or sensor and thus no termination or splitting[4] takes place which is inline with the ZV scheme. However, there is a clear difference between direct sampling from $p_{\text{zv}}(\omega|\,\mathbf{x})$ in the ZV scheme and ADRRS, where we can sample from an arbitrary $p(\omega|\,\mathbf{x})$. While the former has zero variance, i.e. it solves the integral with a single sample, ADRRS only strives for variance reduction through RR-implied rejection and splitting at $\mathbf{y}$.

**Path guiding methods.** Path guiding methods strive to sample scattering directions from a pdf $p(\omega)$ that closely approximates the ZV pdf $p_{\text{zv}}(\omega)$ [Vorba et al., 2014, Hey and Purgathofer, 2002, Jensen, 1995]. Path guiding and ADRRS work in synergy to approximate the optimal ZV scheme even more closely. To see why, recall the rejection sampling interpretation of ADRRS. On one hand, the better the path guiding distribution (i.e. the closer the proposal pdf $p(\omega)$ to the target ZV pdf $p_{\text{zv}}(\omega)$), the less work for RR and splitting. On the other hand, should the path guiding distribution $p(\omega)$ fail to approximate the ZV pdf $p_{\text{zv}}(\omega)$ closely enough, our ADRRS still steps in to improve the effective particle distribution.

We can explain the compatibility of ADRRS and path guiding from another perspective. We show in Appendix 4.B that ADRRS applied in a zero-variance scheme (emission and scattering follows Eq. (2.11) and Eq. 2.12 respectively) yields again a ZV scheme.

## 4.7 Results

Here, we experimentally validate the theoretical outcomes of Sec. 4.6. Results in Figs. 4.13 and 4.14 confirm the strong variance reduction capabilities of our adjoint-driven Russian roulette and splitting (ADRRS) in path tracing (PT) and light tracing (LT). Additionally, we show the results of the combination of ADRRS with path guiding (PG) described in Chapter 3, which yields a practical and efficient algorithm capable of rendering scenes with complex and difficult visibility. Interestingly, this combination is beneficial even in simple scenes where the overhead of path guiding, coming from excessive path lengths, would offset its advantages if used without ADRRS (Fig. 4.11). We encourage the reader to view the supplemental material (`http://cgg.mff.cuni.cz/~jirka/papers/2016/adrrs/index.htm`) for all our results rendered with PT, LT and also progressive photon mapping (PPM) [Hachisuka et al., 2008].

**Setup.** All images in Figs. 4.13 and 4.14 were rendered for 1 hour on an Intel Core i7-2600K CPU using 8 logical cores. Our implementation is based on the Mitsuba renderer [Jakob, 2010]. We set the maximum path length to 40 bounces. The 'plain' light and path tracing algorithms use classic albedo-based RR from the fifth bounce on, and no splitting.

We include the pre-computation time in the reported total time of all our results. The pre-computation times of PG and our ADRRS with path guiding

[4] In fact, interestingly, *integer splitting* (see Sec. 4.4.1) does not increase variance and thus does not violate ZV scheme. However, any unnecessary splitting increases time per sample (particle path).

(ADRRS + PG) are listed in Table 4.1. Results rendered with our ADRRS-only use the same pre-computation as ADRRS + PG to achieve the same quality of cached irradiance/visual importance. This allows us to compare the effect of adding PG on top of ADRRS. Unlike ADRRS + PG, the PG-only results do not use our ADRRS in the training stage.

| | **ADRRS + PG** | **PG** |
|---|---|---|
| Crytek sponza | 150s | 114s |
| Veach door | 222s | 192s |
| Living room | 462s | 426s |
| Classroom | 588s | 462s |

Table 4.1: Training times of ADRRS + PG and PG-only.

**Scenes.** We use the same three scenes that we described in Sec. 3.5, namely *Living room*, *Veach door* and *Classroom*, without any change. Our fourth scene is the *Crytek Sponza* [2010]. Common to all the scenes is complex visibility with many regions lit only by high-order indirect illumination.

In the **Crytek Sponza** scene with mainly diffuse materials, the sunlight enters the atrium and indirectly illuminates most of the shot. Our ADRRS alone achieves substantial variance reduction in comparison to standard path and light tracing. Using path guiding on top of our ADRRS (ADRRS+PG) yields superior result without any of the spike noise present in path guiding alone.

All the illumination in the **Veach door** scene enters through the door ajar from the back room. Note, that our version of the scene differs from that of Lehtinen et al. [2013] in that the light source size is roughly 250× smaller. This makes our scene more realistic – and substantially more challenging (see Fig. 4.12). In PT, our ADRRS + PG combination significantly reduces the spike noise produced by path guiding alone. However, unlike in the Crytek Sponza scene, some of this noise still remains. This is due to the combination of the small light source and specular reflections on the floor, which effectively disqualifies any next event estimation. When a path guiding distribution on a wall fails to accurately target the caustic-like illumination due to specular light source reflection on the floor,

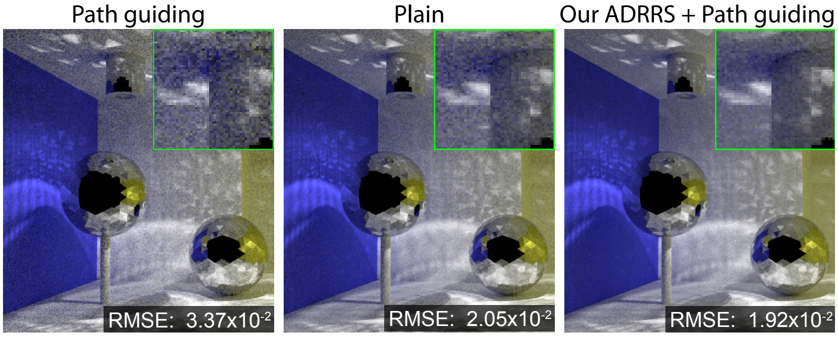


Figure 4.11: Our path guiding (PG) (see Chapter 3) (left) applied in Progressive Photon Mapping (PPM) does not match the efficiency of plain PPM (middle) due to its overhead coming mainly from sampling long paths. PPM with our ADRRS and PG (right) achieves superior results than plain PPM even in this simple scene. All images took 5 minutes to render (including the training time).

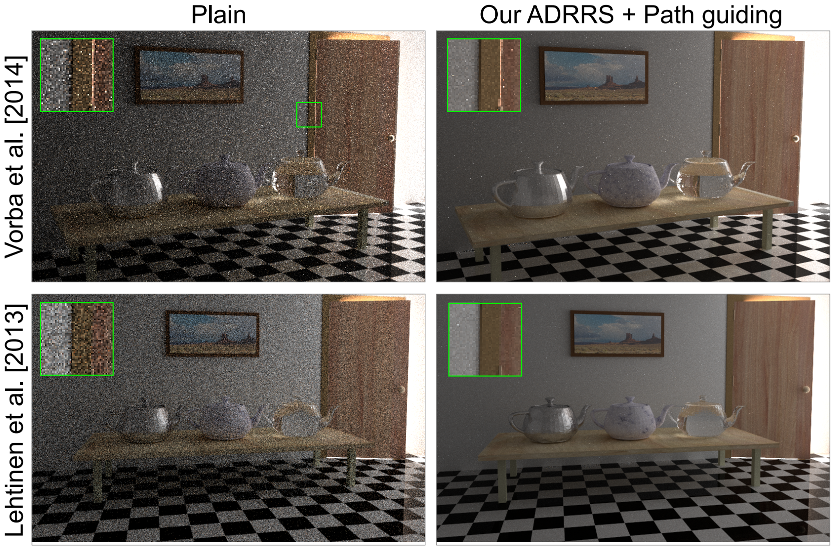


Figure 4.12: Two versions of the Veach Door scene. We rendered 40 light bounces by plain path tracing (left) and path tracing with our ADRRS and path guiding (right). The Vorba et al. version (top) is more challenging than the version of Lehtinen et al. (bottom): its 250× smaller light source is responsible for the spiky noise.

ADRRS cannot remedy the situation by splitting that is decided on the wall (vertex $\mathbf{x}$ in Fig. 4.2). This is because physical splitting on the specular reflection on the floor (vertex $\mathbf{y}$) is ineffective (see “Residual Variance” in Sec. 4.6.1)

Illumination in both the **Living room** and **Classroom** scenes is similar. The Sun is shining through window glass covered by curtains and jalousies respectively, and large parts of the scenes are lit by light after many bounces. While in LT, our ADDRS alone provides excellent results, the results in PT expose the ADRRS limitations discussed in Sec. 4.6.1. Referring back to the rejection sampling interpretation of ADRRS, the proposal distribution could be the floor BSDF, while the target zero-variance distribution is close to a delta-distribution as it encompasses the sunlight passing through the window-panes. ADRRS cannot reduce variance in this case because the physical split can only occur on the glass (vertex $\mathbf{y}$ in Fig. 4.2), where it would be ineffective. Nevertheless, the combination of our ADRRS with path guiding addresses the problem and yields superior results.

**Effect of RR and splitting.** We show separately the effect of our adjoint-driven RR (ADRR) and adjoint-driven splitting in Fig. 4.15. We rendered the Crytek Sponza scene for 20 minutes with a guided path tracer (a), guided path tracer with our ADRR (no splitting) (b), and guided path tracer with full ADRRS (c). The variance reduction in (b) stems from sampling nearly optimal path lengths by our ADRR. The splitting in (c) significantly improves sampling of regions where light is transported through several bounces.

## 4.8 Limitation, Discussion, and Future Work

**Inaccurate adjoint and measurement estimates.** Grossly inaccurate adjoint or measurement estimates can produce increased variance, as in the top right image in Fig. 4.12 under the table. This can happen for example due to light leaks well-

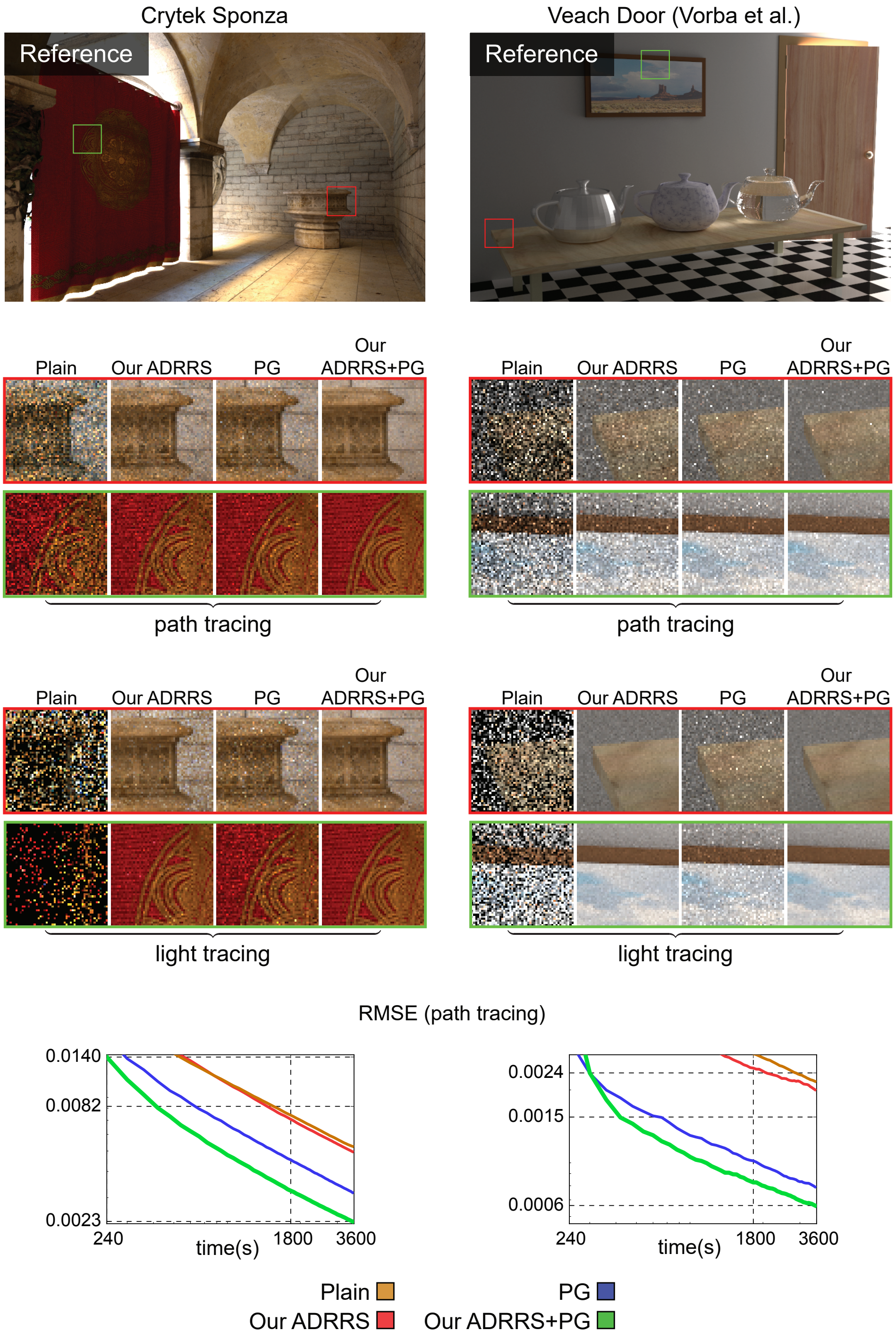


Figure 4.13: We render four scenes dominated by indirect lighting for 1 hour. Results in the figure come from path and light tracing, respectively, with albedo-based RR applied from the fifth bounce on (Plain), our ADRRS, path guiding alone (PG), and our ADRRS with PG. Our ADRRS achieves substantial variance reduction over the albedo-based RR (Plain). Superior results are achieved by complementing our ADRRS with path guiding. The training time is included in the reported 1 hour. Complete images are shown in the supplemental material (`http://cgg.mff.cuni.cz/~jirka/papers/2016/adrrs/index.htm`).

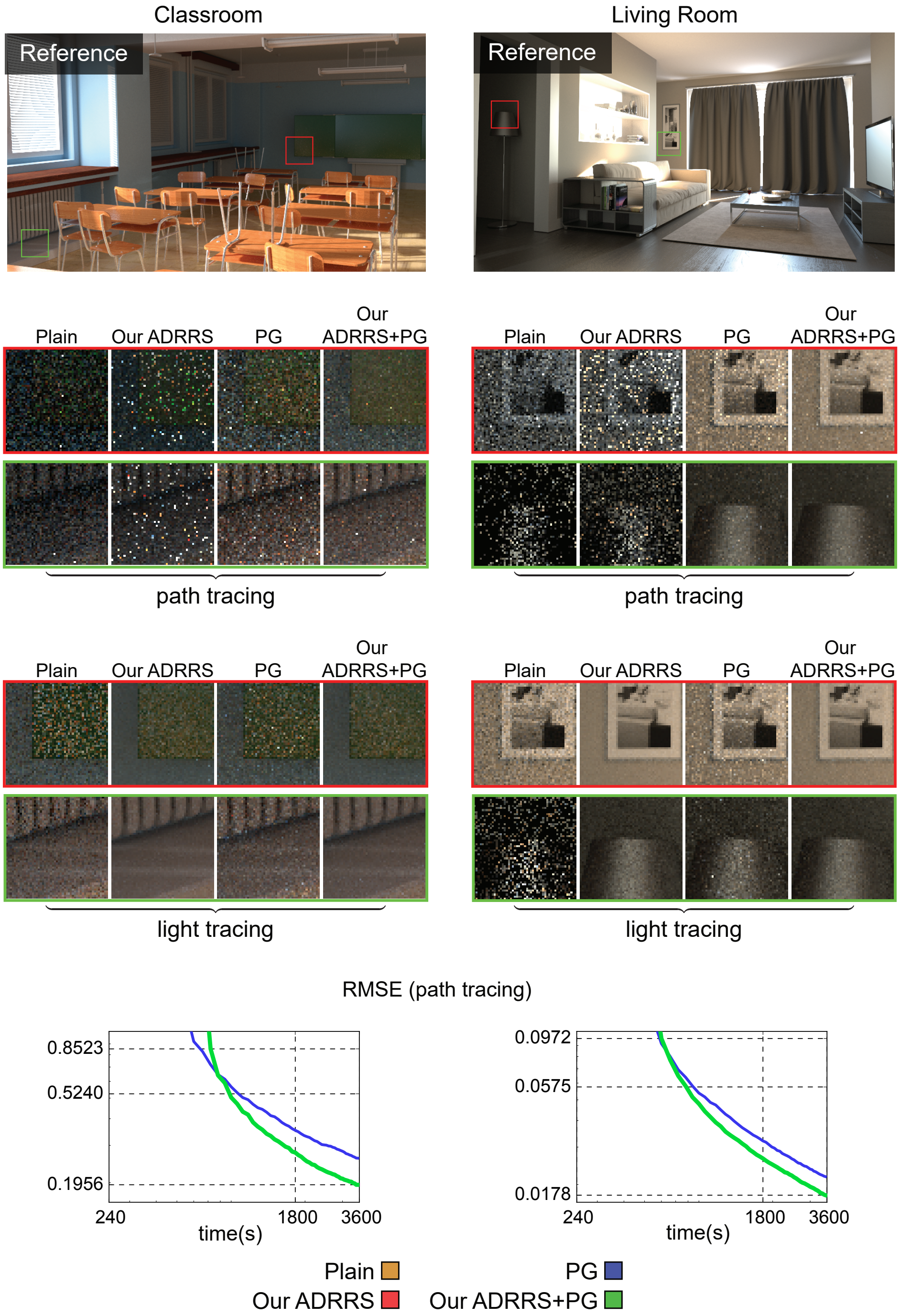


Figure 4.14: We render four scenes dominated by indirect lighting for 1 hour. Results in the figure come from path and light tracing, respectively, with albedo-based RR applied from the fifth bounce on (Plain), our ADRRS, path guiding alone (PG), and our ADRRS with PG. Our ADRRS achieves substantial variance reduction over the albedo-based RR (Plain). Superior results are achieved by complementing our ADRRS with path guiding. The training time is included in the reported 1 hour. Note that the convergence curves for plain and our ADRRS are above the displayed interval. Complete images are shown in the supplemental material (`http://cgg.mff.cuni.cz/~jirka/papers/2016/adrrs/index.htm`).

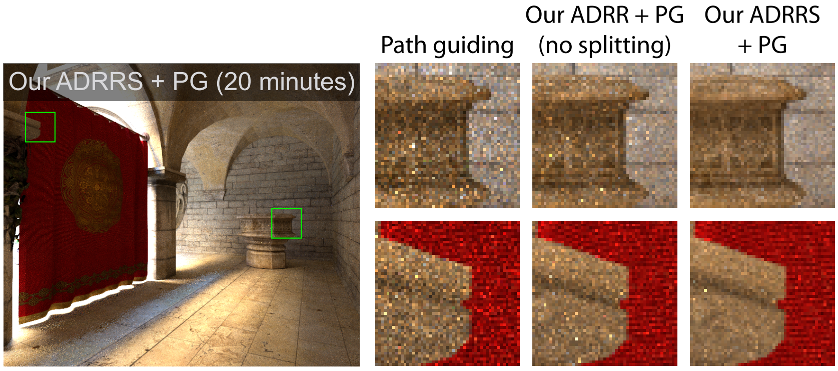


Figure 4.15: Our adjoint-driven RR without splitting (middle) in guided path tracing reduces variance of guided path tracing (left). Application of splitting (right) provides a substantial additional variance reduction in regions reached by light after several events.

known from photon mapping, such as those visible in Fig. 4.8. This limitation is common to all variance reduction techniques based on estimated quantities, and in practice it can be alleviated by adaptive image sampling.

**Efficiency-driven RR and splitting.** While the close relation of our ADRRS to zero-variance path sampling schemes provides a solid justification of its variance reduction properties, nothing in the theory suggests that ADRRS would be optimal with respect to efficiency (inverse of a product of time and variance). As such, an efficiency-driven RR and splitting is an important avenue for future research.

**Splitting and combined estimators.** We have shown that splitting is an effective variance reduction tool in unidirectional path sampling algorithms. It would be interesting to extend its use to bidirectional algorithms based on combining various estimators [Křivánek et al., 2014, Georgiev et al., 2012b, Hachisuka et al., 2012]. A challenge associated with this idea would be the development of proper combination weights that would respect the correlation of split paths due to their common shared prefix [Popov et al., 2015].

**Participating media.** When simulating transport in media, particles typically undergo many more events than on surfaces before making an image contribution. At the same time, whenever a light source or the camera is inside a medium, there can be tremendous variation of the respective adjoint quantity. For those reasons, it is likely that extending our method to participating media would result in a substantially greater increase in efficiency than on surfaces.

## 4.9 Conclusion

We have introduced an approach for selective path termination and splitting that we call adjoint-driven Russian roulette and splitting (ADRRS). The termination and splitting decisions are driven by a pre-computed estimate of the adjoint transport quantity so that a significant variance reduction is achieved. We have provided a theoretical justification of the variance reduction properties – and its limits – by juxtaposing ADRRS to a zero-variance path sampling scheme. To make the method practical and robust, we have introduced the idea of adaptive weight window from the neutron transport field. We have shown that our ADRRS complements the directional importance sampling techniques (path guiding) and together they result in robust and efficient simulations even in fairly simple *unidirectional* methods such as path and light tracing. These are easier to implement than combined path integral estimators such as bidirectional path tracing [Veach, 1997] or vertex connection and merging [Georgiev et al., 2012b, Hachisuka et al., 2012] and are favored in practice for their easy combination with a broad scale of production features.

# Appendix

## 4.A Derivation of Eq. (4.10)

In this appendix, we derive the form of splitting rate $q(\mathbf{y}, \omega_\mathrm{i})$ at a collision $\mathbf{y}$ (Eq. (4.10))

$$\boxed{q(\mathbf{y}, \omega_\mathrm{i}) = \frac{p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x} | \, \mathbf{x})}{p(\omega_\mathrm{o}^\mathrm{x} | \, \mathbf{x})},}$$

that reveals an important relationship between the ideal ZV sampling pdf $p_\mathrm{zv}(\omega_\mathrm{o}^\mathrm{x} | \, \mathbf{x})$ and the actual sampling pdf $p(\omega_\mathrm{o}^\mathrm{x} | \, \mathbf{x})$ at the previous collision $\mathbf{x}$ (see Fig. 2.3). In fact, we derive a more general case

$$q(\mathbf{y}, \omega_\mathrm{i}) = \frac{p_\mathrm{zv}^\mathrm{r}(\omega_\mathrm{o}^\mathrm{x} | \, \mathbf{x})}{p(\omega_\mathrm{o}^\mathrm{x} | \, \mathbf{x})}, \tag{4.11}$$

that is valid also when $\mathbf{y}$ is on a light source or the camera. Here, in the case of light tracing, the function $p_\mathrm{zv}^\mathrm{r}$ is defined as a product of reflected visual importance $W_\mathrm{o}^\mathrm{r}$ at $\mathbf{y}$ into a direction $-\omega$ and BSDF $f_\mathrm{s}$ at $\mathbf{x}$, divided by reflected visual importance $W_\mathrm{o}^\mathrm{r}(\mathbf{x}, \omega_\mathrm{i}^\mathrm{x})$ at $\mathbf{x}$:

$$p_\mathrm{zv}^\mathrm{r}(\omega | \, \mathbf{x}) = \frac{W_\mathrm{o}^\mathrm{r}(\mathbf{y}, -\omega) f_\mathrm{s}(\mathbf{x}, \omega_\mathrm{i}^\mathrm{x} \to \omega) |\cos\theta_\mathrm{o}|}{W_\mathrm{o}^\mathrm{r}(\mathbf{x}, \omega_\mathrm{i}^\mathrm{x})}.$$

The only difference between $p_\mathrm{zv}^\mathrm{r}(\omega | \, \mathbf{x})$ and the zero-variance scattering distribution $p_\mathrm{zv}(\omega | \, \mathbf{x})$ (Eq. (2.12)) is that the function $p_\mathrm{zv}^\mathrm{r}$ does not include direct importance emitted from $\mathbf{y}$. Thus $p_\mathrm{zv}(\omega | \, \mathbf{x}) = p_\mathrm{zv}^\mathrm{r}(\omega | \, \mathbf{x})$ if $\mathbf{y}$ is not on the camera (or a light source in case of path tracing). Without loss of generality, we show validity of Eq. 4.11 only in *light tracing.*

As explained in Sec. 4.3.3, we design our ADRRS so that the particle weight after termination/splitting at a collision $\mathbf{y}$

$$\hat{\nu}(\mathbf{y}, \omega_\mathrm{i}) = \frac{I}{W_\mathrm{o}^\mathrm{r}(\mathbf{y}, \omega_\mathrm{i})} \tag{4.12}$$

is a ratio of the computed solution $I$ and reflected visual importance at $\mathbf{y}$ into the direction $\omega_\mathrm{i}$ from which the particle comes.

From Eq. (4.1)

$$\hat{\nu}(\mathbf{y}, \omega_\mathrm{i}) = \frac{\nu_\mathrm{i}(\mathbf{y}, \omega_\mathrm{i})}{q(\mathbf{y}, \omega_\mathrm{i})},$$

that describes the change of the incident particle weight $\nu_\mathrm{i}$ after temination/splitting, and from Eq. (4.12), we can express the termination/splitting rate $q$ at $\mathbf{y}$ as

$$q(\mathbf{y}, \omega_\mathrm{i}) = \frac{\nu_\mathrm{o}(\mathbf{x}, \omega_\mathrm{o}^\mathrm{x}) W_\mathrm{o}^\mathrm{r}(\mathbf{y}, \omega_\mathrm{i})}{I}. \tag{4.13}$$

Note that we have also used the identity between $\nu_\mathrm{i}$ at $\mathbf{y}$ and $\nu_\mathrm{o}$ at $\mathbf{x}$ (see Fig. 2.3).

To get from Eq. (4.13) to the desired Eq. (4.11), we need to express the outgoing particle weight $\nu_\mathrm{o}(\mathbf{x}, \omega_\mathrm{o}^\mathrm{x})$ at $\mathbf{x}$ in terms of path weight $\hat{\nu}$ after application of ADRRS at $\mathbf{x}$. When the particle scatters at $\mathbf{x}$ into a direction $\omega_{\mathrm{o}}^{\mathrm{x}}$, according to Eq. (2.7), its outgoing weight $\nu_{\mathrm{o}}$ is computed as follows:

$$\nu_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}^{\mathrm{x}}) = \hat{\nu}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}}) \frac{f_{\mathrm{s}}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}} \to \omega_{\mathrm{o}}^{\mathrm{x}}) |\cos \theta_{\mathrm{o}}|}{p(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})}. \tag{4.14}$$

Because we design our ADRRS so that the particle weight invariant (Eq. (4.12)) holds at every collision, it also holds at $\mathbf{x}$:

$$\hat{\nu}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}}) = \frac{I}{W_{\mathrm{o}}^{\mathrm{r}}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}})}.$$

Thus Eq. 4.14 becomes

$$\nu_{\mathrm{o}}(\mathbf{x}, \omega_{\mathrm{o}}^{\mathrm{x}}) = \frac{I}{W_{\mathrm{o}}^{\mathrm{r}}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}})} \frac{f_{\mathrm{s}}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}} \to \omega_{\mathrm{o}}^{\mathrm{x}}) |\cos \theta_{\mathrm{o}}|}{p(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})}. \tag{4.15}$$

Finally, when we insert Eq. (4.15) into Eq. (4.13), we arrive at the splitting rate $q$ that had to been proven:

$$q(\mathbf{y}, \omega_{\mathrm{i}}) = \underbrace{\frac{W_{\mathrm{o}}^{\mathrm{r}}(\mathbf{y}, \omega_{\mathrm{i}}) f_{\mathrm{s}}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}} \to \omega_{\mathrm{o}}^{\mathrm{x}}) |\cos \theta_{\mathrm{o}}|}{W_{\mathrm{o}}^{\mathrm{r}}(\mathbf{x}, \omega_{\mathrm{i}}^{\mathrm{x}})}}_{p_{\mathrm{zv}}^{\mathrm{r}}(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})} \frac{1}{p(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})} \tag{4.16}$$

## 4.B Zero-variance Sampling and ADRRS

In this Appendix, we show that using our ADRRS with infinitesimal weight window, i.e. $s = 1$ (see Sec. 4.4.1), on top of a ZV scheme, does not increase variance and results again in a ZV scheme. Because the zero-variance rules for emission (Eq. (2.11)) and scattering (Eq. (2.12)) are not affected by our ADRRS, we only need to show that our termination/splitting rate $q$ is equal to the zero-variance termination rate $q_{\mathrm{zv}}$.

We start from the form of $q$ given by Eq. (4.11) that we proved in the previous section. We also use the fact that under the ZV scheme the scattering pdf $p(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})$ is equal to $p_{\mathrm{zv}}(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})$ given by Eq. (2.12). Hence we get

$$q(\mathbf{y}, \omega_{\mathrm{i}}) = \frac{p_{\mathrm{zv}}^{\mathrm{r}}(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})}{p_{\mathrm{zv}}(\omega_{\mathrm{o}}^{\mathrm{x}} | \, \mathbf{x})} = \frac{W_{\mathrm{o}}^{\mathrm{r}}(\mathbf{y}, \omega_{\mathrm{i}})}{W_{\mathrm{o}}(\mathbf{y}, \omega_{\mathrm{i}})}. \tag{4.17}$$

Here, the last equality follows from the relationship between $W_{\mathrm{i}}$ and $W_{\mathrm{o}}$ (see Fig. 2.3) which yields the zero-variance survival probability (Eq. (2.13)). This shows that application of our ADRRS with infinitesimal weight window preserves all zero-variance rules and does not increase variance[5].

**Discussion of our implementation.** Note that unlike the zero-variance scheme, in our implementation of ADRRS, we always count contribution from implicitly hit light sources which theoretically increases variance if light sources scatter light. The reason is that the ZV scheme does not allow contribution from particles that have reached a light source (or sensor), unless the particle is terminated. This stochastic contribution with probability $1 - q_{\mathrm{zv}}$ is necessary to achieve the

[5] Provided that we use the exact adjoint and not its approximation.

zero-variance with contribution equal to $I$. This suggests that in our ADRRS, we should ideally apply RR/splitting before we count a possible contribution from $\mathbf{y}$ and count the contribution only when the particle is terminated.

In practice, the introduced variance by deterministic contribution has a very limited impact because usually the intensity of a light source is by orders of magnitude higher than scattered light. Furthermore, the maximal contribution at this case is equal to the computed solution $I$ and thus cannot result in so called "fireflies" (i.e. pixels by orders of magnitude brighter than their neighbours). To see this we express the value of our deterministic contribution when a particle reaches the sensor (it holds analogously for light sources). The weight of a particle reaching a collision at $\mathbf{y}$ is equal to $\frac{I}{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}$ (Eq. (2.15)). To get the deterministic contribution when $\mathbf{y}$ is on a sensor we multiply the particle weight by emitted visual importance $W_\mathrm{o}^\mathrm{e}$:

$$I\frac{W_\mathrm{o}^\mathrm{e}(\mathbf{y},\omega_\mathrm{i})}{W_\mathrm{o}(\mathbf{y},\omega_\mathrm{i})}. \tag{4.18}$$

Because $W_\mathrm{o} = W_\mathrm{o}^\mathrm{e} + W_\mathrm{o}^\mathrm{r}$, the contribution is the highest when the sensor does not scatter any energy and it is equal to the solution $I$.

# Chapter 5

# Conclusion

## 5.1 Summary

In this thesis, we tackled a fundamental weakness of Monte Carlo light transport simulation: slow convergence in the presence of indirect illumination. We proposed an approach based on an approximation of the ideal zero-variance sampling scheme that prescribes the exact sampling process of entire paths of transported particles.

While sampling with zero variance cannot be achieved without prior knowledge of the computed solution itself, it has been shown before that even an approximate knowledge of the adjoint transport in the scene (e.g. radiance for path tracing) can result in significant variance reduction. In our approach, we pre-computed such an approximation to adjoint transport and we used it to drive an approximate zero-variance sampling scheme. We split this thesis into two major parts, each of them describing a different aspect of the zero-variance sampling scheme.

In the first part, we proposed a new *direction sampling* for scattered particles and *position sampling* for emitted particles, where we learned sampling distributions represented by mixture of Gaussians from an on-line stream of adjoint particles. Unlike previous approaches, we showed the ability to handle directional distributions of both high and low angular frequency. Additionally, our method can cope with scenarios where only a tiny fraction of all emitted adjoint particles are available in important locations in the early stages of the learning process.

In the second part, we used the pre-computed adjoint transport to design the optimal *termination and splitting* rate for every particle position and direction in the scene. We adopted ideas from neutron transport where similar Russian roulette and splitting schemes have been used as standard variance reduction tools for many years. However, in our work, for the first time, we clearly showed a theoretical connection between the zero-variance sampling and particle path termination and splitting based on the expected contribution of the particle to the final solution. Our theoretical analysis predicts that both methods, the direction sampling developed in the first part and the adjoint-driven Russian roulette (termination) and splitting developed in the second part, can be combined to form a sampling scheme superior to each method alone. We also designed a practical algorithm that combines both methods and we showed results supporting the outcomes of our theoretical analysis.

We believe that our algorithms will find their application, among other, in architectural visualization and the movie industry, where light transport simulation in virtual scenes with ever increasing complexity is used for producing photo-realistic imagery. We also believe that our work and achieved results will encourage further research into sampling methods based on zero-variance sampling schemes and also into use of machine learning methods in light transport simulations. We hope that such a research could further reduce excessively long simulation times reported across different industries.

## 5.2 Future Directions in Light Transport Research

We presume that future research will aim at further improvements of unidirectional algorithms so that they can match the capabilities of their more complex (and complicated) bidirectional counterparts such as bidirectional path tracing, or vertex connection and merging a.k.a. unified path space. While the bidirectional estimators have recently been implemented and used in the movie industry [Seymour, 2014a,b], it turns out that they suffer from a few practical disadvantages. First, their maintenance is quite difficult in the agile production renderers. Second, it is very challenging or sometimes even impossible to support some non-physical production features which violate the adjoint nature of light transport.

Nevertheless, in this thesis, we were able to show that guided unidirectional methods which use our direction importance sampling (Chapter 3) almost match the qualities of guided bidirectional estimators. This result suggests that it might be possible to develop an efficient, purely unidirectional algorithm that would cope with challenging light transport. Note that our importance sampling algorithms that we presented are also bidirectional in the sense that we compute the transport approximation from the particles traced in the opposite direction (for example in path tracing, we use photons) and thus still more or less suffer from aforementioned disadvantages.

Developing an efficient purely unidirectional method without any use of particles from the opposite direction requires developing algorithms that would adaptively learn adjoint transport from the forward particles – that is from the same particles (samples), that are subsequently used for rendering. Recently, Dahm and Keller [2017] has taken a step in this direction when they formulated light transport as a reinforcement learning problem. Another important ingredient to achieve the outlined goal are further improvements to the adjoint approximation of light transport in both spatial and directional domain. Herholtz and colleagues [2016] have recently taken a step in this direction by introducing a product importance sampling of BSDFs and incident light. Similar improvements will result in better importance sampling schemes closer to the ideal zero-variance solution.

Another important direction of the future research is, undoubtedly, an extension of our work to participating media. While light transport in participating media is an integral part of production renderers in the movie industry and also in architectural visualisation, it is still subject to active research as the simulation times are currently extremely high.

# Bibliography

James Arvo and David Kirk. Particle transport and image synthesis. In *Proc. SIGGRAPH '90*, pages 63–66. ACM, 1990. ISBN 0-89791-344-2. doi: 10.1145/97879.97886. URL `http://doi.acm.org/10.1145/97879.97886`.

Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. A significance cache for accelerating global illumination. *Computer Graphics Forum*, 31(6): 1837–51, 2012.

Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. Importance driven environment map sampling. *IEEE Trans. Vis. Comput. Graphics*, 19, 2013. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.258.

Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, 2(6):pp. 1201–1225, 1974. ISSN 00905364. URL `http://www.jstor.org/stable/2958339`.

Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

Mark R. Bolin and Gary W. Meyer. An error metric for Monte Carlo ray tracing. In *In Rendering Techniques '97*, 1997.

E. Booth, T. Genesis of the weight window and the weight window generator in MCNP - a personal history. Technical Report LA-UR-06-5807, July 2006.

E. Booth, T. Common misconceptions in Monte Carlo particle transport. *Applied Radiation and Isotopes*, 70, 2012a.

E. Booth, T. and S. Hendricks, J. Importance estimation in forward MC calculations. *Nuc. Tech./Fusion*, 5(1), 1984. ISSN 0272-3921.

T.E. Booth. A sample problem for variance reduction in MCNP. Technical report, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, 1985a. URL `http://books.google.cz/books?id=4fpcHQAACAAJ`.

Thomas E. Booth. Monte Carlo variance comparison for expected-value versus sampled splitting. *Nucl. Sci. Eng.*, 89(4), 1985b.

Thomas E. Booth. Common misconceptions in Monte Carlo particle transport. *Applied Radiation and Isotopes*, 70(7), 2012b. ISSN 0969-8043. doi: http://dx.doi.org/10.1016/j.apradiso.2011.11.037. URL `http://www.sciencedirect.com/science/article/pii/S0969804311005914`.

B. C. Budge, J. C. Anderson, and K. I. Joy. Caustic forecasting: Unbiased estimation of caustic lighting for global illumination. *Computer Graphics Forum*, 27(7):1963–70, 2008.

Olivier Cappé. *Online Expectation Maximisation*, pages 31–53. John Wiley & Sons, Ltd, 2011. ISBN 9781119995678. doi: 10.1002/9781119995678.ch2. URL http://dx.doi.org/10.1002/9781119995678.ch2.

P. H. Christensen. Adjoints and importance in rendering: An overview. *IEEE Trans. Vis. Comput. Graphics*, 9(3):329–340, 2003. ISSN 1077-2626. doi: 10.1109/TVCG.2003.1207441. URL http://dx.doi.org/10.1109/TVCG.2003.1207441.

Per H. Christensen and Wojciech Jarosz. The path to path-traced movies. *Foundations and Trends® in Computer Graphics and Vision*, 10(2):103–175, October 2016. ISSN 1572-2740. doi: 10.1561/0600000073.

David Cline, Justin Talbot, and Parris Egbert. Energy redistribution path tracing. *ACM Trans. Graph.*, 24(3), 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073330.

David Cline, Daniel Adams, and Parris Egbert. Table-driven adaptive importance sampling. *Computer Graphics Forum*, 27, 2008. doi: 10.1111/j.1467-8659.2008.01249.x. URL http://dx.doi.org/10.1111/j.1467-8659.2008.01249.x.

Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *SIGGRAPH Comput. Graph.*, 18(3), January 1984. ISSN 0097-8930. doi: 10.1145/964965.808590. URL http://doi.acm.org/10.1145/964965.808590.

Robert L. Cook, Loren Carpenter, and Edwin Catmull. The Reyes image rendering architecture. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, 1987.

Ken Dahm and Alexander Keller. Learning light transport the reinforced way. *CoRR*, abs/1701.07403, 2017.

Aldous David and Vazirani Umesh. "Go with the winners" algorithms. In *Proc. 35th IEEE Symp. on Found. of Comp. Sci.*, 1994.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

John Duchi. Derivations for linear algebra and optimization, 2014. http://www.cs.berkeley.edu/∼jduchi/projects/general_notes.pdf (retrieved in year 2014).

Philip Dutré and Yves Willems. Importance-driven Monte Carlo light tracing. In *Eurographics Workshop on Rendering*, 1994.

Philip Dutré and Yves Willems. Potential-driven Monte Carlo particle tracing for diffuse environments with adaptive probability density functions. In *EG Workshop on Rendering*, 1995.

Philip Dutré, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination.* A. K. Peters, 2nd edition, 2006.

Ronald Fisher. Dispersion on a sphere. *Proc. Royal Society of London. Series A. Math. Phys.*, 217(1130):295–305, 1953.

Václav Gassenbauer, Jaroslav Křivánek, and Kadi Bouatouch. Spatial directional radiance caching. *Computer Graphics Forum (EGSR 2009)*, 28(4):1189–1198, 2009. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2009.01496.x. URL `http://dx.doi.org/10.1111/j.1467-8659.2009.01496.x`.

J. Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans Audio Speech Lang Processing*, 2(2):291–298, 1994. ISSN 1063-6676. doi: 10.1109/89.279278.

Iliyan Georgiev. Implementing vertex connection and merging. Technical report, Saarland University, 2012. URL `http://www.iliyan.com/publications/ImplementingVCM`.

Iliyan Georgiev and Philipp Slusallek. Simple and Robust Iterative Importance Sampling of Virtual Point Lights. *Proceedings of Eurographics (short papers)*, 2010.

Iliyan Georgiev, Jaroslav Křivánek, Stefan Popov, and Philipp Slusallek. Importance caching for complex illumination. *Computer Graphics Forum*, 31:701–10, 2012a. Proc. of Eurographics.

Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.*, 31(6), 2012b. ISSN 0730-0301.

Peter Grassberger. Go with the winners: a general Monte Carlo strategy. In *Comp. Phys. Commun.*, volume 147, pages 64–70, 2002.

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Trans. Graph.*, 27(5), December 2008. ISSN 0730-0301.

Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A path space extension for robust light transport simulation. *ACM Trans. Graph.*, 31(6), 2012.

Alireza Haghighat and John C. Wagner. Monte Carlo variance reduction with deterministic importance functions. *Progress in Nuclear Energy*, 42 (1):25 – 53, 2003. ISSN 0149-1970. doi: http://dx.doi.org/10.1016/S0149-1970(02)00002-1. URL `http://www.sciencedirect.com/science/article/pii/S0149197002000021`.

J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods.* Chapman and Hall, New York, 1964.

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. Product importance sampling for light transport path guiding. *Computer Graphics Forum*, 35(4):67–77, 2016.

H. Hey and W. Purgathofer. Importance sampling with hemispherical particle footprints. In *SCCG*, 2002.

J. Hoogenboom, Eduard. Zero-variance Monte Carlo schemes revisited. *Nucl. Sci. Eng.*, 160:1–22, 2008.

J. Hoogenboom, Eduard and Dávid Légrády. A critical review of the weight window generator in MCNP. *Monte Carlo 2005 Topical Meeting*, April 2005.

Wenzel Jakob. Mitsuba renderer, 2010. http://mitsuba-renderer.org.

Wenzel Jakob and Steve Marschner. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.*, 31(4), July 2012. ISSN 0730-0301.

Wenzel Jakob, Christian Regg, and Wojciech Jarosz. Progressive expectation-maximization for hierarchical volumetric photon mapping. *Computer Graphics Forum*, 30(4):1287–1297, 2011. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2011.01988.x. URL `http://dx.doi.org/10.1111/j.1467-8659.2011.01988.x`.

Henrik Wann Jensen. Importance driven path tracing using the photon map, 1995.

Henrik Wann Jensen. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, pages 21–30, London, UK, UK, 1996. Springer-Verlag. ISBN 3-211-82883-4. URL `http://dl.acm.org/citation.cfm?id=275458.275461`.

Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping.* A. K. Peters, Ltd., Natick, MA, USA, 2001. ISBN 1-56881-147-0.

Herman Kahn. Use of different Monte Carlo sampling techniques. In *Symp. on Monte Carlo Methods.* New York: Wiley, 1956.

Herman Kahn and E. Harris, T. Estimation of particle transmission by random sampling. In *Nat. Bur. of Stand. Appl. Math. Ser.*, volume 12, pages 27–30, 1951.

James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20 (4):143–150, August 1986. ISSN 0097-8930. doi: 10.1145/15886.15902. URL `http://doi.acm.org/10.1145/15886.15902`.

M. H. Kalos. Importance sampling in Monte Carlo shielding calculations: I. neutron penetration through thick hydrogen slabs. In *Nuclear Science and Engineering*, volume 16, pages 227–234, 1963.

Ondřej Karlík. Corona Renderer, 2009. http://corona-renderer.com.

C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Comp. Graph. Forum (Proc. of Eurographics)*, 21(3):531–540, 2002.

A. Keller, L. Fascione, M. Fajardo, I. Georgiev, P. Christensen, J. Hanika, C. Eisenacher, and G. Nichols. The path tracing revolution in the movie industry. In *ACM SIGGRAPH 2015 Courses*, SIGGRAPH '15, pages 24:1–24:7, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3634-5. doi: 10.1145/2776880.2792699. URL http://doi.acm.org/10.1145/2776880.2792699.

Alexander Keller and Ingo Wald. Efficient importance sampling techniques for the photon map. In *Proc. Fifth Fall Workshop Vision, Modeling, and Visualisation*, pages 271–279, 2000.

John T. Kent. The Fisher-Bingham distribution on the sphere. *Journal of the Royal Statistical Society. Series B*, 44(1):71–80, 1982. ISSN 00359246. URL http://www.jstor.org/stable/2984712.

Jaroslav Křivánek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. Radiance caching for efficient global illumination computation. *IEEE Trans. Vis. Comp. Graph.*, 11(5), 2005.

Jaroslav Křivánek, Kadi Bouatouch, Sumanta N. Pattanaik, and Jiří Žára. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Eurographics Symposium on Rendering*, pages 127–138, June 2006. ISBN 3-905673-35-5.

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 Trans. Graph.*, 33(4): 1–13, August 2014. ISSN 0730-0301.

Jaroslav Křivánek and Eugene d'Eon. A zero-variance-based sampling scheme for Monte Carlo subsurface scattering. In *ACM SIGGRAPH 2014 Talks*, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2960-6. doi: 10.1145/2614106.2614138. URL http://doi.acm.org/10.1145/2614106.2614138.

Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In *Proc. of Compugraphics '93*, 1993.

Eric P Lafortune and Yves D Willems. A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Eurographics Workshop on Rendering*, pages 11–20. 1995.

Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. Gradient-domain Metropolis light transport. *ACM Trans. Graph.*, 32(4), 2013.

G. P. Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Comp. Physics*, 27:192–203, 1978.

Percy Liang and Dan Klein. Online EM for unsupervised models. In *Human Language Technologies (NAACL '09)*, pages 611–619. Association for Computational Linguistics, 2009. ISBN 978-1-932432-41-1. URL http://dl.acm.org/citation.cfm?id=1620754.1620843.

I. Lux and László Koblinger. *Monte Carlo particle transport methods: neutron and photon calculations.* CRC Press, 1991.

Frank Meinl. Crytek sponza, 2010. http://www.crytek.com/cryengine/cryengine3/downloads.

Vincent Pegoraro, Carson Brownlee, Peter S. Shirley, and Steven G. Parker. Towards interactive global illumination effects via sequential Monte Carlo adaptation. In *IEEE Symposium on Interactive Ray Tracing*, pages 107–114, 2008a.

Vincent Pegoraro, Ingo Wald, and Steven G. Parker. Sequential Monte Carlo adaptation in low-anisotropy participating media. *Computer Graphics Forum*, 27(4):1097–1104, 2008b.

Ingmar Peter and Georg Pietrek. Importance driven construction of photon maps. In *Rendering Techniques*, pages 269–80, 1998.

Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation.* Morgan Kaufmann Publishers Inc., San Francisco, CA, 2nd edition, 2010. ISBN 0123750792, 9780123750792.

Stefan Popov, Ravi Ramamoorthi, Frédo Durand, and George Drettakis. Probabilistic connections for bidirectional path tracing. *Computer Graphics Forum (Proc. of EGSR)*, 34(4), 2015.

Masa-Aki Sato and Shin Ishii. On-line EM algorithm for the normalized Gaussian network. *Neural Comput.*, 12(2), February 2000. ISSN 0899-7667. doi: 10.1162/089976600300015853. URL `http://dx.doi.org/10.1162/089976600300015853`.

Mike Seymour. Manuka: Weta digital's new renderer, 2014a. http://www.fxguide.com/featured/manuka-weta-digitals-new-renderer/.

Mike Seymour. Renderman/RIS and the start of next 25 years, 2014b. http://www.fxguide.com/featured/rendermanris-and-the-start-of-next-25-years/.

Peter Shirley and Kenneth Chiu. A low distortion map between disk and square. *J. Graph. Tools*, 2(3):45–52, December 1997. ISSN 1086-7651. doi: 10.1080/10867651.1997.10487479. URL `http://dx.doi.org/10.1080/10867651.1997.10487479`.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall, London, 1986.

J Spanier and M. Gelbard, Ely. *Monte Carlo principles and neutron transport problems.* Addison-Wesley, 1969.

J. Steinhurst and A. Lastra. Global importance sampling of glossy surfaces using the photon map. In *IEEE Symposium on Interactive Ray Tracing*, pages 133–138, 2006.

Frank Suykens and Yves D. Willems. Density control for photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, London, UK, 2000. Springer-Verlag. ISBN 3-211-83535-0. URL `http://dl.acm.org/citation.cfm?id=647652.732120`.

László Szécsi, László Szirmay-Kalos, and Csaba Kelemen. Variance reduction for Russian roulette. *Journal of WSCG*, 2003.

László Szirmay-Kalos and György Antal. Go with the winners strategy in path tracing. In *Journal of WSCG.*, volume 13, 2005. URL `http://hdl.handle.net/11025/1454`.

Yu-Ting Tsai, Chin-Chen Chang, Qing-Zhen Jiang, and Shr-Ching Weng. Importance sampling of products from illumination and BRDF using spherical radial basis functions. *The Visual Computer*, 24(7), 2008.

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

Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Proc. Eurographics Rendering Workshop*, 1994.

Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *SIGGRAPH '97*, 1997.

Jakob J Verbeek, Jan RJ Nunnink, and Nikos Vlassis. Accelerated EM-based clustering of large data sets. *Data Mining and Knowledge Discovery*, 13(3): 291–307, 2006.

Jiří Vorba and Jaroslav Křivánek. Bidirectional photon mapping. In *Central European Seminar on Computer Graphics (CESCG)*, 2011.

Jiří Vorba and Jaroslav Křivánek. Adjoint-driven russian roulette and splitting in light transport simulation. *ACM Transactions on Graphics (SIGGRAPH 2016)*, 35(4), July 2016.

Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (SIGGRAPH 2014)*, 33(4), July 2014.

C. Wagner, J. *Acceleration of Monte Carlo shielding calculations with an automated variance reduction technique and parallel processing.* PhD thesis, The Pennsylvania State Univ., 1997.

C. Wagner, J. and A. Haghighat. Automated variance reduction of Monte Carlo shielding calculations using the discrete ordinates adjoint function. In *Nulc. Sci. Eng.*, volume 128, 1998.

Gregory J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92*, pages 265–272. ACM, 1992. ISBN 0-89791-479-1. doi: 10.1145/133994.134078. URL `http://doi.acm.org/10.1145/133994.134078`.

Gregory J Ward, Francis M Rubinstein, and Robert D Clear. A ray tracing solution for diffuse interreflection. In *SIGGRAPH '88*, number 4, pages 85–92. ACM, 1988.

X-5 Monte Carlo team. MCNP – A general Monte Carlo N-particle transport code, version 5. Technical Report LA-UR-03-1987, Los Alamos National Laboratory, April 2003.

Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical Gaussians. *ACM Transactions on Graphics*, 32(6): 209:1–209:11, 2013.

Q. Xu, J. Sun, Z. Wei, Y. Shu, S. Messelodi, and J. Cai. Zero variance importance sampling driven potential tracing algorithm for global illumination. In *Journal of WSCG. 2001*, volume 9, 2001. ISBN 80-7082-712-4.

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **ADRRS** | adjoint-driven Russian roulette and splitting |
| **BDPT** | bidirectional path tracing |
| **BSDF** | bidirectional scattering distribution function |
| **CPU** | central processing unit |
| **EM** | expectation maximization |
| **GMM** | Gaussian mixture model |
| **KL** | Kullback-Leiber |
| **LT** | light tracing |
| **MAP** | maximum a posteriory |
| **MC** | Monte Carlo |
| **MIS** | multiple importance sampling |
| **ML** | maximum likelihood |
| **MLT** | Metropolis light transport |
| **PDF** | probability distribution function |
| **PPM** | progressive photon mapping |
| **PT** | path tracing |
| **RR** | Russian roulette |
| **SDS** | specular-diffuse-specular |
| **TP** | training pass |
| **UPS** | unified path space |
| **VCM** | vertex connection and merging |
| **vMF** | von Mises-Fisher |
| **ZV** | zero variance |

# Attachments

## Project Pages

- On-Line Learning of Parametric Mixture Models for Light Transport Simulation: `http://cgg.mff.cuni.cz/~jirka/papers/2014/olpm/index.htm`
- Adjoint-Driven Russian Roulette and Splitting in Light Transport Simulation: `http://cgg.mff.cuni.cz/~jirka/papers/2016/adrrs/index.htm`

## Supplemental Images and Plots

- On-Line Learning of Parametric Mixture Models for Light Transport Simulation: `http://cgg.mff.cuni.cz/~jirka/papers/2014/olpm/On-line LearningPPMinLTS-supp_img.pdf`
- Adjoint-Driven Russian Roulette and Splitting in Light Transport Simulation: `http://cgg.mff.cuni.cz/~jaroslav/papers/2016-adrrs/supplemental_material/index.html`