
Path Sampling Techniques for Efficient Light Transport Simulation

Iliyan Georgiev

Thesis for obtaining the title of Doctor of Engineering Science
of the Faculty of Natural Science and Technology I
of Saarland University

Saarbücken, Germany, 2015



UNIVERSITÄT
DES
SAARLANDES

Dean of faculty

Univ.-Prof. Dr. Markus Bläser
Saarland University, Saarbrücken, Germany

Committee chair

Prof. Dr. Sebastian Hack
Saarland University, Saarbrücken, Germany

Reviewers

Prof. Dr. Philipp Slusallek
Saarland University, Intel VCI, and DFKI, Saarbrücken, Germany

Prof. Dr. Hans-Peter Seidel
MPI Informatik, Saarbrücken, Germany

Dr. Wojciech Jarosz
Disney Research, Zürich, Switzerland

Academic assistant

Dr. Andreas Nonnengart
DFKI, Saarbrücken, Germany

Date of defense

June 10th, 2015

Revision 1, June 14th, 2015

© 2015 Iliyan Georgiev. All rights reserved.

“Imagination is more important than knowledge. For knowledge is limited to all we now know and understand, while imagination embraces the entire world, and all there ever will be to know and understand.”

– Albert Einstein

Abstract

Reproducing the interactions between light and matter in a physically accurate way can significantly improve the realistic appearance of synthetic images, however such effects can be very computationally expensive to simulate. Pressed by strict requirements on image quality and visual realism, industrial applications have recently moved away from using legacy rasterization-based rendering solutions to fully embrace physically-based Monte Carlo methods. This dramatic shift has rekindled the interest in developing new and robust light transport simulation algorithms that can efficiently handle a wide range of scenes with complex materials and lighting – a problem that we address in this thesis.

State-of-the-art Monte Carlo methods solve the global illumination problem by sampling random light transport paths in the scene via ray tracing. We analyze the efficiency of these methods, devise new path sampling techniques for rendering surface and volumetric light scattering, and develop novel means of leveraging illumination coherence via path reuse. This results in several practical rendering algorithms that produce images with less noise and remain more resilient to variations in the scene configuration than existing methods. The improved efficiency of these algorithms comes from the use of new and diverse sampling techniques, each specialized for handling a different set of lighting effects. Their robustness is due to the adaptive combination of these techniques in a way that preserves their individual strengths.

Kurzfassung

Die physikalisch korrekte Simulation der Interaktion von Licht mit Materie kann den realistischen Eindruck von synthetisch generierten Bildern zwar deutlich verbessern, erfordert allerdings einen sehr hohen Berechnungsaufwand. Trotzdem hat die Industrie aufgrund der hohen Anforderungen an Bildqualität und Realismus die bisherigen approximativen Verfahren aufgegeben und ist vollständig auf die neuen physikalisch-basierte Monte-Carlo-Verfahren umgestiegen. Dieser dramatische Umbruch hat aber zu einem großen Interesse an neuen und robusten Algorithmen geführt, die auch Szenen mit komplexen Materialien und Beleuchtungssituationen effizient berechnen können. Genau solche Algorithmen sind das Thema dieser Dissertation.

Moderne Monte-Carlo-Verfahren simulieren die Beleuchtung in einer Szene, indem sie mögliche Transportpfade von Licht statistisch auf Basis von Ray-Tracing ermitteln. Wir analysieren die statistische Effizienz dieser Methoden, schlagen neue Sampling-Techniken für Oberflächen und Volumen vor und entwickeln neue Methoden, um die Kohärenz in der Beleuchtung besser ausnutzen zu können. Daraus entstehen verschiedene praktische Algorithmen zur Bildsynthese, die weniger Rauschen zeigen und weniger anfällig für Änderungen in der Szene sind als bisherige Ansätze. Die verbesserte Effizienz der neuen Algorithmen wird durch eine geschickte Kombination neuer Sampling-Techniken erzielt, die jeweils auf bestimmte Beleuchtungssituationen spezialisiert sind. Eine adaptive Kombination dieser Techniken, die deren jeweiligen Stärken erhält, führt dann zu der notwendigen Robustheit der Ansätze.

Acknowledgements

This thesis could not have been possible without the help, inspiration, and support of many people over the course of my life and academic career.

First of all, I would like to express my sincerest gratitude to my supervisor Prof. Dr. Philipp Slusallek for introducing me to the fascinating field of computer graphics and for his patient guidance and mentorship through to completion of this thesis. I am immensely grateful to Saarland University, Max-Planck Institute for Informatics, and Intel Visual Computing Institute for providing me with financial support to work and travel.

Very special thanks are due to my main collaborator and good friend Jaroslav Křivánek, whose contagious enthusiasm and rigorous approach to research made the many projects we embarked on together inevitably successful. I am also truly fortunate to have had the opportunity to work with Wojciech Jarosz, Toshiya Hachisuka, and Derek Nowrouzezahrai, who generously shared their insights and vast knowledge as we transformed ideas into completed projects.

I would like to extend my gratitude to my friends and colleagues at Saarland University: Stefan Popov, Javor Kalojanov, Tomáš Davidovič, Lukáš Maršálek, Dmitri Rubinstein, Vincent Pegoraro, Andreas and Anna Katharina Hildebrandt, Stefan Nickels, and Johannes Günther for the ideas, discussions, and the pleasure to work with them on various research projects. I would also like to thank Disney Research Zürich and Weta Digital for giving me the opportunity to spend two great summers working in their highly reputable teams and making new friends.

I am grateful to the anonymous reviewers for their shrewd comments and suggestions on improving my papers, and to all the fantastic people in the computer graphics community for making every conference an enjoyable, insightful, and fun experience.

I would like to thank my family for their continued support throughout the years and for encouraging me to pursue my dreams. Finally, I would be remiss not to acknowledge my girlfriend Zlatina for her love and comfort, for the many weekends sacrificed for my work, and for the playful comments she secretly put in my source code while I was away from the keyboard.

Publications

Parts of this thesis are based on published works that I have co-authored with others and use passages of text from these works with the explicit permission of the co-authors. I was the primary investigator and author of all papers listed below.

- Chapter 5 is based on the work that appears in the paper:
Iliyan Georgiev and Philipp Slusallek. “Simple and robust iterative importance sampling of virtual point lights”. In *Proceedings of Eurographics (short papers)*, 2010.

- Chapter 6 is a reproduction of the material published in the article:
Iliyan Georgiev, Jaroslav Křivánek, Stefan Popov, and Philipp Slusallek. “Importance caching for complex illumination”. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2), 2012.
- Chapter 7 is based on the work published in the following papers:
Iliyan Georgiev, Jaroslav Křivánek, and Philipp Slusallek. “Bidirectional light transport with vertex merging”. In *SIGGRAPH Asia sketches*, 2011.
Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. “Light transport simulation with vertex connection and merging”. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 31(6), 2012.
- Chapter 8 is based on the technical report:
Iliyan Georgiev. “Implementing vertex connection and merging”. Technical report, Saarland University, 2012.
- Chapter 9 is based on the technical paper:
Iliyan Georgiev, Jaroslav Křivánek, Toshiya Hachisuka, Derek Nowrouzezahrai, and Wojciech Jarosz. “Joint importance sampling of low-order volumetric scattering”. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 32(6), 2013.

Contents

Abstract	I
Acknowledgements	V
List of figures	XI
1 Introduction	1
1.1 Summary of original contributions	3
1.2 Thesis outline	4
2 Monte Carlo integration	5
2.1 Random variables	5
2.1.1 Discrete random variables	5
2.1.2 Continuous random variables	6
2.1.3 Expected value	7
2.1.4 Variance and standard deviation	7
2.2 Integral estimators	8
2.3 Estimator error and convergence rate	9
2.3.1 Error	9
2.3.2 Bias	9
2.3.3 Mean squared error	9
2.3.4 Efficiency	10
2.3.5 Consistency and convergence rates	10
2.4 Sampling random variables	11
2.4.1 Transforming between distributions	11
2.4.2 CDF inversion method	12
2.4.3 Multivariate distributions	12
2.4.4 Global vs. local sampling	13
2.4.5 Other methods	13
2.5 Variance reduction techniques	14
2.5.1 Stratified sampling	15
2.5.2 Adaptive sampling	15
2.5.3 Russian roulette and splitting	15
2.5.4 Importance sampling	16
2.5.5 Mixture importance sampling	17
2.5.6 Multiple importance sampling	17
3 Mathematical models of light transport	21
3.1 Light models	21
3.2 Basic radiometry	22
3.2.1 Radiant power or flux	22

3.2.2	Irradiance and radiosity	22
3.2.3	Radiance	23
3.2.4	Importance	23
3.3	Wavelength dependency and color	23
3.3.1	Luminance	24
3.4	Emission, propagation, and scattering	24
3.4.1	Emission	25
3.4.2	Vacuum propagation	25
3.4.3	Medium propagation	26
3.4.4	Visibility	26
3.4.5	Surface scattering	27
3.4.6	Medium scattering	28
3.5	Integral formulation of light transport	29
3.5.1	The measurement equation	29
3.5.2	The surface rendering equation	29
3.5.3	The volume rendering equation	30
3.5.4	Path integral formulation	31
3.5.5	Path classification	32
4	Monte Carlo solutions for the path integral	35
4.1	Path integral estimators	35
4.2	Path sampling techniques	36
4.2.1	Random walks	36
4.2.2	Unidirectional sampling	40
4.2.3	Vertex connection	40
4.3	Many-light rendering	43
4.3.1	The many-light pixel estimator	44
4.3.2	Discussion	44
4.4	Bidirectional path tracing	45
4.5	Specialized techniques for participating media	46
4.5.1	Equi-angular sampling	47
4.5.2	Virtual ray lights	47
4.6	Photon density estimation	49
4.6.1	Photon mapping radiance estimator	49
4.6.2	Progressive photon mapping	50
4.7	Other methods	52
4.7.1	Incident illumination importance sampling	52
4.7.2	Exploiting coherence	52
4.7.3	Density estimation in participating media	53
4.7.4	Markov chain Monte Carlo	54
5	Importance-driven distribution of virtual point lights	55
5.1	Probabilistic VPL acceptance	56
5.1.1	Formal derivation	56
5.1.2	VPL acceptance probability	57
5.2	Results	58
5.3	Discussion	58
6	Importance caching for many-light rendering	61

6.1	Algorithm overview	62
6.1.1	Importance caching	63
6.2	VPL sampling distributions	65
6.2.1	Full contribution	65
6.2.2	Unoccluded contribution	66
6.2.3	Bounded contribution	66
6.2.4	Conservative distribution	67
6.3	Bilateral combination of sampling distributions	67
6.3.1	Column combination	68
6.3.2	Row combination	69
6.3.3	Distribution optimization	69
6.4	Results	70
6.4.1	Technique comparison	70
6.4.2	Numerical convergence	72
6.4.3	Glossy materials	73
6.4.4	High-quality preview rendering	74
6.5	Discussion	75
7	Vertex connection and merging	77
7.1	Photon mapping as a path integral estimator	79
7.1.1	Extended path space formulation	79
7.1.2	Regular path space formulation	81
7.2	Efficiency of different path sampling techniques	84
7.3	A combined light transport algorithm	85
7.3.1	Mathematical formulation	86
7.3.2	Algorithm	87
7.4	Achieving consistency	88
7.4.1	Progressive radius reduction	88
7.4.2	Asymptotic error analysis	89
7.4.3	Discussion	91
7.5	Results	92
7.5.1	Setup	92
7.5.2	Visual comparison	93
7.5.3	Image quality metrics	94
7.5.4	Numerical convergence	94
7.6	Discussion	95
8	Implementing vertex connection and merging	101
8.1	Notation	101
8.2	Recursive path weight evaluation	104
8.2.1	Partial subpath weights	105
8.2.2	Recursive formulation	106
8.3	Practical implementation	107
8.3.1	Subpath vertex data	108
8.3.2	Full path weight	108
8.3.3	Reverse pdf evaluation	109
8.4	Special cases	110
8.4.1	Infinite light sources	110

8.4.2	Orthographic cameras	111
8.4.3	Point and directional light sources	111
8.4.4	Specular materials	111
8.4.5	Bidirectional path tracing	112
8.4.6	Bidirectional photon mapping	113
8.5	Extensions	113
8.5.1	Per-path merging radii	113
8.5.2	Memory-efficient implementation	114
8.5.3	Motion blur and spectral rendering	115
8.6	Discussion	116
9	Joint path sampling in participating media	119
9.1	Path sampling in media	120
9.1.1	Existing techniques	121
9.1.2	Relationship to neutron transport	123
9.2	Joint path vertex sampling	124
9.2.1	Factorizations of the joint pdf	124
9.2.2	Sampling from the joint distributions	126
9.3	Analytic sampling	126
9.3.1	Derivation of U_1	127
9.3.2	Derivation of U_2	128
9.3.3	Derivation of U_3	128
9.3.4	Joint unidirectional pdf	129
9.3.5	Unidirectional sampling techniques	129
9.3.6	Discussion	130
9.4	Tabulated sampling	130
9.4.1	General approach	130
9.4.2	Unidirectional factorization	131
9.4.3	Bidirectional factorization	134
9.5	Applications and results	137
9.5.1	Unidirectional path tracing	138
9.5.2	Virtual ray lights	139
9.5.3	Bidirectional path tracing	139
9.6	Discussion	140
10	Conclusion	147
	Bibliography	153

List of Figures

3.1	The geometry of a radiance measurement	23
3.2	Light emission, propagation, and scattering	25
3.3	Classifications of surface and medium scattering distributions	28
3.4	Path measurement contribution	31
4.1	Random walk	38
4.2	Propagation distance sampling	38
4.3	Scattering direction sampling	39
4.4	Random walk based path sampling techniques	41
4.5	Vertex connection	41
4.6	Comparison of path sampling based rendering algorithms	42
4.7	Geometry term singularity in vertex connection	44
4.8	Sampling techniques in bidirectional path tracing	45
4.9	Specular-diffuse-specular path sampling	46
4.10	Equi-angular sampling and virtual ray lights	47
4.11	Equi-angular sampling and virtual ray lights rendering comparison	48
4.12	Photon density estimation and progressive photon mapping	50
5.1	VPL distribution in a highly occluded scene	57
5.2	Importance-driven VPL distribution rendering results	59
6.1	Probabilistic VPL selection	63
6.2	Importance caching algorithm overview	64
6.3	VPL sampling distributions	65
6.4	VPL contribution geometry term bounds	66
6.5	Combining VPL sampling distributions	68
6.6	STUDY HALL scene rendering comparison	71
6.7	STUDY HALL scene technique contributions	71
6.8	STUDY HALL scene error plots	72
6.9	BUDDHAS scene rendering comparison	73
6.10	Interactive previews with importance caching	74
7.1	Path classification based combination of BPT and PM	78
7.2	Three different measurement contribution functions	80
7.3	Interpretation of photon mapping as a sampling technique	82
7.4	SDS path sampling efficiency	85
7.5	VCM path sampling techniques	86
7.6	VCM algorithm pseudocode	87
7.7	Vertex merging radius reduction scheme analysis	89
7.8	Vertex merging contributions to the progressive VCM estimate	91
7.9	VCM error convergence plots	94

7.10	VCM limitations	95
7.11	LIVING ROOM scene comparison	97
7.12	BATHROOM scene comparison	98
7.13	CAR scene comparison	99
7.14	MIRROR BALLS scene comparison	100
8.1	Path and pdf notation in VCM implementation	103
8.2	Reverse pdf evaluation	109
8.3	Handling infinite lights	110
9.1	Illustration of existing medium path sampling techniques	122
9.2	Joint path vertex sampling problem statement	125
9.3	Unidirectional factorization pdfs	127
9.4	Unidirectional pdf tabulation	132
9.5	Bidirectional pdf tabulation	135
9.6	Comparison of single- and double-scattering sampling techniques	137
9.7	DRAGON scene path tracing comparison	142
9.8	LIGHTHOUSE scene path tracing comparison	143
9.9	Heterogeneous medium path tracing comparison	143
9.10	DRAGON scene higher-order scattering comparison	144
9.11	PARK scene correlated and uncorrelated VRL comparison	144
9.12	STAGE scene bidirectional path tracing comparison	145

Introduction



Sight is arguably the most important sense that allows humans to interact with the outside world. It has been estimated that over eighty percent of our perception, cognition, and activities are mediated through vision. Unsurprisingly, the human effort to understand the nature of light and to capture and recreate images of the surrounding environment has a long history.

Over two thousand years ago, the Greeks believed that it is the eyes that emanate light rays toward the objects being perceived. Even under this false assumption, Euclid correctly postulated that light traveled along straight lines and described the laws of reflection. Many other properties of light have been discovered since then, and modern physics has confirmed that light is, in fact, transmitted by photons from light sources to the visible objects. Scattered by these objects, the photons reach our eyes where minuscule photoreceptors record their energy before our brain finally creates an image.

Until two hundred years ago, the only way to obtain a persistent picture of a physical scene was through hand drawing. Medieval paintings depicted objects and creatures using simplistic forms whose disproportionate sizes and shapes signified their importance or relative positions. The first major steps toward realistic rendering were taken by Renaissance artists like da Vinci and Dürer who studied proportions and perspective. It was soon thereafter recognized that attention to shading aspects like shadows and light reflections is required to produce plausible renderings of real objects. By the early nineteenth century, realistic painting was aided by optical devices like the camera lucida (Latin for “light room”) and camera obscura (“dark room”, a.k.a. pinhole camera), which created a superimposition of the scene onto a drawing surface which could then be manually traced by the artist. The photographic camera subsequently replaced the drawing surface in the camera obscura with a light-sensitive plate, making the process of capturing persistent images of physical scenes fully automatic. Modern cameras sport arrays of electronic light sensors that digitize the captured images. Since the invention of photography, drawing and painting evolved into modern, not necessarily photorealistic art, but remained the only way to create images of virtual (i.e. non-existent) environments until only a few decades ago.

The advent of programmable computers in the second half of the twentieth century gave birth to the field computer graphics. Early rendering algorithms produced line drawings of three-dimensional polygonal models. This approach was superseded by raster graphics which enabled shading objects initially by attributing a single color per polygon and later by computing simple local lighting for each image pixel. The 1980s saw two major breakthroughs toward photorealistic rendering. The first was the introduction of the ray tracing method by Whitted [158], who demonstrated accurate reflections, refractions, and direct illumination with shadows from point (i.e. infinitesimally small) light sources. Whitted’s algorithm was simple and elegant but could not reproduce many commonly seen effects in photographs, such as motion blur, soft shadows, or any kind of indirect (i.e. bounced) illumination. The second breakthrough came a few years later with the development of stochastic ray tracing [17] and the subsequent rigorous mathematical formalization of light transport as an integration problem [66].

These developments enabled the accurate computation of the full global illumination in a scene, which in turn made rendering all the aforementioned effects possible, albeit with varying efficiency.

Propelled by rapid technological advancements, computer graphics became mainstream in the 1990s. Computer games immersed players in complex interactive 3D worlds, and the first full-length computer animation films came out. Due to the significant computational cost of ray tracing at the time, those applications had to rely on more hardware-friendly rasterization-based rendering methods. These methods could reproduce only a limited set of effects efficiently, and more complex light interactions were being approximated using ad-hoc models. Physically-based rendering remained mostly confined to academic research. Over the years, the pervasive use of computer-generated imagery nourished a growing demand for visual realism that became increasingly difficult to satisfy with rasterization-based methods. Meanwhile, research had made significant progress in light transport theory and new methods for rendering a wide range of surface and volumetric lighting effects had been developed. About a decade ago, stochastic ray tracing finally became a viable alternative and in the past few years has almost completely replaced legacy rendering technology in all but real-time graphics applications. Physically-based rendering is now widely adopted in visual arts, industrial design, the media and entertainment industries, which have all mastered the production of synthetic images that are virtually indistinguishable from real photographs, although often at a substantial computational cost.

Rendering a photorealistic image of a virtual scene on a computer requires a simulation of the global light transport in the scene, all the way from the light sources, through scattering events at objects' surfaces and in participating media, to the eye of the virtual observer. The physical laws that must govern this simulation have been studied for centuries, and the rendering problem is by now theoretically well understood. The value of every image pixel can be mathematically expressed as the sum of the differential energy contributions of all possible photon trajectories in the scene that start on a light source and end in the observer's eye. The challenge of computing these pixel integrals efficiently lies in the fact that in most typical scenes only a very small fraction of all emitted photons eventually make their way to the eye. The goal of a global illumination algorithm is thus to find that small set of relevant light transport paths that make actual energy contributions to the image. State-of-the-art methods are based on Monte Carlo integration and compute an estimate for every pixel by sampling a number of random paths in the scene via ray tracing. This approach has the advantage of being conceptually simple and able to reproduce all possible lighting effects. However, random sampling leads to noise in the rendered image. Reducing the amount of noise to an acceptable level can typically be achieved by simply increasing the number of sampled paths, but this comes at the cost of a corresponding linear increase in computation time. Developing methods that can more efficiently find relevant paths is an important problem that has been an active area of research in the past three decades.

The various Monte Carlo rendering methods available today differ primarily in the procedures they use to construct light transport paths, called *path sampling techniques*. The performance of such a method in different scene configurations is largely determined by the efficiency of the sampling techniques it employs. Intuitively, the efficiency measures the number of relevant paths a technique can find per unit of time. Two general approaches exist for improving this efficiency. One aims to accelerate path generation, typically by reusing (parts of) a sampled path to cheaply construct other similar paths, thereby amortizing the sampling computational effort. Path reuse makes many-light [18] and photon density estimation [63] methods efficient in capturing diffuse indirect and caustic illumination respectively. Alternatively, knowledge about the scene can be used to guide the sampling toward important regions and increase the chance of finding relevant paths, thereby concentrating the computational effort where it is likely to pay off most. By virtue of importance sampling, the path tracing algorithm excels at reproducing sharp reflections and direct illumination [127]. Unfortunately, it is very difficult – and likely impossible – to design a single technique that can efficiently sample all possible light

transport effects, and therefore none of the aforementioned methods is robust to all kinds of scene configurations. As a result, some effects often remain noisy in the rendered image or require expensive additional sampling to be accurately reproduced using the methods' limited arsenals of sampling techniques.

The problem of handling scenes with complex lighting can be attacked by employing a larger set of path sampling techniques, each tailored to a specific effect. The key to robustness with this approach is to combine the contributions of the different techniques in a way that preserves the qualities of each. The bidirectional path tracing algorithm achieves this by adaptively weighting these contributions using multiple importance sampling [142]. This intelligent combination makes the algorithm one of the most versatile rendering methods available. However, bidirectional path tracing is also notoriously inefficient on scenes where light interacts multiple times between specular and non-specular surfaces, because none of its many sampling techniques can find these paths with a sufficiently high probability. The problem of finding relevant light transport paths becomes even more difficult in scenes with participating media, where the space of all possible paths is comparatively much larger. Existing volumetric path sampling techniques, which are adaptations from surface-based rendering, can be particularly inefficient in anisotropically scattering media. Handling a wide range of input scene configurations remains an open problem, and devising efficient path sampling techniques is especially challenging for multi-bounce focused illumination effects which cover a small fraction of the entire path space.

The goal of the work presented in this thesis is to develop efficient path sampling techniques for light transport simulation in scenes containing surfaces and participating media. To this end, we start by casting a number of existing methods into the path integral framework and analyze the efficiency of their corresponding path sampling techniques. To address their weaknesses, we devise new techniques for rendering surface and volumetric light scattering and develop novel schemes for combining importance sampling and path reuse. We augment several rendering algorithms with our techniques to improve their efficiency on scenes with complex lighting and produce images with less noise. We also improve the robustness of these algorithms by adaptively combining these diverse techniques, each specialized in handling a different set of light transport effects, via multiple importance sampling.

1.1 Summary of original contributions

The work presented in this thesis builds upon a number of prior works. Below we highlight our major contributions and results.

- **Importance-driven distribution of virtual point lights.** Many-light methods decompose path sampling into a virtual point light (VPL) distribution stage and a subsequent rendering stage that computes the contributions of all VPLs to the surfaces seen from the eye. Our first contribution is an importance-driven VPL sampling algorithm which produces a VPL distribution relevant for the chosen viewpoint, such that every VPL brings roughly the same amount of energy to the image. We achieve this by probabilistically accepting or rejecting VPL candidates based on an on-the-fly estimation of their image contribution. Our method can efficiently find good VPL sets in scenes with difficult visibility configurations, sometimes resulting in rendering speed-ups of over an order of magnitude compared to traditional VPL sampling.
- **Importance caching for many-light rendering.** Our second contribution is a method that aims to improve the efficiency of rendering a large number of VPLs by exploiting the illumination coherence in the scene. The idea is to cache the exact contributions of all VPLs at a sparse set of locations in the scene and then reuse these evaluations in the form of importance to probablis-

4 Section 1.2: Thesis outline

tically select the few most relevant VPLs at other nearby locations. Several importance distributions built at each cache location ensure that sampling remains robust around illumination discontinuities. We combine the many cached distributions gathered around every query location using a novel multiple importance sampling heuristic. The resulting *importance caching* algorithm can deliver significant noise reduction in scenes with complex occluded illumination. Unlike most other caching-based methods, our approach does not introduce additional bias, which allows it to produce high-quality results progressively with a bounded memory footprint.

- **Vertex connection and merging.** Our third contribution is based on a novel interpretation of photon density estimation as a Monte Carlo path sampling technique. This reformulation makes it possible, for the first time, to explain in a formal manner the relative efficiency of photon mapping and bidirectional path tracing – two algorithms that have so far been considered conceptually incompatible solutions to the light transport problem. More importantly, it allows us to employ multiple importance sampling to seamlessly integrate these methods into a more robust, unified practical rendering algorithm, which we call *vertex connection and merging*. We devise a progressive version of this algorithm that is consistent and efficiently handles a wide variety of lighting conditions, ranging from direct illumination, diffuse and glossy inter-reflections, to specular-diffuse-specular light transport. We show that this algorithm inherits the high asymptotic performance of bidirectional path tracing for most light path types, while benefiting from the efficiency of photon mapping for complex specular lighting effects.
- **Joint path sampling in participating media.** Our final contribution begins with an efficiency analysis of path sampling in participating media showing that traditional incremental vertex-by-vertex path construction can lead to high variance and slow error convergence. We then address this problem by devising *joint importance sampling* of path vertices in participating media. This leads to a set of new sampling routines to explicitly construct single- and double-scattering in anisotropically-scattering media. We demonstrate the benefit of our techniques by integrating them into various rendering algorithms, which brings a variance reduction of up to three orders of magnitude compared to prior methods.

1.2 Thesis outline

This thesis is organized into ten chapters. In Chapter 2 we review some relevant theoretical background in Monte Carlo integration, specifically integral estimators, error measures and variance reduction techniques, and set the basic mathematical notation that we will use throughout the remainder of the thesis. Chapter 3 introduces the physical laws of light propagation and scattering on surfaces and in participating media. We present the path integral formulation of light transport which will later serve as the theoretical basis of our contributions. In Chapter 4 we describe how Monte Carlo can be used to estimate the path integral and analyze the efficiency of existing path sampling techniques. In Chapter 5 we formulate our importance-driven virtual point light (VPL) distribution method, which we then complement in Chapter 6 with our importance caching scheme to improve the efficiency of rendering thousands of VPLs. Chapter 7 presents our reformulation of photon mapping as a path sampling technique and its combination with bidirectional path tracing into our unified vertex connection and merging (VCM) algorithm. Chapter 8 then discusses some important aspects of the practical implementation of the VCM algorithm. We finally focus on volumetric rendering in Chapter 9, where we analyze the shortcomings of existing volumetric path construction schemes and devise our new joint path importance sampling techniques. In Chapter 10 we summarize the contributions of this thesis and discuss some related follow-up work as well as possible avenues for future developments.

Monte Carlo Integration

2

Photorealistic image synthesis requires the physically-based simulation of light transport in a virtual three-dimensional environment. As we will see later in Chapter 3, this light transport problem can be formulated as an integration problem that is conceptually simple but difficult to solve in practice. The function to be integrated is defined on a high-dimensional space and has a complex shape that contains discontinuities and singularities. Since the behavior of this function depends on the input scene, analytic integration is generally infeasible and one has to resort to numerical integration instead. Unfortunately, the high dimensionality of the integral makes standard deterministic integration rules impractical, as their computational cost grows exponentially with the dimension. The best-known method for computing high-dimensional integrals is Monte Carlo integration, which is based on random sampling. Monte Carlo integration can handle almost any function, does not suffer from the curse of dimensionality, and is the basis of all state-of-the-art global illumination rendering algorithms.

In this chapter we give a brief introduction to Monte Carlo integration and basic concepts in probability theory. We define the terminology that we will be using in the following chapters and summarize some variance reduction techniques that have proved useful in computer graphics. We only focus on definitions and methods relevant to the contributions of this thesis and refer the interested reader to classic Monte Carlo literature for a more comprehensive introduction to the topic [44, 129, 69].

2.1 Random variables

In probability and statistics, a random variable is a variable whose possible values are numerical outcomes of a random phenomenon. There are two types of random variables: discrete and continuous. Unlike other mathematical variables, each possible value of a random variable has an associated probability (if discrete) or a probability density (if continuous).

2.1.1 Discrete random variables

A discrete random variable X can take a finite number M of possible values. Each possible outcome x'_i , for $i = 1, \dots, M$, has an associated probability $p(x'_i) \in [0, 1]$, where $p(x)$ is the *probability mass function* associated with the random variable X . The *cumulative distribution function* (CDF),

$$P(x) = \Pr \{X \leq x\} = \sum_{x' \leq x} p(x'), \quad (2.1)$$

6 Section 2.1: Random variables

gives the probability of X taking any value smaller than or equal to x . The corresponding notions for a discrete random multidimensional vector $\bar{X} = (X_1, \dots, X_k)$ are the *joint probability mass function* $p(x_1, \dots, x_k)$ and the *joint CDF*

$$P(x_1, \dots, x_k) = \Pr \{ X_i \leq x_i \text{ for all } i = 1, \dots, k \} \quad (2.2a)$$

$$= \sum_{x'_1 \leq x_1} \dots \sum_{x'_k \leq x_k} p(x'_1, \dots, x'_k). \quad (2.2b)$$

2.1.2 Continuous random variables

A real-valued, i.e. continuous, random variable X defined on the real line \mathbb{R} is characterized by its *probability density function*, or *pdf*. The pdf $p(x)$ is defined such that the probability that the variable takes a value x' in the infinitesimally small interval $[x, x + dx]$ is $p(x)dx$. The corresponding *cumulative distribution function* (CDF) provides a slightly more intuitive notion:

$$P(x) = \Pr \{ X \leq x \} = \int_{-\infty}^x p(x') dx'. \quad (2.3)$$

The pdf gives a relative probability that X takes a value in the (infinitesimally small) neighborhood of x , whereas the CDF gives the absolute probability that X takes any value smaller than or equal to x . From Equation 2.3 it follows that, for any two constants $a < b$,

$$\Pr \{ a \leq X \leq b \} = \int_a^b p(x') dx'. \quad (2.4)$$

The corresponding notions for a random multidimensional vector $\bar{X} = (X_1, \dots, X_k)$ defined on \mathbb{R}^k are the *joint pdf* $p(x_1, \dots, x_k)$ and the *joint CDF*

$$P(x_1, \dots, x_k) = \Pr \{ X_i \leq x_i \text{ for all } i = 1, \dots, k \} \quad (2.5a)$$

$$= \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_k} p(x'_1, \dots, x'_k) dx'_1 \dots dx'_k, \quad (2.5b)$$

from which it follows that

$$\Pr \{ a_i \leq X_i \leq b_i \text{ for all } i = 1, \dots, k \} = \int_{a_1}^{b_1} \dots \int_{a_k}^{b_k} p(x'_1, \dots, x'_k) dx'_1 \dots dx'_k. \quad (2.6)$$

More generally, for a random variable X with pdf $p(x)$ defined on an arbitrary domain Ω with measure function $\mu(x)$, its *probability measure*, or *probability distribution*, is defined as

$$P(D) = \Pr \{ X \in D \} = \int_D p(x) d\mu(x) \quad (2.7)$$

for any measurable set $D \subseteq \Omega$. The probability measure satisfies $P(\Omega) = 1$.

2.1.3 Expected value

The *expected value*, or *mean*, of a discrete random variable $Y = f(X)$ is defined as

$$E[Y] = \sum_{i=1}^M f(x'_i) p(x'_i). \quad (2.8)$$

The corresponding definition for a continuous random variable $Y = f(X) \in \Omega$ is

$$E[Y] = \int_{\Omega} f(x) p(x) d\mu(x). \quad (2.9)$$

From the above definitions it follows that, for any constant c ,

$$E[cY] = cE[Y], \quad (2.10)$$

and also, for a set of random variables Y_1, \dots, Y_k ,

$$E \left[c \sum_{i=1}^k Y_i \right] = c \sum_{i=1}^k E[Y_i]. \quad (2.11)$$

2.1.4 Variance and standard deviation

An important characteristic of a random variable is its *variance*, which measures the deviation of the outcomes from the expected value. The variance is defined as the second central moment of the variable about its mean:

$$\text{Var}[X] = E[(X - E[X])^2] \quad (2.12a)$$

$$= \int_{\Omega} (x^2 - 2xE[X] + E[X]^2) p(x) dx \quad (2.12b)$$

$$= E[X^2] - 2E[X]^2 + E[X]^2 = E[X^2] - E[X]^2. \quad (2.12c)$$

The above result also holds for discrete random variables, and the derivation is analogous. From the definition of variance it follows that, for any constant c ,

$$\text{Var}[cX] = c^2 \text{Var}[X]. \quad (2.13)$$

The *standard deviation* is the square root of the variance:

$$\sigma[X] = \sqrt{\text{Var}[X]}. \quad (2.14)$$

Note that the unit of the standard deviation are the same as the unit of the random variable itself, while for the variance it is the square of the variable's unit.

2.2 Integral estimators

Let us now consider the following definite integral over some domain Ω :

$$I = \int_{\Omega} f(x) d\mu(x). \quad (2.15)$$

The idea of Monte Carlo integration is to approximate this integral by defining a random variable whose expected value is the value of the integral. We define

$$\hat{I}_1(X) = \frac{f(X)}{p(X)}, \quad (2.16)$$

where the subscript signifies that \hat{I}_1 is a function of one random variable X defined on Ω with pdf $p(x)$. \hat{I}_1 itself is a random variable and its expected value is

$$\mathbb{E} [\hat{I}_1(X)] = \mathbb{E} \left[\frac{f(X)}{p(X)} \right] = \int_{\Omega} \frac{f(x)}{p(x)} p(x) d\mu(x) = \int_{\Omega} f(x) d\mu(x) = I. \quad (2.17)$$

\hat{I}_1 is called a *primary estimator* for I . The above equation holds under the following two conditions:

- $p(x) = 0$ only if $f(x) = 0$. This ensures that all non-zero values of f are sampled with non-zero probability.
- Whenever $p(x) = 0$, then $\hat{I}_1(X)$ must be defined as zero (to avoid a division by zero in Equation 2.16).

A *secondary estimator* for I is defined as

$$\hat{I}_N(X_1, \dots, X_N) = \frac{1}{N} \sum_{i=1}^N \hat{I}_1(X_i) = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}, \quad (2.18)$$

which computes the average of the primary estimators constructed from N independent and identically distributed random variables X_1, \dots, X_N . The expected value of the secondary estimator \hat{I}_N is also I :

$$\mathbb{E} [\hat{I}_N(X_1, \dots, X_N)] = \mathbb{E} \left[\frac{1}{N} \sum_{i=1}^N \hat{I}_1(X_i) \right] = \frac{1}{N} \sum_{i=1}^N \mathbb{E} [\hat{I}_1(X_i)] = \frac{1}{N} \sum_{i=1}^N I = I. \quad (2.19)$$

Secondary estimators of the form in Equation 2.18 can be evaluated progressively starting from a single primary estimate and incrementally accumulating new estimates into a running average. This scheme is based on a recursive formulation of the secondary estimator:

$$\underbrace{\hat{I}_N(X_1, \dots, X_N)}_{\text{new cumulative estimate}} = \frac{1}{N} \sum_{i=1}^N \hat{I}_1(X_i) = \frac{N-1}{N} \underbrace{\hat{I}_{N-1}(X_1, \dots, X_{N-1})}_{\text{old cumulative estimate}} + \frac{1}{N} \underbrace{\hat{I}_1(X_N)}_{\text{new estimate}}. \quad (2.20)$$

As we will see in the following section, the approximation error of a secondary estimator is inversely proportional to the square root of the number of samples, i.e. primary estimates, it uses. The above recursive formulation allows us to increase the accuracy of a given estimate by progressively accumulating new samples.

Note the simplicity and generality of the Monte Carlo method, which requires only two basic operations to produce an estimate of an integral: (1) a procedure for sampling points in the integration domain Ω and (2) point-wise evaluation of the integrand $f(x)$ and the pdf $p(x)$. Another important property is that its computational complexity of this method does not directly depend on the dimensionality of the integral.

2.3 Estimator error and convergence rate

It is important to understand that when using Monte Carlo integration, any actual estimate obtained by taking a random instance of an estimator only *approximates* the sought “true” integral value, even if its expectation is equal to that true value. In practice it is also useful to employ estimators whose expected values do not exactly equal the true value, especially when these estimators are cheaper to evaluate (a notable example is the photon mapping light transport estimator, which we will discuss in Chapters 4 and 7). It is therefore important to quantify and study the error of Monte Carlo estimators, which can provide insights into how to improve their accuracy and consequently obtain better approximations.

2.3.1 Error

The error of an integral estimate, i.e. a random instance of an estimator, is the difference between the value of the estimate \hat{I} and the true value I :

$$\text{Error}[\hat{I}] = \hat{I} - I. \quad (2.21)$$

Since the estimator is a random variable, its error is also a random variable whose behavior can be analyzed using common statistical tools, as we describe below.

2.3.2 Bias

The *bias* of a Monte Carlo estimator \hat{I} is the expected value of its error, which is also equal to the difference between the expected value and the true value I :

$$\text{Bias}[\hat{I}] = \text{E}[\hat{I} - I] = \text{E}[\hat{I}] - I. \quad (2.22)$$

An estimator whose expected value equals I has zero bias and is thus called *unbiased*, in contrast to *biased* estimators which have non-zero bias. The biased primary estimators used in rendering typically have the same form as the one in Equation 2.16, but evaluate a different function $g(x)$ which approximates $f(x)$.

2.3.3 Mean squared error

The *mean squared error* (MSE) of an estimator measures its average squared error:

$$\text{MSE}[\hat{I}] = \text{E}[(\hat{I} - I)^2] \quad (2.23a)$$

$$= \text{E}[\hat{I}^2] - 2\text{E}[\hat{I}]I + I^2 \quad (2.23b)$$

$$= (\text{E}[\hat{I}^2] - \text{E}[\hat{I}]^2) + (\text{E}[\hat{I}]^2 - 2\text{E}[\hat{I}]I + I^2) \quad (2.23c)$$

$$= \text{Var}[\hat{I}] + \text{Bias}[\hat{I}]^2. \quad (2.23d)$$

The MSE is the second moment of the estimator about the true value and incorporates both the variance of the estimator and its bias. The unit of measurement is the square of the unit of the quantity being estimated.

10 Section 2.3: Estimator error and convergence rate

The *root mean squared error* (RMSE) of an estimator is the square root of its MSE:

$$\text{RMSE} [\hat{I}] = \sqrt{\text{MSE} [\hat{I}]} = \sqrt{\text{Var} [\hat{I}] + \text{Bias} [\hat{I}]^2}. \quad (2.24)$$

The RMSE has the same unit as the quantity being estimated. The smaller the RMSE of an estimator is, the higher is the confidence that a random instance of this estimator closely approximates the desired quantity.

From Equation 2.23 it follows that for unbiased estimators the MSE is equal to the variance, which for a secondary estimator \hat{I}_N (Eq. 2.18) is

$$\text{MSE} [\hat{I}_N] = \text{Var} [\hat{I}_N] = \text{Var} \left[\frac{1}{N} \sum_{i=1}^N \hat{I}_1(X_i) \right] = \frac{1}{N^2} \sum_{i=1}^N \text{Var} [\hat{I}_1(X_i)] = \frac{1}{N} \text{Var} [\hat{I}_1(X_1)], \quad (2.25)$$

provided that the variance of the primary estimator \hat{I}_1 is finite. Also, for unbiased secondary estimators the RMSE is equal to the standard deviation:

$$\text{RMSE} [\hat{I}_N] = \sigma [\hat{I}_N] = \frac{1}{\sqrt{N}} \sigma [\hat{I}_1(X_1)]. \quad (2.26)$$

Knowing the standard deviation of a secondary estimator, we can obtain probabilistic bounds on the deviation of a random instance from the mean using the *central limit theorem*. This theorem states that the distribution of the values of \hat{I}_N converges to a normal distribution as N approaches infinity:

$$\lim_{N \rightarrow \infty} \Pr \{ \hat{I}_N - E [\hat{I}_N] \leq t \sigma [\hat{I}_N] \} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-x^2/2} dx. \quad (2.27)$$

This means that, for a large enough N , the probability that a random unbiased integral estimate \hat{I}_N will differ from the true value by more than three standard deviations is only about 0.3%.

2.3.4 Efficiency

By increasing the number of samples in a secondary estimator we can reduce its variance and thus its error. Doing so, however, linearly increases the computational cost of evaluating the estimator. Ideally, we would like to construct estimators whose error and computational cost are both small. This trade-off is described by the *efficiency* of a Monte Carlo estimator:

$$\varepsilon [\hat{I}] = \frac{1}{\text{MSE} [\hat{I}] \text{Time} [\hat{I}]}, \quad (2.28)$$

where $\text{Time} [\hat{I}]$ is the time required to obtain a random instance of \hat{I} . The lower the MSE obtained in a given fixed running time is, the more efficient the estimator is.

2.3.5 Consistency and convergence rates

A secondary Monte Carlo estimator is called *consistent* if the estimate converges almost surely to the correct solution as the number of samples approaches infinity:

$$\lim_{N \rightarrow \infty} \Pr \{ \hat{I}_N - I = 0 \} = 1. \quad (2.29)$$

Consistency is a very important property for practical applications as it ensures that the accuracy of a secondary Monte Carlo estimate can be improved arbitrarily by simply accumulating more random samples.

For unbiased secondary estimators of the form in Equation 2.18, whose expectation is equal to the correct solution, consistency is guaranteed by the *strong law of large numbers*, which states that the sample mean almost surely converges to the true mean as $N \rightarrow \infty$. In addition, from Equation 2.26 it follows that the *error convergence rate* of such estimators is $O(N^{-1/2})$. This means that in order to reduce the error by half, we need to quadruple the number of random samples, provided that the variance of the primary estimator is finite. The law of large numbers still holds when that variance is infinite, but the convergence rate is lower in this case. We will discuss some examples in Chapter 9.

A secondary estimator constructed from a biased primary estimator is generally inconsistent, since it only converge to the expected value of the primary estimator with an infinite number of samples. It may therefore seem logical to always prefer unbiased estimators. However, the use of carefully constructed biased estimators can be very beneficial in practice, as they can be more *efficient*, i.e. yield estimates with lower error at the same computational cost as an unbiased estimator. In rendering, this efficiency is often achieved by reusing samples for the estimation of multiple integrals (each corresponding to the value of one image pixel), motivated by the observation that neighboring pixels often have highly correlated integrals. Amortizing the sampling effort over many pixels can thus deliver cheap variance reduction, albeit at the cost of introducing systematic error (bias). As we will discuss in Chapters 4 and 7, some biased estimators can be made consistent by ensuring that, as the number of samples increases, both their variance vanishes and their expected value approaches the true value.

2.4 Sampling random variables

In order to obtain an actual estimate from a Monte Carlo integral estimator, we need to devise a *sampling procedure*, or *sampling technique*, for generating random points inside the integration domain. Having sampled an instance y of a random variable Y , we can use it to evaluate the estimator and obtain an integral estimate. In this section we review some techniques for sampling random variables according to a given probability density function.

2.4.1 Transforming between distributions

Sampling techniques usually operate by taking an input a random variable X defined on some domain with some distribution and transforming it to the domain of Y using a transformation function T :

$$Y = T(X), \quad (2.30)$$

where T must be a bijection, i.e. a one-to-one mapping from the domain of X to the domain of Y . As a direct consequence of this property, T must be strictly monotone (otherwise different X values would map to the same Y value). Thus, the derivative of T is either strictly positive or strictly negative, which implies the following relationship between the CDFs of X and Y :

$$P_Y(y) = P_Y(T(x)) = P_X(x). \quad (2.31)$$

12 Section 2.4: Sampling random variables

Differentiating the above equation with respect to y gives the pdf relationship¹

$$p(y) = \frac{p(x)}{|J_T|}, \quad (2.32)$$

where $|J_T|$ is absolute value of the determinant of T 's Jacobian matrix. For the case where X and Y are one-dimensional (i.e. scalar) variables, we have $|J_T| = \left| \frac{dy}{dx} \right|$.

2.4.2 CDF inversion method

We will usually want to prescribe the distribution $p(y)$ explicitly. Therefore, knowing the pdf of the input variable X , we need to derive a transformation T that will yield the desired distribution for Y . This follows directly from Equation 2.31:

$$y = T(x) = P_Y^{-1}(P_X(x)), \quad (2.33)$$

where P_Y^{-1} is the inverse CDF of Y . In a practical implementation, the input variable is most often a “canonical” uniform random number ξ in the interval $[0; 1]$ produced by a (pseudo-)random number generator [35]. In this case the above equation simplifies to

$$y = P_Y^{-1}(\xi), \quad (2.34)$$

since the CDF of the canonical input variable ξ is simply its value.

2.4.3 Multivariate distributions

To sample from a given joint distribution with pdf $p(y_1, \dots, y_k)$, we can use the relationship

$$p(y_1, \dots, y_k) = p(y_1, \dots, y_{k-1})p(y_k|y_1, \dots, y_{k-1}), \quad (2.35)$$

where $p(y_1, \dots, y_{k-1})$ is the *marginal* joint pdf of y_1, \dots, y_{k-1} and $p(y_k|y_1, \dots, y_{k-1})$ is the *conditional* pdf of y_k given y_1, \dots, y_{k-1} . The marginal pdf is obtained by integrating out y_k from the full joint pdf:

$$p(y_1, \dots, y_{k-1}) = \int_{\Omega} p(y_1, \dots, y_k) dy_k. \quad (2.36)$$

We can recursively expand the relationship in Equation 2.35 to obtain a product, i.e. a chain, of univariate marginal and conditional pdfs:

$$p(y_1, \dots, y_k) = p(y_1)p(y_2|y_1) \dots p(y_k|y_1, \dots, y_{k-1}). \quad (2.37)$$

The expressions for the conditional pdfs above can be efficiently obtained via successive marginalization, using a relationship that follows from Equations 2.35 and 2.36:

$$p(y_i|y_1, \dots, y_{i-1}) = \frac{p(y_1, \dots, y_i)}{p(y_1, \dots, y_{i-1})} = \frac{p(y_1, \dots, y_i)}{\int_{\Omega} p(y_1, \dots, y_i) dy_i}, \quad (2.38)$$

¹Strictly speaking, instead of $p(x)$ and $p(y)$ we should write $p_X(x)$ and $p_Y(y)$ to denote that X and Y have different pdfs. We will omit such subscripts when ambiguities can be resolved by the names of the arguments.

starting from $i = k$ and reusing the result of $\int_{\Omega} p(y_1, \dots, y_i) dy_i$ in the nominator of the expression for $p(y_{i-1} | \dots)$. Once we have obtained the expressions for all univariate pdfs in the chain, we can derive their corresponding canonical transformations by inverting their CDFs (Eq. 2.34). Using these transformations we can generate a random vector $\bar{y} = (y_1, \dots, y_k)$ with density $p(y_1, \dots, y_k)$ by successively drawing independent canonical random numbers ξ_i and computing the individual scalar variables y_i .

Equation 2.37 represents one *factorization* of the given joint pdf into a chain of marginal and conditional univariate pdfs. Each possible factorization corresponds to a different order of sampling the scalar variables y_i . A common application in graphics is sampling a point on a 2D domain from a joint density $p(x, y)$ defined by an image or a procedural texture [109, page 724]. This joint pdf can be factorized in two ways:

$$p(x, y) = p(x)p(y|x) = p(y)p(x|y). \quad (2.39)$$

For the case of sampling a pixel in a discrete image, the two possible factorizations correspond to either first choosing the column x from a marginal distribution over all rows and then choosing the row y from a distribution conditioned on the column x , or vice versa.

2.4.4 Global vs. local sampling

We will refer to the above method for generating a random multi-dimensional vector \bar{y} from a prescribed joint distribution as *global sampling*, as it requires strict coordination between the sampling decisions for all scalar variables y_i . It is also possible to generate a random vector by choosing the conditional distribution for each y_i directly. This *local sampling* approach is generally much simpler than global sampling, as it does not require marginalization. It is also practical, as it provides freedom to choose any pdf for each subsequent y_i , usually conditioning it only on the (few) previously sampled variable(s). In fact, as we will discuss later in Section 4.2, this is the standard way to sample paths for estimating the light transport integral (Eq. 3.27). However, a significant disadvantage of local sampling is that it does not provide means to directly control the shape of the final joint distribution of the generated vector, which can result in a high-variance integral estimator. In Chapter 9, we will use global sampling to construct light transport paths in participating media and obtain an efficiency gain of up to several orders of magnitude over local sampling.

2.4.5 Other methods

The CDF inversion method from Section 2.4.2 is generally the most efficient approach to sample from a given pdf. However, it requires the derivation of an analytic expression for the inverse of the CDF, which is possible only for relatively simple pdfs. Below we review some alternative, albeit less efficient, sampling methods that only require point-wise evaluation of the target pdf.

Arguably the simplest method to sample from a given pdf $p(x)$ is *rejection sampling*. This method uses an “envelope” distribution with pdf $q(x)$ that can be easily sampled from (e.g. the uniform distribution), and also requires finding a constant c such that $p(x) \leq cq(x), \forall x \in \Omega$, i.e. such that $cq(x)$ envelops $p(x)$. First, two random samples are drawn: a tentative sample x_i with pdf $q(x_i)$ and a uniform random number $\xi \in [0; 1)$. Then x_i is accepted as a realization of the target distribution $p(x)$ if $\xi \leq \frac{p(x_i)}{cq(x_i)}$. This method is very general, but its efficiency crucially depends on the choice of $q(x)$ and c – if $cq(x)$ is not

14 Section 2.5: Variance reduction techniques

a tight bound on $p(x)$ then many tentative samples will be rejected before a valid realization of $p(x)$ is drawn.

A popular and sophisticated variant of rejection sampling is *Metropolis sampling* [92], which belongs to the class of Markov chain Monte Carlo methods. This method aims to distribute samples from an unknown target pdf $p(x)$ proportional to a given non-negative function $f(x)$. The method produces a sequence, or a chain, of samples $\{X_i\}_{i=0}^N$, where the first one, X_0 , is chosen arbitrarily. To generate X_i , a random *mutation* is performed on the previous sample X_{i-1} , yielding a new tentative sample X'_i . Then X_i is set to either X_{i-1} or X'_i , based on a carefully chosen probability which is a function of $f(X_{i-1})$, $f(X'_i)$, and the probability densities for mutating from X_{i-1} to X'_i and from X'_i to X_{i-1} . The distribution of the samples in the chain converges to the target distribution in the limit $N \rightarrow \infty$. In contrast to traditional rejection sampling, every Metropolis sample is a valid sample. By correlating successive samples, this method can efficiently explore high-intensity regions of the target distribution. However, this correlation may come at the cost of a decreased error convergence rate of the secondary integral estimator constructed from the samples $\{X_i\}_{i=0}^N$. The efficiency of the method crucially depends on the initial state of the chain and the particular mutation strategy used. Seeding the process with a sample in a high-intensity region and developing a good mutation strategy for the specific problem can therefore greatly improve its performance.

When it is not strictly required to sample exactly from the given distribution, a practical and efficient alternative can be to tabulate the pdf (i.e. evaluate it at a finite number of locations) and construct an approximation from which samples can be drawn easily. The approximation is usually piece-wise analytical – an array of simple, constant or linear, functions that interpolate the values between the sampled locations. The inverse CDF is then also piece-wise analytical and can be used to transform canonical uniform random numbers. This *numerical CDF inversion* method can be much more efficient than rejection or Metropolis sampling when the approximation is accurate and the required preprocessing time and memory storage can be amortized by drawing a large number of samples. Unfortunately, this method suffers from the “curse of dimensionality” – the cost of building the approximation generally increases exponentially with dimension, which limits its application to low-dimensional sampling problems. In Chapter 9 we will use numerical inversion to sample light transport paths in anisotropically scattering media from high-dimensional joint distributions. To make this method practical, we will reduce the effective dimensionality of the tabulation by exploiting unique symmetries in the geometric configurations of light transport paths in participating media.

2.5 Variance reduction techniques

The most straightforward way to improve the accuracy of a secondary Monte Carlo estimator (Eq. 2.18) is to reduce its variance by increasing the number of samples N , as we discussed in Section 2.3.5. Unfortunately, this is also the *least efficient* way – since the computational cost and the variance of a secondary estimator respectively increase and decrease linearly with N , its efficiency (Eq. 2.28) is independent of the number of samples.

Devising efficient estimators is a major topic in Monte Carlo and rendering research and is also the main focus of this thesis. In this section we review some commonly used existing techniques that aim to lower the variance of an estimator for the same sampling effort. These techniques are based on improving the distribution of the samples over the integration domain. The different approaches mostly complement each other; we focus our discussion on importance sampling as it forms the basis of our contributions.

2.5.1 Stratified sampling

Recall that in order to obtain an actual Monte Carlo integral estimate (Eq. 2.16), we need to choose a probability distribution on the integration domain and draw samples from it. The simplest distribution is the *uniform* (i.e. constant) distribution $p(x) = 1/|\Omega|$, where $|\Omega|$ denotes the volume of the integration domain Ω . The resulting method is often called *blind* Monte Carlo integration as it does not utilize any information about the function being integrated.

The easiest way to distribute points uniformly on some domain is to scale canonical uniform random numbers in $[0; 1)$ to the size of the domain. However, this approach can lead to a poor sample distribution that produces a bad approximation of the integral. Random samples often clump together, leaving large holes and possibly missing important features of the integrand. The goal of *stratified* sampling is to mitigate this problem by enforcing a more even sample distribution. This is achieved by subdividing the domain Ω into N non-overlapping strata $\Omega_1, \dots, \Omega_N$ of equal size and placing one or more random samples inside each stratum. As a result, the sample distribution is much more uniform and the clumping significantly reduced. It has been shown that stratification can never increase the variance compared to pure random sampling; in fact, stratification can often increase the variance convergence rate of an estimator [93].

Stratified sampling is a very general variance reduction technique. Since it is an *uninformed* method, it can be applied to any Monte Carlo problem. It is widely used in practice, often in combination with other, *informed* variance reduction methods, which exploit some knowledge about the integrand. In the remainder of this section we will review some of these methods.

2.5.2 Adaptive sampling

The basic idea behind *adaptive sampling* is to gather information about the integrand during sampling and use it to concentrate the computational effort where it is likely to pay off most. A simple application of this idea is to keep a running estimate of the variance and terminate sampling as soon as the variance drops below a certain threshold. When computing multiple integrals, e.g. one for each pixel in a rendered image, this technique allows for allocating a larger portion of the sample budget to the “more difficult” integrals. This method can introduce bias that can be avoided via two-stage sampling [75]. Some more advanced methods detect where the integrand has high variation and place more samples in such regions [99, 40], while others construct and progressively adapt sampling distributions on-the-fly [32, 102]. Such methods can significantly boost efficiency, however they also suffer from the aforementioned “curse of dimensionality” as they need to maintain multi-dimensional data structures that store sampling information.

2.5.3 Russian roulette and splitting

Russian roulette and *splitting* are two related methods that aim to increase the sampling density in regions where the integrand is large and decrease it where the integrand is small [2]. They bare similarity with importance sampling (discussed below), and unlike some adaptive sampling methods are unbiased.

16 Section 2.5: Variance reduction techniques

The idea of Russian roulette (RR) is to replace an estimator \hat{I} with a new estimator

$$\hat{I}^{\text{RR}} = \begin{cases} \frac{1}{p} \hat{I} & \text{if } \xi < p \\ 0 & \text{otherwise,} \end{cases} \quad (2.40)$$

where $\xi \in [0; 1)$ is a canonical uniform random number. This formulation allows us to probabilistically avoid the potentially expensive evaluation of \hat{I} when we can expect it to have low contribution. The evaluation probability $p \in (0; 1]$ should ideally be low when the value of \hat{I} is predicted to be low, and high when it is predicted to be high. Note that the introduced random decision actually results in an estimator with a *higher* variance. Nevertheless, \hat{I}^{RR} can still have a higher efficiency than \hat{I} if the average time required to evaluate it is lower. In light transport simulation, this technique is particularly useful for terminating random walks in a scene, as we will discuss in Chapter 4.

Instead of trying to reduce the evaluation cost, splitting replaces a primary multi-dimensional estimator $\hat{I}(X_1, \dots, X_k)$ with a secondary estimator

$$\hat{I}^{\text{split}}(X_1, \dots, X_k) = \frac{1}{N} \sum_{i=1}^N \hat{I}(X_1, \dots, X_s, X_{s+1,i}, \dots, X_{k,i}), \quad (2.41)$$

which fixes the values of the first s random variables and draws N independent realizations of the remaining $k - s$ variables. Similarly to Russian roulette, the splitting factor N can be chosen based on a rough prediction for the value of \hat{I} . While this splitting increases the evaluation cost of \hat{I} , it can improve efficiency for high-dimensional integration when the integrand has little variation in the first s dimensions. Rendering algorithms often use path splitting, e.g. for estimating the illumination at a point by sampling multiple light sources or by spawning random secondary paths from the point.

Splitting and Russian roulette can be combined by replacing each primary estimator in the sum in Equation 2.41 by a corresponding \hat{I}^{RR} estimator. Carefully choosing the splitting factor and the Russian roulette evaluation probability makes it possible to concentrate the sampling effort in important regions of the integration domain.

2.5.4 Importance sampling

From the very definition of Monte Carlo integration (Eq. 2.16) it is apparent that the choice of sampling pdf affects the variance of the estimator. In particular, the values of an estimator vary a lot when high integrand values $f(x)$ are sampled with low probability $p(x)$ and when low values of $f(x)$ are sampled with high probability.

Importance sampling is an informed variance reduction technique whose goal is to choose an optimal sampling pdf that minimizes the estimator's variance. Interestingly, for any given integrand $f(x)$ the ideal sampling pdf is known: it is exactly proportional to the integrand, i.e. $p(x) = cf(x)$, where

$$c = \frac{1}{\int_{\Omega} f(x) d\mu(x)} \quad (2.42)$$

is a normalization constant which ensures that $p(x)$ integrates to one. Using this pdf yields a *zero-variance* primary estimator,

$$\hat{I}(X) = \frac{f(X)}{p(X)} = \frac{1}{c}, \quad (2.43)$$

which has a constant value for all sample points X . Slightly counter-intuitively, this means that we can obtain the exact answer with just a single sample. However, using this pdf is impractical (and actually does not even make sense), as it requires already knowing the value of the desired integral in order to compute the normalization constant c (Eq. 2.42). Nevertheless, significant variance reduction can be achieved by using a pdf p whose shape is similar to f . This is typically accomplished by discarding or approximating some of the terms of f in order to obtain a simplified function whose normalization constant and inverse CDF can be derived easily. Variance can be reduced further by additionally stratifying the canonical uniform random numbers transformed with the inverse CDF (Sec. 2.5.1).

Importance sampling is a very powerful technique as it can achieve substantial variance reduction at a low cost when the generation of samples from the chosen distribution is computationally efficient. In rendering, importance sampling is particularly useful for computing illumination integrals involving sampling sharp light scattering distributions (e.g. BSDFs) or light sources with non-uniform emission. The main contributions of this thesis are novel importance sampling techniques for solving the light transport integral that we will introduce in the following chapter.

2.5.5 Mixture importance sampling

A good importance sampling distribution should ideally be closely proportional to the integrand, but unfortunately deriving such a pdf can be difficult even for simple functions. As an alternative, complex sampling distributions can be constructed by composing, or mixing, a number of simple distributions. One way to achieve this is via *mixture importance sampling* whose primary estimator

$$\hat{I}^{\text{mix}}(X) = \frac{f(X)}{\sum_{i=1}^n w_i p_i(X)} \quad (2.44)$$

uses a pdf $p(x) = \sum_{i=1}^n w_i p_i(x)$ that is a convex combination of pdfs, i.e. with $w_i \geq 0$ and $\sum_{i=1}^n w_i = 1$. To evaluate this estimator, a random pdf p_i is first chosen with probability w_i and then a sample is drawn from the that pdf, effectively sampling from the mixture p . Alternatively, the number of samples from each pdf can be deterministically set to $n_i = w_i N$, where $N = \sum_{i=1}^n n_i$ is the total number of samples. This yields a technique known as *deterministic mixture sampling*:

$$\hat{I}_N^{\text{mix}} = \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{p_i(X_{i,j})}. \quad (2.45)$$

where $X_{i,j}$ are independent random variables with pdf p_i . This estimator can have a slightly lower variance than a secondary estimator that averages N independent realizations of the primary estimator in Equation 2.44. The difference comes from eliminating the randomness in the number of samples taken from each p_i .

2.5.6 Multiple importance sampling

An even more powerful method for combining different distributions was developed by Veach and Guibas [144], called *multiple importance sampling* (MIS). The multi-sample MIS estimator combines the estimators of n sampling techniques:

$$\hat{I}_N^{\text{MIS}} = \sum_{i=1}^n \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})}, \quad (2.46)$$

18 Section 2.5: Variance reduction techniques

where w_i is a weighting function associated with the pdf p_i , and n_i is the number of samples taken from p_i . This estimator is unbiased as long as the family of weighting functions w_i , referred to as the *weighting heuristic*, fulfills the following two conditions:

- $\sum_{i=1}^n w_i(x) = 1$ whenever $f(x) \neq 0$, and
- $w_i(x) = 0$ whenever $p_i(x) = 0$.

Veach [142] showed that the *power heuristic* is a provably good choice for a weighting heuristic in terms of minimizing the variance of the combined estimator in Equation 2.46:

$$w_i(x) = \frac{[n_i p_i(x)]^\beta}{\sum_{k=1}^n [n_k p_k(x)]^\beta}, \quad (2.47)$$

where the exponent $\beta \geq 0$ is a parameter. The special case for $\beta = 1$ is called the *balance heuristic*, for which the estimator simplifies to

$$\hat{I}_N^{\text{MIS}} = \sum_{i=1}^n \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{n_i p_i(X_{i,j})}{\sum_{k=1}^n n_k p_k(X_{i,j})} \frac{f(X_{i,j})}{p_i(X_{i,j})} = \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{\sum_{k=1}^n (n_k/N) p_k(X_{i,j})}, \quad (2.48)$$

where $N = \sum_{k=1}^n n_k$. This estimator is equivalent to the deterministic mixture importance sampling estimator (Eq. 2.45) which can consequently be considered a special case of multiple importance sampling. Veach [142] proved that no other weighting heuristic can result in an estimator with a much lower variance than the balance heuristic.

Another good choice is the *maximum heuristic* which effectively partitions the integration domain into disjoint regions and uses only the technique with the highest sampling probability for each region:

$$w_i(x) = \begin{cases} 1 & \text{if } n_i p_i(x) = \max_k [n_k p_k(x)], \\ 0 & \text{otherwise.} \end{cases} \quad (2.49)$$

While this heuristic generally results in a higher-variance estimator than the power heuristic, in practice it can often be more *efficient* as it allows us to avoid the evaluation of the integrand $f(x)$ for samples x with zero weight².

It is also possible to combine n different techniques into a primary MIS estimator. The one-sample MIS estimator is defined as:

$$\hat{I}^{\text{MIS}} = w_I(X_I) \frac{f(X_I)}{c_I p_I(X_I)}. \quad (2.50)$$

Here, we first choose a technique by drawing a random integer $I \in \{1, \dots, n\}$ from a discrete distribution $\{c_1, \dots, c_n\}$, where c_i is the probability of choosing pdf p_i , with $\sum_{i=1}^n c_i = 1$. Then, a random sample X_I is drawn from the distribution p_I . This estimator is unbiased under the same two conditions on the weighting functions w_i above. Veach [142] proved that in this one-sample case no other weighting heuristic can result in lower variance than the balance heuristic. As in the multi-sample case, using this heuristic corresponds to sampling from a mixture distribution, yielding the estimator in Equation 2.44 with $w_i = c_i$.

Multiple importance sampling (MIS) is particularly useful when integrating complex functions for which it is difficult to derive one single good sampling distribution. For such cases, MIS provides a general and powerful framework for combining the estimates of a number of distributions while

²Note that this optimization is not possible with the power heuristic which only yields weights $w_i(x) = 0$ when $p_i(x) = 0$, and such points x are never sampled by technique i in the first place.

preserving the qualities of each, often resulting in a substantial variance reduction over using just one technique. The key to the robustness of the combination heuristics described above is the assignment of weights proportional to the sampling pdfs of the individual techniques. This weighting scheme is based on the observation that a larger pdf value most often results in a lower-error estimate. That same observation also makes it possible to analyze the relative efficiency of different sampling techniques by simply comparing the values of their pdfs at a given point x , as we will do repeatedly in the following chapters. We will also make extensive use of multiple importance sampling to combine various Monte Carlo techniques for solving the light transport problem.

Mathematical Models of Light Transport

3

Photographic cameras and the human eyes are comprised of a large number of light sensors. When we release the camera shutter or open our eyes, each sensor records the light falling on it. The data collected by all sensors constitutes an image of the visible environment. This incident light has been originally emitted by some light source and has traveled throughout the environment by potentially bouncing, i.e. scattering, a number of times at objects' surfaces and/or in participating media before reaching a sensor.

In this thesis we are interested in the problem of producing a digital image of a virtual three-dimensional environment from the perspective of a virtual observer. An image is a rectangular array (matrix) of color values, each corresponding to the response of a light sensor. Before we can generate such an image, the environment first needs to be modeled on a computer. This involves obtaining a representation of the scene geometry as well as defining the appearance of the objects (i.e. their light scattering properties), the emission characteristics of the light sources, and the light sensitivity of the observer's eye. With this in place, we can perform a simulation of the light transport in the virtual scene and compute the response of each sensor. This process of synthesizing an image from a given scene description is called *rendering* and is the focus of our work.

The key to rendering a photorealistic image is to compute the lighting in the scene in a physically plausible way. This means that the light transport simulation must obey the same laws that govern the propagation and scattering of light in the real world. Having a formal definition of the rendering problem based on these laws, numerical solutions can be devised to obtain realistic-looking images.

In this chapter, we introduce the concepts and definitions required to describe the problem that every global illumination rendering algorithm must solve. We present a formulation of light transport as an integration problem to which we will develop practical numerical solutions in the following chapters. The covered material is described at a fairly high level and is not meant to be exhaustive. For an in-depth introduction to the foundations of physically-based rendering we refer the reader to the distinguished books of Dutré et al. [30] and Pharr and Humphreys [109].

3.1 Light models

The human effort to understand the nature of light has a long history and many theories have been proposed over the centuries [48]. The most comprehensive model available today, *quantum optics*, explains all known light phenomena at a submicroscopic (electron) level; these include black body radiation, fluorescence, and phosphorescence. A simplification of this model, *wave optics*, captures

22 Section 3.2: Basic radiometry

effects that arise when light interacts with objects of size comparable to its wavelength, such as diffraction and polarization. However, neither of these models is commonly used in computer graphics as they are usually considered too detailed for the purpose of image synthesis.

In this thesis we adhere to an even more simplified but widely used light model – *geometric optics*, or *ray optics*. This model allows us to simulate most light phenomena we see in our daily life in a plausible way while making some useful simplifying assumptions about the behavior of light. Most importantly, geometric optics assumes that (1) light travels along straight lines and that (2) light propagates and scatters instantaneously in the environment. The first assumption allows us to use *ray tracing* techniques to efficiently track light particles in a scene. The second assumption allows us to formulate the rendering problem as an integral over a steady-state (i.e. equilibrium) distribution of light energy in the scene. This simplified model cannot account for optical effects such as diffraction, interference, or phosphorescence. However, it provides an accurate approximation when the wavelength of light is small compared to the size of the structures with which light interacts, which is typically the case for the scenes considered in most practical applications.

3.2 Basic radiometry

Under the geometric optics model, the task of a global illumination algorithm is to measure the eye sensor response to the equilibrium distribution of light in the scene. To do this, we first need to compute this equilibrium distribution, which requires an understanding of the physical quantities that represent light energy. Radiometry is the science that studies electromagnetic radiation, including visible light. Below we describe some relevant radiometric quantities that we will make use of in this thesis.

3.2.1 Radiant power or flux

Flux expresses the total energy flow, incident at or exiting from a surface or a volume, per unit time:

$$\Phi = \frac{dQ}{dt}. \quad (3.1)$$

Flux is measured in watts (joules/sec). A common use of this quantity is to specify the total emission power of light sources, e.g. 60W for a typical light bulb (assuming that all the electric energy consumed by the bulb is actually emitted as light).

3.2.2 Irradiance and radiosity

Irradiance represents the flux arriving at a surface point \mathbf{x} , per unit surface area:

$$E(\mathbf{x}) = \frac{d\Phi}{d\mathbf{x}}, \quad (3.2)$$

where we use $d\mathbf{x}$ is a shorthand notation for the differential area measure. Radiosity, B , represents power *leaving* a surface, and has the same definition as irradiance. Both quantities are functions of position, \mathbf{x} , and are measured in watts/m². They can be used to compactly encode the light equilibrium in a directionally-independent form, which is especially practical for scenes with mostly diffuse (i.e. uniformly scattering) objects.

3.2.3 Radiance

Radiance is the most basic quantity used in global illumination algorithms, as it also accounts for the directionality of light flow. Radiance represents the incident or exitant energy at a point \mathbf{x} in a given direction ω , per unit surface area perpendicular to the direction and per unit solid angle. It is measured in watts/(steradian \cdot m²):

$$L(\mathbf{x} \leftarrow \omega) = \frac{d^2\Phi}{|\cos\theta|d\mathbf{x}d\omega}. \quad (3.3)$$

Here, $d\omega$ denotes differential solid angle:

$$d\omega = |\sin\theta|d\theta d\phi, \quad (3.4)$$

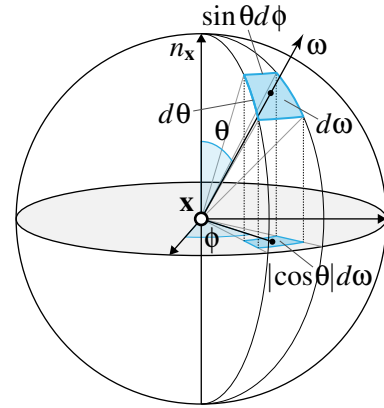


Figure 3.1: Spherical coordinates.

and $\cos\theta = n_{\mathbf{x}} \cdot \omega$, where $n_{\mathbf{x}}$ is the surface normal at \mathbf{x} (see Figure 3.1). The cosine term appears only in the radiance formulation for surface points, and not for points in media. An intuitive explanation for this term is that the power arriving at grazing angles is smeared out over a larger area on the surface.

Radiance encodes the spatio-directional light distribution in a scene. We will denote incident and exitant radiance respectively by $L(\mathbf{x} \leftarrow \omega)$ and $L(\mathbf{x} \rightarrow \omega)$. Integrating these two quantities over the sphere around \mathbf{x} gives the irradiance and radiosity respectively.

3.2.4 Importance

Though not strictly a radiometric quantity, importance is a very useful hypothetical quantity in computer graphics. It is the dual of radiance and expresses the eye's or the camera's sensitivity to light. While radiance is emitted from light sources, importance is "emitted" from the eye. It is denoted by W and is by convention measured in the same units as radiance (although without any physical meaning).

3.3 Wavelength dependency and color

The radiometric quantities described in the previous section are in general not only functions of positions and directions but also of the wavelength of light. When wavelength is specified, *spectral radiance* is the quantity corresponding to radiance. Radiance is then defined as the integral of spectral radiance over the wavelength spectrum:

$$L(\mathbf{x} \leftarrow \omega) = \int_{\text{spectrum}} L(\mathbf{x} \leftarrow \omega, \lambda) d\lambda. \quad (3.5)$$

Since radiance represents the total intensity over all wavelengths, it does not carry any color information. In order to render color images, one should ideally measure spectral radiance in the scene for every wavelength in the visible spectrum. Fortunately, this problem can be simplified by exploiting

24 Section 3.4: Emission, propagation, and scattering

the limitations of the human visual system: the human eye has only three different types of light receptors, each with higher sensitivity respectively to the red, green, and blue regions of the visible light spectrum. According to the tristimulus theory, any perceivable color can be represented by just three scalar values – the responses of the three types of receptors to spectral radiance. For any given point and direction in the scene, we can then perform just three measurements, L_r , L_g , and L_b , one for each of the red, green, and blue components:

$$L_r(\mathbf{x} \leftarrow \omega) = \int_{\text{spectrum}} R(\lambda) L(\mathbf{x} \leftarrow \omega, \lambda) d\lambda, \quad (3.6)$$

where $R(\lambda)$ is the response curve of the red sensor. L_g and L_b are defined analogously. These three values are sufficient for reproducing color in a digital image.

The formulas describing the equilibrium radiance at a point, presented in Section 3.5 below, are generally wavelength-dependent. However, this dependence is due to only a few terms, and the remaining terms come out of the wavelength integral in Equation 3.6. In most practical scenes, the wavelength-dependent terms are the ones that describe the spatial (i.e. positional) variation of light emission and scattering in the scene. This variation is typically specified via three-channel procedural or tabulated textures whose values are already preconvolved with the R , G , and B response curves. This means that in most scenes we can directly measure the “radiance responses” L_r , L_g and L_b . With a slight abuse of terminology, in the remainder of this thesis we will refer to the latter three quantities as radiance. To keep the exposition simple, we will also omit the subscripts, and note that whenever radiance L appears in an expression, three separate measurements, for L_r , L_g , and L_b respectively, must be performed in practice. Note that spectral radiance measurements are still required for rendering effects like dispersion in the presence of materials that scatter light in wavelength-dependent directions.

3.3.1 Luminance

We often need to measure the overall brightness appearance of light, regardless of hue. Color theory postulates that the apparent brightness is given by the *luminance*, which can be estimated from L_r , L_g , and L_b [98]:

$$\text{Luminance}[L] = 0.21 L_r + 0.72 L_g + 0.07 L_b. \quad (3.7)$$

Above we have used the coefficients given by Pharr and Humphreys [109, Chap. 5]. Note that most weight is given to the green channel, which accounts for the fact that the human eye is much more sensitive to the green part of the visible light spectrum.

3.4 Emission, propagation, and scattering

Under the geometric optics model, it is useful to think of light in terms of infinitesimally small light particles, or photons. Every photon starts its journey at a light source and travels along a straight line, or ray, until it interacts with a surface of a solid object or with a particle in a medium. Upon interaction, a photon can be either absorbed or scattered in another direction to continue propagating in the scene. In this section we describe the mathematical models for light emission, propagation, and interaction, which are the building blocks for describing the energy equilibrium in a scene.

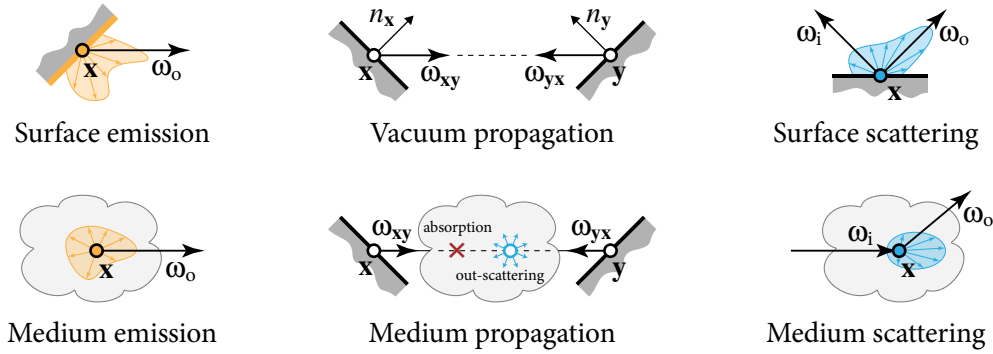


Figure 3.2: Schematic illustration of light emission, propagation, and scattering. Left: Light source emission is described by emission distribution functions. Middle: In a vacuum, the radiance leaving point \mathbf{x} is equal to the radiance arriving at point \mathbf{y} . In participating media, radiance is attenuated due to absorption and out-scattering. Right: Scattering at a surface or in a medium is described by spherical distribution functions similar to the ones for emission.

3.4.1 Emission

Light emission can be mathematically modeled via an *emission distribution function* $L_e(\mathbf{x} \rightarrow \boldsymbol{\omega})$ that defines the emitted radiance at any surface or medium point \mathbf{x} in the scene in direction $\boldsymbol{\omega}$ (see Figure 3.2). Diffuse (i.e. uniform) emission is described by a simple constant distribution over the (hemi-)sphere. Generally, the shape of an emission distribution, also called *emission profile*, can be much more complex, particularly for light sources used in illumination engineering [97]. Both the total emitted power (radiosity) and the shape of the emission profile can vary across the light's surface or volume.

Similarly to light source emission, the spatio-directional importance emission of the eye can be described by a corresponding distribution function $W_e(\mathbf{x} \rightarrow \boldsymbol{\omega})$ which models the response of the sensors.

3.4.2 Vacuum propagation

In a vacuum, the law of energy conservation states that all flux leaving a differential surface around a point \mathbf{x} in the direction of a differential surface around another point \mathbf{y} must arrive at that surface. Assuming there is no obstructing geometry between \mathbf{x} and \mathbf{y} , i.e. that the two points are mutually visible, from this law it follows that

$$L(\mathbf{x} \rightarrow \boldsymbol{\omega}_{\mathbf{x}\mathbf{y}}) = L(\mathbf{y} \leftarrow \boldsymbol{\omega}_{\mathbf{y}\mathbf{x}}), \quad (3.8)$$

which means that the radiance leaving point \mathbf{x} in direction $\boldsymbol{\omega}_{\mathbf{x}\mathbf{y}}$ toward point \mathbf{y} is equal to the radiance arriving at point \mathbf{y} from direction $\boldsymbol{\omega}_{\mathbf{y}\mathbf{x}}$ (see Figure 3.2) [30, Sec. 2.3.3].

Using the definition of radiance (Eq. 3.3) and the relation between differential solid angle and differential surface area,

$$d\boldsymbol{\omega}_{\mathbf{x}\mathbf{y}} = d\mathbf{y} \frac{|n_{\mathbf{y}} \cdot \boldsymbol{\omega}_{\mathbf{x}\mathbf{y}}|}{\|\mathbf{x} - \mathbf{y}\|^2}, \quad (3.9)$$

where $n_{\mathbf{y}}$ is the surface normal at \mathbf{y} , we can write the differential power transferred from \mathbf{x} to \mathbf{y} as

$$\frac{d^2\Phi}{d\mathbf{x}d\mathbf{y}} = L(\mathbf{x} \rightarrow \boldsymbol{\omega}_{\mathbf{x}\mathbf{y}})G(\mathbf{x}, \mathbf{y}). \quad (3.10)$$

26 Section 3.4: Emission, propagation, and scattering

The *geometry term* G above accounts for the distance between the differential surfaces and their mutual orientation:

$$G(\mathbf{x}, \mathbf{y}) = \frac{|n_{\mathbf{x}} \cdot \boldsymbol{\omega}_{\mathbf{xy}}| |n_{\mathbf{y}} \cdot \boldsymbol{\omega}_{\mathbf{yx}}|}{\|\mathbf{x} - \mathbf{y}\|^2}. \quad (3.11)$$

Integrating Equation 3.10 over the areas of two mutually visible surface patches gives the total flux transferred between them.

3.4.3 Medium propagation

When media occupy the space between objects, the assumption of constant radiance along straight lines does not necessarily hold any more, because media *participate* in the light transport in the scene. Light interaction with participating media is described via a probabilistic model that treats a medium as a collection of infinitesimally small particles and considers the aggregate behavior of these particles as light travels through the medium.

When a photon enters a medium, it can either miss all particles and continue unaffected, or it can interact with the particles causing a change in the radiance along the photon ray. The probability for an interaction to occur is related to the *extinction coefficient*, σ_t , of the medium. This quantity is the inverse *mean free path* of the medium – the average distance traveled by a photon before interaction, and depends on the density and size of the particles. When an interaction does occur, the photon can be either absorbed or scattered in another direction. The relative probabilities for these two events are given by the *absorption coefficient* σ_a and the *scattering coefficient* σ_s . Both of these coefficients can vary across the medium, in which case the medium is called *heterogeneous*; when they are constant, the medium is *homogeneous*. The relation between these three coefficients is: $\sigma_t(\mathbf{x}) = \sigma_a(\mathbf{x}) + \sigma_s(\mathbf{x})$.

The radiance attenuation along a ray segment due to absorption and out-scattering is given by the *transmittance* (see Figure 3.2):

$$T_r(\mathbf{x}, \mathbf{y}) = e^{-\tau(\mathbf{x}, \mathbf{y})} = e^{-\int_0^{\|\mathbf{x}-\mathbf{y}\|} \sigma_t(\mathbf{x} + t\boldsymbol{\omega}_{\mathbf{xy}}) dt}, \quad (3.12)$$

where the *optical thickness* $\tau(\mathbf{x}, \mathbf{y})$ is the integral of the extinction coefficient along the ray segment between \mathbf{x} and \mathbf{y} . In homogeneous media, the optical thickness simplifies to $\tau(\mathbf{x}, \mathbf{y}) = \sigma_t \|\mathbf{x} - \mathbf{y}\|$. Note that the optical thickness takes values in the interval $[0; \infty)$, whereas for transmittance we have $T_r(\mathbf{x}, \mathbf{y}) \in [0; 1]$. The absence of participating media corresponds to $\sigma_t = 0$, in which case the optical thickness is always zero and the transmittance between any two points is simply 1.

3.4.4 Visibility

Opaque objects can be thought of as infinitely dense media with $\sigma_t = \infty$ which, when intersected by a ray segment, make the transmittance $T_r(\mathbf{x}, \mathbf{y})$, and thus the radiance between \mathbf{x} and \mathbf{y} , drop to zero. However, for convenience we introduce a separate *visibility* term that explicitly accounts for the mutual visibility between the two points:

$$V(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ and } \mathbf{y} \text{ are mutually visible,} \\ 0 & \text{otherwise.} \end{cases} \quad (3.13)$$

We can now generalize Equation 3.10 to describe the differential flux from \mathbf{x} to \mathbf{y} , accounting for both medium transmittance and visibility:

$$\frac{d^2\Phi}{d\mathbf{x}d\mathbf{y}} = L(\mathbf{x}\rightarrow\mathbf{y})G(\mathbf{x},\mathbf{y})T_r(\mathbf{x},\mathbf{y})V(\mathbf{x},\mathbf{y}). \quad (3.14)$$

Note that this formula is symmetric, in the sense that the differential flux in the opposite direction, from \mathbf{y} to \mathbf{x} , is given by simply replacing $L(\mathbf{x}\rightarrow\mathbf{y})$ by $L(\mathbf{y}\rightarrow\mathbf{x})$ above.

3.4.5 Surface scattering

When a photon hits the surface of an object, it can be either absorbed by the object (and converted to another form of energy, such as heat) or scattered in another direction and continue its propagation. The probability for scattering is given by the *albedo* of the surface material at the intersection location, which we will define below. The probability density that a photon coming from direction ω_i at a surface point \mathbf{x} is scattered in a direction ω_o is given by the *bidirectional scattering distribution function* (BSDF):

$$\rho_s(\mathbf{x}, \omega_i, \omega_o) = \frac{dL(\mathbf{x}\rightarrow\omega_o)}{L(\mathbf{x}\leftarrow\omega_i)|n_{\mathbf{x}}\cdot\omega_i|d\omega_i}. \quad (3.15)$$

The BSDF is defined as the ratio between the differential outgoing radiance in direction ω_o and the differential irradiance from direction ω_i . By convention, both ω_i and ω_o point away from the surface. The BSDF determines the appearance of a surface point, i.e. how bright it looks when illuminated from direction ω_i and viewed from direction ω_o .

For any pair of incoming and outgoing directions the BSDF can take any non-negative value. However, every physically-plausible BSDF must have the following two properties:

- *Energy conservation*: This property asserts that a surface cannot scatter more light than it receives. It can be expressed as the following constraint:

$$\alpha_s(\mathbf{x}, \omega_i) = \int_{\mathcal{S}} \rho_s(\mathbf{x}, \omega_i, \omega_o) |n_{\mathbf{x}}\cdot\omega_o| d\omega_o \leq 1, \quad (3.16)$$

where \mathcal{S} is the unit sphere. Here, $\alpha_s(\mathbf{x}, \omega_i)$ is the *surface reflectance*, or *surface scattering albedo*, which is the total probability that a light particle coming from direction ω_i at \mathbf{x} will scatter off the surface in any direction (rather than being absorbed). Note that all real-world materials have albedo strictly smaller than one, meaning that there is always a non-zero probability for a photon to be absorbed.

- *Reciprocity*: This property ensures that the value of the BSDF remains unchanged when the incoming and outgoing directions are swapped:

$$\rho_s(\mathbf{x}, \omega_i, \omega_o) = \rho_s(\mathbf{x}, \omega_o, \omega_i). \quad (3.17)$$

This means that the surface scattering is invariant to the direction of light flow. Many global illumination algorithms rely on this property to simulate the light transport distribution in a scene starting from the eye.

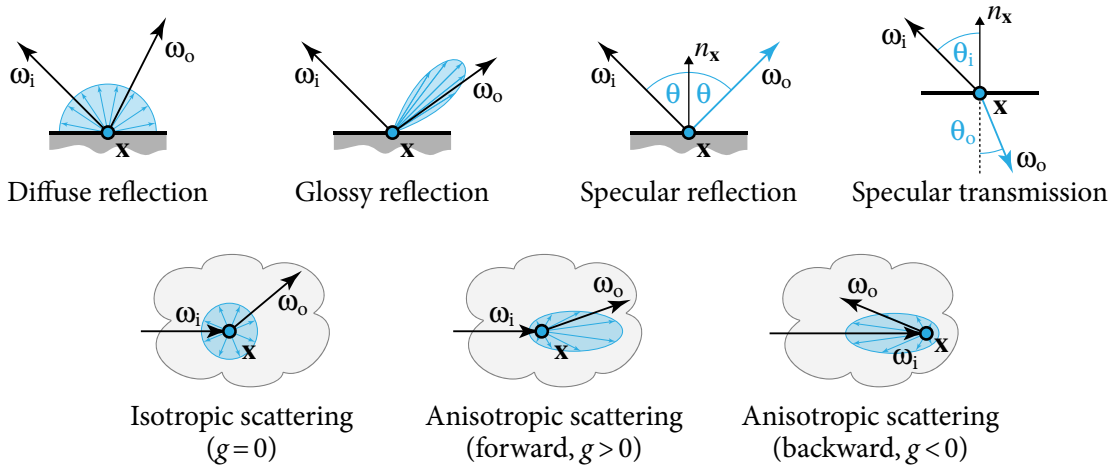


Figure 3.3: Various classifications of surface and medium scattering distribution functions.

BSDF classification

Generally speaking, a surface defines the boundary, or interface, between two dissimilar media. When light strikes a surface it can be either reflected back into the medium it came from or transmitted into the other medium, if the latter is transparent. The BSDF describes how light coming from a particular direction is distributed in all other directions. This scattering distribution can be roughly categorized as *diffuse*, *glossy*, or *specular* (see Figure 3.3). Diffuse BSDFs, which scatter light uniformly, are described by an isotropic, i.e. constant, (hemi-)spherical pdf. Glossy BSDFs scatter light preferentially in a cone of directions. Specular BSDFs scatter light in a single direction and are mathematically modeled via Dirac delta distributions. Integrals involving delta BSDFs must be evaluated with care because their associated measure is not the usual Lebesgue measure, as detailed by Glassner [36, Sec. 13.8.2] and Veach [142, Sec. 5.2]. Recently, Jakob and Marschner [57] presented an interpretation of specular scattering based on path-space manifolds rather than delta distributions.

The BSDFs of real-world materials can have complex shapes and are often modeled as mixtures of the three basic types discussed above. Appearance modeling is a broad research topic that is outside the scope of this thesis. Comprehensive overviews of surface scattering models can be found in the books of Dorsey et al. [26] and Pharr and Humphreys [109, Chap. 8].

3.4.6 Medium scattering

Similarly to the surface case, when a photon traveling through a medium interacts with one of its particles, it can either be absorbed or scattered in another direction. The probability for scattering is given by the *medium scattering albedo*:

$$\alpha_m(\mathbf{x}) = \frac{\sigma_s(\mathbf{x})}{\sigma_t(\mathbf{x})} = \frac{\sigma_s(\mathbf{x})}{\sigma_s(\mathbf{x}) + \sigma_a(\mathbf{x})}. \quad (3.18)$$

The directional scattering distribution is given by the *phase function* of the medium:

$$\rho_m(\mathbf{x}, \omega_i, \omega_o) = \frac{dL(\mathbf{x} \rightarrow \omega_o)}{L(\mathbf{x} \leftarrow \omega_i) d\omega_i}, \quad (3.19)$$

whose definition is very similar to that of the surface BSDF (Eq. 3.15) but without the cosine term in the denominator. In contrast to the BSDF, however, by convention the directions ω_i and ω_o both point in the direction of light flow (see Figure 3.3). Unlike the BSDF, the phase function is an actual probability distribution, i.e. it integrates to one over the sphere, and also usually does not vary with position in the medium. *Isotropic* scattering is described by a constant phase function, $\rho_m \equiv 1/4\pi$, and various distributions have been proposed to model *anisotropic* scattering. One such distribution is the Henyey-Greenstein (HG) function, which is widely used in graphics:

$$\rho_{\text{HG}}(\omega_i, \omega_o) = \frac{1 - g^2}{4\pi [1 + g^2 - 2g(\omega_i \cdot \omega_o)]^{1.5}}. \quad (3.20)$$

The parameter g controls the anisotropy: $g = 0$ makes the distribution constant (isotropic), and when g approaches 1 or -1 the function converges to a forward or backward scattering delta distribution (see Figure 3.3). Note that the HG phase function is one-dimensional and circularly-symmetric, as it depends only on the cosine of the deflection angle between the incident and the outgoing directions.

3.5 Integral formulation of light transport

In the previous sections we described the basic radiometric quantities and principles of light emission, propagation and scattering. We now put all these components together to formulate the light transport problem as an integral over the equilibrium light distribution in the scene.

3.5.1 The measurement equation

The value of each pixel in a rendered image is the response of a corresponding virtual eye sensor. This response is given by the *measurement equation*:

$$I = \int_{\mathcal{M}} \int_{\mathcal{S}} W_e(\mathbf{x} \rightarrow \omega) L(\mathbf{x} \leftarrow \omega) |n_{\mathbf{x}} \cdot \omega| d\omega d\mathbf{x}, \quad (3.21)$$

which expresses the pixel value as an integral of incident radiance multiplied by sensitivity. Here, \mathcal{M} is the union of all surface points in the scene, but the importance emission function W_e effectively restricts the integration to the surface of the sensor. The incident radiance $L(\mathbf{x} \leftarrow \omega)$ at a sensor point is given by the surface rendering equation, described next.

3.5.2 The surface rendering equation

As discussed in Section 3.4, in a vacuum the incident radiance $L(\mathbf{x} \leftarrow \omega)$ at point \mathbf{x} from direction ω is equal to the outgoing radiance at another surface point \mathbf{y} in the opposite direction, $-\omega$. This equality can be expressed as:

$$L(\mathbf{x} \leftarrow \omega) = L(\mathbf{y} \rightarrow -\omega) = L(r(\mathbf{x}, \omega) \rightarrow -\omega), \quad (3.22)$$

where the point \mathbf{y} is given by $\mathbf{y} = r(\mathbf{x}, \omega)$, with $r(\mathbf{x}, \omega)$ being the *ray tracing operator* which returns the first visible point from \mathbf{x} in direction ω . Thus, to solve the measurement equation (Eq. 3.21), we need to compute the outgoing radiance from points visible from the eye in directions back to the eye.

30 Section 3.5: Integral formulation of light transport

The outgoing radiance at a scene point \mathbf{x} in direction ω_o is given by the *rendering equation* (RE) which was first formulated by Kajiya [66] and Immel et al. [54]:

$$L(\mathbf{x} \rightarrow \omega_o) = \underbrace{L_e(\mathbf{x} \rightarrow \omega_o)}_{\text{self-emitted}} + \underbrace{L_s(\mathbf{x} \rightarrow \omega_o)}_{\text{scattered}}. \quad (3.23)$$

This equation states that the outgoing radiance is the sum of the self-emitted radiance at \mathbf{x} and the scattered radiance incident from all directions around \mathbf{x} . The scattered radiance is given by:

$$\underbrace{L_s(\mathbf{x} \rightarrow \omega_o)}_{\text{outgoing}} = \int_S \rho_s(\mathbf{x}, \omega_i, \omega_o) \underbrace{L(\mathbf{x} \leftarrow \omega_i)}_{\text{incident}} |n_{\mathbf{x}} \cdot \omega_i| d\omega_i \quad (3.24a)$$

$$= \int_S \rho_s(\mathbf{x}, \omega_i, \omega_o) \underbrace{L(r(\mathbf{x}, \omega_i) \rightarrow -\omega_i)}_{\text{outgoing}} |n_{\mathbf{x}} \cdot \omega_i| d\omega_i \quad (3.24b)$$

$$= \int_{\mathcal{M}} \rho_s(\mathbf{x}, \omega_{\mathbf{x}\mathbf{y}}, \omega_o) \underbrace{L(\mathbf{y} \rightarrow \omega_{\mathbf{y}\mathbf{x}})}_{\text{outgoing}} G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) d\mathbf{y}. \quad (3.24c)$$

In Equation 3.24b we have used Equation 3.22 to express the incident radiance at one point as the outgoing radiance at another point. Equation 3.24c expresses the scattered radiance as an integral over the union of all scene surfaces (rather than as an integral over the sphere of incoming directions). The geometry term $G(\mathbf{x}, \mathbf{y})$ appears as a result of the change of the integration variable (see Equation 3.9). The visibility term $V(\mathbf{x}, \mathbf{y})$ accounts for the fact that the integral now considers all surface points in the scene and not only the ones visible from \mathbf{x} .

According to Equation 3.24, in order to compute the outgoing radiance at a point \mathbf{x} , we need to recursively compute the outgoing radiance at all surface points that can contribute energy to \mathbf{x} .

3.5.3 The volume rendering equation

In the general case where there are participating media in the scene, the incident radiance $L(\mathbf{x} \leftarrow \omega)$ is given by the *radiative transfer equation* (RTE) [8], also known as the *volume rendering equation*:

$$L(\mathbf{x} \leftarrow \omega) = \underbrace{T_r(\mathbf{x}, \mathbf{y})L(\mathbf{y} \rightarrow -\omega)}_{\text{attenuated surface radiance}} + \quad (3.25a)$$

$$\int_0^d \underbrace{T_r(\mathbf{x}, \mathbf{x}_t) \left[\sigma_a(\mathbf{x}_t)L_e(\mathbf{x}_t \rightarrow -\omega) + \sigma_s(\mathbf{x}_t)L_s(\mathbf{x}_t \rightarrow -\omega) \right]}_{\text{attenuated self-emitted and in-scattered medium radiance}} dt, \quad (3.25b)$$

where $\mathbf{y} = r(\mathbf{x}, \omega)$ is the first visible surface point from \mathbf{x} along ω , the distance to this point is $d = \|\mathbf{x} - \mathbf{y}\|$, and $\mathbf{x}_t = \mathbf{x} + t\omega$. This equation states that the radiance reaching \mathbf{x} from direction ω is equal to the attenuated outgoing radiance from \mathbf{y} , given by the RE (Eq. 3.23), plus the accumulated radiance contribution from any media intersected by the ray segment between \mathbf{x} and \mathbf{y} . The medium radiance is the integral of the attenuated self-emitted and in-scattered radiance along the ray segment. Finally, the in-scattered radiance at point \mathbf{x}_t in a medium is the integral of the incident radiance from all directions around the point, which is the volumetric equivalent of the surface scattering equation (Eq. 3.24):

$$L_s(\mathbf{x}_t \rightarrow -\omega) = \int_S \rho_m(\omega_i, \omega) L(\mathbf{x}_t \leftarrow \omega_i) d\omega_i. \quad (3.26)$$

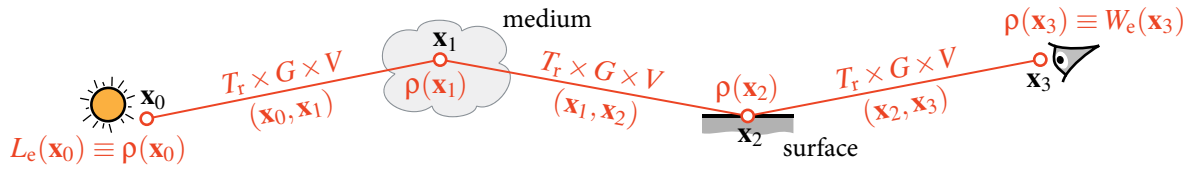


Figure 3.4: Illustration of a length-3 light transport path $\bar{\mathbf{x}} = \mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$, along with the terms of its measurement contribution (colored in orange).

3.5.4 Path integral formulation

The measurement equation (Eq. 3.21), together with the rendering equation (Eq. 3.23) and the radiative transfer equation (Eq. 3.25), provides a well-defined formulation of the global illumination problem as an integral equation over the equilibrium radiance distribution in a scene. Unfortunately, the recursive structure of this formulation gives only a local view of the illumination at a point in the scene and does not provide a flexible framework for developing a wide variety of practical solutions.

Various reformulations of the measurement equation have been proposed, such as adjoint equations using importance as a transport quantity as well as series expansions of these equations using linear operators [30, 142, Chap. 4]. These formulations provide more insight into the light transport problem and lay the ground for developing practical methods for computing the energy equilibrium by tracing particles from the eye or from the lights. However, such methods can simulate transport starting either only from the eye or only from the lights, and it is not obvious from these different formulations when to choose one approach over the others or how to combine different approaches.

To address these theoretical limitations, Veach reformulated the light transport problem as a pure integration problem [142, Chap. 8]. His *path integral* expresses each pixel measurement in a conceptually very simple form:

$$I = \int_{\Omega} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}). \quad (3.27)$$

This integral computes the energy contribution of every *light transport path* $\bar{\mathbf{x}}$ of arbitrary length to a particular pixel. Veach derived the above equation for light transport in vacuum by starting from the measurement equation (Eq. 3.21) and recursively expanding the rendering equation (Eq. 3.23) using the surface form of the scattering equation (Eq. 3.24c). This results in an infinite sum of integrals over the paths of all lengths, which Veach into a single integral over the union of all paths of any possible length Ω . This formulation can be analogously generalized to also handle participating media by expanding the radiative transfer equation (Eq. 3.25). Below we will only define the terms that appear in Equation 3.27 and refer the interested reader to prior works for the full derivation [142, 100, 58, 56].

A light transport path $\bar{\mathbf{x}}$ is defined as a tuple of vertices:

$$\bar{\mathbf{x}} = \mathbf{x}_0\mathbf{x}_1 \dots \mathbf{x}_k, \quad (3.28)$$

where each vertex can lie on a surface or in a medium. The length of a path, k , is the number of its (imaginary) edges, each connecting two consecutive vertices (see Figure 3.4). Ω is the space of all light transport paths of all lengths and $d\mu(\bar{\mathbf{x}})$ is a differential measure on this space, defined for a length- k path as:

$$d\mu(\bar{\mathbf{x}}) = d\mu(\mathbf{x}_0)d\mu(\mathbf{x}_1) \dots d\mu(\mathbf{x}_k), \quad (3.29)$$

32 Section 3.5: Integral formulation of light transport

where the differential measure of a vertex \mathbf{x} is

$$d\mu(\mathbf{x}) = \begin{cases} dA(\mathbf{x}) & \text{if } \mathbf{x} \text{ is on a surface,} \\ dV(\mathbf{x}) & \text{if } \mathbf{x} \text{ is in a medium.} \end{cases} \quad (3.30)$$

Above, $dA(\mathbf{x})$ and $dV(\mathbf{x})$ denote the differential area and volume measures, respectively.

The only thing left to define is the *measurement contribution function* $f(\bar{\mathbf{x}})$, which has the following form:

$$f(\bar{\mathbf{x}}) = \rho(\mathbf{x}_0) T(\mathbf{x}_0 \dots \mathbf{x}_k) \rho(\mathbf{x}_k), \quad (3.31)$$

where the *measurement throughput* T is defined as:

$$T(\mathbf{x}_0 \dots \mathbf{x}_k) = T_r(\mathbf{x}_0, \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) V(\mathbf{x}_0, \mathbf{x}_1) \prod_{i=1}^{k-1} \rho(\mathbf{x}_i) T_r(\mathbf{x}_i, \mathbf{x}_{i+1}) G(\mathbf{x}_i, \mathbf{x}_{i+1}) V(\mathbf{x}_i, \mathbf{x}_{i+1}). \quad (3.32)$$

The vertex scattering term $\rho(\mathbf{x}_i)$ above is defined as:

$$\rho(\mathbf{x}_i) = \begin{cases} L_e(\mathbf{x}_0 \rightarrow \omega_{\mathbf{x}_0 \mathbf{x}_1}) & \text{if } i = 0, \\ W_e(\mathbf{x}_k \rightarrow \omega_{\mathbf{x}_k \mathbf{x}_{k-1}}) & \text{if } i = k, \\ \rho_s(\mathbf{x}_i, \omega_{\mathbf{x}_{i-1} \mathbf{x}_i}, \omega_{\mathbf{x}_i \mathbf{x}_{i+1}}) & \text{if } \mathbf{x}_i \text{ is on a surface,} \\ \rho_m(\mathbf{x}_i, \omega_{\mathbf{x}_{i-1} \mathbf{x}_i}, \omega_{\mathbf{x}_i \mathbf{x}_{i+1}}) \sigma_s(\mathbf{x}_i) & \text{if } \mathbf{x}_i \text{ is in a medium.} \end{cases} \quad (3.33)$$

The definitions of the transmittance term $T_r(\mathbf{x}, \mathbf{y})$ and the visibility term $V(\mathbf{x}, \mathbf{y})$ are given by Equations 3.12 and 3.13 respectively. Finally, $G(\mathbf{x}, \mathbf{y})$ is a generalized geometry term that operates on both surface and medium points:

$$G(\mathbf{x}, \mathbf{y}) = \frac{D(\mathbf{x}, \omega_{\mathbf{x}\mathbf{y}}) D(\mathbf{y}, \omega_{\mathbf{y}\mathbf{x}})}{\|\mathbf{x} - \mathbf{y}\|^2}, \quad (3.34)$$

where the foreshortening term $D(\mathbf{x}, \omega)$ is defined as

$$D(\mathbf{x}, \omega) = \begin{cases} |n_{\mathbf{x}} \cdot \omega| & \text{if } \mathbf{x} \text{ is on a surface,} \\ 1 & \text{if } \mathbf{x} \text{ is in a medium.} \end{cases} \quad (3.35)$$

Figure 3.4 illustrates a length-3 path along with all the terms associated with its measurement contribution.

Equation 3.27 describes the rendering problem as a pure integration problem with a very simple structure. A single expression defines the value of each pixel measurement, as opposed to the RE and the RTE which describe the equilibrium radiance distribution via recursive integral equations. The path integral avoids adjoint formulations, intermediate quantities like radiance and importance, or having to choose between these alternatives. It provides a tidy and flexible foundation for the development of global illumination algorithms and enables the use of general-purpose Monte Carlo integration techniques as well as variance reduction techniques such as multiple importance sampling.

3.5.5 Path classification

It is often useful to classify light transport paths according to the type of scattering distribution at each vertex. We adopt the notation of Heckbert [49] which characterizes each interaction along a path from

a light source (L) to the eye (E) as either diffuse (D) or specular (S). We allow D to also denote any scattering distribution that is non-zero over a finite solid angle (e.g. diffuse or glossy), while S only denotes (delta) distributions defined over a set of directions with a zero measure.

With this notation, any path can be labeled with a string given by the regular expression $L(S|D)^*E$. For example, a length-3 diffuse path is denoted by $LDDE$. On the other hand, a caustic directly seen from the eye is produced by $L(S|D)^*SDE$ paths, and a caustic seen through reflection or refraction is produced by $L(S|D)^*SDS^+E$ paths. Veach extended this notation and used D and S to also describe the emission characteristics of light sources and sensors [142, Sec. 8.3.2], but in this thesis we will adhere to Heckbert's original notation.

Monte Carlo Solutions for the Path Integral

4

The goal of every physically-based rendering algorithm is to solve the measurement equation (Eq. 3.21) for each image pixel or, equivalently, its path integral formulation (Eq. 3.27) which expresses the measurement as the sum of the differential contributions of all possible light transport paths in the scene. In this chapter, we describe the application of Monte Carlo integration to this light transport problem. We start by formulating a few generic estimators for the path integral and introduce the notion of a *path sampling technique*. We then review some existing methods for rendering scenes containing surfaces and participating media and express these methods as different path sampling techniques within the path integral framework. This will lay the ground for the developments in the following chapters where we will devise new path sampling and variance reduction techniques to address some of the limitations of those methods.

4.1 Path integral estimators

Following the standard Monte Carlo definitions (Sec. 2.2), an unbiased primary estimator for the light transport integral in Equation 3.27 has the form

$$\hat{I} = \frac{f(\bar{\mathbf{x}})}{p(\bar{\mathbf{x}})}, \quad (4.1)$$

with $E[\hat{I}] = I$. To obtain an actual pixel estimate we need to first sample a random path $\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$ with a random length k and then evaluate its measurement contribution $f(\bar{\mathbf{x}})$ as well as its pdf which is the joint distribution of its vertices: $p(\bar{\mathbf{x}}) = p(\mathbf{x}_0, \dots, \mathbf{x}_k)$. The path contribution is always given by Equation 3.31, regardless of how $\bar{\mathbf{x}}$ is sampled. Our task therefore reduces to: (1) finding a suitable *path sampling technique* for constructing a random path $\bar{\mathbf{x}}$ and (2) deriving the path probability density function associated with this technique. Each individual path sampling technique yields a different path pdf and thus a different estimator of the form given in Equation 4.1 above.

A secondary estimator for Equation 3.27 can be constructed by averaging the primary estimators of N independent samples, i.e. paths:

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^N \frac{f(\bar{\mathbf{x}}_i)}{p(\bar{\mathbf{x}}_i)}. \quad (4.2)$$

The path integral framework makes a clear distinction between the energy contribution of a path and its sampling probability density, which enables the use of multiple importance sampling (MIS). A multi-

sample MIS estimator for Equation 3.27 has the form

$$\hat{f}^{\text{MIS}} = \sum_{i=1}^n \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(\bar{\mathbf{x}}_{i,j}) \frac{f(\bar{\mathbf{x}}_{i,j})}{p_i(\bar{\mathbf{x}}_{i,j})}, \quad (4.3)$$

which combines the primary estimators of n different path sampling techniques, where n_i denotes the number of paths $\bar{\mathbf{x}}_{i,j}$ sampled from each technique with pdf p_i , weighted by the heuristic w_i .

4.2 Path sampling techniques

Using any of the above estimators, the task of an unbiased Monte Carlo rendering algorithm boils down to sampling light transport paths. Importance sampling theory postulates that the joint density of the path vertices should ideally be proportional to the measurement contribution function $f(\bar{\mathbf{x}})$. Unfortunately, sampling from such a joint distribution is almost never possible in practice since the behavior of f is strongly scene-dependent and generally unpredictable. It can contain discontinuities in the visibility term V , singularities in the geometry term G , as well as arbitrary variations in the transmittance term T_r and the scattering term ρ (see Section 3.5.4 for the definitions of these terms). This makes analytic marginalization and inversion intractable. In fact, the complex shape of the contribution function is the very reason for choosing Monte Carlo integration in the first place. Furthermore, the high – or rather, infinite – dimensionality of the sampling space makes numerical inversion of the entire integral impractical and finding good analytical pdf approximations extremely challenging.

The difficulty with global, or joint, path sampling comes from the required coordination between the sampling distributions for the individual vertices. The complexity of the problem can be greatly reduced by opting for a local sampling approach instead. Most existing rendering algorithms employ techniques that sample vertices incrementally, one by one, by tracing random trajectories throughout the scene. These so-called *random walks* use local sampling decisions that simulate the probabilistic propagation and scattering behavior of real-world photons. Some methods trace random walks from the light sources, others from the eye, and some from both directions. In the remainder of this section we describe the most commonly used path sampling techniques based on random walks.

4.2.1 Random walks

A random walk is a random process that consists of a sequence of discrete steps and is usually “memoryless”: the next state solely depends on the current state and not on the entire history of the process. In computational physics, random walks are used to trace the trajectories of molecules traveling in a liquid or a gas or to study the motions and interactions of neutrons with materials. Random walks that simulate the stochastic physical behavior of particles are called *analog walks* in physics literature [129, 90].

Random walks were introduced to computer graphics by Kajiya [66] in the same work that formulated the rendering equation (Eq. 3.23). Most existing Monte Carlo rendering algorithms use random walks to construct light transport paths incrementally, choosing the sampling distribution for the next vertex based on information locally available at the current vertex. A walk typically starts by sampling the initial path vertex either on a light source or on the eye lens, after which the path is extended with new vertices via a sequence of propagation and scattering sampling decisions. Eventually the process terminates – either deterministically or randomly. The walk is analog if its sampling decisions exactly model the physical transport of light in the scene, as detailed below.

Path notation

Recall from Section 3.5.4 that a length- k path is denoted by $\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$, where vertex \mathbf{x}_0 is on a light source and \mathbf{x}_k is on the eye lens. We will use also a different, redundant path notation to explicitly indicate the direction of the random walk that sampled the path. We will use $\bar{\mathbf{y}} = \mathbf{y}_0 \dots \mathbf{y}_k$ to denote paths sampled starting from a light source and $\bar{\mathbf{z}} = \mathbf{z}_k \dots \mathbf{z}_0$ for paths sampled from the eye. This notation conveniently indexes the vertices in the order of their generation – vertex $\mathbf{x}_0 \equiv \mathbf{y}_0 \equiv \mathbf{z}_k$ is on a light source and $\mathbf{x}_k \equiv \mathbf{y}_k \equiv \mathbf{z}_0$ is on the eye lens. The difference in the indexing is important to keep in mind when evaluating the terms of the measurement contribution function (Eq. 3.31) which are defined unambiguously in the \mathbf{x} -notation.

In the remainder of this subsection we adhere to the \mathbf{x} -notation, but thanks to the symmetric definitions of the scattering term¹ and the path measurement throughput (Sec. 3.5.4), all equations below remain valid when \mathbf{x} is replaced by \mathbf{y} or \mathbf{z} .

Subpaths and sampling throughput

As we trace a random walk, we can incrementally evaluate the measurement contribution of the sampled path. To this end, we first define the measurement contribution of a *subpath* $\mathbf{x}_0 \dots \mathbf{x}_i$ of $\bar{\mathbf{x}}$:

$$f_i(\bar{\mathbf{x}}) = \begin{cases} 1 & \text{if } i = 0, \\ f_{i-1}(\bar{\mathbf{x}}) \rho(\mathbf{x}_{i-1}) T_r(\mathbf{x}_{i-1}, \mathbf{x}_i) G(\mathbf{x}_{i-1}, \mathbf{x}_i) V(\mathbf{x}_{i-1}, \mathbf{x}_i) & \text{otherwise,} \end{cases} \quad (4.4)$$

where the definitions of the four contribution terms on the right-hand side were given in Section 3.5.4. The corresponding subpath pdf is²

$$p_i(\bar{\mathbf{x}}) = \begin{cases} p(\mathbf{x}_0) & \text{if } i = 0, \\ p_{i-1}(\bar{\mathbf{x}}) p(\mathbf{x}_i | \mathbf{x}_0, \dots, \mathbf{x}_{i-1}) & \text{otherwise.} \end{cases} \quad (4.5)$$

In practice, the pdf for each vertex \mathbf{x}_j above is typically conditioned on at most the one or two preceding subpath vertices.

The *sampling throughput*, a.k.a. “energy”, C_i associated with each path vertex \mathbf{x}_i is defined as the measurement contribution of the subpath $\mathbf{x}_0 \dots \mathbf{x}_i$ divided by its pdf. The sampling throughput is essentially a partial evaluation of a corresponding primary estimator (Eq. 4.1):

$$C_i(\bar{\mathbf{x}}) = \frac{f_i(\bar{\mathbf{x}})}{p_i(\bar{\mathbf{x}})} = \begin{cases} \frac{1}{p(\mathbf{x}_0)} & \text{if } i = 0, \\ C_{i-1}(\bar{\mathbf{x}}) \frac{\rho(\mathbf{x}_{i-1}) T_r(\mathbf{x}_{i-1}, \mathbf{x}_i) G(\mathbf{x}_{i-1}, \mathbf{x}_i) V(\mathbf{x}_{i-1}, \mathbf{x}_i)}{p(\mathbf{x}_i | \mathbf{x}_0, \dots, \mathbf{x}_{i-1})} & \text{otherwise.} \end{cases} \quad (4.6)$$

For a random walk to be analog, the vertices \mathbf{x}_i should be sampled in succession such that the sampling throughput remains constant, just like the power of a real-world photon does not change as it travels in the environment and interacts with matter. This can be achieved by importance sampling the product of the four measurement contribution terms in Equation 4.4, as we describe next. The process of sampling a random walk is illustrated graphically in Figure 4.1.

¹Note that care must be taken to ensure that the scattering term $\rho(\mathbf{x})$ is symmetric for refractive BSDFs [142, Chap. 7].

²Strictly speaking, we should denote the pdfs of \mathbf{x}_0 and \mathbf{x}_i by $p_{\mathbf{x}_0}(\mathbf{x}_0)$ and $p_{\mathbf{x}_i | \mathbf{x}_0 \dots \mathbf{x}_{i-1}}(\mathbf{x}_i | \mathbf{x}_0 \dots \mathbf{x}_{i-1})$, respectively, to make it clear that the distribution for every vertex is generally different. For the sake of keeping the notation succinct, we will omit the pdf subscripts, and sometimes also the conditional arguments, since usually these are unambiguously inferred by the context.

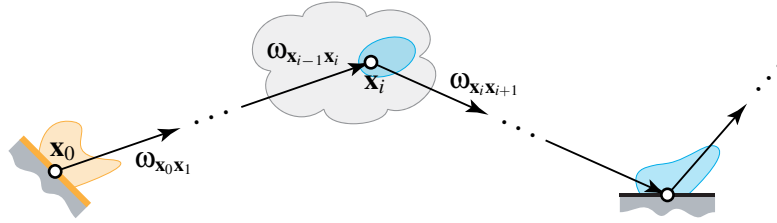


Figure 4.1: A random walk typically begins by sampling a vertex on a light source (as shown here) or the eye lens, along with an outgoing direction. The path is then incrementally extended with new vertices by locally importance sampling their associated terms in the path measurement contribution.

Emission

When a random walk starts on the surface or in the medium of a light source, the first path vertex \mathbf{x}_0 is sampled with an (unconditional) area/volume pdf $p(\mathbf{x}_0)$ proportional to the total emitted power (i.e. radiosity) at that point. Then, the direction $\omega_{\mathbf{x}_0 \mathbf{x}_1}$ toward the next (still unknown) vertex \mathbf{x}_1 is sampled, with a pdf $p(\omega_{\mathbf{x}_0 \mathbf{x}_1} | \mathbf{x}_0)$ proportional to the directional emission $\rho(\mathbf{x}_0 \rightarrow \omega_{\mathbf{x}_0 \mathbf{x}_1})$. With these sampling distributions, the density of the outgoing rays from a light source is proportional to the density of the photons emitted by it. The sampling decisions are analogous when starting on the eye lens.

Propagation

Given a ray $(\mathbf{x}_{i-1}, \omega_{\mathbf{x}_{i-1} \mathbf{x}_i})$, the next vertex on the path \mathbf{x}_i is determined by sampling a propagation distance t_i along the ray. In media, the distance pdf is typically chosen to be proportional to the transmittance, $T_r(\mathbf{x}_{i-1}, \mathbf{x}_i)$ (Eq. 3.12) [122, 87]. For homogeneous media, i.e. with a constant extinction coefficient σ_t , a closed-form transmittance sampling routine exists:

$$t_i = -\frac{\ln \xi}{\sigma_t}, \quad (4.7)$$

where $\xi \in [0; 1)$ is a canonical uniform random number. This expression is obtained by normalizing Equation 3.12 and inverting the CDF of the resulting pdf. In heterogeneous media, $\sigma_t(\mathbf{x})$ can be an arbitrary function, which makes importance sampling the transmittance much less straightforward. Commonly used methods are *ray marching*, which steps along the ray to numerically invert the CDF of T_r [64, 100, 80], and *Woodcock tracking*, which is based on rejection sampling [159, 114, 161, 136].

To determine the final location of \mathbf{x}_i , in addition to sampling a random distance in the medium pierced by the given ray, we need to find the distance to the closest surface intersection using ray tracing (see Figure 4.2). The propagation distance t_i is then set to the smaller of the two distances, and its pdf is

$$p(t_i | \mathbf{x}_{i-1}, \omega_{\mathbf{x}_{i-1} \mathbf{x}_i}) = \begin{cases} \sigma_t T_r(\mathbf{x}_{i-1}, \mathbf{x}_i) & \text{if } \mathbf{x}_i \text{ is in a medium} \\ T_r(\mathbf{x}_{i-1}, \mathbf{x}_i) & \text{if } \mathbf{x}_i \text{ is on a surface.} \end{cases} \quad (4.8)$$

With this pdf, the density of sampled points along the ray is proportional to the density of particle interactions with the medium. Note that if the ray does not pierce any media, then $T_r(\mathbf{x}_{i-1}, \mathbf{x}_i) = 1$, and thus the closest surface intersection is chosen deterministically with probability 1. The full conditional

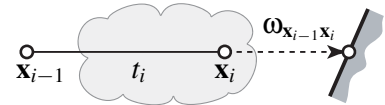


Figure 4.2: Propagation distance sampling.

pdf of \mathbf{x}_i is

$$p(\mathbf{x}_i | \omega_{\mathbf{x}_{i-2}\mathbf{x}_{i-1}}, \mathbf{x}_{i-1}, \omega_{\mathbf{x}_{i-1}\mathbf{x}_i}) = p(\omega_{\mathbf{x}_{i-1}\mathbf{x}_i} | \mathbf{x}_{i-1}, \omega_{\mathbf{x}_{i-2}\mathbf{x}_{i-1}}) p(t_i | \mathbf{x}_{i-1}, \omega_{\mathbf{x}_{i-1}\mathbf{x}_i}) V(\mathbf{x}_{i-1}, \mathbf{x}_i) \frac{D(\mathbf{x}_i, \omega_{\mathbf{x}_{i-1}\mathbf{x}_i})}{\|\mathbf{x}_{i-1} - \mathbf{x}_i\|^2}, \quad (4.9)$$

where the pdf of the ray direction $\omega_{\mathbf{x}_{i-1}\mathbf{x}_i}$, given by emission or scattering sampling (described below), depends on the incoming direction $\omega_{\mathbf{x}_{i-2}\mathbf{x}_{i-1}}$ at vertex \mathbf{x}_{i-1} , unless $\mathbf{x}_{i-1} \equiv \mathbf{x}_0$ is on a light source. Note that this procedure implicitly importance samples the visibility term, since the sampled vertex \mathbf{x}_i is always visible from \mathbf{x}_{i-1} . The last term in the above equation converts the pdf of \mathbf{x}_i to the volume/area measure [142, Sec. 8.2.2.2] (see also the relation in Equation 3.9).

Finding the closest surface point along a ray involves testing all objects in the scene for an intersection. A vast body of research has been devoted to accelerating this computationally intensive ray tracing operation. We refer the interested reader to recent literature for surveys on this topic [45, 109, 110, 147].

Scattering

Given a path vertex \mathbf{x}_i with an incident direction $\omega_{\mathbf{x}_{i-1}\mathbf{x}_i}$, a random direction $\omega_{\mathbf{x}_i\mathbf{x}_{i+1}}$ from \mathbf{x}_i toward the next path vertex \mathbf{x}_{i+1} is sampled with a pdf that should be ideally proportional to the product of the scattering and foreshortening terms at \mathbf{x}_i :

$$p(\omega_{\mathbf{x}_i\mathbf{x}_{i+1}} | \mathbf{x}_i, \omega_{\mathbf{x}_{i-1}\mathbf{x}_i}) \propto \rho(\mathbf{x}_i, \omega_{\mathbf{x}_{i-1}\mathbf{x}_i}, \omega_{\mathbf{x}_i\mathbf{x}_{i+1}}) D(\mathbf{x}_i, \omega_{\mathbf{x}_i\mathbf{x}_{i+1}}), \quad (4.10)$$

which were defined in Equations 3.33 and 3.35 respectively. With such a pdf, the density of sampled directions is proportional to the directional density of particles scattered off the point of collision \mathbf{x}_i .

When the scattering term ρ at \mathbf{x}_i is constant, the pdf of $\omega_{\mathbf{x}_i\mathbf{x}_{i+1}}$ need only be proportional to the foreshortening term D . This corresponds to the case where \mathbf{x}_i is on a diffuse surface or in an isotropically scattering medium. Simple analytic sampling routines exist for the resulting cosine-weighted hemisphere and uniform sphere distributions, respectively [28].

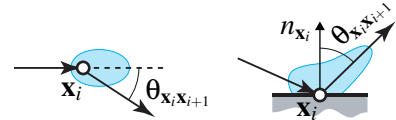


Figure 4.3: Scattering direction sampling.

For the Henyey-Greenstein medium scattering distribution, a closed-form direction sampling routine for the deflection angle exists (see Figure 4.3 left):

$$\cos \theta_{\mathbf{x}_i\mathbf{x}_{i+1}} = \frac{1}{2g} \left(1 + g^2 - \left(\frac{1 - g^2}{1 - g + 2g\xi} \right)^2 \right). \quad (4.11)$$

Analytic importance sampling routines also exist for simple surface scattering distributions (BSDFs) [109, Sec. 14.5]. Unfortunately, more general BSDFs cannot be exactly importance sampled, most notably microfacet models that are expressed as products of non-trivial terms [16]. Moreover, sampling the product with the foreshortening (i.e. cosine) term is generally difficult even with simple BSDFs. Therefore, random walks in most typical scenes are not strictly analog in practice.

One special case are perfectly specular BSDFs, which can be easily sampled as they only scatter light in a single, well-defined direction with probability one [109, Sec. 8.2]. If a specular direction is sampled at vertex \mathbf{x}_i , the throughput C_{i+1} at vertex \mathbf{x}_{i+1} is computed by simply multiplying C_i by the specular BSDF albedo (which for a perfect mirror is one) and the transmittance $T_r(\mathbf{x}_i, \mathbf{x}_{i+1})$.

Termination

In order to maintain a finite computation time, every random walk must eventually terminate. This trivially happens upon interaction with a non-scattering surface or medium, or when a ray escapes the scene. Otherwise, the walk can be deterministically terminated after a predefined number of scattering events. This however may introduce bias in the final estimator since some longer light transport paths may never be sampled. Furthermore, this approach can be inefficient as it can waste a lot of computational effort on sampling paths that end up having very small measurement contributions.

To address these issues, Arvo and Kirk [2] proposed to stochastically terminate random walks using Russian roulette (Sec. 2.5.3). First, a *continuation probability* $P_i^{\text{cont}} = \alpha_i$ is defined at each vertex \mathbf{x}_i , where α_i is the surface (Eq. 3.16) or medium (Eq. 3.18) scattering albedo. A canonical random number $\xi \in [0; 1)$ is then drawn and the walk is continued if $\xi < P_i^{\text{cont}}$, multiplying the pdf of the next path vertex \mathbf{x}_{i+1} by P_i^{cont} . Otherwise the walk is terminated, with probability $P_i^{\text{term}} = 1 - P_i^{\text{cont}}$. This termination mechanism makes the expected length of a random walk the same as the average length of a path traveled by a real-world particle before absorption, which ensures that the walk remains analog.

4.2.2 Unidirectional sampling

Having sampled a path $\bar{\mathbf{z}}$ of length k with a random walk from the eye, we can readily obtain an unbiased primary pixel estimator for this *unidirectional sampling* (US) technique:

$$\hat{I}^{\text{US}}(\bar{\mathbf{z}}) = \frac{f(\bar{\mathbf{z}})}{p(\bar{\mathbf{z}})} = \frac{f_k(\bar{\mathbf{z}})\rho(\mathbf{z}_k)}{p_k(\bar{\mathbf{z}})P_k^{\text{term}}} = \frac{C_k(\bar{\mathbf{z}})\rho(\mathbf{z}_k)}{P_k^{\text{term}}}. \quad (4.12)$$

Evaluating this estimator only requires the additional evaluation of the scattering term at the last path vertex \mathbf{z}_k . A more efficient approach is to evaluate the emission at every path vertex:

$$\hat{I}^{\text{US}}(\bar{\mathbf{z}}) = \sum_{i=1}^k C_i(\bar{\mathbf{z}})\rho(\mathbf{z}_i). \quad (4.13)$$

This approach effectively constructs a separate primary estimator for the integral over every path length i and sums them up all. Note that the above sum generally runs to infinity, but for any randomly chosen path length k all estimates $i > k$ are zero due to the Russian roulette termination.

The resulting algorithm, illustrated in Figure 4.4a, is known as *unidirectional path tracing*. Figure 4.6a shows an image rendered using the estimator in Equation 4.13. Even though this simple unbiased method can sample all lighting effects in the scene, it suffers from high variance when the light source is small. Since the emission term at the last path vertex is not importance sampled, the estimator has a non-zero value only when the random walk hits a light source, which often occurs with a very low probability. The adjoint method, which traces paths from the light sources, can be even less efficient, since the eye lens is usually much smaller than the lights, and randomly hitting it has a probability close to zero (and equal to zero when a pinhole camera with an infinitesimally small aperture is used).

4.2.3 Vertex connection

In order to obtain a lower-variance pixel estimator when both the light source and the eye lens have small areas, it is crucial to importance sample both the radiance and importance emissions. This can

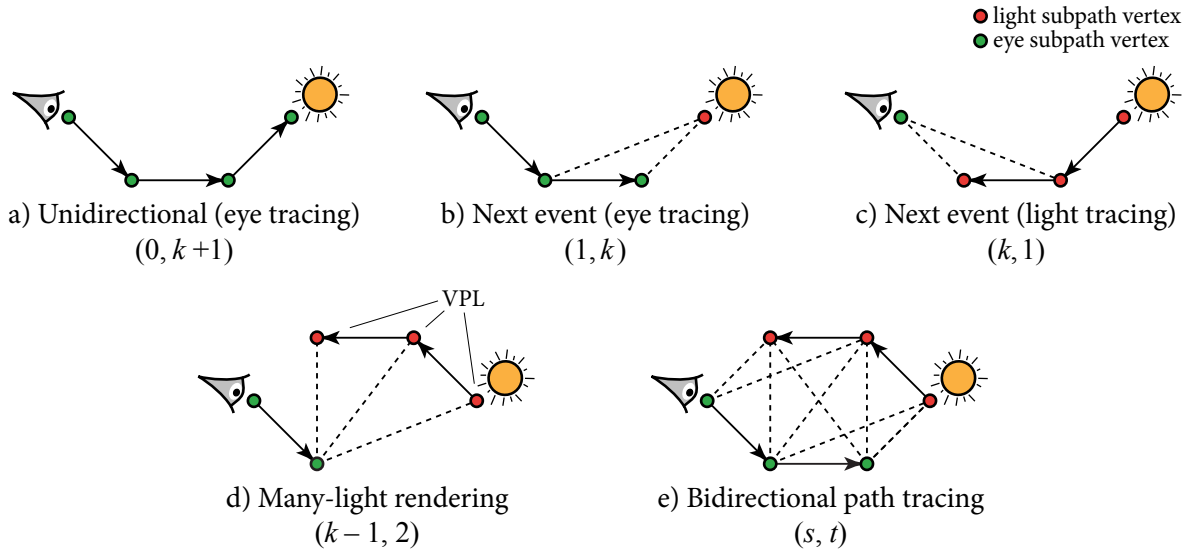


Figure 4.4: Schematic illustration of the path sampling techniques employed by various methods. Bidirectional path tracing combines a wide variety of techniques via multiple importance sampling (MIS).

be achieved by tracing independent random walks from both ends and then connecting the resulting subpaths to complete a full path. By varying the lengths of the light and eye subpaths many different light transport estimators can be constructed.

Assume we have traced a light subpath \bar{y} with s vertices and an eye subpath \bar{z} with t vertices. We can connect their endpoints, \mathbf{y}_{s-1} and \mathbf{z}_{t-1} , to obtain a full path $\bar{\mathbf{x}}_{s,t} = \mathbf{y}_0 \dots \mathbf{y}_{s-1} \mathbf{z}_{t-1} \dots \mathbf{z}_0$ of length $k = s + t - 1$. We follow Veach [142] and denote this *vertex connection* path sampling technique by (s, t) and illustrate it in Figure 4.5. Its corresponding primary pixel estimator reads

$$\hat{I}_{s,t}^{\text{VC}}(\bar{\mathbf{x}}_{s,t}) = \frac{f(\bar{\mathbf{x}}_{s,t})}{p_{s,t}(\bar{\mathbf{x}}_{s,t})} = \frac{f_{s-1}(\bar{y}) C_{s-1,t-1}(\bar{y}, \bar{z}) f_{t-1}(\bar{z})}{p_{s-1}(\bar{y}) p_{t-1}(\bar{z})} \quad (4.14a)$$

$$= C_{s-1}(\bar{y}) C_{s-1,t-1}(\bar{y}, \bar{z}) C_{t-1}(\bar{z}), \quad (4.14b)$$

where the *connection throughput* $C_{u,v}(\bar{y}, \bar{z})$ is defined as

$$C_{u,v}(\bar{y}, \bar{z}) = \rho(\mathbf{y}_u) T_r(\mathbf{y}_u, \mathbf{z}_v) V(\mathbf{y}_u, \mathbf{z}_v) G(\mathbf{y}_u, \mathbf{z}_v) \rho(\mathbf{z}_v). \quad (4.15)$$

Since the subpaths \bar{y} and \bar{z} are sampled independently, the pdf of the full path is simply the product of the subpath pdfs:

$$p_{s,t}(\bar{\mathbf{x}}) = p_{s-1}(\bar{y}) p_{t-1}(\bar{z}). \quad (4.16)$$

As a consequence, this technique does not importance sample any of the terms in the connection throughput $C_{s-1,t-1}(\bar{y}, \bar{z})$. Furthermore, when the BSDF at \mathbf{y}_{s-1} or \mathbf{z}_{t-1} is a delta distribution, the connection throughput, and thus the whole estimator, is zero. Vertex connection can therefore only be used to connect two non-specular vertices, i.e. to construct *DD* path segments.

Finally, it is worth noting that this so-called “connection” is only conceptual and also deterministic – the full path is constructed by simply concatenating the vertex tuples of the light and eye subpaths and the connection throughput is evaluated along the straight line, or edge, connecting their endpoints.

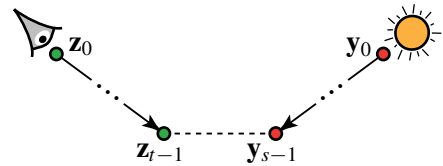


Figure 4.5: Vertex connection.

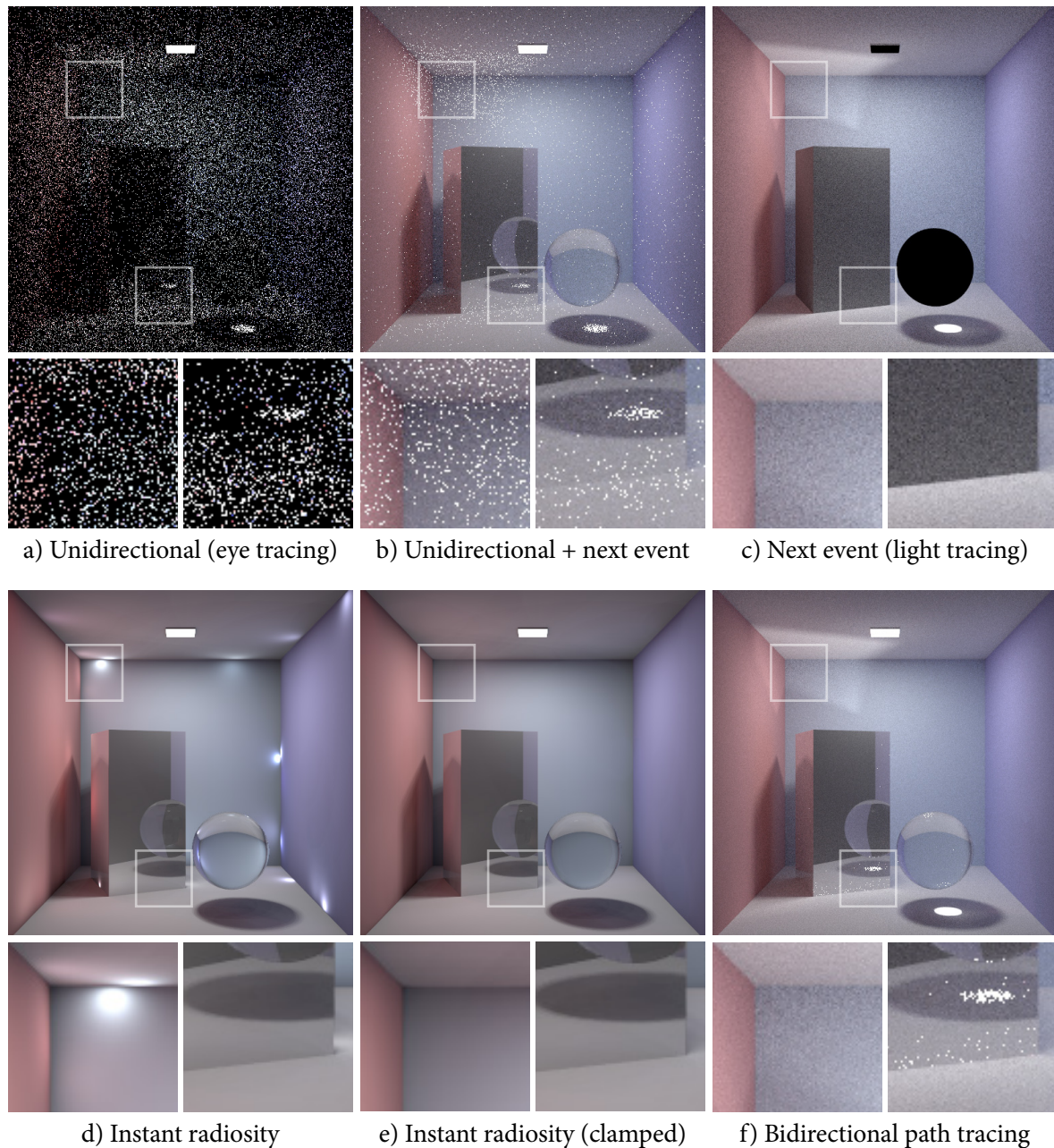


Figure 4.6: A comparison of different rendering methods with 40 paths per pixel. *a) Unidirectional sampling performs poorly on this scene, as it relies on randomly hitting the small light source. b) Adding next event estimation significantly improves efficiency, but caustic paths are still sampled unidirectionally. c) Light tracing is much more efficient for caustics, however it cannot render perfectly specular materials since specular path vertices cannot be connected to the eye. d) By correlating the paths constructed through different image pixels, instant radiosity produces a smooth image, which however lacks caustics and suffers from singularities at geometric corners. e) Clamping the virtual point light (VPL) contributions removes the singularities at the cost of darkening the illumination around corners. f) Bidirectional path tracing is much more robust thanks to the multiple importance sampling (MIS) combination of many techniques; however it still relies on unidirectional sampling to render caustics seen through specular reflection or refraction. Note that all images (except for clamped instant radiosity) have the same average brightness, but in some of them the energy is concentrated in a small number of very bright pixels.*

Next event estimation

With the (s, t) vertex connection technique notation, unidirectional sampling from a light source and from the eye correspond to the special vertex connection cases $(k + 1, 0)$ and $(0, k + 1)$, respectively, where the eye and light subpaths have zero vertices. Another special case of vertex connection are the $(k, 1)$ and $(1, k)$ techniques, corresponding to eye and light subpaths with one vertex, respectively. These techniques are often referred to as *next event estimation*. The unidirectional path sampling method from Section 4.2.2 can be significantly improved by creating a connection between every path vertex and a randomly sampled vertex on a light source (or on the eye lens, respectively). Next event estimation was first used in graphics by Kajiya [66] who traced paths from the eye. Most of today's production rendering engines are based on unidirectional path sampling from the eye, combining the $(0, k + 1)$ and $(1, k)$ techniques via multiple importance sampling (MIS).

The next event estimation techniques are illustrated in Figure 4.4b and Figure 4.4c. When the light source is small, the $(1, k)$ techniques sample most paths with much higher probability than the unidirectional $(0, k + 1)$ techniques. The improved efficiency can be seen in Figure 4.6b, where the image has been rendered using an MIS combination of unidirectional sampling and next event estimation. Unfortunately, that image still suffers from noise due to the inefficient handling of $LS(S|D)^*E$ paths which cannot be sampled by the $(1, k)$ techniques. These include caustic paths and are sampled unidirectionally, often with very low probability.

Figure 4.6c shows an image rendered with an adjoint next event estimation method that samples paths from the light sources using $(k, 1)$ techniques. The algorithm, called *light tracing* and proposed by Dutré et al. [29], can sample some caustic paths ($L(S|D)^*SDE$) more efficiently. Unfortunately, $(k, 1)$ vertex connections cannot render specular objects directly seen from the eye (which correspond to $L(S|D)^*SE$ paths).

In the following sections we review some more advanced global illumination rendering methods that make use of the unidirectional sampling and vertex connection techniques we described above.

4.3 Many-light rendering

An interesting class of bidirectional light transport simulation methods, called *many-light methods*, was pioneered by Keller's instant radiosity algorithm [73]. Instant radiosity runs in two stages. In the first stage, a number of subpaths are sampled from the light sources and stored. In the second, final rendering stage, a secondary estimator is constructed for each pixel by tracing an eye subpath and connecting its first non-specular vertex to the vertices of all light subpaths. This is illustrated in Figure 4.4d. The light subpath vertices can be viewed as *virtual point lights* (VPLs) that "illuminate" the objects seen from the eye. The first stage distributes these VPLs in the scene, which effectively reduces the global illumination problem to the problem of computing direct illumination from many point lights in the second stage. The conceptual simplicity and the ease of implementation of this method (also on graphics hardware) have facilitated its adoption in both interactive and offline applications. Below we formalize instant radiosity's pixel estimator and discuss the limitations of this approach.

4.3.1 The many-light pixel estimator

We now express many-light rendering as Monte Carlo estimation of the path integral (Eq. 3.27). Given an eye subpath $\bar{\mathbf{z}} = \mathbf{z}_0 \dots \mathbf{z}_{t-1}$ with t vertices, only the last of which is non-specular (i.e. the subpath terminates at the first non-specular surface), a many-light method evaluates the secondary estimator

$$\hat{f}^{\text{ML}} = \frac{1}{N} \sum_{i=1}^N \sum_{j=0}^{s_i-1} \hat{f}_{j,t}^{\text{VC}}(\mathbf{y}_{i,0} \dots \mathbf{y}_{i,j} \mathbf{z}_{t-1} \dots \mathbf{z}_0). \quad (4.17)$$

This estimator averages the primary vertex connection estimators (Eq. 4.14) constructed by connecting the eye subpath vertex \mathbf{z}_{t-1} to every vertex $\mathbf{y}_{i,j}$ of N light subpaths $\bar{\mathbf{y}}_i = \mathbf{y}_{i,0} \dots \mathbf{y}_{i,s_i-1}$ with s_i vertices each. In the absence of visible specular surfaces, the eye subpath has $t = 2$ vertices.

We can merge the two sums in Equation 4.17 to rewrite the pixel estimator in a more common many-light form which considers each light subpath vertex as a (virtual) light source:

$$\hat{f}^{\text{ML}} = \sum_{k=1}^M \frac{1}{N} \hat{f}_{j_k,t}^{\text{VC}}(\mathbf{y}_{i_k,0} \dots \mathbf{y}_{i_k,j_k} \mathbf{z}_{t-1} \dots \mathbf{z}_0) \quad (4.18a)$$

$$= \sum_{k=1}^M \underbrace{\frac{1}{N} C_{j_k}(\bar{\mathbf{y}}_{i_k})}_{\text{VPL energy}} \underbrace{C_{j_k,t-1}(\bar{\mathbf{y}}_{i_k}, \bar{\mathbf{z}})}_{\text{connection throughput}} \underbrace{C_{t-1}(\bar{\mathbf{z}})}_{\text{eye subpath importance}} = \sum_{k=1}^M h(\mathbf{x}, \mathbf{v}_k), \quad (4.18b)$$

which computes the contributions $h(\mathbf{v}_k, \mathbf{x})$ of $M = \sum_{i=1}^N s_i$ virtual point lights (VPLs). On the right-hand side we have used a compact notation to denote the VPL vertices by $\mathbf{v}_k \equiv \mathbf{y}_{i_k,j_k}$ and the eye subpath vertex by $\mathbf{x} \equiv \mathbf{z}_{t-1}$. Note that in the two equations above, the eye subpath $\bar{\mathbf{z}}$ is different for each pixel estimator \hat{f}^{ML} , whereas the set of light subpaths $\{\bar{\mathbf{y}}_i\}_{i=1}^N$ is the same for all pixel estimators.

4.3.2 Discussion

A consequence of correlated path sampling, e.g. reusing the same set of VPLs for all pixels, is that variance manifests itself as structured low-frequency noise in the image, which can sometimes be less objectionable than the high-frequency noise produced by uncorrelated sampling. Nevertheless, many VPLs may need to be sampled in order to reduce this noise, which can significantly increase the computational cost of the second (rendering) stage. In Chapters 5 and 6 we will present techniques to reduce this cost by only evaluating the contributions of a few carefully selected VPLs.

One shortcoming of exclusively using vertex connections in many-light rendering is related to the fact that the geometry term in the path contribution diverges as the distance between two connected subpath vertices approaches zero (Eq. 3.34). Since the geometry term in the connection throughput is not importance sampled, the variance of the primary vertex connection estimator explodes to infinity when the subpath endpoints are very close to each other (Fig. 4.7). The visual effect is illustrated in Figure 4.6d, where geometric corners are plagued by high-intensity spots in the rendered image. The common approach to remedy this problem is to clamp the distance $d = \|\mathbf{x} - \mathbf{v}_k\|$ between the two vertices to some minimum value, $d^* = \max(d_{\text{min}}, d)$, and evaluate the geometry term with d^* . As shown in Figure 4.6e, this clamping produces a smooth image but introduces bias which translates to slightly underestimated illumination

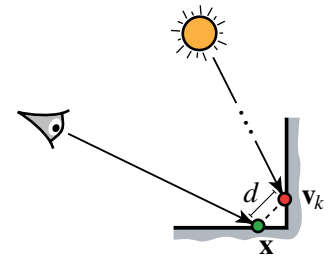


Figure 4.7: The vertex connection estimator diverges around corners.

around geometric corners. Instead of clamping, Kollig and Keller [78] completely discard short connections and then continue the eye subpath to “gather” the unaccounted for close-range illumination. This method effectively combines different vertex connection techniques with varying subpath lengths, making sure that each full path is sampled by a technique that has a long enough connection segment.

To render specular objects using many-light methods, every subpath from the eye is extended until the first diffuse object is encountered. However, because full paths are always constructed via connections, only the $L(S|D)^*DDS^*E$ subset of the full light transport in the scene can be captured. As a result, many-light methods cannot render caustics from specular objects, as seen in Figure 4.6. And while caustics from glossy objects can technically be captured by these methods, they are generally inefficient for such effects. This is because the pixel estimator suffers from high variance when high-frequency BSDFs are involved in the vertex connection throughput which is not importance sampled (Eq. 4.15).

4.4 Bidirectional path tracing

Path tracing and many-light methods sample every light transport path using one, pre-determined vertex connection technique. However, each different (s, t) technique can efficiently sample only a certain kind of paths and no single technique importance samples all terms in the measurement contribution function (Eq. 3.31). This makes these methods very sensitive to the scene configuration as the techniques they employ can only capture a subset of all lighting effects efficiently. That is, these methods are not robust to variations in geometry and materials.

Most light transport paths can be sampled by more than one vertex connection technique. The idea behind *bidirectional path tracing* (BPT) is to employ all possible techniques for every path [85, 143]. For each pixel, BPT starts by sampling two independent subpaths, one from a light source and one from the eye. After that, a number of full paths are constructed by connecting every pair of light and eye subpath vertices, as illustrated in Figure 4.4e. With this scheme, any full path of length k can be sampled in up to $k + 2$ ways, each corresponding to a different $(s, t = k - s)$ vertex connection technique with a different pdf. The number of possible techniques is smaller when the path contains specular vertices, since such vertices cannot be involved in connections. Figure 4.8 shows all possible techniques³ for $k = 3$.

Since different techniques are efficient for different kinds of paths, Veach and Guibas [144] combine their primary estimators via multiple importance sampling (MIS), in order to preserve the qualities of each:

$$\hat{I}^{\text{BPT}} = \sum_{s \geq 0} \sum_{t \geq 0} w_{s,t}(\bar{\mathbf{x}}_{s,t}) \hat{I}_{s,t}^{\text{VC}}(\bar{\mathbf{x}}_{s,t}). \quad (4.19)$$

Recall that the power heuristic (Eq. 2.47) aims to promote efficient techniques and demote inefficient ones by assigning weights that are inversely proportional to their corresponding path pdfs. In BPT

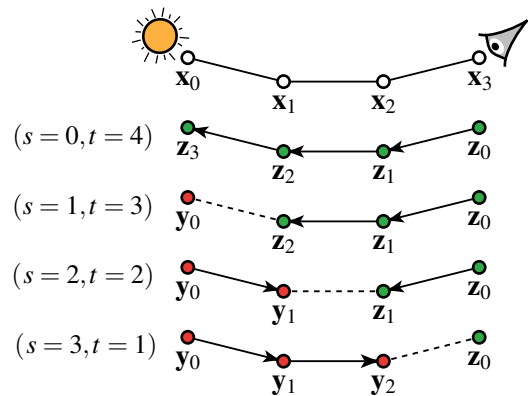


Figure 4.8: A full path constructed using different subpath connection techniques.

³We omit the $(k + 1, 0)$ technique, as it is rarely useful in practice due to the relatively small size of the eye sensors.

this helps avoid infinite variance due to geometric singularities, since techniques that perform connections along short path segments have relatively low pdfs and are automatically down-weighted. The resulting combined estimator thus handles a wide range of lighting effects much more robustly than each individual technique alone, as it avoids potential high variance induced by low path sampling pdfs. This can be clearly seen in Figure 4.6f, where the BPT image has much lower noise levels than any other image.

The use of MIS is the key to the robustness of BPT, as it can automatically find a good mixture of the available sampling techniques for each individual light transport path. However, MIS cannot help when none of the combined techniques can efficiently sample a certain kind of paths. A prominent example are $L(SD)^+SE$ paths, which are not sampleable by any $(s > 0, t > 0)$ vertex connection technique, because they do not contain DD segments where a vertex connection could be established. Such specular-diffuse-specular (SDS) light interactions correspond to caustics seen through a reflection or refraction. In BPT, these paths can only be found by $(0, k + 1)$ unidirectional sampling (Fig. 4.9) and with a very low probability when the light source is small. The resulting noise can be seen in Figure 4.6f. Kollig and Keller [77] call this the *problem of insufficient techniques*. In Chapter 7 we will address this problem by augmenting BPT with path sampling techniques from photon mapping (discussed below in Section 4.6) which can be much more efficient for SDS paths.

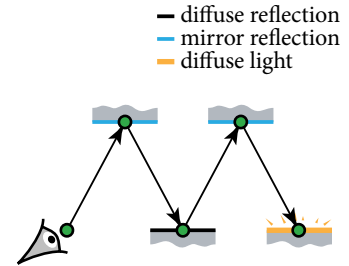


Figure 4.9: In BPT, specular-diffuse-specular (SDS) paths can be sampled only unidirectionally.

4.5 Specialized techniques for participating media

So far we have focused most of our discussion on surface rendering. However, the light transport problem is generally more difficult in the presence of participating media, since light can scatter at any point along a ray in a medium and not only at the closest surface (Eq. 3.25). All methods described in the three sections above can, in fact, handle volumetric scattering by sampling path vertices in media during the random walks. As we described in Section 4.2.1, the propagation distance along a ray is typically sampled with a pdf proportional to transmittance. When connecting two subpaths, the vertex connection estimator also evaluates the transmittance along the connection segment (Eq. 4.15), which in the absence of media is equal to one.

Unfortunately, in participating media the vertex connection estimator is even more prone to high variance in the geometry term. This is because the endpoints of two subpaths can end up arbitrarily close to each other *anywhere in space*, and not only around surface concavities. Moreover, since the scattering term at neither endpoint is importance sampled, the variance is further increased in the common case of anisotropic volumetric scattering.

The root of the above problems lies the fact that the light and eye subpaths are sampled completely independently and their endpoints are connected deterministically. Thus, there is no “freedom” left for the estimator to importance sample the connection throughput. Below we describe two recently proposed techniques that take advantage of the extra dimension in media and perform connections by not only considering the vertices of the subpaths but also their entire (semi-infinite) *segments*. This provides the necessary freedom to importance sample some terms in the connection throughput. In Chapter 9 we will revisit these techniques, improve upon them, and integrate them as general subpath connection techniques into the methods described in the previous sections.

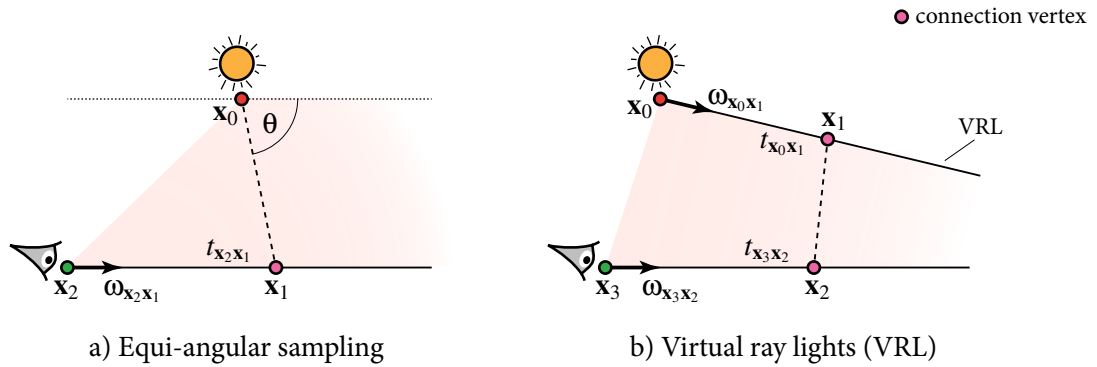


Figure 4.10: Illustration of the equi-angular and virtual ray light sampling techniques.

4.5.1 Equi-angular sampling

Kulla and Fajardo [80] considered the problem of rendering single scattering in participating media by sampling a length-2 path $\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2$ given a vertex \mathbf{x}_0 on a light source and an eye ray $(\mathbf{x}_2, \omega_{\mathbf{x}_2\mathbf{x}_1})$ (see Figure 4.10a). They observed that while the transmittance along the eye ray is bounded in the interval $[0; 1]$, the geometry term between the vertices \mathbf{x}_0 and \mathbf{x}_1 can have arbitrarily high variation. Therefore, instead of sampling the propagation distance $t_{\mathbf{x}_2\mathbf{x}_1}$ proportionally to transmittance (as traditionally done in a random walk from the eye), Kulla and Fajardo use a pdf proportional to $G(\mathbf{x}_0, \mathbf{x}_1)$, i.e. the inverse squared distance between \mathbf{x}_0 and \mathbf{x}_1 :

$$p(t_{\mathbf{x}_2\mathbf{x}_1} | \mathbf{x}_0, \mathbf{x}_2, \omega_{\mathbf{x}_2\mathbf{x}_1}) \propto G(\mathbf{x}_0, \mathbf{x}_1) = \frac{1}{\|\mathbf{x}_0 - \mathbf{x}_1\|^2}. \quad (4.20)$$

Sampling from this pdf results in a uniform distribution of angles θ between the segment $\mathbf{x}_0\mathbf{x}_1$ and the eye ray direction $\omega_{\mathbf{x}_2\mathbf{x}_1}$. Hence, Kulla and Fajardo called the technique *equi-angular sampling*. This technique has in fact appeared in the neutron transport literature a long time ago for solving the adjoint problem of estimating the incident flux at a point detector located in a medium [67, 115].

The equi-angular distribution can result in a lower-variance estimator than transmittance sampling when the eye ray passes close to the light source, as can be seen in Figure 4.11b. However, this sampling distribution can be less efficient in dense or heterogeneous media or when the distance between the eye ray and the light source is large, in which case $G(\mathbf{x}_0, \mathbf{x}_1)$ has relatively low variation. Transmittance sampling may still be preferred in such cases, so Kulla and Fajardo combine the two techniques via multiple importance sampling (MIS).

4.5.2 Virtual ray lights

To ameliorate the geometric singularities that plague virtual point light methods, Novák et al. [96] proposed *virtual ray lights* – a specialized many-light approach for rendering multiple scattering in media. A virtual ray light (VRL) is a semi-infinite segment on a light subpath sampled via a traditional random walk. The energy transport between a VRL and an eye ray is a double integral along both rays and can be estimated via Monte Carlo integration. The problem is illustrated in Figure 4.10b for the case of double scattering, where the task is to sample vertices \mathbf{x}_1 and \mathbf{x}_2 along the rays $(\mathbf{x}_0, \omega_{\mathbf{x}_0\mathbf{x}_1})$

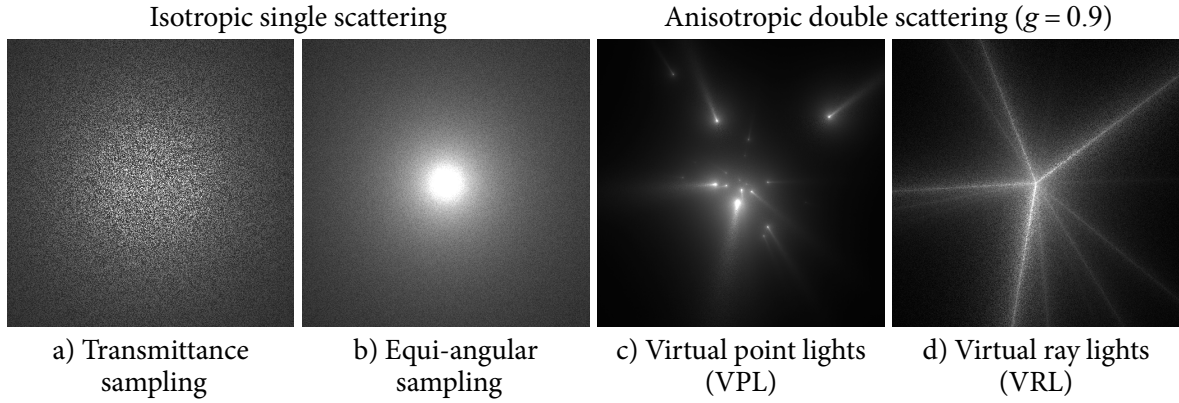


Figure 4.11: A comparison of equi-angular sampling and VRLs to traditional vertex connection based methods on a scene with a homogeneous medium and an isotropic point light source. The left two images were rendered with 4 paths per pixel. The right two images were rendered with a fixed set of 24 virtual point and ray lights, respectively, and 16 eye rays per pixel. Note that the two left and the two right images have the same expected values, respectively. The difference in the appearance of the two right images is due to the different types of sampling correlation in the VPL and VRL methods. The anisotropic emission shape of the virtual point lights can be seen in in the center right image, whereas each line in right-most image corresponds to one virtual ray light.

and $(\mathbf{x}_3, \omega_{\mathbf{x}_3\mathbf{x}_2})$ respectively. Novák et al. sample the corresponding propagation distances $t_{\mathbf{x}_0\mathbf{x}_1}$ and $t_{\mathbf{x}_3\mathbf{x}_2}$ from a joint distribution that is approximately proportional to the product of the geometry and scattering terms in the connection throughput between \mathbf{x}_1 and \mathbf{x}_2 :

$$p(t_{\mathbf{x}_0\mathbf{x}_1}, t_{\mathbf{x}_3\mathbf{x}_2} | \mathbf{x}_0, \omega_{\mathbf{x}_0\mathbf{x}_1}, \mathbf{x}_3, \omega_{\mathbf{x}_3\mathbf{x}_2}) \propto G(\mathbf{x}_1, \mathbf{x}_2) \rho(\mathbf{x}_1) \rho(\mathbf{x}_2) = \frac{\rho(\mathbf{x}_1) \rho(\mathbf{x}_2)}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}. \quad (4.21)$$

The approximate proportionality stems from the various simplifications made by Novák et al. [96] in order to make sampling from this joint distribution practical. The distance $t_{\mathbf{x}_0\mathbf{x}_1}$ along the VRL is sampled first, from an analytical marginal distribution that assumes isotropic scattering in the medium (i.e. disregards the scattering terms above) and that the eye ray is a fully infinite line. The distance $t_{\mathbf{x}_3\mathbf{x}_2}$ along the eye ray is then sampled from a distribution conditioned on vertex \mathbf{x}_1 . For the case of isotropic scattering the pdf is the same as the equi-angular pdf (Eq. 9.14) – vertex \mathbf{x}_1 in Figure 4.10b corresponds to \mathbf{x}_0 Figure 4.10a. When the medium phase function is anisotropic, the conditional pdf is not constant in the angular domain anymore, and Novák et al. tabulate the product of the two scattering terms $\rho(\mathbf{x}_1) \rho(\mathbf{x}_2)$ for a number of values for the angle θ . In Chapter 9, we will improve upon these approximations and will generalize the VRL method into a full bidirectional path tracer.

Figure 4.11 compares the vertex connection based VPL method to the VRL method. The joint VRL distribution (Eq. 4.21) achieves good importance sampling when the distance between two rays is relatively small or when the medium phase function is highly anisotropic. However, the technique does not importance sample any of the transmittance terms along the length-3 subpath involved in the connection. Depending on the geometry configuration and media properties, transmittance sampling of either propagation distances may lead to a lower-variance estimator, so Novák et al. [96] combine the various possible sampling combinations via MIS.

4.6 Photon density estimation

A popular class of bidirectional rendering algorithms was pioneered by Arvo [1] who was the first to propose tracing particle trajectories from the light sources. He stored virtual photons on the scene surfaces and then estimated the irradiance at points seen from the eye by computing the density of the deposited energy per unit area. This method received a lot of attention due to its ability to efficiently render caustics and was consequently refined and formalized as *photon density estimation* [126, 149]. Just like many-light methods, photon density estimation methods sample subpaths from the light sources and treat their vertices as lighting samples. However, instead using these samples to approximate the exitant illumination in the scene (i.e. by treating them as virtual light sources), density estimation methods view the samples as an approximation of the *incident* radiance distribution on the scene surfaces. In the remainder of this section, we present the most popular photon density estimation algorithm today – photon mapping, along with the recent progressive formulations of its pixel estimator that have made this algorithm consistent and even more practical.

4.6.1 Photon mapping radiance estimator

Photon mapping is a two-stage rendering algorithm proposed by Jensen [62]. In the first stage, N subpaths are traced from the light sources and their vertices (photons) are organized into a spatial data structure – a kd-tree or a uniform grid, called a *photon map*. In the second stage the photon map is used to estimate the scattered outgoing radiance (Eq. 3.24) at surface points \mathbf{z} sampled from the eye:

$$\hat{L}_s(\mathbf{z} \rightarrow \omega_o) = \frac{1}{N} \sum_{\mathbf{y} \in P_r} K_r(\mathbf{y}, \mathbf{z}) \rho_s(\mathbf{z}, \omega_{\mathbf{y}}, \omega_o) C_{i_{\mathbf{y}}}(\bar{\mathbf{y}}), \quad (4.22)$$

where P_r is the set of photons \mathbf{y} in the r -neighborhood around \mathbf{z} , which can be efficiently found via a range search in the photon map. Note that the BSDF ρ_s is evaluated at \mathbf{z} but with the incident direction at $\omega_{\mathbf{y}}$ used as the incoming direction. $C_{i_{\mathbf{y}}}(\bar{\mathbf{y}})$ is the sampling throughput of the light subpath $\bar{\mathbf{y}}$ at the photon vertex \mathbf{y} (Eq. 4.6). The weighting kernel K_r must integrate to one over the surface area in the r -neighborhood around \mathbf{z} . An estimator for the measurement equation (Eq. 3.21) is obtained by tracing an eye path through the pixel and evaluating the above radiance estimator at the first non-specular vertex, as done in many-light methods. The algorithm is schematically illustrated in Figure 4.12a.

The photon mapping estimator is biased as it computes the outgoing radiance at point via averaging the incident illumination in its vicinity. The search radius r is a parameter that trades off the variance and the bias of the estimator. Jensen [62] sets r to the distance between \mathbf{z} and its n th nearest photon; alternatively it can be set manually or made proportional to the area of the pixel footprint⁴ at \mathbf{z} .

Jensen's [62] original derivation of the photon mapping estimator was based on intuition and mathematical manipulations; more rigorous derivations have been proposed since then [142, 109, 31, 58]. While the behavior of this estimator in different lighting configurations is relatively well understood on an intuitive level, in-depth efficiency analyses and comparisons to the pure Monte Carlo methods from the previous sections have not been available. In Chapter 7 we will express this estimator in the path integral framework, which will provide new insights into the efficiency of the photon mapping method.

⁴The pixel footprint is the surface region around \mathbf{z} that projects to one pixel on the screen [53].

- light subpath vertex (photon)
- eye subpath vertex (query point)

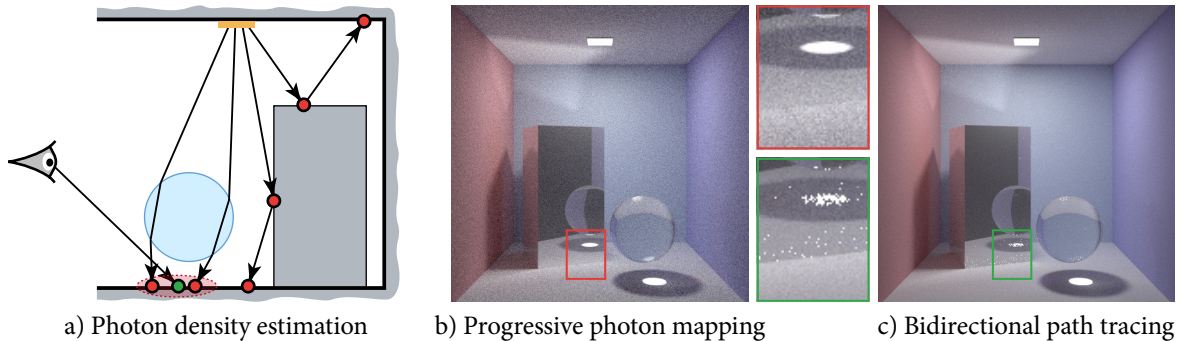


Figure 4.12: An illustration of the photon mapping algorithm (a) and a rendering comparison between progressive photon mapping (b) and bidirectional path tracing (c) on the scene from Figure 4.6.

4.6.2 Progressive photon mapping

While the photon tracing stage in photon mapping provides an unbiased estimate of the incident radiance distribution in the scene, the final outgoing radiance estimator (Eq. 4.22) introduces bias by blurring this distribution. This estimator can be made consistent by shrinking the search radius to zero as the number of sampled photons approaches infinity [63]. This guarantees that both its variance and the bias vanish in the limit. Unfortunately, this also means that convergence to the correct solution comes at the cost of a photon map with an infinitely large memory footprint.

Recently, Hachisuka et al. [41] showed that consistency can in fact be achieved without the need to store infinitely many photons in a single photon map. Their *progressive photon mapping* (PPM) algorithm goes about this by storing at every query point \mathbf{z} a local radius and cumulative photon contributions which are updated in successive independent photon tracing iterations. At each iteration, a new photon map is built, the contributions of the newly sampled photons are accumulated to every query point, and the photon map is finally discarded. Hachisuka et al. [41] showed that by shrinking the query radii at an appropriately chosen rate after every iteration, the radiance estimates converge to the correct values while keeping the memory footprint bounded. Hachisuka and Jensen [38] subsequently proposed *stochastic progressive photon mapping* (SPPM) to store the cumulative statistics with each pixel rather than at a fixed set of query points. While the PPM formulation requires the query points \mathbf{z} to remain fixed over the course of rendering, SPPM allows for tracing a new set of eye subpaths at each iteration which enables consistent estimation of the final pixel values.

Probabilistic formulation

More recently, Knaus and Zwicker [76] presented an (S)PPM formulation that avoids maintaining photon statistics altogether. Their practical algorithm is a simple extension of traditional photon mapping that averages independently rendered images using progressively shrunk query radii. Their consistent pixel estimator has the form

$$\hat{I}_N^{\text{PPM}} = \frac{1}{N} \sum_{i=1}^N \hat{I}_{r_i}^{\text{PM}}. \quad (4.23)$$

Here, $\hat{I}_{r_i}^{\text{PM}}$ is a primary estimator for the measurement equation that uses the photon mapping radiance estimator (Eq. 4.22) with a query radius r_i computed as

$$r_i = r_1 \sqrt{\left(\prod_{k=1}^{i-1} \frac{k + \alpha}{k} \right) \frac{1}{i}}, \quad (4.24)$$

where r_1 is the radius at the first iteration and $\alpha \in (0; 1)$ is a user parameter that controls the radius reduction rate for the subsequent iterations. In Section 7.4.1 we will derive a simpler radius reduction formula that is asymptotically equivalent to the one above.

Knaus and Zwicker [76] proved that the secondary estimator in Equation 4.23 converges to the true pixel value under the assumption that the photons in all progressive iterations are sampled from the same probability distribution. Their proof is thus incompatible with methods that (adaptively) change the photon sampling distribution, such as the Markov chain approach of Hachisuka and Jensen [39], whose consistency requires additional proofs.

Figure 4.12b shows an image produced by the (S)PPM variant of Knaus and Zwicker [76] next to the bidirectional path tracing (BPT) result from Figure 4.6f. Both images have been rendered progressively by recursively evaluating their corresponding secondary pixel estimators (Eq. 2.20) and use roughly the same number of light subpaths. We see that PPM captures the reflected caustics (produced by SDS paths) in this scene much more efficiently than BPT which on the other hand delivers a smoother result almost everywhere else. In Chapter 7, our formulation of the primary photon mapping pixel estimator $\hat{I}_{r_i}^{\text{PM}}$ as a path integral estimator of the form in Equation 4.1 will allow us to combine these two rendering methods via multiple importance sampling into a consistent algorithm that preserves their complementary advantages.

Asymptotic performance

The secondary pixel estimator in Equation 4.23 is asymptotically equivalent to that of SPPM [38]. Knaus and Zwicker [76] derived the asymptotic behavior of both the primary and the secondary estimators:

$$\text{Var} [\hat{I}_{r_i}^{\text{PM}}] = O(i^{1-\alpha}) \quad \text{Bias} [\hat{I}_{r_i}^{\text{PM}}] = O(i^{\alpha-1}) \quad (4.25a)$$

$$\text{Var} [\hat{I}_N^{\text{PPM}}] = O(N^{-\alpha}) \quad \text{Bias} [\hat{I}_N^{\text{PPM}}] = O(N^{\alpha-1}). \quad (4.25b)$$

The MSE of the secondary estimator, which measures its total expected error, is thus:

$$\text{MSE} [\hat{I}_N^{\text{PPM}}] = \text{Var} [\hat{I}_N^{\text{PPM}}] + \text{Bias}^2 [\hat{I}_N^{\text{PPM}}] = O(N^{-\alpha}) + O(N^{2(\alpha-1)}), \quad (4.26)$$

which has a maximum rate of $O(N^{-2/3})$, reached for $\alpha = 2/3$. This means that (S)PPM converges to the correct solution asymptotically slower than the unbiased vertex connection estimators from Section 4.2, and thus slower than every unbiased rendering algorithm we have discussed so far, including BPT. However, as seen in Figure 4.12, PPM can often produce a lower-error estimate with a finite number of samples, especially in regions with SDS paths. The combined rendering algorithm that we describe in Chapter 7 inherits the higher error convergence rate of BPT while preserving the efficiency of PPM for SDS paths.

4.7 Other methods

A large variety of Monte Carlo methods have been proposed for solving the light transport problem, and our discussions throughout this thesis are focused on those that are most closely related to the developments presented in the following chapters. In this section we review some other prior work related to our contributions.

4.7.1 Incident illumination importance sampling

As we discussed in Section 2.5.4, the ideal sampling density for Monte Carlo integration is the one exactly proportional to the integrand. And while it is practically infeasible to sample from such a high-dimensional distribution to solve the full light transport problem, it has been shown that it is possible to construct approximate distributions that achieve close proportionality for certain lower-dimensional sub-problems.

The most sophisticated unbiased importance sampling methods available construct an explicit representation of a pdf for sampling the incident illumination at a given surface point. Ideally, this pdf would be the product of all terms under the scattering integral (Eq. 3.24). In practice, not all of these terms are known or cheap to compute, so the pdf typically includes only some of them. In particular, most methods aim to sample from the product of the BSDF and the unoccluded incident radiance, while the visibility term is usually omitted as it is the most expensive to evaluate. The most common approach is to build a compact tabulation of this product for direct illumination [12, 14, 10] as well as for indirect illumination from virtual point lights [154]. Rousselle et al. [118] also include a conservative visibility term. The photon map has also been used for importance sampling directions in random walks, where the distribution is derived from the photons cached at nearby locations [61, 52, 108, 132].

Some methods use *importance resampling* to estimate the illumination from an environment map at a point [5, 137]. With this approach, a number of candidate directions are first sampled from the BSDF, which are then weighted and resampled to obtain a distribution that is approximately proportional to the product of the BSDF and the environment radiance.

It is also possible to reconstruct an approximation of the ideal distribution during sampling, using the contributions of the already evaluated samples. This *sequential Monte Carlo* approach has been used for sampling the incident illumination at points on surfaces [102] and in participating media [103].

The virtual point light importance caching method that we present in Chapter 6 bears similarity to the pdf reconstruction approach of Pegoraro et al. [102]. It also makes use of importance resampling to handle glossy surfaces more efficiently. In addition, like most of the aforementioned methods, our joint path sampling techniques presented in Chapter 9 aim to importance sample the product of several path contribution terms, but with the difference that we derive the sampling distributions through analytical or numerical marginalization instead of caching and reconstruction.

4.7.2 Exploiting coherence

In most typical scenes the incident illumination on surfaces and in media often varies piece-wise smoothly. In such cases, the scattering integrals at nearby locations in space are strongly correlated, as are the measurement integrals of neighboring pixels in the image. Many approaches have been

proposed to exploit these correlations by reusing Monte Carlo samples for the estimation of multiple illumination integrals, thereby amortizing the sampling effort and improving rendering efficiency.

Irradiance caching [157, 156] computes accurate indirect irradiance estimates at a set of adaptively chosen locations in the scene and interpolates these estimates at points outside this set. By considering a larger number of less accurate estimates, Kontkanen et al. [79] extended this approach to adaptive irradiance filtering over surfaces. These methods can significantly improve efficiency over brute-force sampling, but are limited to diffuse inter-reflections as the interpolation used does not account for directionality in the incident illumination. *Radiance caching* [81, 34, 50] addresses this shortcoming by storing a compact incident radiance representation at each cache location that captures this directionality. However, it still assumes that the illumination varies smoothly everywhere except around geometric discontinuities, which is not the case in the presence of small sources of strong indirect illumination. Křivánek et al. [82] ameliorated this problem by improving the interpolation strategies and the schemes for adaptive placement of illumination records in (ir)radiance caching. Nevertheless, these methods introduce bias to the solution which is impossible to eliminate. The importance caching method that we describe in Chapter 6 is also designed to exploit illumination coherence but in a way that avoids interpolation bias and is robust to discontinuities.

On top of product importance sampling of BRDF and environment map illumination [10], Clarberg and Akenine-Möller [11] add a control variate term that includes interpolated visibility. This reduces noise in occluded regions, but since the sampling distribution remains unchanged, the effect of variance reduction is limited. The table-driven adaptive importance sampling method of Cline et al. [15] reduces variance by sharing importance information across neighboring pixels. This approach is however prone to high variance at discontinuities, which Cline et al. clamp in a biased way.

Illumination can also be interpolated in screen space. Suykens and Willems [135] reconstruct an image from bidirectional path tracing samples using adaptive density estimation on the image plane. Their algorithm progressively shrinks the 2D image filter as more samples are accumulated, which bears similarity to progressive photon mapping. Such screen-space de-noising algorithms have recently received renewed interest in research due to their ability to reconstruct a smooth, albeit biased, result from a very small number of samples. Some approaches operate entirely on screen-space sample information [125, 119, 120, 89], some rely on auxiliary feature information for detecting discontinuities [19, 94, 121], and some also adaptively distribute samples on the image plane based on the estimated error [119, 120, 89, 121]. Screen-space filtering can be applied on top of almost any Monte Carlo rendering method [162], including the ones we present in the following chapters.

4.7.3 Density estimation in participating media

Photon density estimation has also been successfully applied to rendering participating media. Jensen and Christensen [64] presented a simple extension of the photon mapping estimator (Eq. 4.22) to compute the outgoing radiance at points in participating media. Their *volumetric photon mapping* algorithm stores photons in the volumes of the media, which are then gathered around points sampled along eye rays via ray marching. Knaus and Zwicker [76] later showed that the volumetric radiance estimator can be easily made consistent using the same progressive query radius reduction scheme from their progressive photon mapping formulation. Our vertex connection and merging algorithm, presented in Chapter 7, also benefits from these two straightforward extensions.

Jarosz et al. [58] improved the efficiency of volumetric photon mapping by gathering photons around entire eye segments, or beams, avoiding the costly ray matching. Sun et al. [134] extended this idea

to a radiance estimator that considers the entire segments of the light subpaths rather than only their vertices (i.e. the photons). Concurrently with Sun et al., Jarosz et al. [59] formalized various volumetric radiance estimators based on *photon point* and *photon beam* density estimation in a unified framework. Jarosz et al. [60] later proposed a progressive photon beam estimator that achieves consistency in the spirit of the progressive photon mapping formulation of Knaus and Zwicker [76]. These formulations inspired the development of the virtual ray light and virtual beam light methods of Novák et al. [96, 95]. Building on these ideas, in Chapter 9 we propose novel sampling techniques for constructing light transport paths in participating media that use both points and segments as sampling primitives.

4.7.4 Markov chain Monte Carlo

The formalization of light transport simulation as a pure integration problem [144] allowed Veach and Guibas to leverage Metropolis sampling for rendering scenes with complex lighting [145, 142]. Their *Metropolis light transport* (MLT) algorithm was the first application of Markov chain Monte Carlo (MCMC) integration to image synthesis. It mutates entire light transport paths in the high-dimensional path space with a target density approximately proportional to the measurement contribution function (Eq. 3.31). The initial path in a chain is generated by an ordinary Monte Carlo path sampling technique, such as vertex connection (Sec. 4.2.3). Veach and Guibas proposed a number of mutation strategies that perturb individual path vertices or regenerate entire subpaths again using traditional path sampling techniques.

Metropolis sampling can significantly outperform ordinary, uncorrelated Monte Carlo methods when the energy is primarily concentrated in small isolated (e.g. caustic) regions of the path space, thanks to its ability to efficiently explore such regions. Having recognized this strength, research effort has since focused on extending and further improving the MLT algorithm. Pauly et al. [100] generalized MLT to handle participating media and augmented its mutation arsenal with new strategies. Kelemen et al. [72] proposed to perform the mutations in the unit hypercube of random numbers that traditional sampling techniques map to paths in the path space. While the original MLT algorithm of Veach and Guibas [145] renders an image using a single long chain of paths, Cline et al. [13] showed that using a large number of short chains lowers the sample correlation and improves stratification, which is beneficial in scenes with more uniform lighting. More recently, Jakob and Marschner [57] recognized that light transport paths containing specular vertices lie on submanifolds of the full path space. They designed a mutation strategy that “walks” on these submanifolds to explore caustic paths more efficiently than prior mutation strategies.

Metropolis sampling has also been successfully used to distribute photons and virtual point lights (VPLs) in regions of importance to the image [33, 124, 39, 9]. Our VPL distribution method, described in the following chapter, relies on ordinary sampling methods to achieve the same goal.

In this thesis we focus on ordinary Monte Carlo path sampling techniques, noting that Markov chain methods also rely on such techniques for constructing and mutating light transport paths. We therefore view MCMC methods as orthogonal to the contributions of this work. Indeed, we expect that enhancing our techniques through the use of MCMC samplers will bring further efficiency improvements.

Importance-driven Distribution of Virtual Point Lights

5

Many rendering applications in industrial design, architectural visualization, and film production require accurate global illumination simulation in scenes with complex distributions of geometry and lighting. Monte Carlo integration is currently the best-known method for solving that problem and many-light rendering, which we introduced in Section 4.3, is one such approach that can handle scenes with diffuse and moderately glossy materials. Many-light methods reduce the global illumination problem to the problem of computing direct illumination from a set of virtual point lights (VPLs). The efficiency of this approach largely depends on two factors: (1) the quality of the VPL distribution in the scene, i.e. how relevant the VPL set is to the regions seen from the eye, and (2) the cost of computing the contributions of all VPLs to every pixel. In this chapter, we address the problem of distributing a set of VPLs relevant to the chosen viewpoint. In Chapter 6 we will propose a method to efficiently compute an image from this VPL set.

Keller's instant radiosity [73] samples VPLs with a density roughly proportional to the flux density in the scene. As a result, many VPLs are usually concentrated in the areas around the light sources, and in highly occluded scenes, where light scatters at least a few times before reaching the eye, most of these VPLs do not contribute any energy to the image. Moreover, VPLs that bring some energy are sampled with low probability, leading to high variance. Since the VPL distribution stage is typically much cheaper than the final rendering stage, it is worth spending some extra effort to distribute a set of VPLs that are relevant to the image.

Importance-driven tracing of light subpaths has been first employed in the context of photon mapping. Peter and Pietrek [107] proposed building an "importon" map from the vertices of subpaths traced from the eye. This map is then used to guide the sampling of photon paths toward regions of importance to the image. Keller and Wald [74] use the importon map to probabilistically store photons in the photon map. For many-light rendering, Wald et al. [148] trace paths from the eye and count the number of times each light source is hit to estimate its importance prior to VPL distribution. Matrix row-column sampling (MRCS) [46] adaptively selects a few important VPLs by first clustering all sampled VPLs based on their image contributions and then electing a representative one from each cluster. Bidirectional instant radiosity (BIR) [123] samples VPL locations from both the eye and the light sources and then resamples them proportionally to their estimated image contributions, keeping the most relevant ones for the final rendering. Metropolis instant radiosity (MIR) [124] uses Metropolis sampling to generate a set of VPLs that contribute an equal amount of energy to the image. While MRCS and BIR can find relevant VPLs in highly occluded scenes, their implementation is fairly involved and requires tuning a number of parameters. On the other hand, MIR is more robust to scenes with difficult visibility and has a small memory footprint. However, it is prone to producing poorly stratified VPL sets due to inherent correlations in the underlying Metropolis sampler.

In this chapter, we propose a practical importance-driven VPL distribution algorithm that is a simple extension to the traditional random walk based distribution of Keller [73]. The idea is to stochastically accept or reject each VPL on-the-fly with a probability proportional to its estimated importance to the final image. As a result, more VPLs are concentrated in regions that illuminate the visible parts of the scene, which comes at the cost of a negligible overhead to the overall rendering performance. Unlike some previous methods, this algorithm is very simple to implement and is well suited to progressive rendering. We demonstrate its efficiency on scenes with difficult visibility.

5.1 Probabilistic VPL acceptance

The computational cost of the many-light estimator (Eq. 4.18) is proportional to the number of VPLs distributed in the scene. As illustrated in Figure 5.1a, in highly occluded scenes most of these VPLs contribute no energy to the image. We can take advantage of this fact and quickly discard VPLs with low estimated contributions during the VPL distribution stage. In this section we describe how this can be achieved without introducing bias to the image.

5.1.1 Formal derivation

Our method replaces instant radiosity’s deterministic VPL creation by a stochastic acceptance decision based on a carefully chosen probability. In the spirit of Keller and Wald [74], we transform the VPL contribution function $h(\mathbf{x}, \mathbf{v}_k)$ (Eq. 4.18) by introducing a Russian roulette decision with *acceptance probability* $P_{\text{acc}}(\mathbf{v}_k) \in (0; 1]$. We start by rewriting the many-light estimator:

$$\hat{I}^{\text{ML}} = \sum_{k=1}^M h(\mathbf{x}, \mathbf{v}_k) = \sum_{k=1}^M h(\mathbf{x}, \mathbf{v}_k) \frac{P_{\text{acc}}(\mathbf{v}_k)}{P_{\text{acc}}(\mathbf{v}_k)} \quad (5.1a)$$

$$= \sum_{k=1}^M h(\mathbf{x}, \mathbf{v}_k) \frac{1}{P_{\text{acc}}(\mathbf{v}_k)} \int_0^1 \chi_{[0; P_{\text{acc}}(\mathbf{v}_k)]}(t) dt, \quad (5.1b)$$

where we express $P_{\text{acc}}(\mathbf{v}_k)$ as an integral over the characteristic function of the interval $[0; P_{\text{acc}}(\mathbf{v}_k)]$. One-sample Monte Carlo estimation of this integral gives

$$\int_0^1 \chi_{[0; P_{\text{acc}}(\mathbf{v}_k)]}(t) dt \approx \chi_{[0; P_{\text{acc}}(\mathbf{v}_k)]}(\xi) = \begin{cases} 1 & \text{if } \xi < P_{\text{acc}}(\mathbf{v}_k) \\ 0 & \text{else,} \end{cases} \quad (5.2)$$

where $\xi \in [0, 1)$ is a canonical uniform random variable. Plugging this result back into Equation 5.1, we arrive at the following modified many-light pixel estimator:

$$\hat{I}^{\text{IML}} = \sum_{k=1}^M \begin{cases} h(\mathbf{x}, \mathbf{v}_k) \frac{1}{P_{\text{acc}}(\mathbf{v}_k)} & \text{if } \xi < P_{\text{acc}}(\mathbf{v}_k) \\ 0 & \text{else.} \end{cases} \quad (5.3)$$

Note that this modification can in fact only increase the variance of the pixel estimator. Nevertheless, by choosing an appropriate acceptance probability, VPLs with low contributions can be discarded early on and not be considered in the final rendering stage, which in turn improves the overall efficiency. We next describe how to choose the acceptance probability for each VPL.

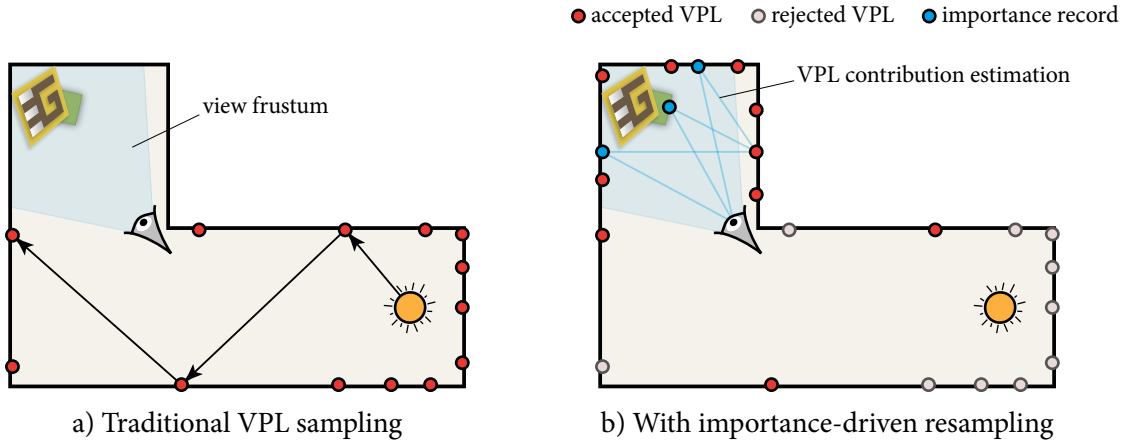


Figure 5.1: VPL distribution in a highly occluded scene. a) Traditional sampling produces many VPLs that do not contribute energy to the image. b) Our importance-driven resampling method concentrates more VPLs in areas relevant to the visible parts of the scene.

5.1.2 VPL acceptance probability

To be able to efficiently discard VPLs during the distribution stage, as in previous work [148, 46, 123, 124] we make the practical assumption that the contribution of each VPL does not vary much across different image pixels. Under this assumption, the image variance would be zero if all M VPLs contribute the same amount of luminance Φ/M , where Φ is the total image luminance.

Assume we have an estimate $\hat{\Phi}$ of the final image luminance as well as an estimate $\hat{\Phi}_k$ of the contribution of each VPL candidate \mathbf{v}_k . We define the acceptance probability for \mathbf{v}_k as

$$P_{\text{acc}}(\mathbf{v}_k) = \max \left(\min \left(M \frac{\hat{\Phi}_k}{\hat{\Phi}}, 1 \right), P_{\text{acc}}^{\min} \right), \quad (5.4)$$

where $P_{\text{acc}}^{\min} > 0$ is a parameter. Using this probability, VPLs with estimated contribution higher than desired are trivially accepted with unmodified contribution (due to $P_{\text{acc}}(\mathbf{v}_k) = 1$ in such cases), thus avoiding excessive variance. In contrast, VPLs with comparatively low contributions are likely to be rejected. As a result, more VPLs are concentrated in the areas around the visible parts of the scene.

The minimum acceptance probability P_{acc}^{\min} in Equation 5.4 can be viewed as a trade-off parameter between run-time and importance sampling efficiency. If $P_{\text{acc}}^{\min} \geq 1$, the resulting VPL set is identical to the one created by instant radiosity, as all VPLs are trivially accepted. A value close to zero leads to more aggressive importance sampling. We now describe how to compute the two luminance estimates required for the evaluation of Equation 5.4.

VPL contribution estimation

Prior to VPL distribution, we trace a number of subpaths from the eye, in the order of a few hundred, terminating each one at the first non-specular surface. The endpoints of these subpaths are the equivalent to importons in the photon mapping setting and we call them *importance records* (see Figure 5.1b). This is the only storage required by our algorithm. During VPL sampling from the light sources, the image contribution $\hat{\Phi}_k$ of each VPL \mathbf{v}_k is estimated by connecting \mathbf{v}_k to all importance records. This last step is equivalent to rendering a random subset of all image pixels using a single VPL.

Image luminance estimation

The second quantity required by the VPL acceptance probability is the estimate of the total image luminance $\hat{\Phi}$. In a progressive rendering setup, $\hat{\Phi}$ can be directly obtained from the running image estimate. For the first rendering iteration, $\hat{\Phi}$ is set to zero. This results in no importance sampling for the initial VPL set, as P_{acc} is conservatively clamped to 1 for every VPL. Over time, as the running image estimate becomes more accurate, the acceptance probabilities become less conservative. As the number of rendering iterations grows, the importance sampling becomes more aggressive and the VPL distribution gradually shifts towards the regions of importance to the viewpoint.

In case the image needs to be rendered in a single pass, $\hat{\Phi}$ can be computed using the importance records and a number of pilot VPLs or with some other method such as (bidirectional) path tracing. Note that the estimate $\hat{\Phi}$ need not be very precise for the method to work; however its accuracy does affect the efficiency of the importance sampling.

5.2 Results

We have implemented the above described VPL resampling algorithm in a progressive rendering system where the viewpoint and the objects can be manipulated interactively. When the interaction is stopped, a high quality image is obtained by averaging the results of independent many-light rendering iterations. Below we summarize the results obtained on three different scenes with varying illumination and occlusion characteristics.

The EG scene, shown in the top row of Figure 5.2, has one point light source placed around the corner behind the camera, as illustrated schematically in Figure 5.1. It takes more than two bounces off the walls on average for the light to reach the regions visible from the eye. On this scene, our resampling method discards 93% of the VPL candidates to achieve the same visual quality as traditional VPL sampling but with a much lower number of VPLs. This average acceptance probability of 0.07 corresponds to more than an order of magnitude increase in efficiency in the rendering stage.

Figure 5.2c shows the popular SPONZA palace scene illuminated by one directional light source. The image was rendered with an average VPL acceptance probability of 0.28.

LIVING ROOM is a typical architectural scene, shown in Figure 5.2d, where sunlight can reach the interior only through the blinds on the windows. Almost all the illumination in this scene is indirect. Traditional sampling places most VPLs on the outside walls; our resampling method efficiently discards these VPLs early on and only keeps about 23% of all candidates.

5.3 Discussion

Our experiments show that in highly occluded scenes, for the same number of light subpaths, our importance-driven VPL distribution method can produce an image very similar to that of traditional instant radiosity [73] but using a small fraction of the generated VPLs. Due to the probabilistic rejection based on connecting every VPL candidate to all importance records, our method requires the sampling of more light subpaths and tracing more rays than instant radiosity to produce the same num-

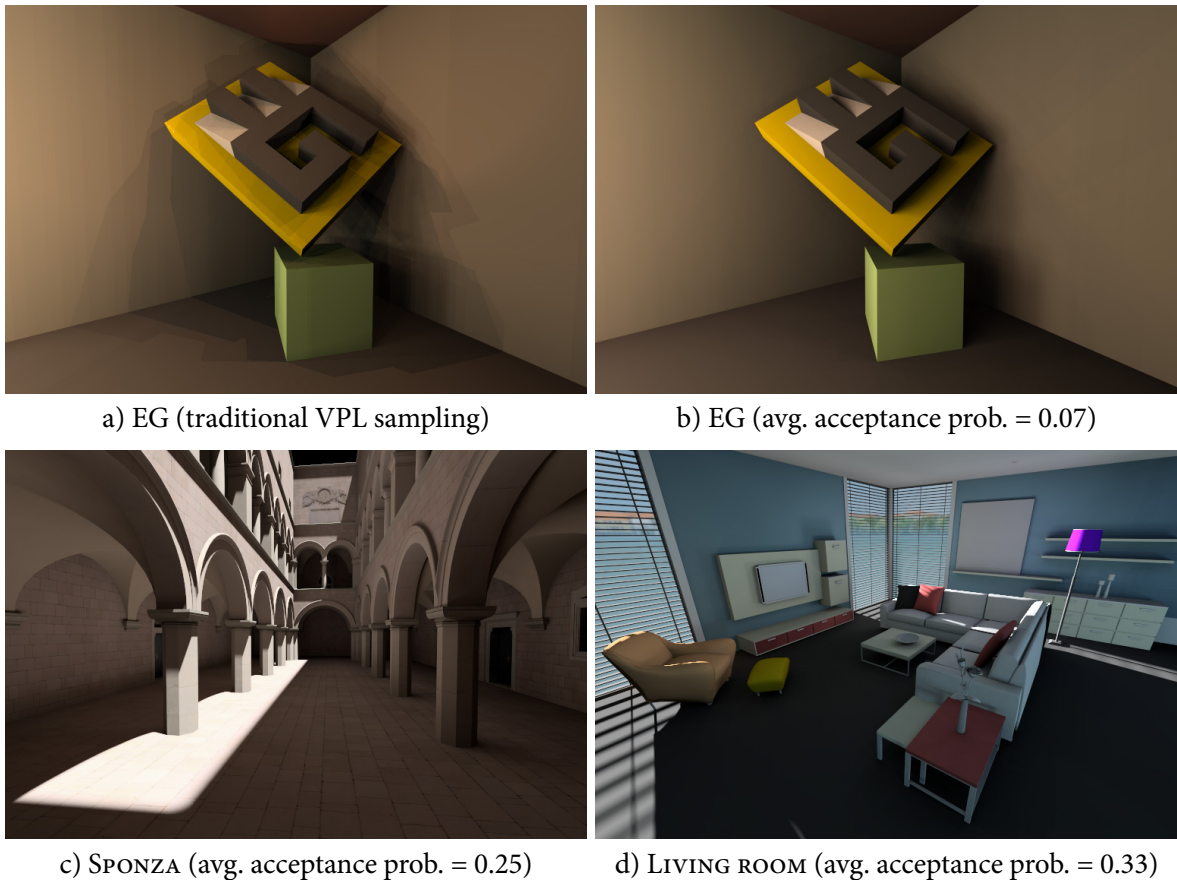


Figure 5.2: Three scenes rendered with our importance-driven VPL resampling method. *a)* In this scene, most VPLs produced by traditional sampling do not contribute any energy to the image (see the schematic illustration in Figure 5.1). *b)* Our resampling method produces a much smooth result with the same number of VPLs, where only 7% of all proposed VPLs were accepted. *c)* The SPONZA scene rendered using $4\times$ fewer VPLs than traditional sampling with the same image quality. *d)* In the LIVING ROOM scene, 77% of the VPL candidates are discarded by our algorithm.

ber of VPLs. However, we found that this overhead does not increase the total image time noticeably in practice, because the rendering stage remains significantly more expensive than our VPL distribution stage. And because our method is a simple extension to instant radiosity, the good VPL distribution typically obtained by using low-discrepancy sampling patterns can be preserved in the regions of importance to the image. This is more difficult to achieve with previous methods [123, 124] which rely on multiple or heavily correlated sampling techniques.

Our method introduces two parameters: the minimum acceptance probability P_{acc}^{\min} and the number of importance records N^{IR} . In all our tests we used $P_{\text{acc}}^{\min} = 0.05$ and $N^{\text{IR}} = 100$, and did not see benefit from fine-tuning these parameters per scene. However, in scenes with high geometric or lighting variation in the visible regions, a larger number of importance records may be required to obtain accurate estimates for the the VPL contributions $\hat{\Phi}_k$.

Our algorithm is easy to implement on top of an existing many-light renderer. It is also easy to parallelize since VPLs are resampled on-the-fly as they are generated and each VPL is subjected to rejection independently from all the others. Even though the required intermediate storage for the importance

records is low, this memory footprint can in fact be completely eliminated by sampling a new set of importance records when estimating the image contribution of each VPL. While doing so would double the computational cost of that estimation, we expect the increased overhead to remain negligible.

In large-scale scenes with many light sources, where the eye sees only a small part of the scene (e.g. one room in a building), our method can be combined with the technique of Wald et al. [148] to estimate the image importance of each light source prior to sampling the VPLs. In a progressive rendering setup, the accuracy of this importance estimation can be improved after every iteration. As an alternative, the importance of each light source can be derived from its actual contribution to the running image estimate.

Finally, our method is best suited to scenes where the lighting distribution in the visible scene regions is mostly uniform. When this distribution is highly non-uniform, a large number of VPLs may be required to obtain a smooth result, since in such cases different VPL subsets are important to different image regions. However, large VPL counts can make the rendering stage prohibitively expensive. We will address this problem in the following chapter by devising a method to efficiently find the few most important VPLs at every visible surface point in the scene.

Importance Caching for Many-light Rendering

6

Many-light rendering is an attractive and flexible approach for solving the global illumination problem: a coarse, artifact-free approximation can be quickly obtained with a few VPLs, while increasing the number of VPLs to a sufficient level yields a high-fidelity image. However, in scenes with complex visibility configuration and lighting distribution, a large number of VPLs is often required to obtain an accurate solution. This can substantially increase the computational cost of the rendering stage, where traditionally every VPL is connected to the endpoint of every eye subpath (Eq. 4.18), even though only a small fraction of all VPLs contribute significantly to each pixel. In Chapter 5 we showed how to distribute VPLs in the scene according to their total image contribution. In this chapter, we go one step further and attack the problem of estimating the value of each pixel from thousands of VPLs.

Various methods have been proposed to improve the scalability of many-light rendering since Keller introduced the basic instant radiosity algorithm [73]. Some approaches use adaptive clustering to handle the many VPLs necessary to render scenes with complex lighting. The lightcuts method [150, 151] computes a pixel estimate from a cut in a VPL tree. The cut is derived by bounding the contributions of entire VPL clusters locally at each surface point, assuming full visibility. Wang and Akerlund [154] use a VPL tree to randomly select a few VPLs based on the product of the unoccluded incident VPL radiance and the BSDF at the surface point. By clustering VPLs together and employing heuristics to roughly estimate their contributions, these methods aim to efficiently find a small representative set of VPLs at every surface point. However, like most importance sampling methods we discussed in Section 4.7.1, they compute this representative set independently at each point, without taking advantage of the fact that the VPL contributions usually vary piece-wise smoothly over surfaces. Moreover, the VPL set can be unnecessarily large in highly occluded scenes as the heuristics disregard occlusion. Reconstruction cuts [150] address the former issue by interpolating lightcut contributions across image pixels, while Herzog et al. [50] combine lightcuts and radiance caching to further improve the efficiency and the accuracy of the interpolation. While interpolation exploits illumination coherence, it also introduces bias in the image that is impossible to completely eliminate. Interactive many-light methods achieve high efficiency through even more aggressive interpolation, visibility approximations, and optimized implementation on graphics hardware [116, 24, 117].

In this chapter, we propose a stochastic approach for handling large numbers of VPLs. Instead of evaluating the contributions of all VPLs to every pixel, we compute a Monte Carlo estimate of this sum. The key to achieving efficiency with this approach is to select a small number of VPLs for each pixel from a discrete distribution that is ideally proportional to their contributions. We amortize the costly construction of these probability distributions by sharing them among pixels in a way that exploits illumination coherence. At a sparse set of locations in the scene, we evaluate the exact contributions of all VPLs and derive several types of distributions from these evaluations. Multiple cached distributions are then used for selecting VPLs at other locations, and we combine all estimators using a bilateral multiple importance sampling framework that employs a novel aggressive weighting heuristic.

Our method handles direct and indirect illumination simultaneously, and the VPL sampling distributions consider all terms of the measurement contribution function, including visibility. We demonstrate that this can deliver significant variance reduction in occluded scenes with complex lighting, where visibility is often a major source of variance. Since our approach does not introduce interpolation bias, high-fidelity results can be obtained progressively with a bounded memory footprint. We also demonstrate a simplified version of our method that can produce accurate low-noise previews at interactive rates.

6.1 Algorithm overview

Our goal is to compute a Monte Carlo estimate of the light transport integral (Eq. 3.27) for each pixel in the image. We attack this problem via many-light rendering, which amortizes the path sampling cost by sharing a common set of light subpaths among all pixel integrals. We render the image progressively, accumulating the results of independent sampling iterations. At the beginning of each iteration, we create a set of M virtual point lights (VPLs) using the resampling method from Chapter 5. Even though the obtained VPL set is relevant to the chosen viewpoint, evaluating the contributions of all VPLs can be prohibitively expensive when their number is in the order of thousands or more. Our solution is based on an unbiased Monte Carlo estimation of the many-light sum in Equation 4.18:

$$\hat{I}^{\text{ML}} = \sum_{k=1}^M h(\mathbf{x}, \mathbf{v}_k) \approx \sum_{k=1}^m \frac{h(\mathbf{x}, \mathbf{v}_k)}{P(\mathbf{v}_k | \mathbf{x})} = \sum_{k=1}^m \frac{B(\mathbf{x}, \mathbf{v}_k) L(\mathbf{x}, \mathbf{v}_k) G(\mathbf{x}, \mathbf{v}_k) V(\mathbf{x}, \mathbf{v}_k)}{P(\mathbf{v}_k | \mathbf{x})} = \hat{I}^{\text{IC}}, \quad (6.1)$$

which evaluates the contributions of a much smaller number $m \ll M$ of VPLs, chosen from a discrete probability distribution P conditioned on the eye subpath vertex \mathbf{x} . In the interest of keeping the notation in the remainder succinct, in the above equation we have regrouped some of the terms in the VPL contribution $h(\mathbf{x}, \mathbf{v}_k)$ as follows: G and V are the regular geometry (Eq. 3.11) and visibility (Eq. 3.13) terms, respectively. Without loss of generality, we have assumed that the transmittance between \mathbf{x} and \mathbf{v}_k is one, i.e. that there are no participating media in the scene. The terms B and L incorporate the sampling throughputs and scattering terms at the eye subpath vertex \mathbf{x} and the VPL \mathbf{v}_k , respectively:

$$B(\mathbf{x}, \mathbf{v}_k) = C_{t-1}(\bar{\mathbf{z}}) \rho_s(\mathbf{z}_{t-1}, \boldsymbol{\omega}_{\mathbf{z}_{t-2}\mathbf{z}_{t-1}}, \boldsymbol{\omega}_{\mathbf{z}_{t-1}\mathbf{y}_{i_k,j_k}}) \quad (6.2)$$

$$L(\mathbf{x}, \mathbf{v}_k) = C_{j_k}(\bar{\mathbf{y}}) \rho_s(\mathbf{y}_{i_k,j_k}, \boldsymbol{\omega}_{\mathbf{y}_{i_k,j_k-1}\mathbf{y}_{i_k,j_k}}, \boldsymbol{\omega}_{\mathbf{y}_{i_k,j_k}\mathbf{z}_{t-1}}) \frac{1}{N \cdot P_{\text{acc}}(\mathbf{v}_k)}. \quad (6.3)$$

The throughput, scattering, and light subpath count normalization terms above come from the many-light estimator in Equation 4.18, with $\mathbf{x} \equiv \mathbf{z}_{t-1}$ and $\mathbf{v}_k \equiv \mathbf{y}_{i_k,j_k}$. The L term in addition incorporates the VPL acceptance probability $P_{\text{acc}}(\mathbf{v}_k)$ (Eq. 5.4).

Similarly to the probabilistic VPL acceptance decision in Chapter 5, the Monte Carlo estimation of the sum in Equation 6.1 can only increase the variance of the pixel estimator. And again, efficiency can be greatly improved if the sampling distribution P is chosen carefully. In fact, making P exactly proportional to the VPL contribution $h(\mathbf{x}, \mathbf{v}_k)$ results in a zero-variance estimate which could be obtained using a just single VPL per pixel, as shown in Figure 6.1b. Unfortunately, while sampling from this particular distribution is actually feasible, it is impractical since computing the distribution requires evaluating the contribution of every VPL \mathbf{v}_k at point \mathbf{x} , which defeats the purpose of estimating the sum via Monte Carlo in the first place.

A common approach to making the VPL importance distribution cheaper to compute is to exclude the visibility term from it [5, 12, 154]. However, as Figure 6.1c demonstrates, doing so in occluded scenes

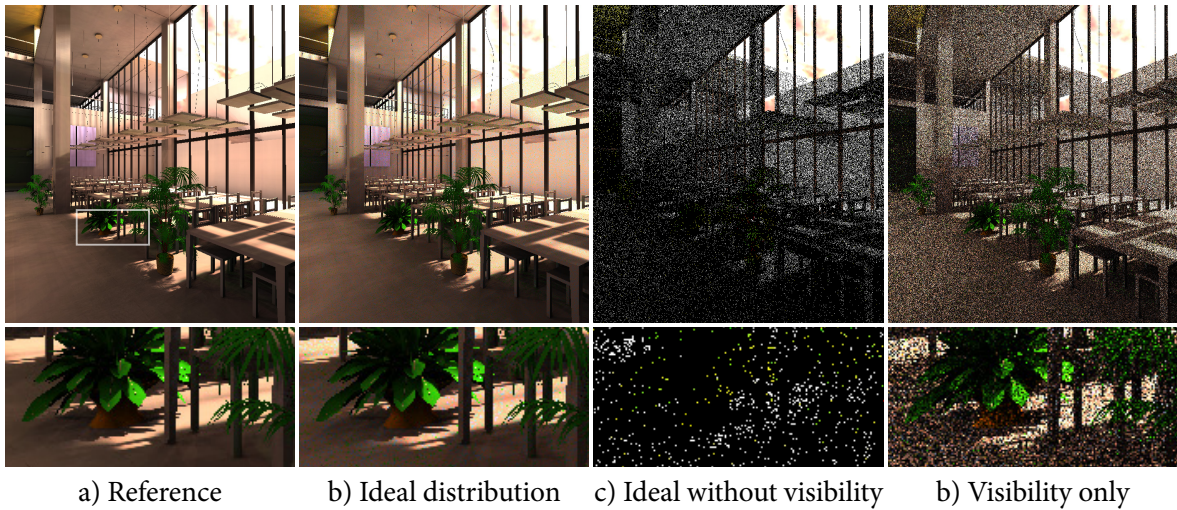


Figure 6.1: A scene lit by an environment map, with direct and indirect illumination approximated by 6000 VPLs. a) Rendering all VPLs gives the exact result. b) Choosing a single random VPL per pixel proportionally to its actual contribution results in perfect importance sampling. c) Excluding visibility from this ideal distribution leads to excessive variance. d) Using only visibility as an importance distribution yields a better result, highlighting the importance of accounting visibility when choosing the VPL.

can destroy proportionality and lead to a dramatic increase in variance. Moreover, with this approach the total rendering time is still dominated by the construction of the distribution.

Following the above observations, we propose a solution that:

1. samples VPLs from a distribution that accounts for visibility,
2. exploits illumination coherence to amortize the distribution construction over many pixels, and
3. remains robust at discontinuities.

6.1.1 Importance caching

Our approach is based on the idea of taking a set of evaluated VPL contributions at one location and reusing these evaluations in the form of *importance* at other locations. At every rendering iteration, we first perform full VPL evaluation at a sparse set of surface points \mathbf{r}_j generated by tracing random rays from the eye. Each of these *importance records* (IRs) stores the contributions $h(\mathbf{r}_j, \mathbf{v}_k)$ of all VPLs \mathbf{v}_k . The set of contributions at \mathbf{r}_j is then normalized to a probability distribution $P_j(\mathbf{v}_k)$ for sampling VPLs at nearby surface points. P_j represents the *fully* evaluated VPL contributions, including visibility. Reusing this distribution at other locations allows for exploiting illumination coherence in an unbiased way. Thus, instead of constructing a costly and only approximate distribution at every shading point \mathbf{x} , we create ideal distributions at a small number of points – the importance records. These distributions are stored in memory as easy-to-build one-dimensional discrete cumulative distribution functions (CDFs). The full algorithm is graphically illustrated in Figure 6.2.

During the final rendering stage, at each eye shading point \mathbf{x} we first find the R nearest importance records \mathbf{r}_j according to the following distance metric:

$$d(\mathbf{x}, \mathbf{r}_j) = \|\mathbf{x} - \mathbf{r}_j\| + \lambda \sqrt{1 - n_{\mathbf{x}} \cdot n_{\mathbf{r}_j}}, \quad (6.4)$$

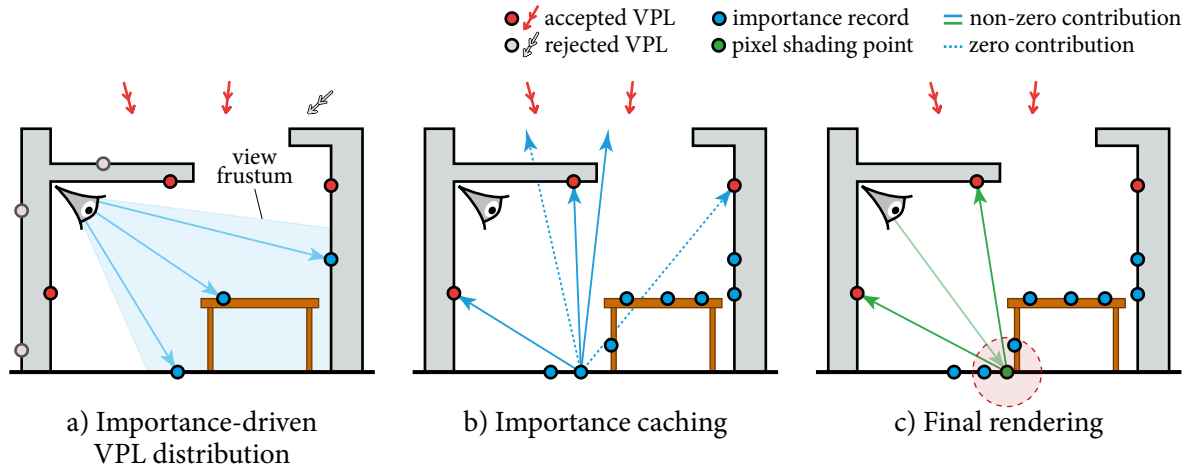


Figure 6.2: Overview of our algorithm. a) We begin by distributing a set of VPLs in the scene using the importance-driven method from Chapter 5. b) We then evaluate the contributions of all VPLs to every importance record (IR) sampled from the eye, and derive a few VPL importance distributions from these evaluations. c) During the final rendering stage, at each shading point we use the distributions gathered from the few closest IRs to importance sample the VPL contributions at that point.

where $n_{\mathbf{x}}$ and $n_{\mathbf{r}_j}$ are the surface normals at \mathbf{x} and \mathbf{r}_j , respectively. Similar metrics are common in illumination interpolation methods, e.g. irradiance caching [157], where λ trades off the importance of Euclidean distance and difference in surface orientation. The IRs are organized in a range-search kd-tree. Lookups are performed using a two-step filtering approach similar to the one of Clarberg and Akenine-Möller [11]. We set the nearest neighbor count to $R = 3$ and use $\lambda = 0.5/D$ where D is the scene’s bounding box diagonal. We use the distributions from the closest IRs to evaluate the pixel estimator \hat{I}^{IC} (Eq. 6.1). This way, the pre-sampling setup for \mathbf{x} is reduced to a nearest neighbor search.

When reused at a spatially close location \mathbf{x} , we can expect that, while not ideal anymore, the distribution $P_j(\mathbf{v}_k)$ from an importance record \mathbf{r}_j will most often be closely proportional to $h(\mathbf{x}, \mathbf{v}_k)$. This leads to good importance sampling in such cases. However, the proportionality decreases at discontinuities, which can in turn significantly increase the variance. We combat this problem in two ways.

When reusing P_j across an illumination discontinuity, important VPL contributions at \mathbf{x} might be assigned low (sometimes even zero) sampling probability, e.g. if a light source was in shadow at \mathbf{r}_j . Therefore, in addition to this aggressive distribution, at each IR \mathbf{r}_j we build three more increasingly conservative distributions derived from $h(\mathbf{r}_j, \mathbf{v}_k)$ that are less likely to miss important new VPL contributions in the vicinity of \mathbf{r}_j . We devise these sampling distributions by identifying the different situations that cause changes in VPL contribution due to variations in position and orientation (Sec. 6.2).

Second, we combine the many distributions borrowed from nearby IRs around each shading point \mathbf{x} in a way that tries to preserve the qualities of each distribution. We perform a “bilateral” multiple importance sampling combination using a novel α -max heuristic that weights the distributions according to a specified prioritization (Sec. 6.3).

Our method does not distinguish between the different kinds of VPLs, e.g. finite (from area lights) and infinite (from environment lights) or direct and indirect. This way we effectively importance sample the *total* incident illumination at every pixel shading point. In addition, thanks to the unbiased exploitation of coherence, we can discard all cached data at the end of every rendering iteration to obtain a high-fidelity solution by progressively averaging independently rendered images.

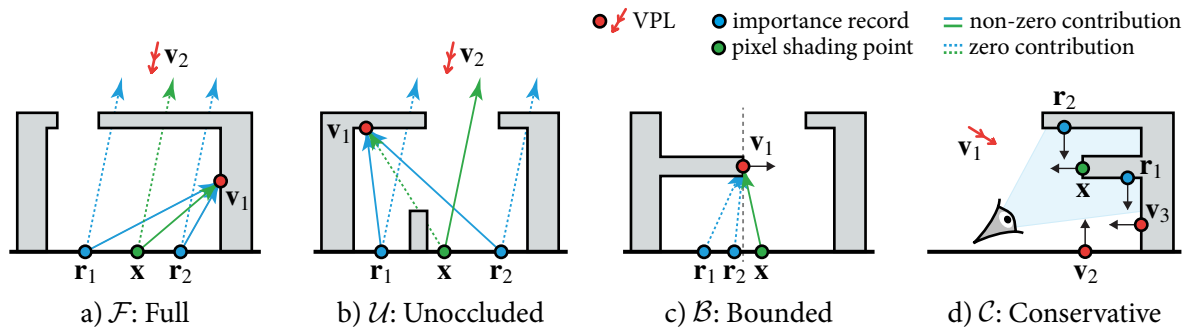


Figure 6.3: Four situations encountered when reusing data from the importance records (IRs) \mathbf{r}_1 and \mathbf{r}_2 at the shading point \mathbf{x} . At each IR we define four distributions designed to discover important VPL contributions in each situation. a) In the ideal case of locally smooth illumination, full contribution sampling (\mathcal{F}) can achieve very close proportionality. b) The unoccluded distribution (\mathcal{U}) is robust to VPL contribution changes due to variation in occlusion. c) Bounded contribution sampling (\mathcal{B}) in addition tries to discover new important contributions due to orientation changes. d) The conservative uniform sampling (\mathcal{C}) handles the rare cases where the importance information at the nearest IRs is irrelevant at the shading point \mathbf{x} .

6.2 VPL sampling distributions

Recall that each importance record (IR) \mathbf{r}_j stores the local contributions of all VPLs \mathbf{v}_k as a normalized discrete probability distribution $P(\mathbf{v}_k | \mathbf{r}_j)$. Given a nearby shading point \mathbf{x} , if the illumination is locally smooth, it is likely that $P(\mathbf{v}_k | \mathbf{r}_j)$ will be roughly proportional to $h(\mathbf{x}, \mathbf{v}_k)$, leading to good importance sampling. However, in regions around illumination discontinuities there can be little or no correlation, leading to excessive variance. High variance is mainly caused by high-energy regions in $h(\mathbf{x}, \mathbf{v}_k)$ not being present in $h(\mathbf{r}_j, \mathbf{v}_k)$ and consequently sampled with low probabilities.

We identify the causes for changes in VPL contributions between surface points, and at each IR we build four increasingly conservative importance distributions from the VPL evaluations. These distributions help the sampling remain robust when reusing importance across discontinuities.

6.2.1 \mathcal{F} : Full contribution

As discussed above, the most straightforward distribution to define at each IR \mathbf{r}_j is the one obtained by normalizing the contributions $h(\mathbf{r}_j, \mathbf{v}_k)$ of all VPLs. This full contribution distribution (Fig. 6.3a), which we denote by $\mathcal{F}(\mathbf{v}_k | \mathbf{r}_j)$, is our most aggressive distribution and is also the one with the highest importance sampling potential. It can in fact achieve perfect proportionality in some cases, e.g. on flat diffuse surfaces illuminated by unoccluded (infinite) directional light sources, in which case $h(\mathbf{x}, \mathbf{v}_k)$ is independent of the position \mathbf{x} . \mathcal{F} often discovers the largest fraction of the energy at \mathbf{x} as we will show in Section 6.4 below.

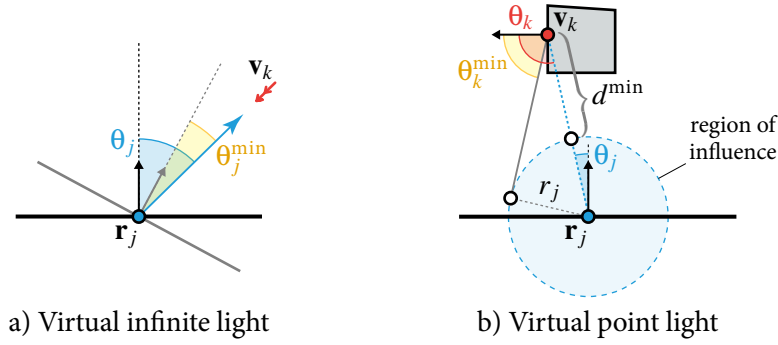


Figure 6.4: Upper bounds for the geometry term in the connection throughput between a VPL \mathbf{v}_k and points within the region of influence of an importance record \mathbf{r}_j (see Section 6.2.3).

6.2.2 \mathcal{U} : Unoccluded contribution

It sometimes happens that all IRs \mathbf{r}_j in the region around a shading point \mathbf{x} agree on the importance of a particular VPL \mathbf{v}_k , though inconsistently with its actual contribution at \mathbf{x} . In scenes with complex geometry distribution, these inconsistencies are often caused by changes in the visibility term, i.e. $V(\mathbf{x}, \mathbf{v}_k) \neq V(\mathbf{r}_j, \mathbf{v}_k)$. False positives (e.g. \mathbf{v}_1 in Figure 6.3b) increase variance but are usually not too problematic as their contribution at \mathbf{x} is simply zero. On the other hand, if a false negative (e.g. \mathbf{v}_2 in Figure 6.3b) is sampled with a low probability, variance can explode.

The unoccluded contribution distribution $\mathcal{U}(\mathbf{v}_k | \mathbf{r}_j)$ is a conservative modification of \mathcal{F} that is designed to discover VPL contributions potentially missed by \mathcal{F} due to changes in visibility. This distribution is also built from the VPL contributions at \mathbf{r}_j but assuming full visibility, i.e. by normalizing the unoccluded contributions $B(\mathbf{r}_j, \mathbf{v}_k)L(\mathbf{r}_j, \mathbf{v}_k)G(\mathbf{r}_j, \mathbf{v}_k)$ of all VPLs. The resulting sampling technique is particularly good at finding small-scale illumination features, such as a bright spot on a surface produced by a thin light beam passing through a small window or a collection of leaves (see Figure 6.6).

6.2.3 \mathcal{B} : Bounded contribution

Sometimes all IRs \mathbf{r}_j around \mathbf{x} may falsely suggest a low or zero contribution of \mathbf{v}_k due to differences in surface orientation. Figure 6.3c illustrates such a case, where $G(\mathbf{x}, \mathbf{v}_k) > G(\mathbf{r}_j, \mathbf{v}_k)$. The actual contribution $h(\mathbf{x}, \mathbf{v}_k)$ might in fact be small, but if sampled with a too low probability, variance can increase.

The bounded contribution distribution $\mathcal{B}(\mathbf{v}_k | \mathbf{r}_j)$ is a conservative extension of \mathcal{U} targeting orientation-induced false negatives. Knowing that each IR will be used in a small local neighborhood, we build the \mathcal{B} distribution from the contributions $B(\mathbf{r}_j, \mathbf{v}_k)L(\mathbf{r}_j, \mathbf{v}_k)G^{\max}(\mathbf{r}_j, \mathbf{v}_k)$. Here, $G^{\max}(\mathbf{r}_j, \mathbf{v}_k)$ is the upper bound for the geometry term between the VPL \mathbf{v}_k and points \mathbf{x} in the neighborhood of the importance record \mathbf{r}_j (see Figure 6.4). If \mathbf{v}_k is a sample on an infinite light source, we have $G^{\max}(\mathbf{r}_j, \mathbf{v}_k) = \cos \theta_j^{\min}$, where θ_j^{\min} is the minimum angle between $n_{\mathbf{r}_j}$ and the VPL direction in the region of influence of \mathbf{r}_j , whose estimation we describe below. Otherwise, \mathbf{v}_k is a surface point and we need to find an upper bound for the full geometry term $G(\mathbf{r}_j, \mathbf{v}_k) = \frac{\cos \theta_j \cos \theta_k}{d^2}$, where $d = \|\mathbf{r}_j - \mathbf{v}_k\|$.

We bound each term in $G(\mathbf{r}_j, \mathbf{v}_k)$ individually, i.e. we compute $G^{\max}(\mathbf{r}_j, \mathbf{v}_k) = \frac{\cos \theta_j^{\min} \cos \theta_k^{\min}}{(d^{\min})^2}$. For this we need the radius r_j of the region of influence of \mathbf{r}_j . We estimate r_j by dividing the radius of the eye ray

footprint at \mathbf{r}_j by the screen-space IR density. The distance bound is then $d^{\min} = \max(0, d - r_j)$, where $d = \|\mathbf{r}_j - \mathbf{v}_k\|$. For the bound θ_k^{\min} we find the maximum change $\Delta\theta_k = -\text{asin}(r_j/d)$. Is it difficult to find a bound for θ_j without analyzing the geometry surrounding \mathbf{r}_j . We compute θ_j^{\min} using a maximum change of $\Delta\theta_j = -30^\circ$, which worked better in our tests than the conservative $\theta_j^{\min} = 0^\circ$.

The \mathcal{B} distribution usually finds VPLs with small contributions. It does so, however, with a sufficiently high probability to avoid excessive variance (see middle row in Figure 6.6).

6.2.4 \mathcal{C} : Conservative distribution

When the nearest IRs around \mathbf{x} are far way in terms of both position and orientation, false negatives can occur that neither \mathcal{F} , \mathcal{U} , or \mathcal{B} handle. An example is shown in Figure 6.3d. Such situations arise when strong illumination falls on a surface that has a small screen-space footprint and unique orientation in its neighborhood. For such rare cases where no useful importance information can be reliably extracted from any of the nearest IR, we use the most conservative uniform sampling distribution \mathcal{C} which assigns equal probability to all VPLs.

We can still detect such potentially problematic cases by looking at the distances (Eq. 6.4) between the shading point \mathbf{x} and its closest IRs. If the average distance is above a certain threshold, we increase the number of VPLs chosen from \mathcal{C} . This decreases variance in a brute-force yet adaptive way.

6.3 Bilateral combination of sampling distributions

During the final rendering stage, at every pixel shading point \mathbf{x} we find the R nearest importance records (IRs) and use the four distributions cached at each of them to evaluate the \hat{I}^{IC} estimator in Equation 6.1. We combine the resulting estimates by transforming \hat{I}^{IC} into a multiple importance sampling (MIS) estimator. While MIS was originally developed for continuous distributions, it trivially applies to the discrete case as well. In order to avoid bias we must ensure that every VPL \mathbf{v}_k with a non-zero contribution is chosen with a non-zero probability; however, all \mathcal{F} , \mathcal{U} , and \mathcal{B} can have zeros. Fortunately, MIS does not require each individual distribution to be non-zero everywhere [142]. Thus, including \mathcal{C} is sufficient to guarantee the unbiasedness of the resulting combined estimator.

Once we have gathered $4R$ distributions from the R nearest IRs, we can construct an MIS estimator by choosing an appropriate weighting heuristic. Figure 6.5a shows a matrix arrangement of the distributions cached at the $R = 3$ nearest IRs around the red-marked point in Figure 6.6. As expected, the three distributions in each row correlate closely, while the four increasingly conservative distributions in each column have dissimilar structures. Because Veach's [142] MIS heuristics are solely based on sampling probabilities, they cannot exploit this additional information, which may hint to when a particular distribution is better than the others. Choosing one of Veach's heuristics to weight all $4R$ distributions can thus be sub-optimal.

We propose a bilateral, two-stage combination of the distributions in the matrix using different heuristics schemes for the rows and columns in order to better preserve the qualities of each distribution. We first construct an MIS estimator that mixes the R distributions in each row. This way, we conceptually collapse all columns into one. The four distributions in this resulting column are subsequently combined using a novel heuristic to form the final MIS estimator.

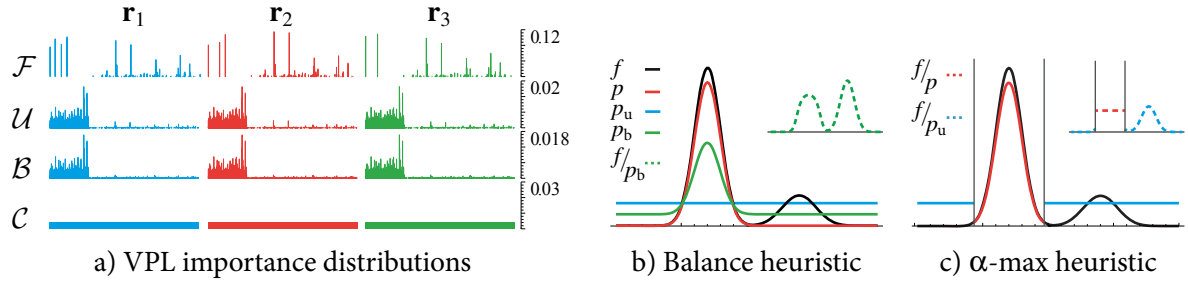


Figure 6.5: Combining VPL sampling distributions. a) A matrix arrangement of the four distributions (columns) at the three closest records (rows) to the red-marked point in the left image in Figure 6.6. Notice the correlation among the columns. Direct illumination VPLs are in the beginning of the VPL list, hence the higher probabilities in the left. b) A balance heuristic mixture p_b of a distribution p and a defensive uniform distribution p_u ruins the local proportionality of p to the integrand f in the left part of the domain, while still resulting in lower than optimal probabilities outside the peak. c) By adaptively partitioning the sampling domain, our novel α -max heuristic avoids mixing and instead selects for each partition the distribution with closest proportionality in a user-controlled way.

6.3.1 Column combination

An MIS pixel estimator that combines all R distributions in row i using a column weighting heuristic $w_{i,j}^{\text{col}}$ reads:

$$\hat{I}_i^{\text{IC}} = \sum_{j=1}^R \frac{R}{n_i} \sum_{k=1}^{n_i/R} w_{i,j}^{\text{col}}(\mathbf{v}_{i,j,k}) \frac{h(\mathbf{x}, \mathbf{v}_{i,j,k})}{P_i(\mathbf{v}_{i,j,k} | \mathbf{r}_j)}, \quad i = 1, \dots, 4 \quad (6.5)$$

where $P_1 = \mathcal{F}$, $P_2 = \mathcal{U}$, $P_3 = \mathcal{B}$, $P_4 = \mathcal{C}$, taking n_i/R samples from each of the R distributions.

Since we do not have reliable means to detect all discontinuities, e.g. shadow penumbrae, we cannot determine which IRs correlate most with the shading point \mathbf{x} . Therefore, we avoid aggressive weighting for combining the columns j and opt for the safest combination – the balance heuristic:

$$w_{i,j}^{\text{col}}(\mathbf{v}) = \frac{P_i(\mathbf{v} | \mathbf{r}_j)}{\sum_{l=1}^R P_i(\mathbf{v} | \mathbf{r}_l)}. \quad (6.6)$$

The balance heuristic corresponds to sampling from the mixture distribution (see Section 2.5.6)

$$P_i(\mathbf{v}) = \frac{1}{R} \sum_{l=1}^R P_i(\mathbf{v} | \mathbf{r}_l). \quad (6.7)$$

This combination can be interpreted as importance interpolation at the shading point \mathbf{x} that gives equal weight to all IRs. We can construct the mixture CDF for P_i implicitly during sampling, by traversing the individual CDFs in a synchronized binary search and averaging their elements. Compared to sampling from each component separately, mixture sampling requires R times fewer random numbers and facilitates sample stratification. Equation 6.5 now simplifies to an ordinary secondary Monte Carlo estimator:

$$\hat{I}_i^{\text{IC}} = \frac{1}{n_i} \sum_{k=1}^{n_i} \frac{h(\mathbf{x}, \mathbf{v}_{i,k})}{P_i(\mathbf{v}_{i,k})}. \quad (6.8)$$

6.3.2 Row combination

In order to obtain the final pixel estimator, we combine the four row estimators \hat{I}_i^{IC} using a row weighting heuristic w_i^{row} :

$$\hat{I}^{\text{IC}} = \sum_{i=1}^4 \frac{1}{n_i} \sum_{k=1}^{n_i} w_i^{\text{row}}(\mathbf{v}_{i,k}) \frac{h(\mathbf{x}, \mathbf{v}_{i,k})}{P_i(\mathbf{v}_{i,k})}. \quad (6.9)$$

For w_i^{row} the balance heuristic is not necessarily the best choice. Each of the \mathcal{F} , \mathcal{U} , \mathcal{B} , and \mathcal{C} techniques is designed to achieve as close as possible proportionality, i.e. to make the problem *low-variance* [142], in different situations. Figure 6.5b illustrates that averaging a locally proportional distribution and a uniform distribution can destroy the qualities of both over the whole domain. In such cases, more aggressive heuristics can perform better, e.g. the power or max heuristics [142]. These heuristics weight the techniques proportionally to their probabilities under the assumption that higher probabilities lead to lower variance. While this approach avoids extreme variance, it can still ruin local proportionality. This can be seen in Figure 6.5c, where the uniform distribution p_u has higher probability around the boundaries of the middle partition, yet p alone results in a zero-variance estimator for that entire partition.

The α -max heuristic

We often have additional knowledge about the distributions, e.g. we expect certain distributions to perform well more often than others. We propose an α -max heuristic to exploit this knowledge:

$$w_s^\alpha(x) = \begin{cases} 1, & \text{if } w_i^\alpha(x) = 0, \text{ for } 1 \leq i < s, \text{ and } p_s(x) \geq \max_{s < i \leq n} \alpha_i p_i(x), \\ 0, & \text{otherwise,} \end{cases} \quad (6.10)$$

where x has been sampled from p_s , and each of the n distributions p_i has an associated priority index i and confidence value $\alpha_i \in (0; 1]$. Setting all confidence values $\alpha_i = 1$ gives Veach's max heuristic, while α_1 is a redundant parameter that we set to 1.

The α -max heuristic partitions the sampling domain by assigning each distribution to regions where its probability is larger than that of the lower-priority distributions scaled by their confidence values. This controlled partitioning can better preserve proportionality if some a-priori information about the distributions is available. This can lead to more effective variance reduction than heuristics based solely on probabilities. Figure 6.5c right demonstrates this, with the ordering $(\alpha_p, \alpha_{p_u}) = (1, 0.2)$. It also illustrates how our α -max heuristic, with α_{p_u} being a trade-off parameter, can be used as an alternative to defensive importance sampling [51] which also suffers from mixture proportionality deterioration. This deterioration can be remedied effectively by the α -max heuristic, as we will further demonstrate in Section 6.4.

For the distributions from Section 6.2 we use the ordering $\mathcal{F}, \mathcal{U}, \mathcal{B}, \mathcal{C}$. This prioritization follows the increasing conservativeness of the distributions, ranking \mathcal{F} highest.

6.3.3 Distribution optimization

Veach [142] argues that zero-weight heuristics waste computation on generating samples that are subsequently ignored. However, the α -max heuristic can in fact be more computationally efficient than,

Method	Preprocess			Rendering		
	VPL	CDF	Sampling	Shading	Ray tracing	Other
Our importance caching	3%	14%	30%	3%	42%	8%
Resampled importance sampling	0%	-	3%	70%	24%	3%

Table 6.1: Fractional time break-down for the $FUBC^\alpha$ and RIS images in Figure 6.6 rendered in 20 sec. The “Other” column includes the time for eye ray tracing and nearest importance record search.

e.g., the power heuristic. Since weights are based on cheap to compute sampling probabilities, we can avoid the costly contribution evaluation (which involves ray tracing) of VPLs whose MIS weight is zero.

The above observation can be further exploited to avoid generating samples that would end up being discarded in the first place. This can be achieved by reducing the redundancy, i.e. overlap, of the distributions being combined. To this end, we modify the distributions at each IR \mathbf{r}_j right after their construction as follows. For every VPL \mathbf{v}_k and every P_i (corresponding to $\mathcal{F}, \mathcal{U}, \mathcal{B}, \mathcal{C}$), we multiply $P_i(\mathbf{v}_k | \mathbf{r}_j)$ by $w_i^\alpha(\mathbf{v}_k)$ and then re-normalize the corresponding CDF. This way we partition the sampling domain (the set of VPLs \mathbf{v}_k), so that exactly one of the four distributions at \mathbf{r}_j has a non-zero value for every VPL. This in turn increases the sampling probabilities after re-normalization. Note that since the optimization is performed at each record independently, it does not entirely eliminate all redundancies in the final mixture distributions (Eq. 6.7) used during the final rendering, where the α -max weighting is applied again.

6.4 Results

We implemented our importance caching method in a basic ray tracer and performed tests on a mid-range 4-core Intel Core i7-860 CPU. The images in each comparison were rendered in equal time of 20 sec at 1024×768 resolution. All images were produced by progressively averaging three independent rendering iterations, each using 2700 importance records and 8000 VPLs. Table 6.1 summarizes the average fractional time spent in the different steps of our algorithm compared to resampled importance sampling (explained below).

6.4.1 Technique comparison

Figure 6.6 shows the STUDY HALL scene illuminated by the high frequency St. Peter’s cathedral environment map, rendered with full global illumination using different methods. The configuration of this scene allows us to compare all sampling distributions and combination strategies as it exhibits a variety of lighting conditions. These include smooth direct and indirect illumination, small-scale lighting and geometric features, as well as occlusion.

Each label in Figure 6.6 denotes the distributions and weighting heuristic used to produce the corresponding image. β denotes the power heuristic (Eq. 2.47) with $\beta = 2$, ‘d’ denotes defensive importance

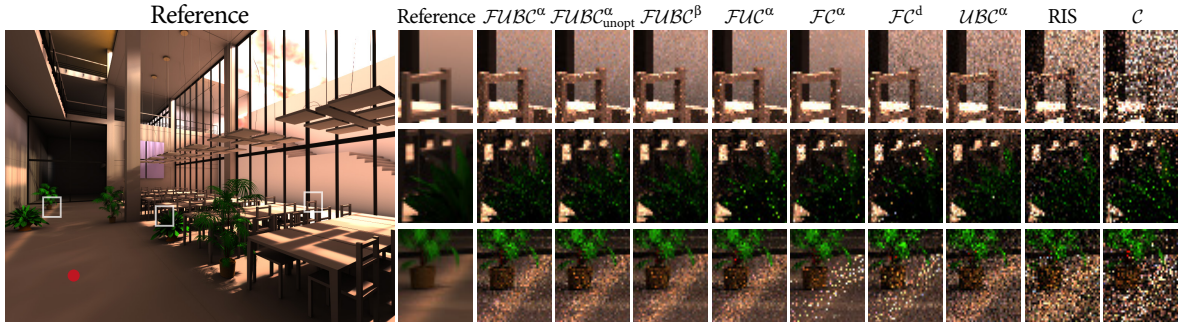


Figure 6.6: A comparison of various rendering configurations on a scene with small-scale geometry features illuminated by a high-frequency environment map. The label above each zoom-in denotes the sampling distributions and combination heuristic employed. For each image we sample 35 VPLs per pixel per progressive rendering iteration, distributed manually among the distributions to achieve the best result. The reference image was rendered using $FUBC^\alpha$ in 1000 sec, while the other images took 20 sec.

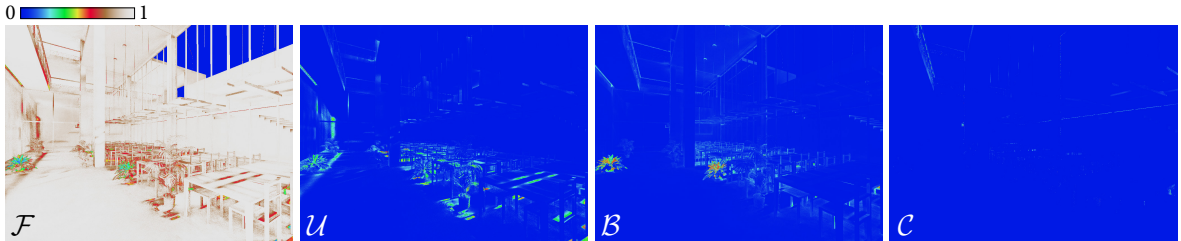


Figure 6.7: Fractional contributions of the individual VPL sampling distributions to the reference image in Figure 6.6. With the so chosen α -max parameters, \mathcal{F} contributes 93.8% of the energy in the image, \mathcal{U} and \mathcal{B} account for 3% each, leaving 0.2% of the contribution to \mathcal{C} .

sampling with 0.4 weight for the conservative (uniform) distribution \mathcal{C} , and α denotes the use of our α -max heuristic with $(\alpha_{\mathcal{F}}, \alpha_{\mathcal{U}}, \alpha_{\mathcal{B}}, \alpha_{\mathcal{C}}) = (1, 0.5, 0.5, 0.3)$.

Overall, the full contribution sampling technique \mathcal{F} performs remarkably well in regions with smooth illumination, as it exploits any available coherence (Fig. 6.6 insets). In such regions it samples the VPL contribution function $h(\mathbf{x}, \mathbf{v}_k)$ with close proportionality, yielding a low-variance estimate. However, this proportionality is ruined when averaging \mathcal{F} with other distributions. The power heuristic (Fig. 6.6, $FUBC^\beta$) and especially defensive importance sampling (Fig. 6.6, FC^d) increase variance over α -max in such regions by 45% and 330% respectively.

When \mathcal{F} misses important contributions it is often due to changes in occlusion between the locations of the importance records and the shading point. While \mathcal{C} does not provide high enough sampling probabilities, \mathcal{U} is more efficient at discovering new unoccluded VPLs, as seen when comparing the FC^α and FUC^α images (particularly the bottom row in Figure 6.6).

The $FUBC^\beta$ and $FUBC^\alpha_{\text{unopt}}$ technique combinations do not use distribution optimization scheme from Section 6.3.3. As a result, \mathcal{B} does not capture some orientation-induced false negatives effectively because it is too conservative. After distribution optimization, the average increase in probability for \mathcal{F} , \mathcal{U} , \mathcal{B} , \mathcal{C} is respectively 0.05%, 20%, 454%, 50%. The significant increase in \mathcal{B} is mostly due to the eliminated redundancies with \mathcal{U} (the redundancies can be seen in Figure 6.5a). The best result is then produced by $FUBC^\alpha$, with low-variance regions of smooth illumination and no “firefly” pixels.

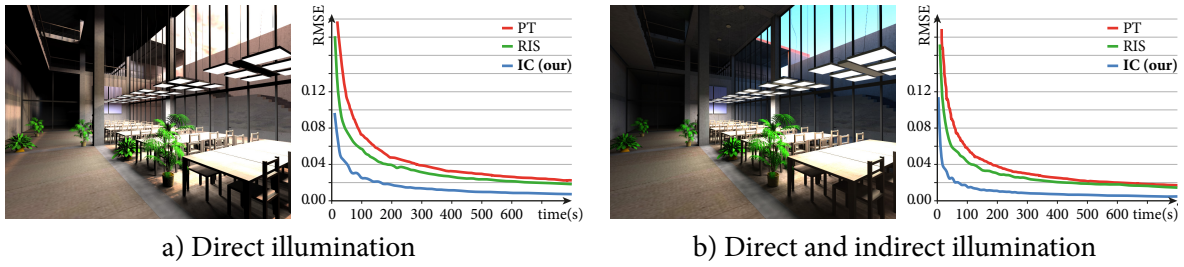


Figure 6.8: The *STUDY HALL* scene illuminated by 36 area lights and an environment map. The RMSE convergence plots compare our importance caching (IC) algorithm against path tracing (PT) and resampled importance sampling (RIS). a) Direct illumination only, using the St. Peter’s cathedral environment map. b) Full global illumination, using a sunset environment map.

The benefit of including occlusion in the VPL sampling distributions can be seen when comparing the $FUBC^\alpha$ image to the UBC^α image whose variance is $10\times$ higher. We also compare against resampled importance sampling (RIS), as well as against uniform VPL sampling (i.e. \mathcal{C} alone) which produces a similar result to path tracing. For RIS, we first evaluate the unoccluded contributions of S randomly chosen VPLs at every shading point and then draw $S/20$ samples from an on-the-fly constructed CDF over these evaluations. Even though UBC^α relies on sometimes inaccurate borrowed distributions, it remains superior to RIS as it utilizes importance information for *all* VPLs, while RIS chooses the initial S candidates blindly with a uniform probability. Increasing the RIS candidate count S has a negative impact on the computational performance without delivering a noticeable improvement in image quality.

The reference image in Figure 6.6 has been produced using our $FUBC^\alpha$ method with a large number of rendering iterations. Figure 6.7 visualizes the fractional contributions of the individual distributions to that image. \mathcal{F} achieves close proportionality almost everywhere and consequently contributes almost 94% of the total energy in the image. \mathcal{U} and \mathcal{B} capture almost all of the remaining energy (3% each), leaving a very small fraction, 0.2%, to \mathcal{C} . It is important to note that \mathcal{U} , \mathcal{B} , and \mathcal{C} together require about $6\times$ more samples than \mathcal{F} due to their increasing conservativeness (which is slightly ameliorated by the optimization scheme from Section 6.3.3). Still, the overall energy contribution to the image from these distributions is comparatively small.

6.4.2 Numerical convergence

In Figure 6.8 we demonstrate the ability of our importance caching method to simultaneously handle large numbers and different types of light sources. We also measure its error convergence against path tracing and resampled importance sampling for direct and full global illumination separately. The graphs on the right of each image compare the root mean squared error (RMSE) of the three methods. Due to the geometry term clamping in the connection throughput of indirect illumination VPLs (Sec. 4.3.2), in order to produce meaningful plots the path tracer has been slightly modified so that all three methods converge to the same biased solution. In both cases, IC consistently outperforms PT and RIS with an average variance reduction of $9\times$ and $4.8\times$ respectively for the left image and $25\times$ and $9\times$ for the right image.

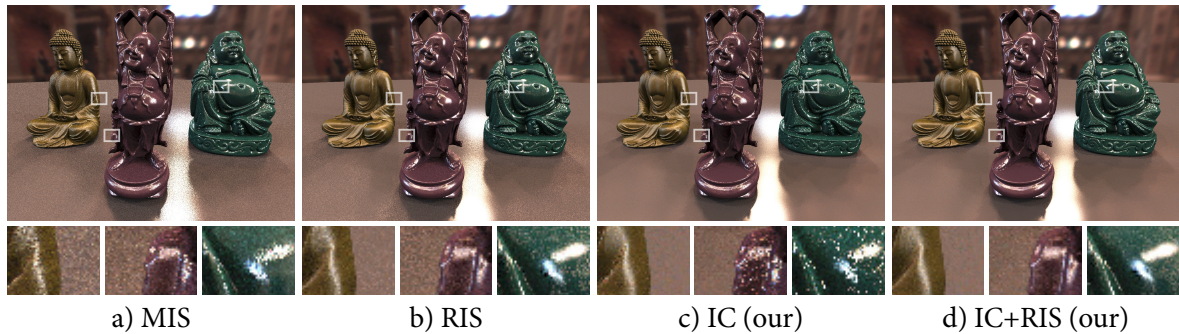


Figure 6.9: Three glossy Buddha statues with Phong BxDF exponents 10, 80, 300 (from left to right) on a glossy ground with exponent 500, rendered in 20 sec using: a) MIS-combined BxDF/environment importance sampling; b) Resampled importance sampling (RIS); c) Our importance caching algorithm (IC); d) Combined IC and RIS (IC+RIS). While this scene is a failure case for our IC method, it still outperforms RIS and MIS in regions with occlusion and/or smooth illumination. The IC+RIS combination achieves the best overall quality.

6.4.3 Glossy materials

Figure 6.9 shows three Buddha statues with increasingly specular Phong BxDFs [86] illuminated by the St. Peter’s cathedral environment map. We compare our importance caching algorithm to resampled importance sampling with the same sampling setup we use for the STUDY HALL scene (Sec. 6.4.1), as well as to an MIS combination of BxDF and environment map importance sampling.

Our importance caching (IC) method is not as well suited to scenes with glossy materials due to the less available coherence to exploit – a result of the high frequencies in the geometry, BxDFs, and illumination. We can still notice that in occluded regions and in regions with low-to-mid frequency BxDFs and geometry distributions, IC outperforms the other two methods, producing very smooth results. However, with highly glossy BxDFs even slight orientation changes can make the importance record data invalid at nearby shading points. In such situations the RIS and MIS algorithms perform better than IC.

Based on the above observations we devised the following heuristic combination of IC and RIS. We first compute the average “BxDF distance” between \mathbf{x} and its R nearest importance records \mathbf{r}_j :

$$d^{\text{BxDF}} = 1 - \frac{1}{R} \sum_{j=1}^R \frac{\rho_s(\mathbf{x}, \bar{\omega}_{\mathbf{r}_j}, \omega_{\mathbf{x}})}{\rho_s(\mathbf{x}, \bar{\omega}_{\mathbf{x}}, \omega_{\mathbf{x}})}, \quad (6.11)$$

where $\omega_{\mathbf{x}}$ is the direction toward the eye, $\bar{\omega}_{\mathbf{x}}$ is its reflection about the surface normal at \mathbf{x} , and $\bar{\omega}_{\mathbf{r}_j}$ is the reflection of the direction toward the eye at \mathbf{r}_j about the surface normal at \mathbf{r}_j . This metric captures orientation differences between the glossy reflection lobes while treating the BxDFs as black boxes. If d^{BxDF} is below 0.5 we use our IC method to sample VPLs at \mathbf{x} , otherwise we use RIS. The combined IC+RIS algorithm (Fig. 6.9d) preserves the qualities of both IC and RIS by heuristically choosing the better method locally at each shading point. Note that this approach further reduces the noise on the statues, because RIS, which is now used more sparingly across the image, can take more samples in the given time.

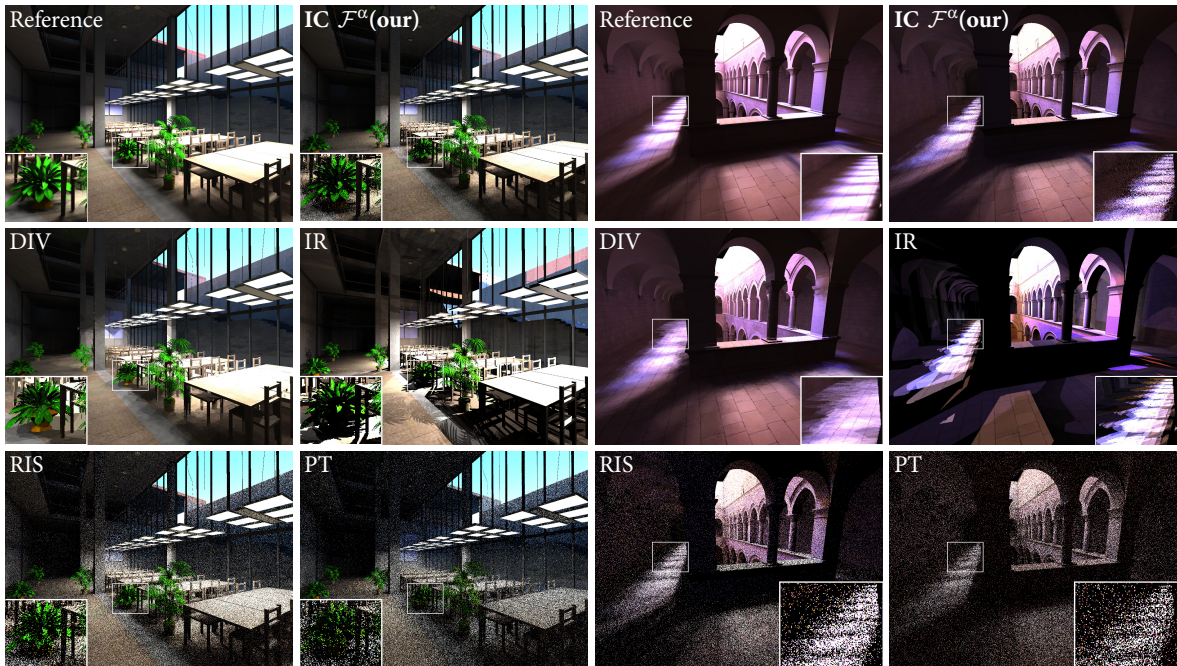


Figure 6.10: A 2-second full global illumination preview quality comparison on the STUDY HALL (left) and SPONZA (right) scenes. Our importance caching algorithm ($IC \mathcal{F}^\alpha$), which here only employs an α -max clamped \mathcal{F} distribution, produces a low-noise image very similar to the reference. Even without the \mathcal{U} , \mathcal{B} , and \mathcal{C} techniques, our method handles orientation and visibility discontinuities reasonably well, which are blurred away by direct importance visualization (DIV). Traditional instant radiosity (IR) suffers from structured noise, while the resampled importance sampling (RIS) and path tracing (PT) images are plagued by uncorrelated noise.

6.4.4 High-quality preview rendering

The results in Figure 6.7 suggest that we can obtain a quick, slightly biased but low-variance image estimate by using a small number of VPLs selected only from distribution \mathcal{F} . The implementation then only needs to store this one distribution at each IR. The α -max heuristic can still be utilized to avoid excessive variance, which in the case of sampling only from \mathcal{F} corresponds to simply discarding samples with probabilities below a certain threshold. If desired, the introduced bias can be compensated for in a controlled way by selectively including \mathcal{U} , \mathcal{B} or \mathcal{C} in the sampling process.

Figure 6.10 demonstrates the high quality achieved with one rendering iteration in 2 seconds using the above described method, $IC \mathcal{F}^\alpha$, on the STUDY HALL and SPONZA scenes, compared to RIS and PT, as well as to traditional instant radiosity [73]. We also include direct importance visualization (DIV) images which are produced by simply interpolating the irradiance cached at the nearest importance records around each shading point. DIV is only slightly faster than $IC \mathcal{F}^\alpha$; the latter evaluates the contribution of only 4 VPLs per pixel, which now only takes about 30% of the total frame time. Although noise-free, DIV blurs away all fine details, while instant radiosity suffers from severe structured noise. On the other hand, RIS and PT produce images with high levels of uncorrelated noise.

6.5 Discussion

Although the sampling framework of our importance caching method is unbiased, the use of VPLs incurs a small systematic error when geometric clamping is used. In addition, distributing VPLs from the light sources becomes very inefficient in the presence of highly glossy BSDFs. Handling such materials with VPLs is a general problem, although solutions have been proposed [78, 47, 22].

Our implementation distributes importance records by uniformly sampling the image plane. We also experimented with distributions that adapt to geometric discontinuities. However, this caused a decrease in IR density in regions of sharp illumination discontinuities, such as shadow penumbrae, leading to a noticeable increase in variance. A better strategy would be to adapt the IR distribution by utilizing variance statistics obtained during rendering, which is an interesting avenue for future work.

To render an image, our algorithm needs about $M \times N_{\text{IR}} \times 4 \times \text{sizeof}(\text{float})$ bytes of memory to store 4 distributions over M VPLs at each of the N_{IR} importance records. For the tests in Figures 6.6, 6.8 and 6.9 this amounted to about 330MB. This memory footprint can be decreased by splitting up the image into tiles that are rendered independently. Alternatively, M and/or N_{IR} can be reduced at the cost of increasing variance.

Vertex Connection and Merging

7

Light transport simulation is a central problem in realistic image synthesis and has been an active area of research for decades. Considerable advances have been made over the years with respect to the efficiency of light transport algorithms. However, such improvements usually come with some sort of bias: Certain types of light interactions are often disregarded or handled inefficiently. For instance, the many-light rendering methods discussed in Chapters 5 and 6 can produce high quality images very quickly but only on scenes with diffuse and not too shiny materials. The limited set of vertex connection techniques employed by these methods necessitates path contribution clamping in order to avoid high-variance image artifacts caused by geometric singularities and glossy BSDFs. The resulting approximations are sometimes acceptable but can lead to severe loss of image fidelity [83]. Such problems can be alleviated to some degree by augmenting many-light methods with more sampling techniques [78, 47, 22, 153, 96, 95], but none of these techniques can efficiently handle caustic paths. Developing truly robust light transport algorithms that can efficiently and accurately capture the global illumination in a variety of scenes remains an important challenge that we address in this chapter.

Bidirectional path tracing (BPT) [85, 143] is among the most versatile light transport algorithms available today in terms of the range of lighting configurations it can handle. The key to its robustness is the provably good combination of various path sampling techniques via multiple importance sampling. However, as we showed in Section 4.4, BPT cannot efficiently capture specular-diffuse-specular (*SDS*) light transport, where the notion of ‘specular’ also includes sharp glossy interactions. This is indeed an important practical limitation because such paths occur in any scene containing specular objects and their image contribution is especially important in some very common cases such as an object enclosed in glass, the interior of a car or a building, etc. The reason for this problem is that the sampling techniques in BPT often find *SDS* paths with very low probability that may even go to zero when point light sources and pinhole cameras are used.

Efficient handling of *SDS* paths, on the other hand, has long been demonstrated with photon mapping (PM) [63]. Recently, progressive photon mapping (PPM) has been shown to converge to the correct solution with a bounded memory footprint [41], as we discussed in Section 4.6. However, PM is inefficient under diffuse lighting and it also has difficulties in scenes with many glossy objects, as has been pointed out by Hašan et al. [47] and Vorba [146]. Together with the relatively low error convergence rate [76], these problems make PPM impractical as a general global illumination solution.

Intuitively, a combination of BPT and PM would be beneficial, since these two methods complement each other in terms of the light transport effects they can efficiently handle, as we saw in Figure 4.12. In fact, PM has traditionally been combined with some of BPT’s path sampling techniques through heuristics such as separating the direct lighting computation, splitting the photons into global and caustic maps, and final gathering. However, as discussed by Veach and Guibas [144], such combination can be far from optimal, and the adaptation of these heuristics to glossy reflectance is not obvious. We

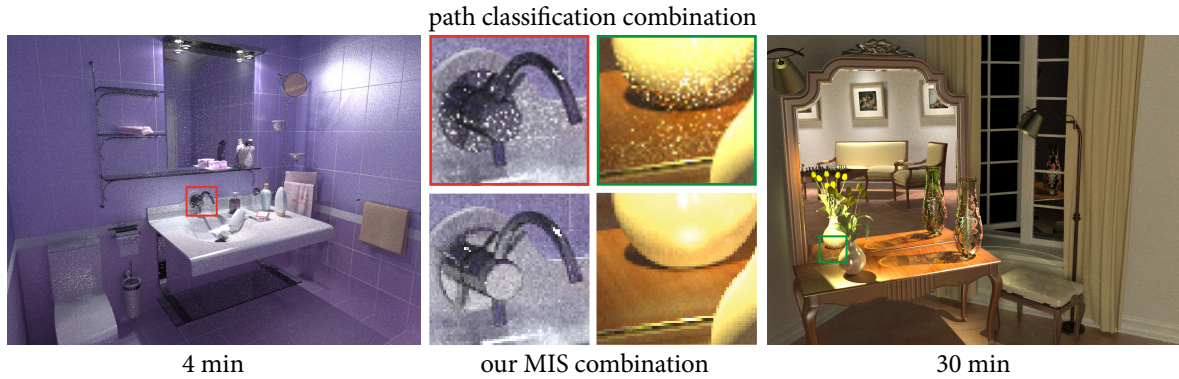


Figure 7.1: Combining BPT and PM via a heuristic classification of paths into caustic and non-caustic (see Section 7.5.1 for explanation) can be far from optimal (top insets). The combination proposed in this chapter is based on multiple importance sampling and can be substantially more robust (bottom insets).

demonstrate this on two scenes in Figure 7.1 where we classify paths into caustic and non-caustic and render these two path types with only PPM and BPT respectively.

Judging from the success of multiple importance sampling (MIS) in improving the robustness of BPT compared to its initial formulation [85, 143] we can expect that an MIS-based combination of BPT and PM would yield a more robust solution than the aforementioned heuristics. Some prior work has already explored the application of MIS to photon mapping. To improve efficiency on scenes with glossy materials, Vorba [146] uses MIS to combine radiance estimates at different eye subpath vertices. However, his path pdf formulation disallows a meaningful MIS combination of PM and BPT estimators, which estimate integrals of different dimension, as we will detail below. Bekaert et al. [3] apply multiple importance sampling on top of their generalized kernel density estimator in a way similar to Vorba [146]. Tokuyoshi [139] improves caustics on glossy surfaces through an MIS-based combination of PM and final gathering estimates. However, a principled and comprehensive combination of BPT and PM has not been shown so far. This is due to important differences in the mathematical frameworks used to formulate these two algorithms – BPT is a Monte Carlo estimator for the path integral and PM is an outgoing radiance estimator based on photon density estimation.

In this chapter, we present a reformulation of photon mapping as a bidirectional path sampling technique for Monte Carlo light transport simulation, circumventing the concept of density estimation altogether. The benefit of this new formulation is twofold. First, it makes it possible to explain, in a formal manner, the relative efficiency of photon mapping and bidirectional path tracing, which have long been considered conceptually incompatible solutions to the light transport problem. Second, it allows us to employ multiple importance sampling to combine BPT and PM in a more robust rendering algorithm that alleviates the problem of insufficient techniques. We demonstrate the efficiency of this combined algorithm in handling a variety of illumination effects ranging from direct illumination and diffuse inter-reflections to *SDS* light transport. We also develop a progressive version of the algorithm that is consistent and retains both the higher $O(1/N)$ mean squared error convergence rate of BPT for most light paths and the efficiency of photon mapping for *SDS* lighting effects.

This chapter is organized as follows. In Section 7.1 we present our reformulation of photon mapping as a bidirectional path sampling technique. In Section 7.2 we analyze the relative efficiency of PM and BPT, which is made possible by this reformulation. In Section 7.3 we present a rendering algorithm that combines techniques from PM and BPT via MIS, which we make progressive and consistent in Section 7.4. Section 7.5 presents an evaluation of this progressive algorithm, followed by a final discussion in Section 7.6.

7.1 Photon mapping as a path integral estimator

Our goal is to combine photon mapping (PM) and bidirectional path tracing (BPT) via multiple importance sampling. The first step is to formulate the two methods in a common mathematical language. For this we choose the path integral formulation of light transport which we introduced in Section 3.5.4 and briefly review below. Since BPT is already defined in this framework, we only need to reformulate the PM method. This involves deriving the integral estimated by PM as well as the corresponding path sampling technique and its associated pdf that we could then plug into the power heuristic.

Without loss of generality, in the following discussion we consider light transport paths of a fixed length k . We start by writing the pixel measurement integral (Eq. 3.27) over all such paths:

$$I_k = \int_{\mathcal{M}^{k+1}} f_k(\bar{\mathbf{x}}) d\mu_{k+1}(\bar{\mathbf{x}}), \quad (7.1)$$

where f_k is the measurement contribution function (Eq. 3.31) for length- k paths $\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$, illustrated graphically in Figure 7.2a. The differential product area measure $d\mu_{k+1}(\bar{\mathbf{x}}) = dA(\mathbf{x}_0) \dots dA(\mathbf{x}_k)$ is defined on the space of all length- k paths $\mathcal{M}^{k+1} = \mathcal{M} \times \dots \times \mathcal{M}$ ($k+1$ times). We will call paths of the form $\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$ *regular paths*, and the function $f_k(\bar{\mathbf{x}})$ *regular contribution function*.

We restrict the following discussion to surface light transport. However, most equations below also apply trivially to medium transport; we will point out important differences along the way.

7.1.1 Extended path space formulation

We now write the primary pixel estimator (i.e. for a single light subpath) resulting from the evaluation of the photon radiance estimator (Eq. 4.22) at the $(s+1)$ -th vertex from the light source using a blurring kernel K_r with support radius r . This vertex \mathbf{y}_s is a photon and the estimator reads

$$\hat{I}_{k,s,r}^{\text{PM}} = C_s(\bar{\mathbf{y}}) K_r(\mathbf{y}_s, \mathbf{z}_{k-s}) \rho_s(\mathbf{z}_{k-s}, \boldsymbol{\omega}_{\mathbf{y}_s \mathbf{z}_{k-s}}, \boldsymbol{\omega}_{\mathbf{z}_{k-s} \mathbf{z}_{k-s-1}}) C_{k-s}(\bar{\mathbf{z}}). \quad (7.2)$$

Here, the BSDF is evaluated at the radiance estimate query point \mathbf{z}_{k-s} on an eye subpath $\bar{\mathbf{z}}$ with $k-s+1$ vertices with a sampling throughput $C_{k-s}(\bar{\mathbf{z}})$ (Eq. 4.6), as done in a practical implementation. The actual value estimated by the above estimator is given by its expectation:

$$I_{k,s,r} = E[\hat{I}_{k,s,r}^{\text{PM}}] = \int_{\mathcal{M}^{k+2}} f_{k,s,r}(\bar{\mathbf{x}}^*) d\mu_{k+2}(\bar{\mathbf{x}}^*), \quad (7.3)$$

which is an integral over paths of the form $\bar{\mathbf{x}}^* = \mathbf{x}_0 \dots \mathbf{x}_s^* \mathbf{x}_s \dots \mathbf{x}_k = \mathbf{y}_0 \dots \mathbf{y}_s \mathbf{z}_{k-s} \dots \mathbf{z}_0$. Here we denote the photon vertex by \mathbf{x}_s^* and the radiance estimate query vertex by \mathbf{x}_s . We refer to such paths $\bar{\mathbf{x}}^*$ as *extended paths*, and call the space \mathcal{M}^{k+2} *extended space* with an associated differential product area measure $d\mu_{k+2}$. The function $f_{k,s,r}$ is the corresponding *extended contribution function*:

$$f_{k,s,r}(\bar{\mathbf{x}}^*) = \rho(\mathbf{x}_0) T(\mathbf{x}_0 \dots \mathbf{x}_s^*) K_r(\mathbf{x}_s^*, \mathbf{x}_s) \rho_s(\mathbf{x}_s, \boldsymbol{\omega}_{\mathbf{x}_s \mathbf{x}_s^*}, \boldsymbol{\omega}_{\mathbf{x}_s \mathbf{x}_{s+1}}) T(\mathbf{x}_s \dots \mathbf{x}_k) \rho(\mathbf{x}_k), \quad (7.4)$$

where the measurement throughput T and the scattering term ρ were defined in Equations 3.32 and 3.33, respectively. We illustrate this notation graphically in Figure 7.2b.

Note the extra area integration in Equation 7.3 compared to the regular path integral (Eq. 7.1) for the same path length k (i.e. edge count). This integration over the possible locations of the photon vertex \mathbf{x}_s^* corresponds to blurring the illumination with the kernel K_r – a well-known effect of density estimation.

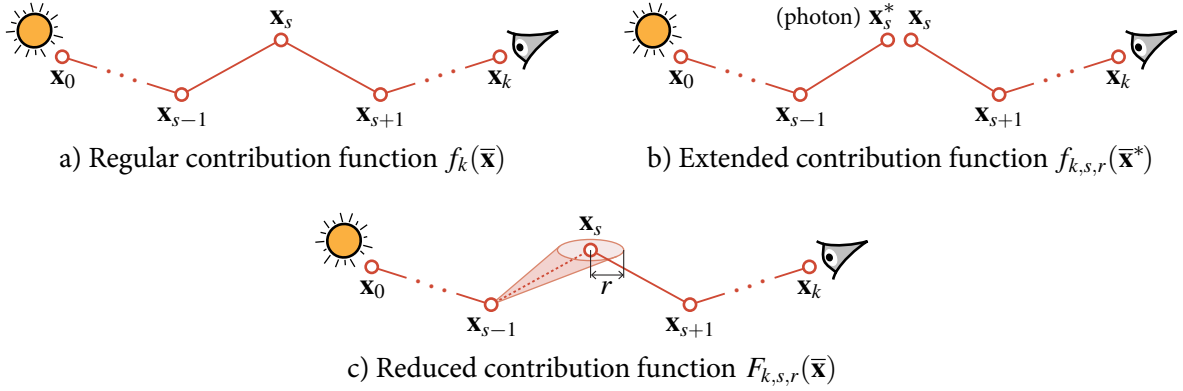


Figure 7.2: Illustration of three different measurement contribution functions. The reduced function operates on regular paths and computes a blurred contribution by integrating the extended function over the photon position \mathbf{x}_s^* within the support of the kernel K_r (see Section 7.1.2).

Sampling technique for extended paths

Having obtained expressions for the path integral estimated by photon mapping (Eq. 7.3) and its corresponding contribution function (Eq. 7.4), we can rewrite Equation 7.2 as an ordinary Monte Carlo estimator:

$$\hat{I}_{k,s,r}^{\text{PM}}(\bar{\mathbf{x}}^*) = \frac{f_{k,s,r}(\bar{\mathbf{x}}^*)}{p(\bar{\mathbf{x}}^*)}. \quad (7.5)$$

The extended path $\bar{\mathbf{x}}^* = \mathbf{x}_0 \dots \mathbf{x}_s^* \mathbf{x}_s \dots \mathbf{x}_k = \mathbf{y}_0 \dots \mathbf{y}_s \mathbf{z}_{k-s} \dots \mathbf{z}_0$ is constructed by concatenating two subpaths $\bar{\mathbf{y}} = \mathbf{y}_0 \dots \mathbf{y}_s$ and $\bar{\mathbf{z}} = \mathbf{z}_0 \dots \mathbf{z}_{k-s}$ at their endpoints, which are sampled independently via random walks from a light source and from the eye, respectively. The pdf of the full path, which is given by the joint distribution of its vertices, is thus simply the product of the two subpath pdfs:

$$p(\bar{\mathbf{x}}^*) = p_s(\bar{\mathbf{y}})p_{k-s}(\bar{\mathbf{z}}), \quad (7.6)$$

where the subpath pdf is given by Equation 4.5.

We have now defined PM as path sampling technique with an associated pdf. An entire family of techniques is obtained by considering different locations along the path for the radiance estimate, $s \in [1, \dots, k-1]$. We call this family of techniques *vertex merging* as one can intuitively think of the estimator $\hat{I}_{k,s,r}^{\text{PM}}$ as merging the path vertices corresponding to a photon and the radiance estimate location within a small neighborhood. This is in contrast to vertex connection (Eq. 4.14) which creates a path edge between two subpath endpoints. Note that in practice photon mapping constructs many extended full paths for every pixel that share the same eye subpath $\bar{\mathbf{z}}$ up to the photon query location.

We could directly use these definitions to apply MIS and combine the results of the PM radiance estimates at different vertices of an eye subpath. Indeed, that would correspond to what has been done by Vorba [146] (with the difference that Vorba uses path vertex pdfs expressed w.r.t. the solid angle measure). However, we *cannot* use the above definitions for an MIS-based combination of BPT and PM. The reason is that for a given path length (i.e. number of edges), the extended paths sampled by PM contain one extra vertex, the photon \mathbf{x}_s^* , compared to the regular paths sampled by BPT. The pdf of an extended path is consequently expressed w.r.t. a different, higher-dimensional product area measure than the pdf of a regular path of the same length. The MIS heuristics expect the pdfs to be expressed w.r.t. the same measure in order to yield a meaningful combination weight.

7.1.2 Regular path space formulation

In order to combine BPT and PM using MIS, we must express the pdfs of same-length paths sampled by BPT and PM w.r.t. the same measure, i.e. both methods should conceptually operate in the space of either the extended or the regular paths. Both options are possible; we choose the space of regular paths, because doing so preserves Veach's original path integral formulation that we already discussed extensively in Chapters 3 and 4. (We will discuss the alternative option in Section 7.6.) This means that we need to express the PM path integral (Eq. 7.3) as an integral over regular, not extended paths. We achieve this by considering the extra area integral in $I_{k,s,r}$ as a nested integration problem to which the path integral is oblivious:

$$I_{k,s,r} = \int_{\mathcal{M}^{k+1}} \left[\int_{\mathcal{M}} f_{k,s,r}(\bar{\mathbf{x}}^*) dA(\mathbf{x}_s^*) \right] d\mu_{k+1}(\bar{\mathbf{x}}) = \int_{\mathcal{M}^{k+1}} F_{k,s,r}(\bar{\mathbf{x}}) d\mu_{k+1}(\bar{\mathbf{x}}). \quad (7.7)$$

Here, $\bar{\mathbf{x}}$ is a regular light path created from an extended path $\bar{\mathbf{x}}^*$ by leaving out the \mathbf{x}_s^* vertex and concatenating all the remaining vertices. The *reduced contribution function* $F_{k,s,r}(\bar{\mathbf{x}})$ blurs the contribution of the regular path $\bar{\mathbf{x}}$ in an r -neighborhood \mathcal{M}_r around \mathbf{x}_s , which is described by an area integral of $f_{k,s,r}$:

$$F_{k,s,r}(\bar{\mathbf{x}}) = \int_{\mathcal{M}_r} f_{k,s,r}(\bar{\mathbf{x}}^*) dA(\mathbf{x}_s^*), \quad (7.8)$$

where $\mathcal{M}_r = \{\mathbf{x} \in \mathcal{M} \mid \|\mathbf{x}_s - \mathbf{x}\| < r\}$. Note that it is the small support of the kernel K_r inside $f_{k,s,r}$ that effectively limits the integration over \mathcal{M} in Equation 7.7 to the r -neighborhood \mathcal{M}_r of \mathbf{x}_s in Equation 7.8. We illustrate the reduced contribution function $F_{k,s,r}$ graphically in Figure 7.2c.

Sampling technique for regular paths

We now interpret photon mapping (vertex merging) as a sampling technique for the Monte Carlo estimation of the reduced path integral (Eq. 7.7), which samples *regular* paths $\bar{\mathbf{x}}$ and uses $F_{k,s,r}(\bar{\mathbf{x}})$ as the contribution function. That is, a PM radiance estimate at vertex \mathbf{x}_s creates regular paths $\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$ via vertex connection between \mathbf{x}_{s-1} and \mathbf{x}_s (one such path for each photon), excluding the photon vertex \mathbf{x}_s^* . The measurement contribution $F_{k,s,r}(\bar{\mathbf{x}})$ of such a path now contains an extra area integration that corresponds to blurring by the kernel K_r , as we described above. We interpret the photon vertex \mathbf{x}_s^* as a Monte Carlo sample used to estimate that integral. More importantly, the photon serves as a Russian roulette random variable that conditions the probabilistic acceptance of the proposed full regular path. The path $\bar{\mathbf{x}}$ is accepted if and only if the photon location \mathbf{x}_s^* is within distance r from the radiance estimate location \mathbf{x}_s . This interpretation, illustrated in Figure 7.3b, is in line with the traditional view and implementation of photon mapping, where photons outside of the search radius are not considered in the radiance estimate.

The pdf of a regular path $\bar{\mathbf{x}}$ generated with the above procedure is

$$p_{\text{VM},s,t}(\bar{\mathbf{x}}) = p_{\text{VC},s,t}(\bar{\mathbf{x}}) P_{\text{acc}}(\bar{\mathbf{x}}), \quad (7.9)$$

where $p_{\text{VC},s,t}(\bar{\mathbf{x}})$ is the pdf for the vertex connection (VC) technique (Eq. 4.16) that samples $\bar{\mathbf{x}}$ by connecting subpaths with endpoints \mathbf{x}_{s-1} and \mathbf{x}_s . The acceptance probability on the right is the probability

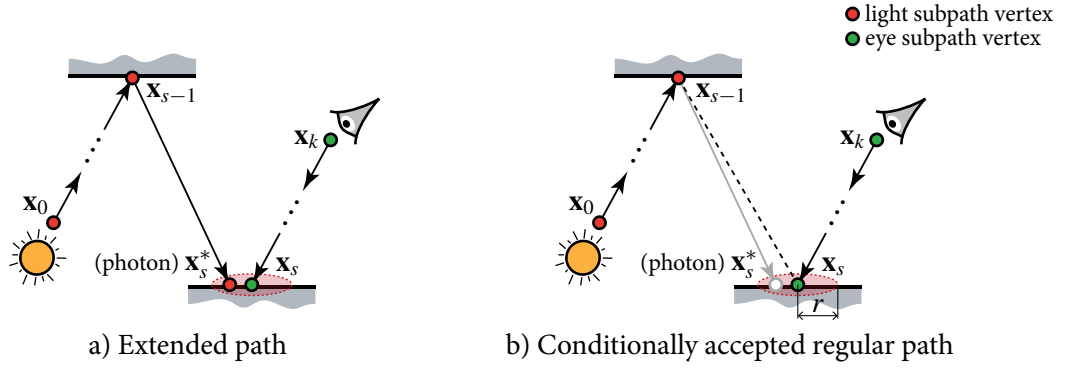


Figure 7.3: Two interpretations of photon mapping as a “vertex merging” path sampling technique. *a)* Photon mapping can be considered to sample an extended path $\bar{\mathbf{x}}^*$ of length k that has $k + 2$ vertices. *b)* To remain compatible with the path integral framework we interpret this process as sampling a regular path $\bar{\mathbf{x}}$ of the same length (i.e. number of edges) that has only $k + 1$ vertices, where the photon \mathbf{x}_s^* is used as a Russian roulette variable. The path is accepted only if \mathbf{x}_s^* lies within distance r to the path vertex \mathbf{x}_s .

for sampling a photon inside the r -neighborhood of \mathbf{x}_s :

$$P_{\text{acc}}(\bar{\mathbf{x}}) = \Pr(\|\mathbf{x}_s - \mathbf{x}^*\| < r) = \int_{\mathcal{M}_r} p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}) d\mathbf{x} \quad (7.10a)$$

$$\approx |\mathcal{M}_r| p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}^*) \approx \pi r^2 p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}^*), \quad (7.10b)$$

where $p(\mathbf{x}_{s-1} \rightarrow \mathbf{x})$ is shorthand for $p(\mathbf{x}|\mathbf{x}_{s-1}, \omega_{\mathbf{x}_{s-2}\mathbf{x}_{s-1}})$. Exact analytic evaluation of the integral in Equation 7.10a is generally impossible. To make the computation feasible, in Equation 7.10b we first assume that the pdf p is constant inside \mathcal{M}_r . This is a common assumption made by the progressive radiance estimation [41] and its asymptotic analysis [76]. It allows us to take $p(\mathbf{x}_{s-1} \rightarrow \mathbf{x})$ out of the integral and replace it by the pdf of the actually sampled photon, $p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*)$, which is known. Second, we make the common photon mapping assumption that \mathcal{M}_r is a disk centered around \mathbf{x}_s with radius r and area πr^2 . The accuracy of the resulting P_{acc} approximation reduces in areas of geometric variation and when p is far from constant, e.g. when \mathbf{x}_{s-1} has a very sharp glossy BSDF. Note, however, that the approximation converges to the true value as the radius r approaches zero. Also note that for \mathbf{x}_s^* inside \mathcal{M}_r coming from a vertex \mathbf{x}_{s-1} with a perfect specular BSDF (e.g. mirror or glass), the acceptance probability is simply $P_{\text{acc}} = 1$ as the location of \mathbf{x}_s^* is sampled deterministically¹.

The final vertex merging path pdf now reads:

$$p_{\text{VM},s,t}(\bar{\mathbf{x}}) = p_{\text{VC},s,t}(\bar{\mathbf{x}}) P_{\text{acc}}(\bar{\mathbf{x}}) = p_{\text{VC},s,t}(\bar{\mathbf{x}}) \int_{\mathcal{M}_r} p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}) d\mathbf{x} \quad (7.11a)$$

$$\approx p_{\text{VC},s,t}(\bar{\mathbf{x}}) [\pi r^2 p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*)]. \quad (7.11b)$$

This pdf formula describes the actual random events that occur during the path construction in photon mapping. Importantly, the pdf is expressed w.r.t. the same product area measure as any length- k regular path sampled by bidirectional path tracing. The pdf of the photon vertex \mathbf{x}_s^* only appears as a part of the approximation of the acceptance probability P_{acc} which itself is ‘unitless’² and therefore has no impact

¹Strictly speaking, connections to perfectly specular vertices are prohibited. However, by allowing approximate connections to the specular “parents” of nearby photons, photon mapping samples caustic paths while introducing little error.

²Considering that the unit of a vertex pdf is $[\text{m}^{-2}]$ and πr^2 is in $[\text{m}^2]$, and so the units cancel out.

on the measure. Note that this approximation makes the path pdf expression (Eq. 7.11b) symmetric, in the sense that it includes the densities of all sampled vertices. Thus, deriving the pdf from the eye tracing side, i.e. interpreting the radiance query vertex \mathbf{x}_s as the Russian roulette variable and the photon vertex \mathbf{x}_s^* as a path vertex, would result in the same path pdf expression.

Note that for the path pdf derivation we only consider the probability for the photon vertex \mathbf{x}_s^* to land inside the *support* of the kernel K_r . The actual shape of the kernel is still accounted for in the path contribution $F_{k,s,r}(\bar{\mathbf{x}})$ (Eq. 7.8).

The above derivation also applies to volumetric photon mapping [64] where the radiance estimate query vertex \mathbf{x}_s is in a scattering medium. The only difference in the final expression for the path pdf in Equation 7.11b is the replacement of πr^2 by $\frac{4}{3}\pi r^3$, which accounts for the fact that the r -neighborhood of \mathbf{x}_s is a sphere in that case.

The acceptance probability term $\pi r^2 p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*)$ in the path pdf is useful for understanding the efficiency of photon mapping, as we will show in Section 7.2 below. The dependence of this term on the radius r will also prove crucial for the good asymptotic performance of our progressive combined algorithm in Section 7.4.

Regular space estimator

We now derive the primary vertex merging estimator for the reduced path integral (Eq. 7.7). We begin with the measurement contribution function $F_{k,s,r}(\bar{\mathbf{x}})$ which is itself defined as an integral (Eq. 7.8). A one-sample Monte Carlo estimate of that integral can be obtained using \mathbf{x}_s^* , i.e. the photon:

$$\hat{F}_{k,s,r}(\bar{\mathbf{x}}) = \frac{f_{k,s,r}(\bar{\mathbf{x}}^*)}{\left[\frac{p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*)}{\int_{\mathcal{M}_r} p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}) d\mathbf{x}} \right]}, \quad (7.12)$$

where we normalize $p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*)$ to obtain a valid pdf over \mathcal{M}_r , since only points \mathbf{x}_s^* inside this set survive the Russian roulette that determines the acceptance of the path, as described above.

We can now construct the final pixel estimator using the path pdf $p_{\text{VM},s,t}(\bar{\mathbf{x}})$ from Equation 7.11a:

$$\hat{I}_{s,t,r}^{\text{VM}}(\bar{\mathbf{x}}) = \frac{\hat{F}_{k,s,r}(\bar{\mathbf{x}})}{p_{\text{VM},s,t}(\bar{\mathbf{x}})} = \frac{f_{k,s,r}(\bar{\mathbf{x}}^*)}{\left[\frac{p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*)}{\int_{\mathcal{M}_r} p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}) d\mathbf{x}} \right] p_{\text{VM},s,t}(\bar{\mathbf{x}})} = \frac{f_{k,s,r}(\bar{\mathbf{x}}^*)}{p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*) p_{\text{VC},s,t}(\bar{\mathbf{x}})} = \hat{I}_{k,s,r}^{\text{PM}}, \quad (7.13)$$

which we denote by $\hat{I}_{s,t,r}^{\text{VM}}$, with $k = s + t - 2$, in order to maintain notational compatibility with the vertex connection estimator (Eq. 4.14). Note that the pdf normalization integral from Equation 7.12 cancels out above when multiplied by the VM path pdf (Eq. 7.11a) and thus it does not need to be approximated. However, the practical approximation of this integral (Eq. 7.11b) will be required later in our combined algorithm for the MIS weight evaluation of different path sampling techniques.

We have arrived at an expression for the photon mapping pixel estimator (Eq. 7.2). This result shows that our reformulation of PM as a sampling technique for regular paths is compatible with the classic view of PM because the final estimators for both views are identical. The importance of our formulation is that it clearly separates the path pdf from the contribution function and is compatible with Veach's [142] path integral formulation (Eq. 3.27). This lays the ground for including vertex merging into MIS estimators of the form given in Equation 4.3 and obtaining a meaningful combination.

7.2 Efficiency of different path sampling techniques

Our vertex merging (VM) formulation provides new insight into the efficiency of the photon mapping (PM) algorithm, which explains why PM performs differently than bidirectional path tracing (BPT) in different lighting conditions. Our formulation provides the pdf for the individual light transport paths sampled by PM. This allows us to analyze the relative efficiency of various path sampling techniques by comparing their sampling probability densities for a given path. This tool has been long available to BPT where, e.g., the power heuristic is based on the observation that a higher pdf most often results in a lower-variance estimate [144]. Vertex merging allows us to include photon mapping in the comparison and reason about its efficiency as a Monte Carlo path sampling technique.

Note in Equation 7.11 that for any path the VM pdf is at most equal to that of vertex connection (VC). This is due to the path acceptance probability $P_{\text{acc}} \leq 1$ (Eq. 7.10). Consider the path in Figure 7.3b as an example. For practical values of r , the merging disk often spans a small solid angle as seen from \mathbf{x}_{s-1} , depending on its distance to that vertex. If the vertex \mathbf{x}_{s-1} is diffuse, P_{acc} is small, as it is equal to the probability of sampling a random ray inside that solid angle. The resulting VM path pdf can then be six or more orders of magnitude lower than the corresponding VC path pdf. It can be also shown that if the merging disk area equals the light source area, then unidirectional sampling (US) and VM can have almost equal pdfs. The intuition is that the probability of hitting the light would be roughly the same as for hitting the merging disk.

Based on the above observations, we can conclude that VM (and thus photon mapping) is not an intrinsically more efficient sampling technique than the BPT techniques. However, the power of VM is its *computational efficiency*, which comes from the evaluation of light transport along approximate trajectories in the extended contribution function (Eq. 7.4). This enables the construction and evaluation of a large number of full paths via simple conditional subpath concatenation that comes at the cost of a single range search in a pre-built acceleration structure – the photon map. Therefore, in cases where the pdfs of other techniques (e.g. unidirectional sampling or vertex connection) are not much higher than that of VM, the latter can result in a significantly lower-error estimate due to its *efficient brute-force variance reduction*. The most prominent example for such cases are *SDS* paths, which in BPT can be sampled only unidirectionally and often with very low probability. The cheap variance reduction in PM is made possible by reusing the same set of light subpaths for the eye subpaths of all pixels, thereby amortizing the light subpath sampling cost over the evaluation of many estimators. We demonstrate this in Figure 7.4 by comparing the image quality achieved by VM without and with light subpath reuse against unidirectional sampling. Without reuse, VM performs similarly to unidirectional path tracing without next event estimation.

Note that we could also use the regular measurement contribution function (Eq. 3.31) in the VM estimator in Equation 7.13. This would effectively turn the estimator into a variant of vertex connection, since the exact connection throughput (Eq. 4.15) along the edge $\mathbf{x}_{s-1}\mathbf{x}_s$ (Fig. 7.3b) would be evaluated, which would also eliminate the blurring bias³. This, in fact, is precisely what has been done by Bekaert et al. [3]. Unfortunately, this modification would also significantly decrease the computational efficiency of the estimator, because a visibility ray would need to be traced along the connection segment of each constructed full path. Moreover, caustics paths due to perfect specular vertices \mathbf{x}_{s-1} would not be captured as vertex connection can only construct *DD* path segments.

³Note that the estimator would still not become completely unbiased, because the pdf normalization integral $\int_{\mathcal{M}_r} p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}) d\mathbf{x}$ would not cancel out and would therefore have to be approximated, e.g. as in Equation 7.10.

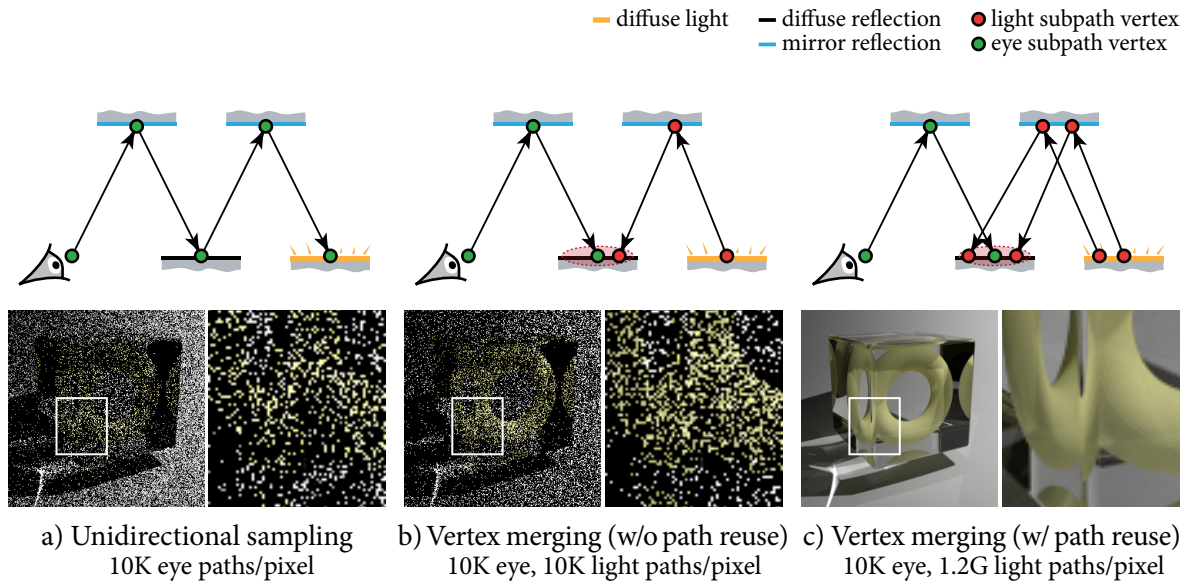


Figure 7.4: Vertex merging explains why photon mapping can be so much more efficient in handling SDS light transport than bidirectional path tracing. It is not because it can find SDS paths with a higher probability, but because it can efficiently reuse light subpaths across pixels. Without reuse, and with equal light source and merging disk areas, photon mapping performs similarly to unidirectional sampling.

7.3 A combined light transport algorithm

In Section 7.1 we reformulated photon mapping (PM) as a path sampling technique, which we call vertex merging. This formulation provides us with a clear definition of the light transport paths sampled by the PM radiance estimate and their associated pdfs which are expressed w.r.t. the same measure as in bidirectional path tracing (BPT). In this section we take advantage of this formulation to integrate PM and BPT into a more robust light transport algorithm via multiple importance sampling (MIS).

Figure 7.5 illustrates the different types of path sampling techniques we have at our disposal. For a path of length k , BPT offers up to $k + 2$ techniques corresponding to performing the vertex connection along different path edges. (Unidirectional sampling is the special case where one of the subpaths has zero length.) Vertex merging adds up to $k - 1$ more techniques which correspond to “merging” at different interior vertices on the path⁴. We combine all these techniques into a unified rendering algorithm, which we call *vertex connection and merging* (VCM).

The VCM formulation described below assumes a fixed merging radius r . Since our method includes biased estimators, its combined estimator will also be biased for any merging radius $r > 0$. In Section 7.4 we will present a progressive variant of the VCM algorithm that converges to the true value as r approaches zero.

⁴This number does not include the two techniques that merge at the light source vertex and the eye vertex respectively, since directly evaluating light emission or sensor sensitivity at such vertices is usually more efficient than merging.

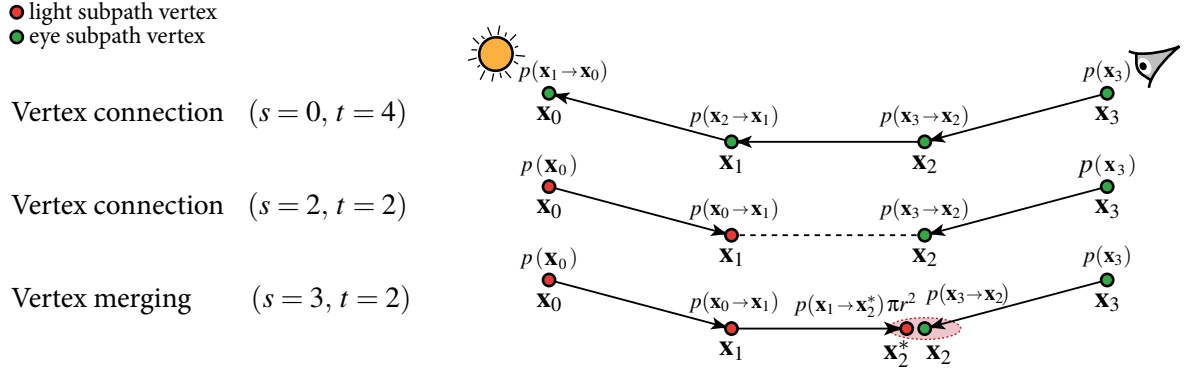


Figure 7.5: Illustration of different path sampling techniques, along with the corresponding pdf terms associated with each vertex. For paths with k edges (here $k = 3$) bidirectional path tracing provides $k + 2$ sampling techniques. (In practice we do not use the unidirectional $(k + 1, 0)$ technique.) Vertex merging brings $k - 1$ new techniques corresponding to merging at the $k - 1$ different interior path vertices.

7.3.1 Mathematical formulation

Thanks to the vertex merging formulation as a path sampling technique with an associated pdf, we can now apply MIS to combine the different techniques from PM and BPT using the power heuristic. Note that the different (s, t) vertex merging estimators use different contribution functions and thus have different expected values. Therefore, using the power heuristic to weight these estimators does not guarantee minimization of the bias of the resulting combined estimator. Here we focus on minimizing the variance only, and leave bias-aware combination as an interesting avenue for future work.

A VCM pixel estimator combines weighted contributions C_{VC} from vertex connection (VC) estimators $\hat{I}_{s,t}^{VC}$ (Eq. 4.14) and weighted contributions C_{VM} from vertex merging (VM) estimators $\hat{I}_{s,t,r}^{VM}$ (Eq. 7.13):

$$\hat{I}^{VCM} = C_{VC} + C_{VM} \quad (7.14a)$$

$$= \frac{1}{n_{VC}} \sum_{j=1}^{n_{VC}} \sum_{s \geq 0, t \geq 0} w_{VC,s,t}(\bar{\mathbf{x}}_{s,t,j}) \hat{I}_{s,t}^{VC}(\bar{\mathbf{x}}_{s,t,j}) + \quad (7.14b)$$

$$\frac{1}{n_{VM}} \sum_{j=1}^{n_{VM}} \sum_{s \geq 2, t \geq 2} w_{VM,s,t}(\bar{\mathbf{x}}_{s,t,j}) \hat{I}_{s,t,r}^{VM}(\bar{\mathbf{x}}_{s,t,j}), \quad (7.14c)$$

which is an extended version of the BPT estimator (Eq. 4.19). It considers an eye subpath through the pixel whose vertices are connected to the vertices of n_{VC} light subpaths and potentially merged with the vertices of n_{VM} light subpaths. The subscript s, t corresponds to a path constructed from a light subpath with s vertices and an eye subpath with t vertices. The power heuristic weight for technique (v, s, t) has the usual form

$$w_{v,s,t}(\bar{\mathbf{x}}) = \frac{n_v^\beta \cdot p_{v,s,t}^\beta(\bar{\mathbf{x}})}{\sum_{s' \geq 0, t' \geq 0} n_{VC}^\beta \cdot p_{VC,s',t'}^\beta(\bar{\mathbf{x}}) + \sum_{s' \geq 2, t' \geq 2} n_{VM}^\beta \cdot p_{VM,s',t'}^\beta(\bar{\mathbf{x}})}, \quad (7.15)$$

where v is either VC or VM. Note that the weight of a technique is amplified by the total number of samples, i.e. light paths, it uses. The computational efficiency of vertex merging allows in practice n_{VM} to be much larger than n_{VC} . Usually, n_{VM} would be the total number of sampled light subpaths for the image, while $n_{VC} = 1$ as in BPT. As discussed in Section 7.2, subpath reuse and the consequent brute-force variance reduction is the key to the efficiency of the vertex merging technique.

```

1: ▶ Renders progressively by averaging the results from  $N$  independent iterations
2: function PROGRESSIVERENDER( $r_1, \alpha, N, image$ )
3:   for  $i = 1$  to  $N$  do
4:      $iterationImage = ZEROIMAGE()$ 
5:      $r_i = r_1 \sqrt{i^{\alpha-1}}$ 
6:     RENDER( $r_i, iterationImage$ )
7:      $image = \frac{i-1}{i} image + \frac{1}{i} iterationImage$ 
8:   end for
9: end function
10:
11: ▶ Renders a single image with a given maximum vertex merging radius
12: function RENDER( $r$ )
13:   ▶ Stage 1: Light path sampling
14:    $lightPaths = TRACELIGHTPATHS(pixelCount)$ 
15:   CONNECTTOEYE( $lightVertices$ )
16:   BUILDRANGESTRUCT( $lightVertices$ )
17:   ▶ Stage 2: Eye path sampling and pixel estimator construction
18:   for  $i = 1$  to  $pixelCount$  do
19:      $eyeVertex = TRACERAY(SAMPLEPIXEL(i))$ 
20:     while  $eyeVertex$  is valid do
21:       ▶ Unidirectional sampling (US)
22:       if  $eyeVertex$  is emissive then
23:         ACCUM( $eyeVertex, US, r, i$ )
24:       end if
25:       ▶ Vertex connection (VC)
26:       for  $lightVertex$  in  $lightPaths[i] \cup SAMPLELIGHTPOINT()$  do
27:         ACCUM(CONNECT( $eyeVertex, lightVertex$ ), VC,  $r, i$ )
28:       end for
29:       ▶ Vertex merging (VM)
30:       for  $lightVertex$  in RANGESEARCH( $eyeVertex, r$ ) do
31:         ACCUM(MERGE( $eyeVertex, lightVertex$ ), VM,  $r, i$ )
32:       end for
33:        $eyeVertex = CONTINUERANDOMWALK(eyeVertex)$ 
34:     end while
35:   end for
36: end function
37:
38: ▶ Accumulates the pixel measurement estimate due to a given path
39: function ACCUM( $path, technique, r, i$ )
40:    $contrib = MEASUREMENTCONTRIBUTION(path, technique, r)$ 
41:    $pdf = PDF(path, technique, r)$ 
42:    $weight = POWERHEURISTIC(path, technique, pdf)$ 
43:    $image[i] += weight * contrib / pdf$ 
44: end function

```

Figure 7.6: Pseudocode for our progressive VCM rendering algorithm, which produces an image given an initial vertex merging radius r_1 and a radius reduction parameter α .

7.3.2 Algorithm

We now describe the practical implementation of our combined VCM estimator from Equation 7.14. Since subpath sampling is expensive, it is desirable to amortize this effort over many primary estimator evaluations. Veach's [142] BPT implementation reuses subpaths by connecting every eye subpath vertex to all vertices on one light subpath. Vertex merging lends itself to a substantially more efficient path reuse scheme that allows to potentially merge every eye subpath vertex with the vertices of a large number of pre-generated light subpaths. To maximize subpath reuse, our algorithm runs in two stages, separating the sampling of the light and eye subpaths, similarly to photon mapping. An outline of the algorithm is given in the RENDER function in Figure 7.6.

In the *first stage*, we trace a number of subpaths from the light sources, connect their vertices to the eye and build a range search data structure over them (lines 14-16). In the *second stage*, we trace an eye subpath for each pixel. Upon sampling an eye subpath vertex, we check if it lies on a light source and accumulate the potential contribution (lines 22-24). We then connect that vertex to the vertices of one of the pre-generated light subpaths, similarly to BPT (lines 26-28). To reduce sampling correlation we follow Veach [142, Sec. 10.3.4.2] and do not store the first vertex of a light subpath and instead connect every eye vertex to a new, randomly sampled point on a light source. Finally, we perform a range search around the eye vertex to merge with all light subpath vertices within its r -neighborhood (lines 30-32) as in photon mapping. We construct an estimate for each generated full path, evaluate its MIS weight, and finally accumulate the weighted estimate into the running pixel estimate (lines 39-44). Random walks are terminated via Russian roulette or when a maximum subpath length is reached. Most of the terms required to evaluate path contributions and MIS weights can be incrementally computed and stored at the vertices of the subpaths as they are traced. We will discuss efficient practical implementations of the algorithm in more detail in Chapter 8.

7.4 Achieving consistency

Apart from the usual variance in the form of noise, the images produced by our combined VCM algorithm contain systematic error (i.e. bias) in the form of blur, which is inherited from vertex merging. We now show how the algorithm can be made consistent by constructing a secondary pixel estimator that progressively reduces the merging radius r , such that the variance and bias vanish in the limit, much like in the progressive photon mapping variant of Knaus and Zwicker [76] (see also Section 4.6.2).

The progressive variant of the VCM pixel estimator (Eq. 7.14) averages the results of N independent rendering iterations:

$$\hat{I}_N^{\text{VCM}} = \frac{1}{N} \sum_{i=1}^N (C_{\text{VC},i} + C_{\text{VM},i}), \quad (7.16)$$

where $C_{\text{VC},i}$ and $C_{\text{VM},i}$ are as in Equation 7.14, but use a new, independent set of light and eye subpaths at each iteration i as well as a new merging radius r_i . The key to making this estimator consistent is to progressively reduce the merging radius as the iteration counter increases and to do so at a rate that ensures that both the variance and the bias of the estimator vanish as $N \rightarrow \infty$. Below we devise our radius reduction scheme and then we carry out an asymptotic analysis that proves the consistency of the resulting estimator and derives its error convergence rate. The pseudocode of the progressive algorithm is given in the `PROGRESSIVERENDER` function in Figure 7.6.

7.4.1 Progressive radius reduction

Our radius reduction scheme is based on the one used by Knaus and Zwicker [76] who compute the radius r_i at iteration i in progressive photon mapping as

$$r_i = r_1 \sqrt{\left(\prod_{k=1}^{i-1} \frac{k + \alpha}{k} \right) \frac{1}{i}}. \quad (7.17)$$

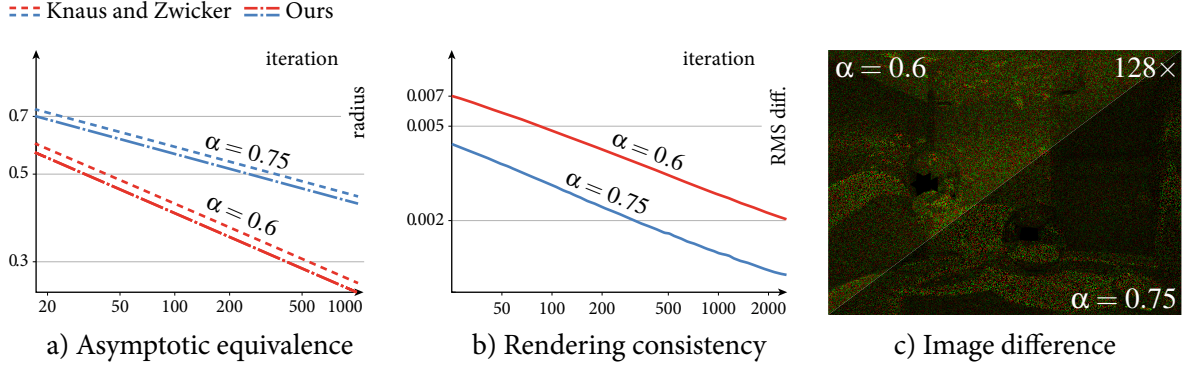


Figure 7.7: A comparison of our new radius reduction scheme against that of Knaus and Zwicker [76], carried out in a progressive photon mapping implementation for two α -parameter values. a) Log-log plots of the radii computed by the two schemes. b) A relative RMS image difference log-log plot of the two schemes, showing that their corresponding estimators converge to the reference solution for both α values. c) A 128 \times -scaled difference image taken after 2500 rendering iterations.

Here, r_1 is the initial radius and $\alpha \in (0; 1)$ is a user parameter. From Appendix E in their paper [76] it follows that $r_i = O(i^{\frac{\alpha-1}{2}})$. From this result we derive the simpler formula

$$r_i = r_1 \sqrt{i^{\alpha-1}}. \quad (7.18)$$

Our scheme is asymptotically equivalent to that of Knaus and Zwicker, but is simpler and easier to compute for each iteration independently. We demonstrate the asymptotic equivalence of the two schemes experimentally in Figure 7.7.

7.4.2 Asymptotic error analysis

An important advantage of BPT over PPM is its higher asymptotic error convergence rate, as we discussed Section 4.6.2. This rate measures how fast the progressive estimate approaches the true value as the number of samples N (rendering iterations in our case) grows to infinity. BPT has a mean squared error (MSE) convergence rate of $O(N^{-1})$ [142]. In Section 4.6.2 we showed that the optimal MSE rate of PPM is $O(N^{-2/3})$, reached for $\alpha = 2/3$. Since our progressive VCM estimator (Eq. 7.16) is a weighted average of vertex connection (VC) and vertex merging (VM) estimators, its MSE rate must necessarily lie somewhere between those of BPT and PPM. We show next that, in fact, the MSE of this combined estimator converges asymptotically as fast as that of BPT for light transport paths that can be sampled by BPT.

We first perform asymptotic simplifications w.r.t. the iteration counter i , using $r_i = O(\sqrt{i^{\alpha-1}})$ from our radius reduction scheme (Eq. 7.18). Knaus and Zwicker [76] have shown that PPM (and thus VM) primary estimators have $\text{Var}[\hat{I}_{s,t,r}^{\text{VM}}] = O(1/r_i^2)$ and $\text{Bias}[\hat{I}_{s,t,r}^{\text{VM}}] = O(r_i^2)$. In contrast, the unbiased VC estimators are independent of r_i :

$$\text{Var}[\hat{I}_{s,t}^{\text{VC}}] = O(1) \quad \text{Bias}[\hat{I}_{s,t}^{\text{VC}}] = 0 \quad (7.19a)$$

$$\text{Var}[\hat{I}_{s,t,r}^{\text{VM}}] = O(i^{1-\alpha}) \quad \text{Bias}[\hat{I}_{s,t,r}^{\text{VM}}] = O(i^{\alpha-1}). \quad (7.19b)$$

Furthermore, for any path $\bar{\mathbf{x}}$ and any s and t , we have $p_{\text{VC},s,t}(\bar{\mathbf{x}}) = O(1)$, as the VC path pdf is independent of r_i . From Equation 7.11b it follows that $p_{\text{VM},s,t}(\bar{\mathbf{x}}) = O(r_i^2) = O(i^{\alpha-1})$. Substituting in

90 Section 7.4: Achieving consistency

Equation 7.15, and using $\alpha - 1 < 0$, we obtain:

$$w_{\text{VC},s,t}(\bar{\mathbf{x}}) = \frac{O(1)}{O(1) + O(i^{\beta(\alpha-1)})} = O(1) \quad (7.20a)$$

$$w_{\text{VM},s,t}(\bar{\mathbf{x}}) = \frac{O(i^{\beta(\alpha-1)})}{O(1) + O(i^{\beta(\alpha-1)})} = O(i^{\beta(\alpha-1)}). \quad (7.20b)$$

Recall that β is the power heuristic exponent. Note that while the weights of the VC techniques are asymptotically constant, the VM weights decrease as the iteration index i increases.

Variance

The variance of the progressive VCM estimator (Eq. 7.16) is

$$\text{Var} [\hat{I}_N^{\text{VCM}}] = \frac{1}{N^2} \sum_{i=1}^N (\text{Var}[C_{\text{VC},i}] + \text{Var}[C_{\text{VM},i}]). \quad (7.21)$$

The sums over subpaths in $C_{\text{VC},i}$ and $C_{\text{VM},i}$ (Eq. 7.14) can be simplified away, as they are independent of i . $\text{Var}[C_{\text{VC},i}]$ and $\text{Var}[C_{\text{VM},i}]$ now reduce to the variances of the primary VC and VM estimators (Eq. 7.19) scaled by their corresponding weights (Eq. 7.20):

$$\text{Var} [\hat{I}_N^{\text{VCM}}] = \frac{1}{N^2} \sum_{i=1}^N [O(1)O(1) + O(i^{2\beta(\alpha-1)})O(i^{1-\alpha})] \quad (7.22a)$$

$$= \frac{1}{N^2} \sum_{i=1}^N O(1) + \frac{1}{N^2} \sum_{i=1}^N O(i^{(2\beta-1)(\alpha-1)}) \quad (7.22b)$$

$$= \frac{1}{N^2} N \cdot O(1) + \frac{1}{N^2} N \cdot O(N^{(2\beta-1)(\alpha-1)}) \quad (7.22c)$$

$$= O(N^{-1}) + O(N^{2\beta(\alpha-1)-\alpha}) = O(N^{-1}), \quad (7.22d)$$

where in the last step we have assumed that $2\beta(\alpha-1) - \alpha < -1$. For practical values of $\beta \geq 1$ this inequality indeed holds, since $\alpha \in (0; 1)$. This means that the variance of the combined estimator is independent of α , and in fact its asymptotic rate is as high as that of BPT.

Bias

For the bias of the progressive VCM estimator we analogously obtain:

$$\text{Bias} [\hat{I}_N^{\text{VCM}}] = \frac{1}{N} \sum_{i=1}^N (\text{Bias}[C_{\text{VC},i}] + \text{Bias}[C_{\text{VM},i}]) \quad (7.23a)$$

$$= \frac{1}{N} \sum_{i=1}^N [0 + O(i^{\beta(\alpha-1)})O(i^{\alpha-1})] = O(N^{(\beta+1)(\alpha-1)}). \quad (7.23b)$$

This means that the bias of the combined estimator diminishes faster than that of PPM for the same α value, since $\beta > 0$ and also $(\alpha - 1) < 0$.

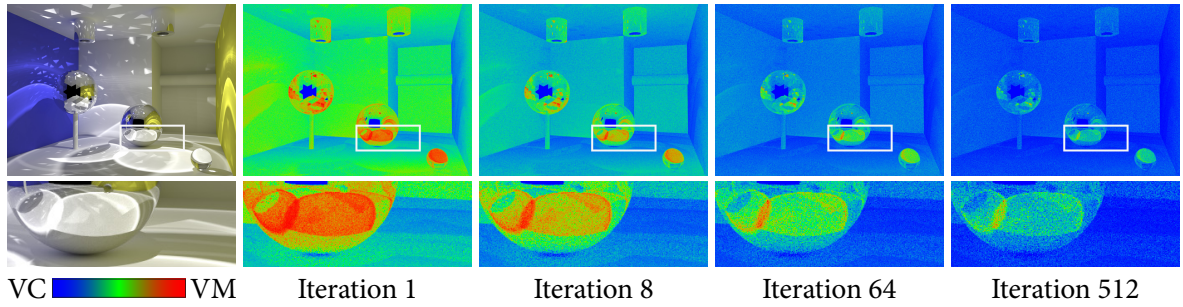


Figure 7.8: Relative contributions of vertex connection (VC) and vertex merging (VM) at different VCM rendering iterations. Note how VM’s contribution to the average estimate diminishes over time.

Mean squared error

Finally, for the MSE of the progressive VCM estimator (Eq. 7.16), which measures the total expected error of the pixel estimate, we get:

$$\text{MSE} [\hat{I}_N^{\text{VCM}}] = \text{Var} [\hat{I}_N^{\text{VCM}}] + \text{Bias}^2 [\hat{I}_N^{\text{VCM}}] \quad (7.24a)$$

$$= O(N^{-1}) + O\left(N^{2(\beta+1)(\alpha-1)}\right). \quad (7.24b)$$

For $\alpha \leq \frac{2\beta+1}{2\beta+2}$, the MSE of the progressive estimator has a maximum convergence rate of $O(N^{-1})$, which is equal to that of unbiased estimators. When using the balance heuristic, i.e. $\beta = 1$, this optimal rate is achieved for *any* $\alpha \in (0; 0.75]$. This means that our algorithm inherits the higher convergence rate of bidirectional path tracing and is thus *asymptotically faster than progressive photon mapping*, whose maximum MSE rate is only $O(N^{-2/3})$. Moreover, this result is achieved for a wide range of values for the parameter α .

7.4.3 Discussion

The intuition behind the high error convergence rate of our progressive VCM estimator is that the contribution of vertex merging (VM) to the running image estimate diminishes as the number of iterations grows, as we demonstrate in Figure 7.8. As we progressively shrink the radius r_i , we increase variance of VM (Eq. 7.19b). The power heuristic automatically compensates for this by assigning a reciprocal weight (Eq. 7.20b). The resulting algorithm is thus asymptotically equivalent to BPT. However, VM brings efficient initial variance reduction, which helps to faster achieve acceptable image quality with a finite number of samples, as we show in the following section.

We note that the above asymptotic result only holds for light transport paths that can be sampled by the unbiased techniques of BPT. The contribution of paths that do not have a DD segment, starting on a point source and ending on a pinhole camera [142], still converge at the slower rate of PPM. A prominent example for such paths are caustics from a point or a directional light source seen through a reflection or refraction.

7.5 Results

We now present an empirical evaluation of our method. We provide a comprehensive set of results for four scenes with different geometry and illumination configurations. For each scene, we provide images and statistics taken after 4 minutes of progressive rendering. These results are assembled in Figures 7.11–7.14 at the end of this section. (Readers of the electronic version are encouraged to zoom in the document for closer inspection of the images.)

7.5.1 Setup

On each of the four scenes in Figures 7.11–7.14 we compare seven rendering algorithms:

- our vertex connection and merging (VCM),
- path tracing (PT) [66],
- bidirectional path tracing (BPT) [143],
- stochastic progressive photon mapping (PPM) [38, 76],
- a combination of BPT and PPM via caustic path classification (BPT-PPM),
- Metropolis light transport [145] (MLT-Veach), and
- the primary-space MLT formulation of Kelemen et al. [72] (MLT-Kelemen).

The BPT-PPM algorithm handles caustic paths with PPM and all other paths with BPT. We classify a path as caustic if it either contains an *SDS* segment or ends with an *SDE* segment (i.e. caustics directly seen from the eye). Our PPM implementation closely follows Knaus and Zwicker [76] and handles glossy BSDFs as described by Hachisuka and Jensen [38].

Rendering is performed progressively using one eye subpath per pixel per iteration with an image resolution of 1024×768 . The numbers in parentheses for each algorithm denote the number of rendering iterations, proportional to the total number of samples (full paths), taken in the given time. Each iteration starts by sampling the same number of light subpaths as there are image pixels (i.e. $\approx 786k$). All light subpaths are reused by PPM and by the vertex merging estimators in VCM for every pixel. For vertex connections in BPT and VCM we assign one light subpath to each pixel, i.e. we set $n_{VC} = 1$ and $n_{VM} \approx 786k$ in the combined VCM estimator (Eq. 7.14). We set the radius reduction parameter to $\alpha = 2/3$ for both PPM and our VCM.

All measurements have been obtained on a 4-core Intel Core i7-860 2.8GHz processor machine. All rendering algorithms have been implemented in the same CPU-based ray tracing framework, with the exception of MLT-Kelemen and MLT-Veach for which we have used the Mitsuba renderer [55]. To improve the fairness of the comparison, we let Mitsuba run $3 \times$ longer than the algorithms implemented in our slightly faster renderer.

For each scene we also show statistics from two image quality metrics: the structural similarity index [155] (SSIM) and the visual difference predictor of Mantiuk et al. [91] (HDR-VDP-2). We have used the authors' implementations with the recommended default parameters, and measured the statistics against the reference images in the middle of each figure.

7.5.2 Visual comparison

We compare unconverged images so that the noise differences are more noticeable. The reference image for the MIRROR BALLS scene was obtained using PPM. Our VCM algorithm was used for all the other reference images, as none of the compared methods was able to provide noise-free results in a reasonable amount of time. For each scene we also include color-coded relative error images between the reference and BPT, PPM, and our VCM, as well as a color-coded image showing the relative contributions of all vertex connection (VC) and vertex merging (VM) techniques to the VCM image.

The LIVINGROOM scene in Figure 7.11 shows an environment with different “illumination scales”. Most of the diffuse illumination comes from far away (behind the camera), resulting in excessive noise in PPM. The objects on the desk are lit by two local area light sources, which, in combination with the mirror and the vases, produce intricate caustic paths that are difficult for BPT but well handled by PPM. Our combined VCM algorithm captures all light interactions efficiently, automatically finding a good balance between the many sampling techniques it has at its disposal.

The BATHROOM scene in Figure 7.12 is a compact environment with moderately glossy tiles and highly glossy chrome elements. It is illuminated by two small area lights, each almost fully enclosed in a metal-glass shell. This configuration poses a challenge to both BPT and PPM. Our VCM handles the entire illumination in the scene robustly, not only because adaptively combines the two methods but also because it includes more sampling techniques: unlike PPM, VCM merges light vertices at *every* eye subpath vertex. The benefit is clearly visible on the water tap.

The highly glossy CAR with specular chrome elements in Figure 7.13 is placed in a semi-open studio environment which is illuminated by two rectangular light sources from above. The car interior is additionally lit by a small rectangular light enclosed in a glass shell. This illumination, seen through the windows, as well as the caustic reflections in the exterior, are difficult for BPT. On the other hand, PPM performs poorly on the diffuse illumination and on the glossy inter-reflections, e.g. between the tire and the fender. Our VCM renders a smooth, low-noise image.

The MIRROR BALLS scene in Figure 7.14 is illuminated by metal-enclosed disc lights on the ceiling whose light is focused by lenses attached to the lamp shades. This entirely caustic illumination, which is additionally reflected in the balls, makes this scene well suited for PPM. Nevertheless, BPT handles the directly visible caustics more efficiently than PPM via vertex connections to the eye, but then relies on inefficient unidirectional sampling to find the reflections of those caustics. Even though VCM is more than $2\times$ slower per iteration than PPM, and thus slightly noisier on the reflected caustics, it once again delivers the image with the highest overall quality.

While a combination of BPT and PPM based on path classification can often produce better results than either method alone, it can be far from optimal, as can be seen when comparing the BPT-PPM images to the results achieved by our VCM algorithm. These results show that such heuristic classification can be far from optimal in the presence of glossy objects as well as for short-range diffuse inter-reflections between specular interactions (i.e. *SDDS* segments) when compared to VCM. We also experimented with different path classification strategies, but none could deliver quality similar to VCM, which employs more path sampling techniques than BPT and PPM combined and also often mixes vertex connection (VC) and vertex merging (VM) techniques with roughly equal weights. Such equal-weight combinations appear in green in the color-coded relative VC-VM contribution images.

Even with a $3\times$ larger time budget, the MLT algorithms cannot efficiently capture the complex light interactions resulting from the various configurations of specular and glossy objects in our scenes. When they occasionally manage to find a high-contribution path, their Metropolis samplers get stuck exploring a small neighborhood region in the path space, which skews the result and produces spurious image artifacts.

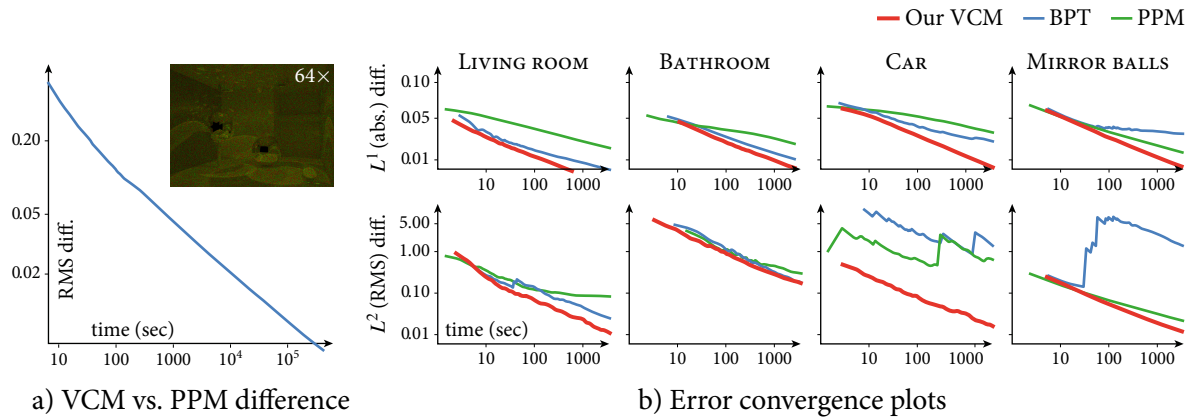


Figure 7.9: Error convergence plots. a) A time plot of the RMS difference between the images produced by our progressive VCM algorithm and PPM on the MIRROR BALLS scene, along with a $64\times$ amplified color-coded difference image taken at the end of the measurement. b) Mean absolute (L^1) and root mean squared (L^2) log-log reference difference plots for VCM, BPT, and PPM on the four scenes shown in Figures 7.11–7.14. The oscillations in the plots are due to the “fireflies” caused by the low sampling probability of certain high-contribution paths which increase the error of the produced image when found occasionally.

7.5.3 Image quality metrics

The bottom row in each of the Figures 7.11–7.14 shows the statistics obtained from the SSIM and HDR-VDP-2 image quality metrics. Overall, our VCM algorithm gets highest visual quality scores from both metrics. There are some slight inconsistencies between the two metrics, due to SSIM operating on gamma-corrected low dynamic range images and HDR-VDP-2 operating on the raw high dynamic range images also taking into account the observer’s luminance adaptation. These inconsistencies are noticeable for PPM on the CAR scene and for BPT-PPM on the MIRROR BALLS scene.

Running the image quality metrics on the MLT images would require separate reference images, due to the slight differences in the materials between Mitsuba and our renderer. Unfortunately, we were not able to obtain such references in reasonable time with the algorithms available in Mitsuba and therefore do not provide these results.

7.5.4 Numerical convergence

To verify that our VCM algorithm converges to the correct solution, we measure the root mean squared (RMS) difference between the images produced by VCM and PPM on the MIRROR BALLS scene. Figure 7.9a shows a plot of the steadily decreasing difference over time. This plot confirms that our algorithm and PPM converge to the same solution, and the $64\times$ amplified color-coded difference image taken at the end of the measurement indicates that any remaining differences are due to random noise.

The slopes of the log-log reference error plots for BPT, PPM, and VCM in Figure 7.9b show that our combined VCM algorithm converges to the reference solution at a higher rate than PPM. The oscillations in the plots are due to the “fireflies” caused by low sampling probability for high contribution paths which rapidly increase the error of the image when found occasionally. For BPT and PPM, these paths mostly involve glossy inter-reflections; for VCM, these are highly glossy and perfectly specular (i.e. LS^+E) unidirectionally sampled paths.

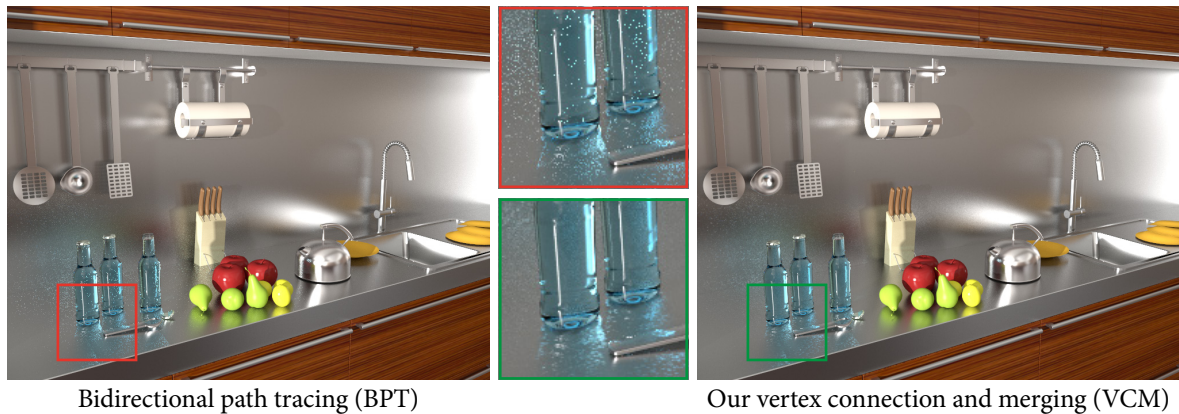


Figure 7.10: A scene dominated by glossy materials rendered with BPT and our progressive VCM in 11 hours. Even though VCM brings a noticeable improvement, the caustics on the glossy plot remain noisy because none of the many techniques it employs can sample those paths with high enough probability.

7.6 Discussion

We conclude this chapter with a short summary of the parameters of our VCM algorithm and its limitations. We also briefly discuss the alternative formulation of the method in the extended path space by Hachisuka et al. [42].

Parameter choice

Two parameters that our progressive VCM algorithm inherits from PPM are the initial merging radius r_1 and its reduction rate parameter α . In our experiments, we globally set r_1 to 0.01% – 0.07% of the scene’s bounding box for both PPM and VCM. For a fair comparison against PPM, we also use $\alpha = 2/3$ for both methods. In general, for VCM we recommend setting r_1 smaller than for PPM and using $\alpha = 0.75$. Such settings introduce less initial bias and maximize the variance convergence rate of vertex merging and thus its efficiency too. In the following chapter we will discuss a better approach for choosing the initial radius adaptively for every pixel.

Limitations

Our VCM algorithm inherits from PM the ability to approximately capture *SDS* paths due to point light sources, which BPT cannot sample. However, since such paths can only be handled by vertex merging, the error convergence rate of their contribution in our progressive VCM is as low as in PPM. This limitation does not apply to the scenes shown in Figures 7.11–7.14, in which we use only physically plausible light sources with finite area. Slightly counter-intuitively, this means that, in the presence of specular objects, area light sources are asymptotically more efficient to handle than point lights.

While our combined VCM algorithm is more robust than each of its ingredients alone, it does not perform better on paths that are poorly handled by both BPT and PM. Such paths are, for example, caustics falling on a highly glossy surface, as we show in Figure 7.10. Even though VCM improves over BPT on this difficult scene, the caustics on the glossy kitchen plot remain noisy even after 11 hours of rendering. Efficient handling of such cases is a challenging avenue for future work.

Finally, since our combined VCM estimator is an MIS estimator, it can be sub-optimal when some of its sampling techniques are much more efficient than the rest. A prominent example are scenes with diffuse and moderately glossy materials, which are well suited for path tracing with next event estimation and/or bidirectional path tracing.

Extended path space formulation

Independently of our work, Hachisuka et al. [42] have also developed a method that combines BPT and PM using multiple importance sampling (MIS). Similarly to our approach, they interpret photon radiance estimation as a path sampling technique and use MIS to combine techniques from PM and BPT. However, as a major difference to our approach, their formulation considers the extended path space (Sec. 7.1.1) and they augment the BPT techniques with virtual random perturbations of path vertices in order to express the vertex connection pdfs in this higher-dimensional space. Interestingly, this formulation yields a combined estimator that is equivalent to our VCM estimator (Eq. 7.14). We believe that this result confirms the validity of both approaches.

Hachisuka et al. [42] also derive provably good MIS weights that account for the bias in some of the techniques. Unfortunately, these weights are difficult to use in practice as they require computing an accurate estimate for the bias of the photon mapping estimator. Our derivations instead focus on the optimal asymptotic behavior of the progressive combined estimator (Eq. 7.16).

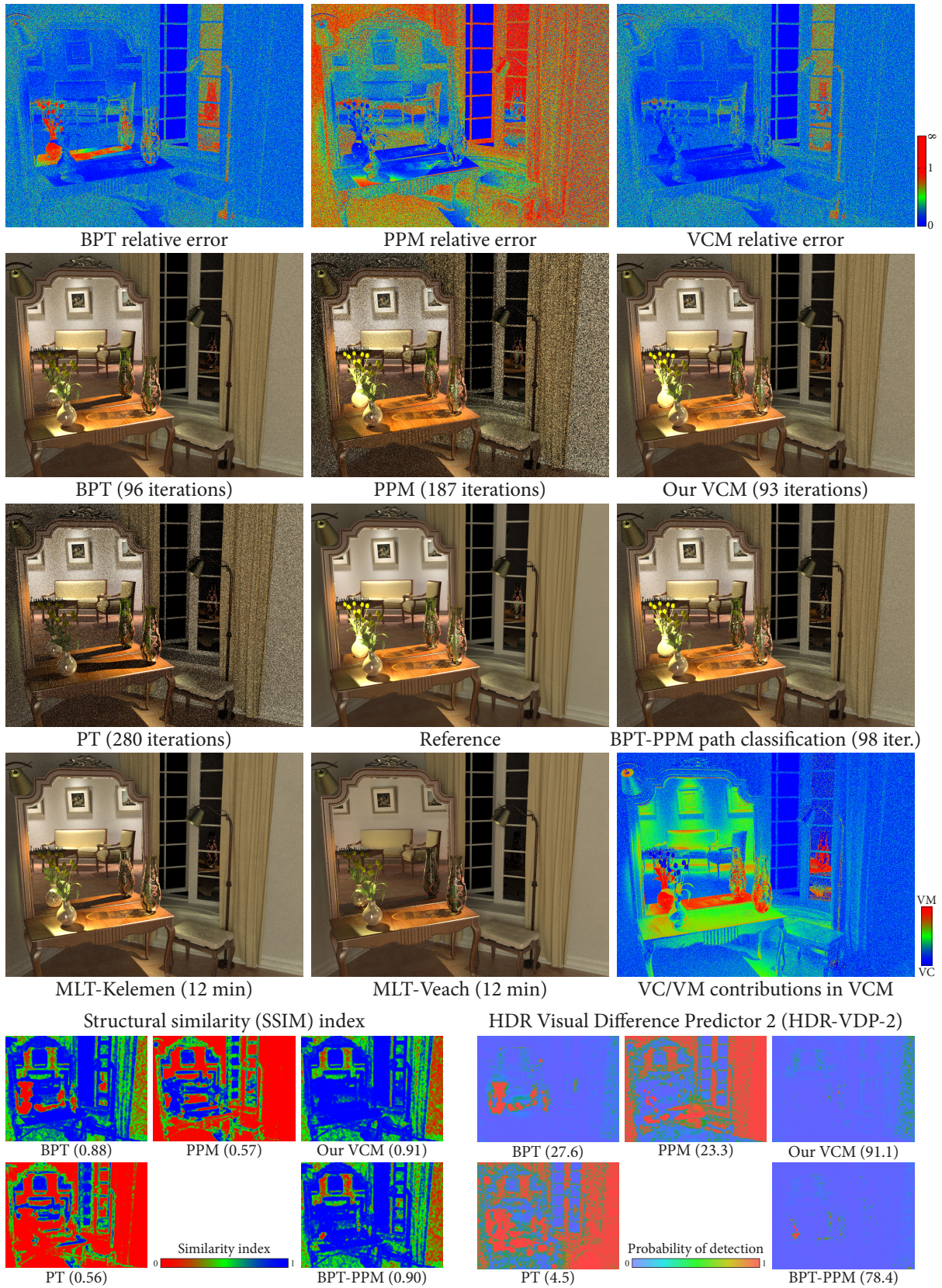


Figure 7.11: A comparison of seven rendering algorithms on the LIVING ROOM scene, described in Section 7.5.1. We also include error images and statistics from two image quality metrics.

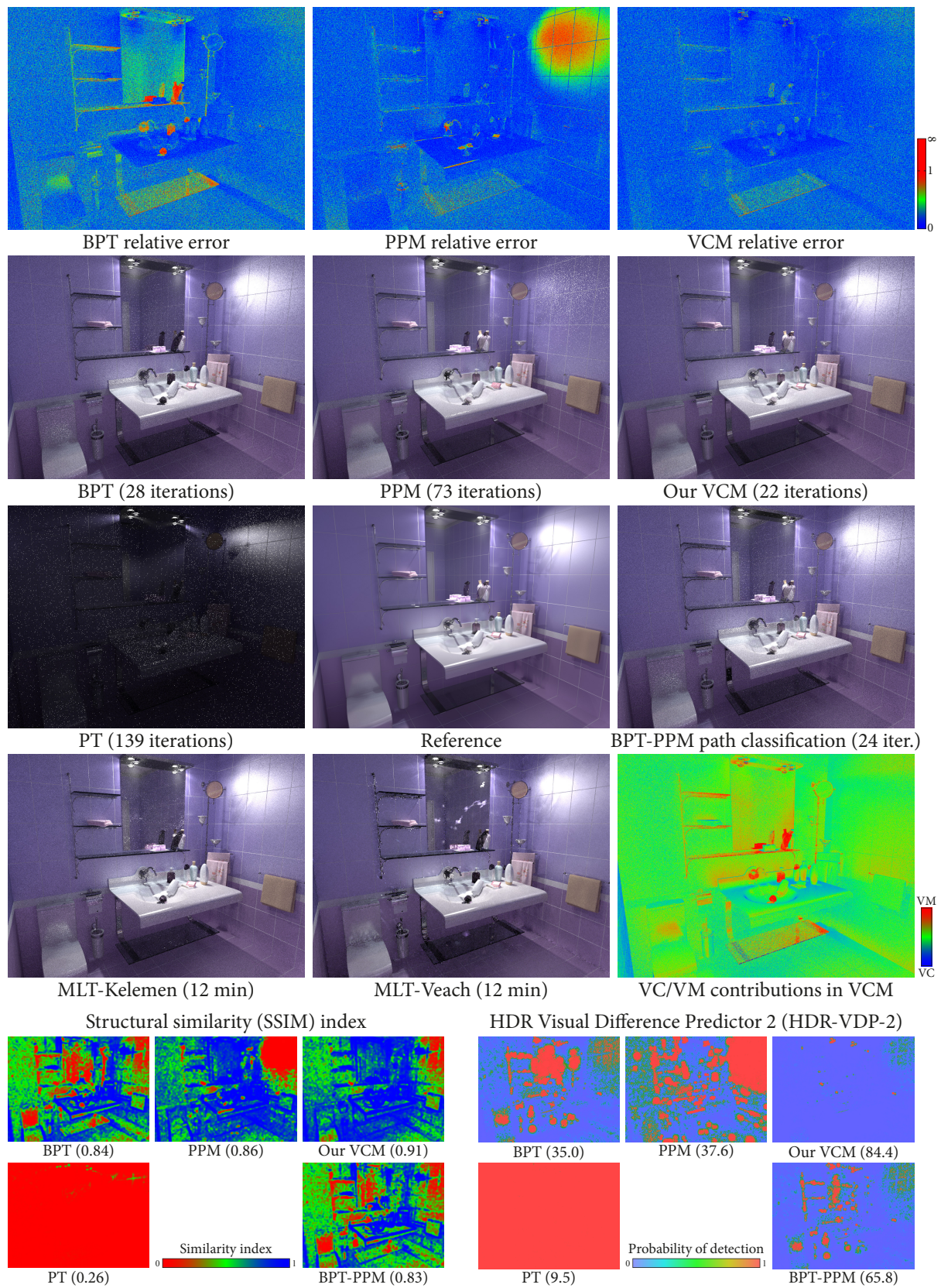


Figure 7.12: A comparison of seven rendering algorithms on the BATHROOM scene, described in Section 7.5.1. We also include error images and statistics from two image quality metrics.

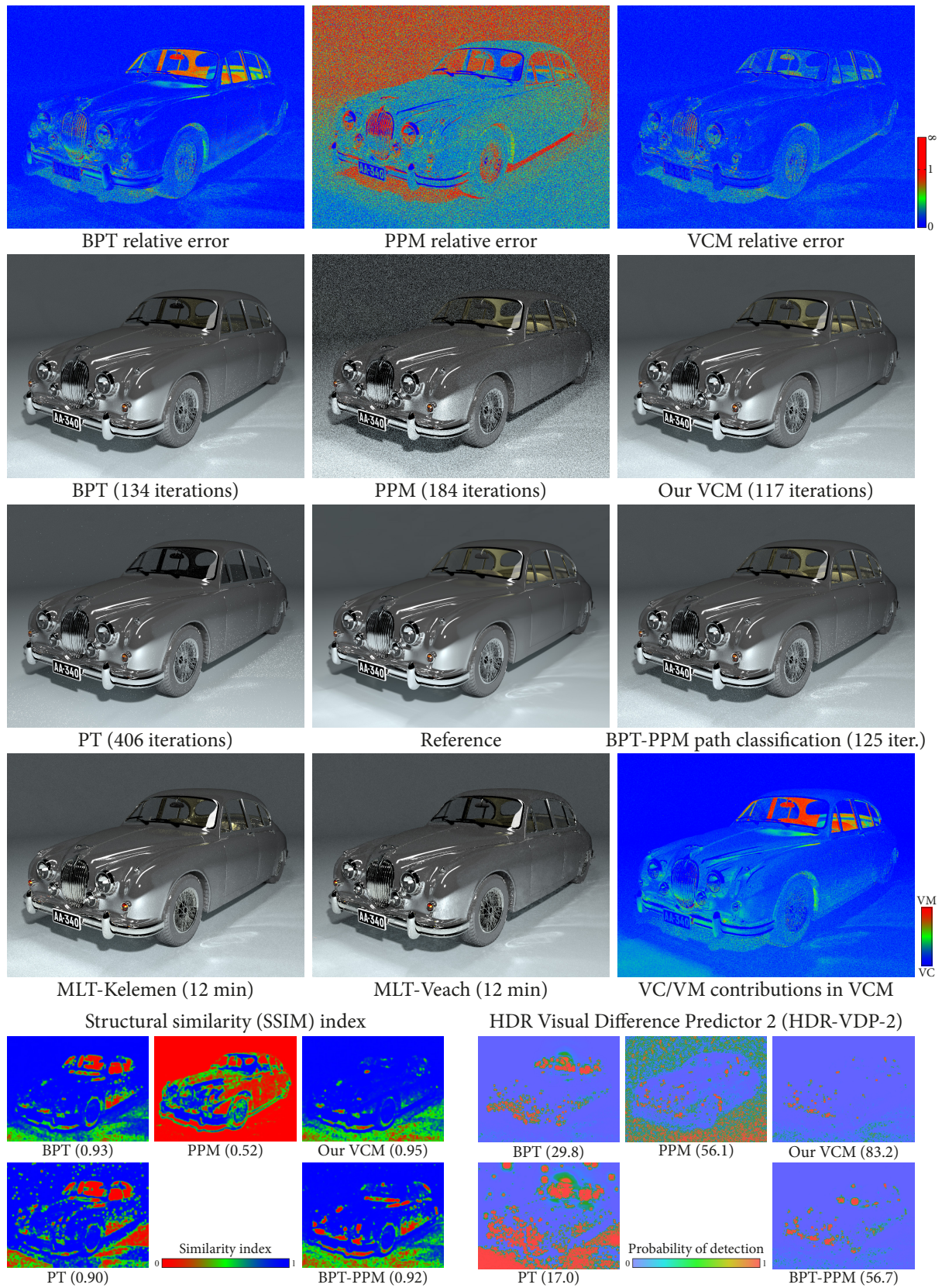


Figure 7.13: A comparison of seven rendering algorithms on the CAR scene, described in Section 7.5.1. We also include error images and statistics from two image quality metrics.

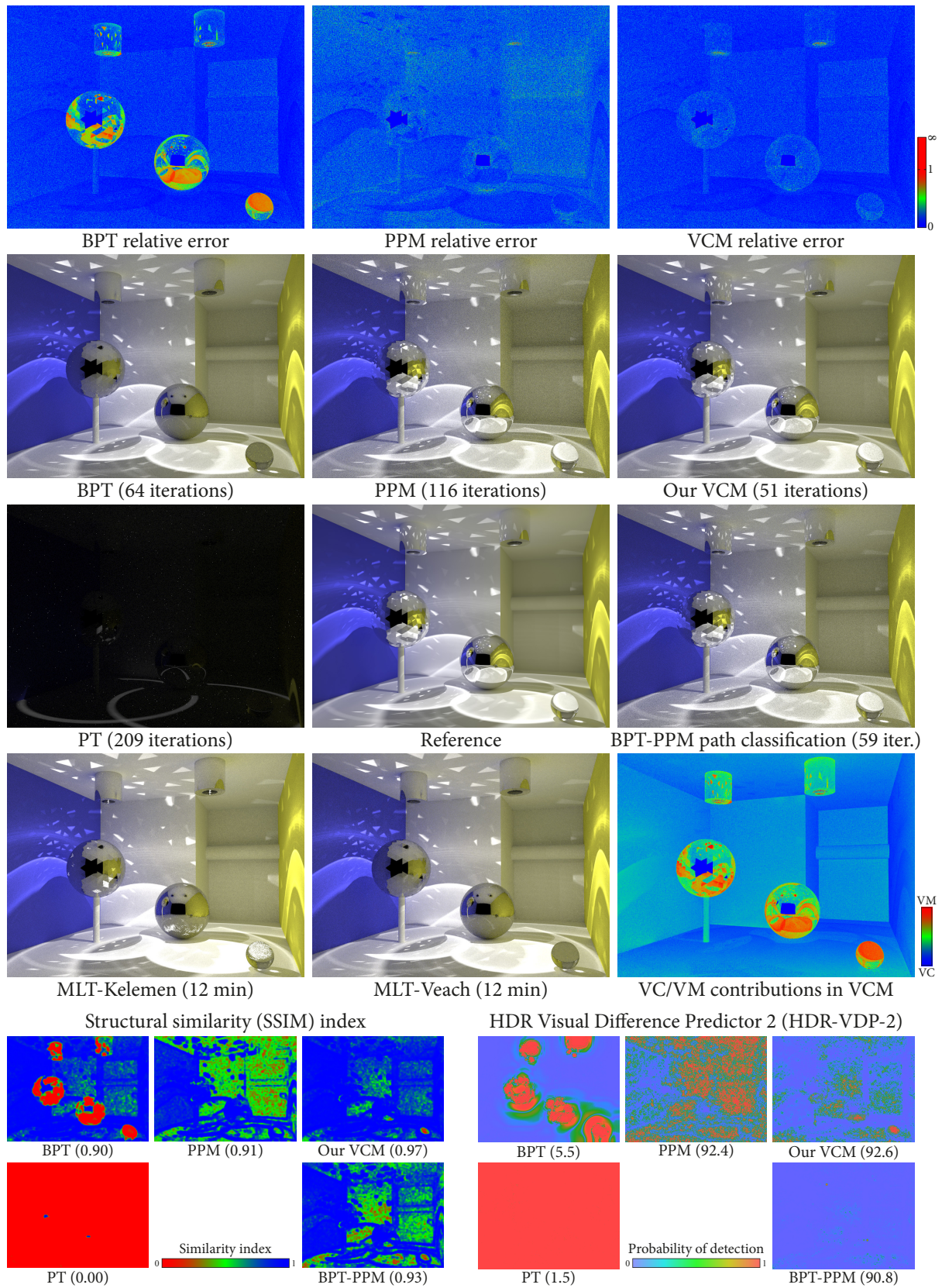


Figure 7.14: A comparison of seven rendering algorithms on the MIRROR BALLS scene, described in Section 7.5.1. We also include error images and statistics from two image quality metrics.

Implementing Vertex Connection and Merging

8

Bidirectional path tracing (BPT) and photon mapping (PM) are arguably the two most versatile physically based rendering algorithms available today. It has been long acknowledged that these two methods complement each other in terms of the types of light transport effects they can efficiently capture. The *vertex connection and merging* (VCM) algorithm we presented in Chapter 7 aims to leverage the advantages of both methods by adaptively combining *vertex connection* techniques from BPT and *vertex merging* techniques from PM via multiple importance sampling. We showed that this unified algorithm can efficiently handle a wide range of lighting effects and can be substantially more robust than either BPT or PM alone, while preserving the higher asymptotic performance of BPT.

In Chapter 7 we focused on the formal derivation, asymptotic analysis, and evaluation of the VCM algorithm. In this chapter we address some important details in its practical implementation, most notably the evaluation of the multiple importance sampling (MIS) weights in the combined pixel estimator (Eq. 7.14). Indeed, correctly implementing MIS is already challenging in BPT, and VCM increases the complexity by including even more sampling techniques. Moreover, the cheap light subpath reuse with vertex merging allows for efficiently constructing large amounts of full paths for each pixel, which in turn significantly increases the impact of path weight evaluation on the overall performance. The traditional BPT-style MIS weight computation [142, Sec. 10.2], which iterates over all vertices of every full path, can therefore become inefficient. We address this problem by devising a method to accumulate and store partial weights at the vertices of light and eye subpaths as they are sampled during the random walks. This allows us to quickly evaluate the weight for any full path by only using data that is readily available at the two vertices that are connected or merged. The MIS weight evaluation then becomes as efficient as the evaluation of the subpath sampling throughput which already benefits from such an accumulation scheme (Sec. 4.2.1). Our method is similar to the one independently developed by van Antwerpen [141] for BPT but in addition accounts for vertex merging techniques.

In addition to the efficient MIS weight evaluation, we discuss how to handle emission and scattering distributions whose definitions involve delta distributions. We also present an alternative, more memory-efficient implementation of the VCM algorithm and describe a few extensions such as using a different merging radius for each light transport path, as well as spectral and motion blur rendering.

8.1 Notation

The technical nature of the content in this chapter requires the extensive use of mathematical notation. In this section, we review our path and subpath notation and introduce a shorthand path vertex pdf notation. Table 8.1 summarizes the most commonly used symbols in this chapter.

Symbol	Description
$\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$	Full length- k path: vertex \mathbf{x}_0 is on a light source, \mathbf{x}_k is on the eye lens
$\bar{\mathbf{y}} = \mathbf{y}_0 \dots \mathbf{y}_{s-1}$	Light subpath with first vertex $\mathbf{y}_0 \equiv \mathbf{x}_0$ on a light
$\bar{\mathbf{z}} = \mathbf{z}_0 \dots \mathbf{z}_{t-1}$	Eye subpath with first vertex $\mathbf{z}_0 \equiv \mathbf{x}_k$ on the eye lens
$\vec{p}_i, \overleftarrow{p}_i$	Forward (i.e. actual) and reverse area pdfs for subpath vertex i ($\vec{p}_i = \vec{p}_{\omega,i} \vec{g}_i$ and $\overleftarrow{p}_i = \overleftarrow{p}_{\omega,i} \overleftarrow{g}_i$)
$\vec{p}_{\omega,i}, \overleftarrow{p}_{\omega,i}$	Forward and reverse solid angle pdfs for subpath vertex i
$\vec{g}_i, \overleftarrow{g}_i$	Forward and reverse pdf conversion factors from solid angle measure to area measure
$p_{VC,s,t}, p_{VC,s}$	Vertex connection (VC) pdf for a length- k path with s light vertices and $t = k+1-s$ eye vertices
$p_{VM,s,t}, p_{VM,s}$	Vertex merging (VM) pdf for a length- k path with s light vertices and $t = k+2-s$ eye vertices
n_{VC}, n_{VM}	Number of samples (i.e. paths) used for vertex connection and vertex merging, respectively
$\eta_{VCM} = \frac{n_{VM}}{n_{VC}} \pi r^2$	Shorthand for common constants that appear in the MIS path weights

Table 8.1: A list of some commonly used symbols in this chapter. Figure 8.1 illustrates the redundant (sub)path notation and the vertex pdf notation.

Paths

In the path integral formulation of light transport, which we introduced in Section 3.5.4, a path $\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$ of length k (edges) is a tuple of $k+1$ vertices, where the vertex \mathbf{x}_0 is on a light source and \mathbf{x}_k is on the eye lens. A Monte Carlo estimator for the path integral (Eq. 3.27) is constructed by sampling a random path $\bar{\mathbf{x}}$ and dividing its measurement contribution $f(\bar{\mathbf{x}})$ by its probability density $p(\bar{\mathbf{x}})$.

As we discussed in Section 4.2, Monte Carlo global illumination methods sample light transport paths by tracing random walks in scene. Bidirectional methods construct a full path $\bar{\mathbf{x}}$ by joining the endpoints of one subpath traced from a light source and another one traced from the eye. We denote a light subpath with s vertices by $\bar{\mathbf{y}} = \mathbf{y}_0 \dots \mathbf{y}_{s-1}$ and an eye subpath with t vertices by $\bar{\mathbf{z}} = \mathbf{z}_0 \dots \mathbf{z}_{t-1}$. Here, the vertex \mathbf{y}_0 is a point on a light source, and \mathbf{z}_0 is on the eye lens. These notations, illustrated in Figure 8.1, were already introduced in Section 4.2.1 but we repeat them here for reference. They are redundant with the \mathbf{x} -notation but conveniently index the vertices in the order of their generation. The forward and reverse vertex pdf notation, defined for $\bar{\mathbf{y}}$ below, also applies to $\bar{\mathbf{z}}$, and this symmetry will later allow us to apply the same MIS weight derivations to both light and eye subpaths.

Forward vertex pdfs

The probability density function (pdf) of a (sub)path describes the joint distribution of its vertices via a chain of marginal conditional pdfs. We denote the pdf of subpath vertex i by

$$\vec{p}_i(\bar{\mathbf{y}}) = \begin{cases} p(\mathbf{y}_0) & \text{if } i = 0, \\ \vec{p}_{\omega,i}(\bar{\mathbf{y}}) \vec{g}_i(\bar{\mathbf{y}}) & \text{otherwise,} \end{cases} \quad (8.1)$$

with

$$\vec{p}_{\omega,i}(\bar{\mathbf{y}}) = \begin{cases} p(\omega_{\mathbf{y}_0 \mathbf{y}_1} | \mathbf{y}_0) & \text{if } i = 1, \\ p(\omega_{\mathbf{y}_{i-1} \mathbf{y}_i} | \mathbf{y}_{i-1}, \omega_{\mathbf{y}_{i-2} \mathbf{y}_{i-1}}) & \text{if } i > 1 \end{cases} \quad (8.2)$$

$$\vec{g}_i(\bar{\mathbf{y}}) = \frac{|\cos \theta_{i,i-1}|}{\|\mathbf{y}_i - \mathbf{y}_{i-1}\|^2}. \quad (8.3)$$

Above, $p(\cdot)$ denotes an unconditional vertex pdf expressed w.r.t. the area measure, e.g. $p(\mathbf{y}_0)$ for sampling \mathbf{y}_0 on a light source. The subscript ω denotes a solid angle pdf. The factor \vec{g}_i converts the pdf

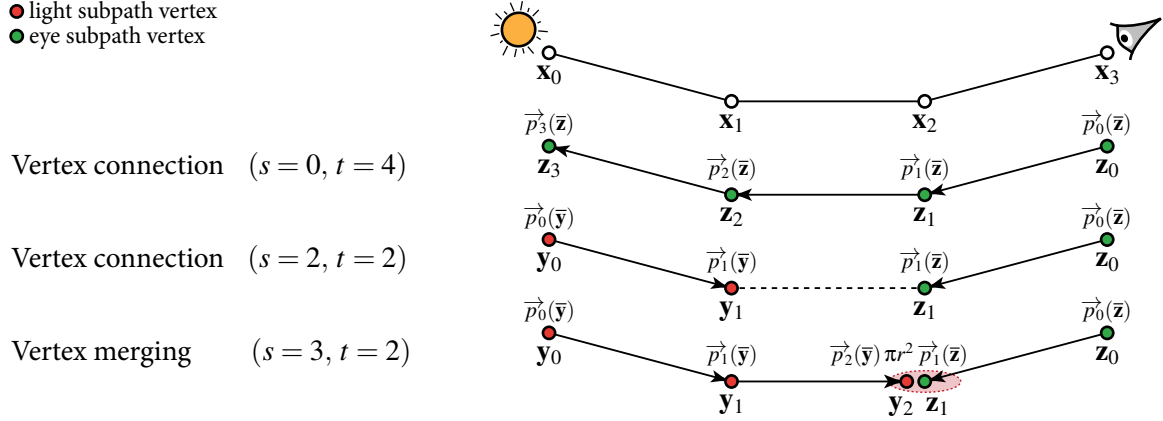


Figure 8.1: Illustration of the vertex connection (VC) and vertex merging (VM) path sampling techniques, along with their associated vertex pdfs. Note that the different VC and VM techniques for sampling length- k paths can be uniquely identified by the number of light subpath vertices s . Note also that in this chapter we consider the “photon” vertex (\mathbf{y}_2 in this example) as part of the light subpath.

measure from solid angle to area, with $\theta_{i,i-1}$ being the angle between the surface normal at \mathbf{y}_i and the vector $\overrightarrow{\mathbf{y}_i \mathbf{y}_{i-1}}$. We call \vec{p}_i forward vertex pdfs (w.r.t. the random walk direction). With this notation, the (joint) pdf of a subpath $\bar{\mathbf{y}}$ with s vertices is:

$$p_{s-1}(\bar{\mathbf{y}}) = p(\mathbf{y}_0, \dots, \mathbf{y}_{s-1}) = \prod_{i=0}^{s-1} \vec{p}_i(\bar{\mathbf{y}}), \quad (8.4)$$

which is similar to the subpath pdf notation we defined in Equation 4.5 but is more concise. All these definitions apply without modification to eye subpaths $\bar{\mathbf{z}}$ as well.

Reverse vertex pdfs

Analogously to the forward vertex pdf notation, we define a *reverse* notation:

$$\overleftarrow{p}_i(\bar{\mathbf{y}}) = \begin{cases} p(\mathbf{y}_k) & \text{if } i = k, \\ \overleftarrow{p}_{\omega,i}(\bar{\mathbf{y}}) \overleftarrow{g}_i(\bar{\mathbf{y}}) & \text{otherwise,} \end{cases} \quad (8.5)$$

with

$$\overleftarrow{p}_{\omega,i}(\bar{\mathbf{y}}) = \begin{cases} p(\omega_{\mathbf{y}_k \mathbf{y}_{k-1}} | \mathbf{y}_k) & \text{if } i = k - 1, \\ p(\omega_{\mathbf{y}_{i+1} \mathbf{y}_i} | \mathbf{y}_{i+1}, \omega_{\mathbf{y}_{i+2} \mathbf{y}_{i+1}}) & \text{if } i < k - 1 \end{cases} \quad (8.6)$$

$$\overleftarrow{g}_i(\bar{\mathbf{y}}) = \frac{|\cos \theta_{i,i+1}|}{\|\mathbf{y}_i - \mathbf{y}_{i+1}\|^2}. \quad (8.7)$$

The arrows make it easier to distinguish between the actual sampling pdf of a vertex, $\vec{p}_i(\bar{\mathbf{y}})$, and its “reverse” pdf, $\overleftarrow{p}_i(\bar{\mathbf{y}})$. That is, the latter denotes the probability density for sampling \mathbf{y}_i in a direction opposite to that of the random walk. In Equation 8.5, for example, \mathbf{y}_k is a light subpath vertex that has landed on the eye lens, and $\overleftarrow{p}_k(\bar{\mathbf{y}}) \equiv p(\mathbf{y}_k)$ denotes the probability density for sampling that same vertex on the lens using a different technique, e.g. to start an eye subpath. These reverse pdfs are required for evaluating the multiple importance sampling path weights in the VCM pixel estimator (Eq. 7.14). Recall that all this notation applies to eye subpaths $\bar{\mathbf{z}}$ as well.

8.2 Recursive path weight evaluation

The heart of our VCM algorithm is the multiple importance sampling (MIS) pixel estimator that combines primary vertex connection estimators \hat{I}^{VC} and primary vertex merging estimators \hat{I}^{VM} :

$$\hat{I}^{\text{VCM}} = \frac{1}{n_{\text{VC}}} \sum_{j=1}^{n_{\text{VC}}} \sum_{s \geq 0, t \geq 0} w_{\text{VC},s,t}(\bar{\mathbf{x}}_{s,t,j}) \hat{I}_{s,t}^{\text{VC}}(\bar{\mathbf{x}}_{s,t,j}) + \quad (8.8a)$$

$$\frac{1}{n_{\text{VM}}} \sum_{j=1}^{n_{\text{VM}}} \sum_{s \geq 2, t \geq 2} w_{\text{VM},s,t}(\bar{\mathbf{x}}_{s,t,j}) \hat{I}_{s,t}^{\text{VM}}(\bar{\mathbf{x}}_{s,t,j}). \quad (8.8b)$$

This estimator considers one eye subpath through the corresponding pixel, whose vertices are connected to the vertices of n_{VC} light subpaths and potentially merged with the vertices of n_{VM} light subpaths. Our implementation uses $n_{\text{VC}} = 1$, and we set n_{VM} to the total number of sampled light subpaths, which for symmetry reasons we choose to be equal to the total number of eye subpaths, i.e. the image resolution.

The power heuristic weight for technique (v, s, t) in Equation 7.15 can be written as:

$$w_{v,s,t}(\bar{\mathbf{x}}) = \frac{1}{\frac{n_{\text{VC}}^{\beta}}{n_v^{\beta}} \sum_{s' \geq 0, t' \geq 0} \frac{p_{\text{VC},s',t'}^{\beta}(\bar{\mathbf{x}})}{p_{v,s,t}^{\beta}(\bar{\mathbf{x}})} + \frac{n_{\text{VM}}^{\beta}}{n_v^{\beta}} \sum_{s' \geq 2, t' \geq 2} \frac{p_{\text{VM},s',t'}^{\beta}(\bar{\mathbf{x}})}{p_{v,s,t}^{\beta}(\bar{\mathbf{x}})}}, \quad (8.9)$$

which takes into account all possible ways of sampling the path $\bar{\mathbf{x}}$ via vertex connection or merging. Here, v is either VC or VM, and the path pdfs for the VC and VM technique families are (see Equations 4.16 and 7.11, respectively)

$$p_{\text{VC},s,t}(\bar{\mathbf{x}}) = p_{s-1}(\bar{\mathbf{y}}) p_{t-1}(\bar{\mathbf{z}}) \quad (8.10a)$$

$$p_{\text{VM},s,t}(\bar{\mathbf{x}}) = p_{s-1}(\bar{\mathbf{y}}) p_{t-1}(\bar{\mathbf{z}}) \pi r^2, \quad (8.10b)$$

where r is the vertex merging radius. It is important to note at this point that, unlike Chapter 7, in this chapter we adhere to the more intuitive definition of vertex merging which considers the “photon” vertex to be the endpoint of the light subpath $\bar{\mathbf{y}}$, as this view conveniently aligns with the actual implementation. With this interpretation, a full VM path (called *extended path* in Chapter 7) constructed by a (VM, s, t) technique is one segment (edge) shorter than a VC path constructed by a (VC, s, t) technique, as illustrated in Figure 8.1.

The naïve way to evaluate the weight $w_{v,s,t}(\bar{\mathbf{x}})$ is to independently compute each path pdf that appears in the denominator in Equation 8.9. For bidirectional path tracing, which uses the same formula, minus the VM sum on the right, Veach [142, p. 306] developed a more efficient scheme by exploiting the fact that many of the terms in the pdf fractions cancel out when the subpath pdfs in Equation 8.10 are expanded into chains of vertex pdfs (Eq. 8.4). He computes the entire VC sum by looping once over the light and eye subpath vertices and accumulating the sum of pdf fractions.

While Veach’s scheme can be easily extended to vertex merging, its efficiency is sub-optimal. First, it makes many redundant computations, since every time a subpath is reused in a different primary estimator all pdf terms associated with its vertices are re-evaluated. Moreover, the computation of these terms requires accessing all subpath vertices in memory. Weight evaluation can therefore become a significant overhead for vertex merging, which relies heavily on subpath reuse and constructs a large number of full paths with a single range search query.

We now note that the unweighted contribution of a path is in fact computed efficiently in both BPT and PM, and thus also in VCM. This is achieved by computing cumulative subpath throughputs during the random walks (Eq. 4.6). Upon connecting or merging two vertices, the corresponding primary estimate for the resulting full path is quickly evaluated using the sampling throughputs stored at those vertices. We set out to derive a similar scheme for the path weight evaluation. We go about this by reformulating the two sums in Equation 8.9 as recursive quantities that can be incrementally computed and cached at the subpath vertices during the random walks.

At this point, readers not interested in the formal derivation of our MIS weight evaluation scheme can skip to Section 8.3. There we discuss its practical implementation, describing the quantities stored at each subpath vertex and how to compute the full path weight from this data.

8.2.1 Partial subpath weights

In order to keep the notation succinct, in the following we will consider paths \bar{x} of length k and will often omit the redundant subscript t denoting the number of eye subpath vertices. Also, without loss of generality, we will assume $\beta = 1$, i.e. that the balance heuristic is used. We now rewrite Equation 8.9 more compactly:

$$w_{v,s,t} = \frac{1}{\frac{n_{VC}}{n_v} \sum_{j=0}^{k+1} \frac{p_{VC,j}}{p_{v,s}} + \frac{n_{VM}}{n_v} \sum_{j=2}^k \frac{p_{VM,j}}{p_{v,s}}}, \quad (8.11)$$

where $p_{v,j}$ is the pdf for sampling a length- k path using a light subpath with j vertices, and $v \in \{VC, VM\}$. We now write the weight for a length- k path in the form

$$w_{v,s,t} = \frac{1}{w_{v,s}^{\text{light}} + 1 + w_{v,s}^{\text{eye}}}, \quad (8.12)$$

where we have split and regrouped the sums in Equation 8.11 to iterate over the light and eye subpath vertices, respectively, and have extracted the term $p_{v,s}/p_{v,s} = 1$ from the appropriate sum. The partial subpath weights $w_{v,s}^{\text{light}}$ and $w_{v,s}^{\text{eye}}$ have slightly different definitions depending on the value of v .

Vertex connection. For paths sampled with $v = VC$, in the weight $w_{VC,s,t}$ we have:

$$w_{VC,s}^{\text{light}} = \sum_{j=0}^{s-1} \frac{p_{VC,j}}{p_{VC,s}} + \frac{n_{VM}}{n_{VC}} \sum_{j=2}^s \frac{p_{VM,j}}{p_{VC,s}} \quad (8.13)$$

$$w_{VC,s}^{\text{eye}} = \sum_{j=s+1}^{k+1} \frac{p_{VC,j}}{p_{VC,s}} + \frac{n_{VM}}{n_{VC}} \sum_{j=s+1}^k \frac{p_{VM,j}}{p_{VC,s}}. \quad (8.14)$$

The sums on the left in Equations 8.13 and 8.14 are obtained by splitting the left sum in Equation 8.11 in two (and extracting the term $p_{VC,s}/p_{VC,s} = 1$), and the sums on the right are obtained by splitting the right sum in Equation 8.11.

Vertex merging. Analogously, for paths sampled with $v = VM$, in the weight $w_{VM,s,t}$ we have:

$$w_{VM,s}^{\text{light}} = \frac{n_{VC}}{n_{VM}} \sum_{j=0}^{s-1} \frac{p_{VC,j}}{p_{VM,s}} + \sum_{j=2}^{s-1} \frac{p_{VM,j}}{p_{VM,s}} \quad (8.15)$$

$$w_{VM,s}^{\text{eye}} = \frac{n_{VC}}{n_{VM}} \sum_{j=s}^{k+1} \frac{p_{VC,j}}{p_{VM,s}} + \sum_{j=s+1}^k \frac{p_{VM,j}}{p_{VM,s}}. \quad (8.16)$$

8.2.2 Recursive formulation

We now reformulate the path weight from Equation 8.12 in a form that is suitable for evaluation in a *forward* manner, i.e. in the order of the generation of the subpath vertices, instead of iterating backwards from the subpath endpoints as suggested by Veach [142, p. 306]. We write the weight as

$$w_{v,s,t} = \frac{1}{\bar{w}_{v,s-1}(\bar{\mathbf{y}}) + 1 + \bar{w}_{v,t-1}(\bar{\mathbf{z}})}, \quad (8.17)$$

where $\bar{w}_{v,s-1}(\bar{\mathbf{y}})$ and $\bar{w}_{v,t-1}(\bar{\mathbf{z}})$ are recursive formulations of the partial light and eye subpath weights $w_{v,s}^{\text{light}}$ and $w_{v,s}^{\text{eye}}$ from the previous subsection. These two recursive quantities can be evaluated incrementally as the subpaths $\bar{\mathbf{y}}$ and $\bar{\mathbf{z}}$ are traced, so they can be readily available for summing up when we connect or merge two vertices. We next derive the two quantities separately for $v = \text{VC}$ and $v = \text{VM}$.

In the following, we will use a shorthand notation to group all parameters and constants that appear in the path weight into a single term:

$$\eta_{\text{VCM}} = \frac{n_{\text{VM}}}{n_{\text{VC}}} \pi r^2. \quad (8.18)$$

Vertex connection

Using the path pdf definitions in Equation 8.10, we follow Veach [142, p. 306] to expand the pdf fractions in Equation 8.13:

$$w_{\text{VC},s}^{\text{light}} = \sum_{j=0}^{s-1} \prod_{i=j}^{s-1} \frac{\overleftarrow{p}_i(\bar{\mathbf{y}})}{\overrightarrow{p}_i(\bar{\mathbf{y}})} + \eta_{\text{VCM}} \sum_{j=2}^s \overleftarrow{p}_{j-1}(\bar{\mathbf{y}}) \prod_{i=j}^{s-1} \frac{\overleftarrow{p}_i(\bar{\mathbf{y}})}{\overrightarrow{p}_i(\bar{\mathbf{y}})}, \quad (8.19)$$

where we use the forward and reverse vertex pdf notations from Equations 8.1 and 8.5. Next, we reformulate the two sums above as recursive quantities (omitting the $\bar{\mathbf{y}}$ arguments for readability):

$$\bar{w}'_{\text{VC},0} = \frac{\overleftarrow{p}_0}{\overrightarrow{p}_0} \quad \bar{w}''_{\text{VC},0} = 0 \quad (8.20a)$$

$$\bar{w}'_{\text{VC},i} = \frac{\overleftarrow{p}_i}{\overrightarrow{p}_i} (1 + \bar{w}'_{\text{VC},i-1}) \quad \bar{w}''_{\text{VC},i} = \overleftarrow{p}_i \left(1 + \frac{1}{\overrightarrow{p}_i} \bar{w}''_{\text{VC},i-1} \right), \quad (8.20b)$$

and write the partial light subpath weight in (Eq. 8.19) as

$$w_{\text{VC},s}^{\text{light}} = \bar{w}'_{\text{VC},s-1}(\bar{\mathbf{y}}) + \eta_{\text{VCM}} \bar{w}''_{\text{VC},s-1}(\bar{\mathbf{y}}). \quad (8.21)$$

We can further combine $\bar{w}'_{\text{VC},i}$ and $\bar{w}''_{\text{VC},i}$ to formulate Equation 8.21 as a single recursive quantity (again omitting the $\bar{\mathbf{y}}$ arguments):

$$\bar{w}_{\text{VC},0} = \frac{\overleftarrow{p}_0}{\overrightarrow{p}_0} \quad (8.22a)$$

$$\bar{w}_{\text{VC},i} = \overleftarrow{p}_i \left(\eta_{\text{VCM}} + \frac{1}{\overrightarrow{p}_i} + \frac{1}{\overrightarrow{p}_i} \bar{w}_{\text{VC},i-1} \right) \quad (8.22b)$$

Finally, thanks to the $\bar{\mathbf{z}}$ notation (Fig. 8.1) which indexes path vertices starting from the eye, the above recursive formula also applies to the partial eye subpath weight in Equation 8.14: $w_{\text{VC},s}^{\text{eye}} = \bar{w}_{\text{VC},t-1}(\bar{\mathbf{z}})$. Plugging $\bar{w}_{\text{VC},s-1}(\bar{\mathbf{y}})$ and $\bar{w}_{\text{VC},t-1}(\bar{\mathbf{z}})$ into Equation 8.17 completes the definition of $w_{\text{VC},s,t}$.

Vertex merging

We now expand equation Equation 8.15 as above:

$$w_{\text{VM},s}^{\text{light}} = \frac{1}{\eta_{\text{VCM}}} \frac{1}{\overrightarrow{p}_{s-1}(\bar{\mathbf{y}})} \sum_{j=0}^{s-1} \prod_{i=j}^{s-2} \frac{\overleftarrow{p}_i(\bar{\mathbf{y}})}{\overrightarrow{p}_i(\bar{\mathbf{y}})} + \sum_{j=2}^{s-1} \prod_{i=j}^{s-1} \frac{\overleftarrow{p}_{i-1}(\bar{\mathbf{y}})}{\overrightarrow{p}_i(\bar{\mathbf{y}})}. \quad (8.23)$$

Using the same methodology as for vertex connection above, we obtain:

$$\bar{w}_{\text{VM},1} = \frac{1}{\overrightarrow{p}_1} \left(\frac{1}{\eta_{\text{VCM}}} + \overleftarrow{p}_0 \frac{1}{\eta_{\text{VCM}} \overrightarrow{p}_0} \right) \quad (8.24a)$$

$$\bar{w}_{\text{VM},i} = \frac{1}{\overrightarrow{p}_i} \left(\frac{1}{\eta_{\text{VCM}}} + \overleftarrow{p}_{i-1} + \overleftarrow{p}_{i-1} \bar{w}_{\text{VM},i-1} \right) \quad (8.24b)$$

Again, using the $\bar{\mathbf{z}}$ notation we can also apply the above formula to formulate the partial eye sub-path weight in Equation 8.16 as a recursive quantity: $w_{\text{VM},s}^{\text{eye}} = \bar{w}_{\text{VM},t-1}(\bar{\mathbf{z}})$. Plugging $\bar{w}_{\text{VM},s-1}(\bar{\mathbf{y}})$ and $\bar{w}_{\text{VM},t-1}(\bar{\mathbf{z}})$ into Equation 8.17 completes the definition of $w_{\text{VM},s,t}$. Recall that in Equation 8.17 we have $t = k + 1 - s$ for paths sampled with VC and $t = k + 2 - s$ for paths sampled with VM.

8.3 Practical implementation

Equations 8.22 and 8.24 define the partial weight quantities $\bar{w}_{\text{VC},i}$ and $\bar{w}_{\text{VM},i}$ associated with the sub-path vertices. Ideally, we want to cache these quantities at the subpath vertices as we trace a random walk. Then, upon connecting or merging any two vertices, the full weight would be obtained by simply summing up their respective $\bar{w}_{\text{VC},i}$ or $\bar{w}_{\text{VM},i}$ quantities, as postulated by Equation 8.17. Unfortunately, this scheme cannot be directly implemented, since $\bar{w}_{\text{VC},i}$ and $\bar{w}_{\text{VM},i}$ require reverse probabilities that are not yet known at the point of sampling subpath vertex i . Specifically, $\overleftarrow{p}_i(\bar{\mathbf{y}})$ depends on the next two vertices via $p_{\sigma}(\mathbf{y}_i | \mathbf{y}_{i+1}, \mathbf{y}_{i+2})$. Similarly, $\overleftarrow{p}_{i-1}(\bar{\mathbf{y}})$ depends on the next subpath vertex, \mathbf{y}_{i+1} .

A practical implementation of the weight evaluation scheme from Section 8.2 is made possible by splitting up the computation of the recursive quantities $\bar{w}_{\text{VC},i}$ and $\bar{w}_{\text{VM},i}$. To this end, at vertex i we cache only (and all) the terms that depend on the subpath vertices sampled up to and including i . We extract three new quantities, d_i^{VC} , d_i^{VM} , and d_i^{VCM} , from $\bar{w}_{\text{VC},i}$ and $\bar{w}_{\text{VM},i}$, postponing the evaluation of the remaining terms until we have sampled the next subpath vertex $i + 1$ or connected/merged vertex i with another vertex:

$$\bar{w}_{\text{VC},i} = \overleftarrow{p}_i \left(\eta_{\text{VCM}} + \underbrace{\frac{1}{\overrightarrow{p}_i}}_{d_i^{\text{VCM}}} + \underbrace{\frac{1}{\overrightarrow{p}_i} \bar{w}_{\text{VC},i-1}}_{\overleftarrow{p}_{\omega,i-1} d_i^{\text{VC}}} \right) \quad (8.25)$$

$$\bar{w}_{\text{VM},i} = \underbrace{\frac{1}{\overrightarrow{p}_i}}_{d_i^{\text{VCM}}} \frac{1}{\eta_{\text{VCM}}} + \overleftarrow{p}_{\omega,i-1} \underbrace{\frac{\overleftarrow{g}_{i-1}}{\overrightarrow{p}_i} (1 + \bar{w}_{\text{VM},i-1})}_{d_i^{\text{VM}}}. \quad (8.26)$$

Here we have used the expansion $\overleftarrow{p}_{i-1} = \overleftarrow{p}_{\omega,i-1} \overleftarrow{g}_{i-1}$, with $\overleftarrow{p}_{\omega,i-1}$ being the reverse solid angle pdf for subpath vertex $i - 1$. Note that d_i^{VCM} appears in the weight for paths sampled with both VC and VM, whereas d_i^{VC} and d_i^{VM} are specific to the VC and VM weights respectively. Also, recall from Section 8.2 that the recursive formulas above apply to both light subpaths $\bar{\mathbf{y}}$ and eye subpaths $\bar{\mathbf{z}}$.

8.3.1 Subpath vertex data

As we trace a subpath, we update and store the quantities d_i^{VCM} , d_i^{VC} and d_i^{VM} at each vertex. The formulas for these quantities are the same for all subpath vertices, with the only exception for \mathbf{y}_1 and \mathbf{z}_1 . The reason is that we do not consider path sampling techniques with zero eye subpath vertices, as the probability of hitting the eye lens is usually very low (or even zero when a pinhole camera model is used). Note also that we do not store the vertices \mathbf{y}_0 and \mathbf{z}_0 , since choosing a new point on a light source or the eye lens for each connection reduces the sampling correlation and is usually cheap. The precise data we store at the subpath vertices is

$$\mathbf{y}_1 : d_1^{\text{VCM}} = \frac{p_0^{\text{connect}}}{p_0^{\text{trace}}} \frac{1}{\overrightarrow{p}_1} \quad \mathbf{z}_1 : d_1^{\text{VCM}} = \frac{p_0^{\text{connect}}}{p_0^{\text{trace}}} \frac{n_{\text{light}}}{\overrightarrow{p}_1} \quad (8.27)$$

$$d_1^{\text{VC}} = \frac{\overleftarrow{g}_0}{p_0^{\text{trace}} \overrightarrow{p}_1} \quad d_1^{\text{VC}} = 0 \quad (8.28)$$

$$d_1^{\text{VM}} = \frac{\overleftarrow{g}_0}{p_0^{\text{trace}} \overrightarrow{p}_1 \eta_{\text{VCM}}} \quad d_1^{\text{VM}} = 0 \quad (8.29)$$

$$\mathbf{y}_i, \mathbf{z}_i : d_i^{\text{VCM}} = \frac{1}{\overrightarrow{p}_i} \quad (8.30)$$

$$d_i^{\text{VC}} = \frac{\overleftarrow{g}_{i-1}}{\overrightarrow{p}_i} \left(\eta_{\text{VCM}} + d_{i-1}^{\text{VCM}} + \overleftarrow{p}_{\omega, i-2} d_{i-1}^{\text{VC}} \right) \quad (8.31)$$

$$d_i^{\text{VM}} = \frac{\overleftarrow{g}_{i-1}}{\overrightarrow{p}_i} \left(1 + \frac{d_{i-1}^{\text{VCM}}}{\eta_{\text{VCM}}} + \overleftarrow{p}_{\omega, i-2} d_{i-1}^{\text{VM}} \right). \quad (8.32)$$

In the equations above, the different vertex pdfs p_0^{connect} and p_0^{trace} account for the fact that different techniques may be used for sampling a vertex on a light source or the eye lens, depending on whether this vertex will be connected to a subpath or used to start a new subpath, respectively. The total number of light subpaths, n_{light} , is the number of samples the eye connection technique (VC, s , 1) takes. We have obtained the expressions for d_i^{VC} and d_i^{VM} by recursively expanding $\overleftarrow{w}_{\text{VC}, i-1}$ and $\overleftarrow{w}_{\text{VM}, i-1}$ in Equations 8.25 and 8.26. Note that the three floating point quantities d_i^{VCM} , d_i^{VC} , and d_i^{VM} are the only MIS weight related data that we need to store with the subpath vertices.

8.3.2 Full path weight

Recall that the weight for a full path constructed from a light subpath $\overline{\mathbf{y}}$ with s vertices and an eye subpath $\overline{\mathbf{z}}$ with t vertices is

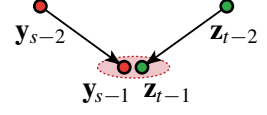
$$w_{v,s,t} = \frac{1}{\overleftarrow{w}_{v,s-1}(\overline{\mathbf{y}}) + 1 + \overleftarrow{w}_{v,t-1}(\overline{\mathbf{z}})}. \quad (8.33)$$

We now show how to compute $\overleftarrow{w}_{v,s-1}(\overline{\mathbf{y}})$ and $\overleftarrow{w}_{v,t-1}(\overline{\mathbf{z}})$ from the three vertex quantities defined above, depending on the sampling technique $v \in \{\text{VC}, \text{VM}\}$. The general-case formulas for VC and VM below (i.e. for $s > 1$ and $t > 1$) follow directly from Equations 8.25 and 8.26. Due to the symmetry in the notation for light and eye subpaths, these general-case formulas are the same for $\overline{\mathbf{y}}$ and $\overline{\mathbf{z}}$.

Vertex merging ($s > 1, t > 1$). A path is constructed by merging the light subpath vertex \mathbf{y}_{s-1} with the eye subpath vertex \mathbf{z}_{t-1} :

$$\bar{w}_{\text{VM},s-1}(\bar{\mathbf{y}}) = \frac{d_{s-1}^{\text{VCM}}}{\eta_{\text{VCM}}} + \overleftarrow{p}_{\omega,s-2} d_{s-1}^{\text{VM}} \quad (8.34)$$

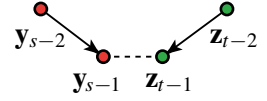
$$\bar{w}_{\text{VM},t-1}(\bar{\mathbf{z}}) = \frac{d_{t-1}^{\text{VCM}}}{\eta_{\text{VCM}}} + \overleftarrow{p}_{\omega,t-2} d_{t-1}^{\text{VM}}. \quad (8.35)$$



Vertex connection ($s > 1, t > 1$). A path is constructed by connecting the light subpath vertex \mathbf{y}_{s-1} to the eye subpath vertex \mathbf{z}_{t-1} :

$$\bar{w}_{\text{VC},s-1}(\bar{\mathbf{y}}) = \overleftarrow{p}_{s-1} (\eta_{\text{VCM}} + d_{s-1}^{\text{VCM}} + \overleftarrow{p}_{\omega,s-2} d_{s-1}^{\text{VC}}) \quad (8.36)$$

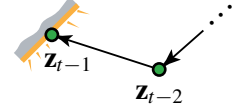
$$\bar{w}_{\text{VC},t-1}(\bar{\mathbf{z}}) = \overleftarrow{p}_{t-1} (\eta_{\text{VCM}} + d_{t-1}^{\text{VCM}} + \overleftarrow{p}_{\omega,t-2} d_{t-1}^{\text{VC}}). \quad (8.37)$$



Vertex connection ($s = 0$). The eye subpath vertex \mathbf{z}_{t-1} is sampled on a light source, i.e. the light subpath has zero vertices:

$$\bar{w}_{\text{VC},s-1}(\bar{\mathbf{y}}) = 0 \quad (8.38)$$

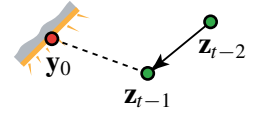
$$\bar{w}_{\text{VC},t-1}(\bar{\mathbf{z}}) = p_{t-1}^{\text{connect}} d_{t-1}^{\text{VCM}} + p_{t-1}^{\text{trace}} \overleftarrow{p}_{\omega,t-2} d_{t-1}^{\text{VC}}. \quad (8.39)$$



Vertex connection ($s = 1$). The eye subpath vertex \mathbf{z}_{t-1} is connected to vertex \mathbf{y}_0 on a light source (a.k.a. next event estimation):

$$\bar{w}_{\text{VC},0}(\bar{\mathbf{y}}) = \frac{\overleftarrow{p}_0(\bar{\mathbf{y}})}{p_0^{\text{connect}}(\bar{\mathbf{y}})} \quad (8.40)$$

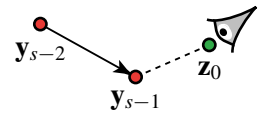
$$\bar{w}_{\text{VC},t-1}(\bar{\mathbf{z}}) = \frac{p_0^{\text{trace}}(\bar{\mathbf{y}})}{p_0^{\text{connect}}(\bar{\mathbf{y}})} \overleftarrow{p}_{t-1} (\eta_{\text{VCM}} + d_{t-1}^{\text{VCM}} + \overleftarrow{p}_{\omega,t-2} d_{t-1}^{\text{VC}}). \quad (8.41)$$



Vertex connection ($t = 1$). The light subpath vertex \mathbf{y}_{s-1} is connected to vertex \mathbf{z}_0 on the eye lens (a.k.a. eye/camera projection):

$$\bar{w}_{\text{VC},s-1}(\bar{\mathbf{y}}) = \frac{p_0^{\text{trace}}(\bar{\mathbf{z}})}{p_0^{\text{connect}}(\bar{\mathbf{z}})} \frac{\overleftarrow{p}_{s-1}}{n_{\text{light}}} (\eta_{\text{VCM}} + d_{s-1}^{\text{VCM}} + \overleftarrow{p}_{\omega,s-2} d_{s-1}^{\text{VC}}) \quad (8.42)$$

$$\bar{w}_{\text{VC},0}(\bar{\mathbf{z}}) = 0. \quad (8.43)$$



Recall that n_{light} is the total number of light subpaths, which is the number of samples this eye connection technique takes.

8.3.3 Reverse pdf evaluation

Having constructed a path by a VC technique, evaluating the pdf for sampling that path by a VM technique is straightforward, as VM can sample all light transport paths that VC can sample. This is illustrated in Figure 8.2 left, where the “merged” vertices coincide at \mathbf{z}_1 .

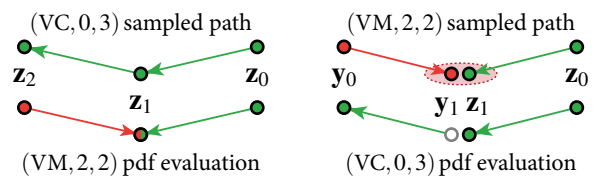


Figure 8.2: Reverse pdf evaluation.

The opposite case, shown in Figure 8.2 right, is slightly more complicated, as each different VM technique evaluates light transport along approximate

trajectories that in principle cannot be exactly sampled by any other VC or VM technique. Nevertheless, given a path sampled by a certain VM technique, we can evaluate reverse pdfs for the vertices around the merging location by respecting the edges of the sampled path. For the case considered in Figure 8.2 right, the unidirectional (VC, 0, 3) path pdf includes the reverse vertex pdf $\overleftarrow{p}_0(\bar{\mathbf{y}})$, which in turn includes the pdf for sampling the direction toward vertex \mathbf{y}_0 (Eq. 8.6). We evaluate this pdf as $p(\omega_{\mathbf{y}_1\mathbf{y}_0} | \mathbf{z}_1, \omega_{\mathbf{z}_0\mathbf{z}_1})$ using the actually sampled directions by the (VM, 2, 2) technique, just like BSDFs are evaluated in the VM measurement contribution function (Eq. 7.4). To evaluate the pdfs of VC techniques that make an actual subpath connection we simply disregard one of the “merged” vertices, i.e. we evaluate $p_{\text{VC},1,2} = \overrightarrow{p}_0(\bar{\mathbf{y}}) \overrightarrow{p}_0(\bar{\mathbf{z}}) \overrightarrow{p}_1(\bar{\mathbf{z}})$ and $p_{\text{VC},2,1} = \overrightarrow{p}_0(\bar{\mathbf{y}}) \overrightarrow{p}_1(\bar{\mathbf{y}}) \overrightarrow{p}_0(\bar{\mathbf{z}})$.

8.4 Special cases

In this section we discuss how to handle infinite light sources and orthographic cameras with multiple importance sampling as well as how to handle with BSDFs and light sources whose definitions involve delta distributions. We also show how to apply our recursive MIS weight evaluation scheme to bidirectional path tracing and bidirectional photon mapping, which are special cases of our vertex connection and merging method.

8.4.1 Infinite light sources

Light sources that are located very (or even infinitely) far away from the rest of the scene geometry are usually not defined via emitting surfaces but via *directional incident radiance* distributions at the scene surfaces. This definition prevents the straightforward use of infinite lights in methods like VCM defined in the path integral framework (Sec. 3.5.4) which is based on area/volume integration.

A common approach to handle infinite lights in bidirectional path tracing is to turn them into finite area lights. This is done by mapping their directional emission onto a large sphere that surrounds the scene and emits inwards diffusely (with the total power scaled appropriately). While such spherical area lights are naturally handled by the path integral, their emission distribution only approximates that of the original infinite light. Enlarging the sphere improves the approximation accuracy but can cause numerical issues due to the large distance between the light and the scene and due to the small solid angles involved in the emission importance sampling for light subpaths (Fig. 8.3a).

The above problems can be avoided by handling infinite lights in their original formulation, i.e. via solid angle integration. Paths starting on an infinite light source now have the form $\bar{\mathbf{x}} = \bar{\mathbf{x}}_0 \mathbf{x}_1 \dots \mathbf{x}_k$, where we have replaced the point \mathbf{x}_0 with a direction $\bar{\mathbf{x}}_0$. We start a light subpath $\bar{\mathbf{y}}$ by first sampling $\bar{\mathbf{y}}_0$ (deterministically for classical directional lights). We then sample a point \mathbf{y}_1^\perp on a plane perpendicular to $\bar{\mathbf{y}}_0$ and project it onto the scene along $\bar{\mathbf{y}}_0$ via

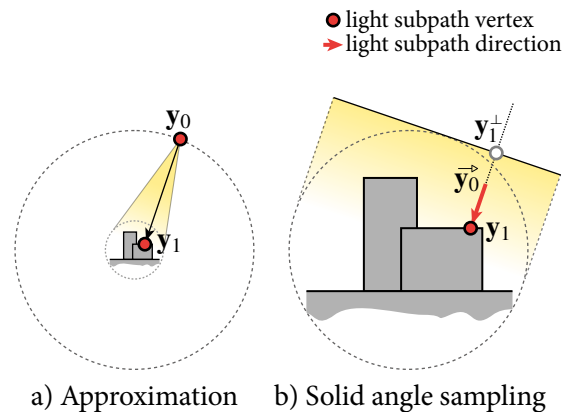


Figure 8.3: Instead of approximating infinite lights via emissive spheres (a), we directly handle them via solid angle integration and derive the corresponding path pdfs for use in MIS (b).

ray tracing to obtain \mathbf{y}_1 (see Figure 8.3b and also Pharr and Humphreys [109, p. 714]). We now write the forward and reverse pdfs corresponding to these sampling decisions. We preserve the pdf notation introduced in Section 8.1 but accommodate for the changes in the path geometry:

- $p_0^{\text{connect}}(\bar{\mathbf{y}}) = p_\sigma^{\text{connect}}(\bar{\mathbf{y}}_0^\rightarrow)$ and $p_0^{\text{trace}}(\bar{\mathbf{y}}) = p_\sigma^{\text{trace}}(\bar{\mathbf{y}}_0^\rightarrow)$ are expressed w.r.t. the solid angle measure,
- $\overleftarrow{p}_0(\bar{\mathbf{y}}) = \overleftarrow{p}_{\omega,0}(\bar{\mathbf{y}}) = p_\sigma(\bar{\mathbf{y}}_0^\rightarrow | \mathbf{y}_1, \omega_{\mathbf{y}_2, \mathbf{y}_1})$ is now a solid angle probability density as well,
- $\overleftarrow{g}_0(\bar{\mathbf{y}}) = 1$, as no pdf measure conversion is needed (a consequence of $\overleftarrow{p}_0(\bar{\mathbf{y}}) = \overleftarrow{p}_{\omega,0}(\bar{\mathbf{y}})$ above),
- $\overrightarrow{p}_1(\bar{\mathbf{y}}) = p(\mathbf{y}_1^\perp) \cos \theta_{1 \rightarrow 0}$, where $\theta_{1 \rightarrow 0}$ is now the angle between the normal at \mathbf{y}_1 and $-\bar{\mathbf{y}}_0^\rightarrow$.

The pdf modifications for $\bar{\mathbf{z}}_k^\rightarrow \equiv \bar{\mathbf{y}}_0^\rightarrow$ and $\mathbf{z}_{k-1} \equiv \mathbf{y}_1$ are symmetric.

8.4.2 Orthographic cameras

Orthographic cameras are the equivalent of infinite directional lights on the eye side. When such a camera model is used, the same modifications we made to the path geometry and the subpath pdfs for infinite lights apply on the eye side as well. That is, the last path vertex \mathbf{x}_k is replaced by a direction $\bar{\mathbf{x}}_k^\rightarrow$, and the pdf modifications above apply with $\bar{\mathbf{z}}_0^\rightarrow \equiv \bar{\mathbf{y}}_k^\rightarrow$.

8.4.3 Point and directional light sources

Omnidirectional and spot lights have infinitely small areas, and directional lights emit in a single direction. Such light sources cannot be hit by random rays, as their emission is defined via a delta distribution, i.e. we have $p_{\text{vc},0,k+1} = 0$. In order to account for this correctly in the MIS path weights, we need to modify the cumulative weight quantities d_1 associated with the light subpath vertex \mathbf{y}_1 . With this change, the three vertex quantities at \mathbf{y}_1 become identical to the ones associated with the eye subpath vertex \mathbf{z}_1 ¹ (Sec. 8.3.1), minus the n_{light} factor in d_1^{VCM} :

$$\mathbf{y}_1 : d_1^{\text{VCM}} = \frac{p_0^{\text{connect}}}{p_0^{\text{trace}}} \frac{1}{\overrightarrow{p}_1} \quad (8.44)$$

$$d_1^{\text{VC}} = 0 \quad (8.45)$$

$$d_1^{\text{VM}} = 0. \quad (8.46)$$


In addition, Equation 8.40 simplifies to $\bar{w}_{\text{vc},0}(\bar{\mathbf{y}}) = 0$. Even though sampling such lights is deterministic as emission is defined only at a single point, or in a single direction, we have kept the “connect” and “trace” pdfs above to account for the potentially different strategies for *choosing* the light source for next event estimation and light subpath tracing, respectively.

8.4.4 Specular materials

Similarly to point and directional lights, scattering at materials like mirror and glass is defined via delta BSDFs that cannot be handled directly by Lebesgue integration and thus neither by general-purpose

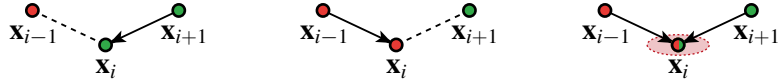
¹Recall that the eye subpath quantities were originally defined differently than the light subpath quantities in Section 8.3.1, because we do not allow light subpaths to randomly hit the eye lens.

numerical methods like Monte Carlo integration. These distributions define a single direction with a non-zero scattering contribution, which is deterministically followed in a random walk. Therefore, the forward and reverse scattering probabilities at a path vertex \mathbf{x}_i with a delta BSDF are simply one:

$$p(\omega_{\mathbf{x}_i \mathbf{x}_{i+1}} | \mathbf{x}_i, \omega_{\mathbf{x}_{i-1} \mathbf{x}_i}) = 1 \quad p(\omega_{\mathbf{x}_i \mathbf{x}_{i-1}} | \mathbf{x}_i, \omega_{\mathbf{x}_{i+1} \mathbf{x}_i}) = 1. \quad (8.47)$$


When the BSDF is a mixture of distributions, at least one of which is a delta distribution (e.g. specular reflection or refraction), the above pdfs become equal to the probability of choosing the particular delta component at \mathbf{x}_i . Note that when randomly hitting the eye lens with a light subpath is disallowed, purely specular (LS^+E) paths can only be sampled unidirectionally from the eye and thus have the trivial MIS weight of one.

Vertices with delta BSDFs also cannot be connected to or merged with other vertices, in the sense that the measurement contribution of the resulting full paths (Eq. 3.31) is always zero. As a result, certain path sampling techniques become invalid and their zero probabilities need to be accounted in the MIS weights. Specifically, a delta BSDF at vertex \mathbf{x}_i makes the following path sampling techniques invalid:

$$p_{VC,i,k-i+1} = 0 \quad p_{VC,i+1,k-i} = 0 \quad p_{VM,i+1,k-i+1} = 0 \quad (8.48)$$


We can account for these zero-probability techniques in the MIS weight as we trace the subpaths. If the scattering event at vertex $i-1$ is specular, then the weight quantities for vertex i simplify to

$$\mathbf{y}_i, \mathbf{z}_i : d_i^{\text{VCM}} = 0 \quad (8.49)$$

$$d_i^{\text{VC}} = \frac{\overleftarrow{g}_{i-1}}{\overrightarrow{p}_i} \overleftarrow{p}_{\omega,i-2} d_{i-1}^{\text{VC}} \quad (8.50)$$

$$d_i^{\text{VM}} = \frac{\overleftarrow{g}_{i-1}}{\overrightarrow{p}_i} \overleftarrow{p}_{\omega,i-2} d_{i-1}^{\text{VM}}. \quad (8.51)$$

Note that with these modifications our scheme correctly handles the case where the BSDF at subpath vertex $i-1$ is a mixture of delta and non-delta components. The type of scattering event chosen at that vertex only influences the quantities d_i for the next vertex, and not the quantities d_{i-1} . Vertex $i-1$ is still stored and used for connection and merging to construct other paths whose measurement contribution is evaluated with the non-delta BSDF components at that vertex. For those paths the MIS weight formulas from Section 8.3.2 apply as usual.

8.4.5 Bidirectional path tracing

The path weight evaluation scheme from Section 8.3 can be easily applied to traditional bidirectional path tracing by restricting the formulas to only account for vertex connection techniques. This is achieved by setting $\eta_{\text{VCM}} = 0$ in Equation 8.31 and Equations 8.36–8.43 and also eliminating the vertex quantity d_i^{VM} . The resulting weight accumulation scheme then becomes nearly identical to the one proposed by van Antwerpen [141].

8.4.6 Bidirectional photon mapping

Bidirectional photon mapping [146] is another special case of VCM that only uses vertex merging techniques. Analogously to the bidirectional path tracing case above, restricting the weighting to vertex merging techniques is as simple as setting the terms involving η_{VCM} in Equations 8.29, 8.32, 8.34 and 8.35 to zero and also eliminating the vertex quantity d_i^{VC} .

8.5 Extensions

We now present three useful extensions to the basic VCM algorithm we described in Section 7.3.2. We first show how to accommodate the use of a different merging radius for each full path in our MIS weight evaluation scheme. We then describe how to interleave the evaluation of the vertex connection and vertex merging estimators to significantly reduce the memory footprint of the algorithm. Finally, we discuss possible ways to incorporate motion blur and spectral rendering in VCM.

8.5.1 Per-path merging radii

The merging radius r is an important VCM parameter that controls the performance of the vertex merging techniques and also their relative weights in the combined MIS estimator (Eq. 7.14). The optimal radius size generally varies across the scene, however our MIS weight accumulation scheme from Section 8.3 implicitly assumes that all vertex merging estimators use the same global radius r . This is due to the dependence of the cumulative subpath vertex quantities d_i^{VC} and d_i^{VM} on r via η_{VCM} (Eq. 8.18). If we use different merging radii at different locations in the scene (e.g. derived from the pixel footprint) and want to obtain meaningful MIS weights, we need to make these vertex quantities independent of r . We achieve this by replacing the vertex quantities d_i^{VC} and d_i^{VM} by two new quantities c_i^{VC} and c_i^{VM} :

$$\mathbf{y}_1 : d_1^{\text{VCM}} = \frac{p_0^{\text{connect}}}{p_0^{\text{trace}}} \frac{1}{\vec{p}_1} \quad \mathbf{z}_1 : d_1^{\text{VCM}} = \frac{p_0^{\text{connect}}}{p_0^{\text{trace}}} \frac{n_{\text{light}}}{\vec{p}_1} \quad (8.52)$$

$$c_1^{\text{VC}} = \frac{\overleftarrow{g}_0}{p_0^{\text{trace}} \vec{p}_1} \quad c_1^{\text{VC}} = 0 \quad (8.53)$$

$$c_1^{\text{VM}} = 0 \quad c_1^{\text{VM}} = 0 \quad (8.54)$$

$$\mathbf{y}_i, \mathbf{z}_i : d_i^{\text{VCM}} = \frac{1}{\vec{p}_i} \quad (8.55)$$

$$c_i^{\text{VC}} = \frac{\overleftarrow{g}_{i-1}}{\vec{p}_i} \left(d_{i-1}^{\text{VCM}} + \overleftarrow{p}_{\omega, i-2} c_{i-1}^{\text{VC}} \right) \quad (8.56)$$

$$c_i^{\text{VM}} = \frac{\overleftarrow{g}_{i-1}}{\vec{p}_i} \left(1 + \overleftarrow{p}_{\omega, i-2} c_{i-1}^{\text{VM}} \right) \quad (8.57)$$

Having c_i^{VC} and c_i^{VM} , the quantities d_i^{VC} and d_i^{VM} for subpath vertex i can be computed on-the-fly:

$$d_i^{\text{VC}} = c_i^{\text{VC}} + \eta_{\text{VCM}} c_i^{\text{VM}} \quad (8.58)$$

$$d_i^{\text{VM}} = \frac{c_i^{\text{VC}}}{\eta_{\text{VCM}}} + c_i^{\text{VM}} \quad (8.59)$$

With this modification, the evaluation of η_{VCM} , and thus of d_i^{VC} and d_i^{VM} , can be postponed to the point when a full path is constructed using a particular eye subpath vertex. This allows us to choose a different radius r for each path. However, we still need to use the same r in the weight for every possible technique that can sample a given full path, so that all weights for that path sum up to one.

In an actual implementation, the merging radius can be computed from the pixel footprint [53] at the first eye subpath vertex with a non-delta BSDF and then reused for all subsequent vertices. When connecting a light subpath vertex to an eye subpath vertex (where the latter can also be on the eye lens), the radius is computed by propagating the pixel footprint from the eye along the resulting full path.

8.5.2 Memory-efficient implementation

The VCM algorithm we presented in Section 7.3.2 is split into two stages, where the first stage traces all light subpaths and stores their vertices, similarly to photon mapping. Since these vertices are also used for connections in the second stage, we need to store their associated BSDF structures as well. However, in production renderers the shading structure at a surface point can be as large as 1KB², which results in a gigabyte of storage per million vertices. This memory issue does not appear in bidirectional path tracing, where every pixel is rendered independently and its corresponding light and eye subpath pair is immediately discarded upon connecting their vertices. Photon mapping can also maintain a low memory footprint, as it always evaluates BSDFs at the *eye* subpath vertices, thereby avoiding the need for storing shading data at the light subpath vertices.

To reduce the memory footprint of the light subpath vertices, we can modify our progressive VCM algorithm so that each rendering iteration operates in a single stage as follows. At every iteration, for each pixel we first trace one light subpath. We then copy its vertices into a separate global list, where for each vertex we only keep the position, incident direction, sampling throughput, and the three cumulative MIS weight quantities. The memory footprint of a vertex is then as large as in photon mapping but with the addition of the three weight quantities. After that, we trace an eye subpath through the pixel. Every eye vertex is connected to the light subpath and also merged with all nearby light vertices that have been stored in the *previous* iteration. Finally, we discard both subpaths. After rendering all pixels, we build a range search acceleration structure over the light vertices in the separate global list, after disposing the structure from the previous iteration. Rendering begins by tracing an initial set of light subpaths that are only used for merging at the first iteration. Alternatively, we can skip merging at the first iteration and scale the VM contributions at the second iteration by $2\times$.

The above modification makes VCM very similar to bidirectional path tracing, with the addition of caching all light vertices at every iteration and reusing them for merging at the following iteration. Interleaving the connection and merging estimator evaluations allows us to reduce the memory footprint to the level of photon mapping, with only three additional floating-point numbers per light vertex.

²Shading structures often store the results from multiple texture queries, local geometric and shading coordinate frames and derivatives, as well as BSDF parameters.

8.5.3 Motion blur and spectral rendering

Our discussion so far has assumed rendering a still image at a given time instant (i.e. infinitely small time window). In reality, however, physical cameras take pictures by opening their shutter for a finite period of time. As a result, any motion in the scene is integrated over this time interval. Simulating this so-called temporal anti-aliasing, or *motion blur*, effect is crucial in animation rendering in order to reproduce visually smooth motion with a finite sequence of images. We have also disregarded the simulation of wavelength-dependent effects, such as chromatic dispersion, which are produced by materials whose scattering properties, e.g. index of refraction, vary with wavelength. These effects cannot be accurately captured with the commonly used red-green-blue spectral representation we described in Section 3.3, which assumes that the directional component of BSDFs is wavelength-independent.

In order to render motion blur and dispersion, one needs to consider a more general form of the path measurement contribution (Eq. 3.31) as a function of time and wavelength, respectively, and perform explicit integration over the shutter interval and the visible light spectrum in the light transport equation (Eq. 3.27). In a Monte Carlo pixel estimator this translates to assigning random time t and wavelength λ values to every sampled path $\bar{\mathbf{x}}$:

$$\hat{I} = \frac{f(\bar{\mathbf{x}}, t, \lambda)}{p(\bar{\mathbf{x}}, t, \lambda)}. \quad (8.60)$$

Below we describe two possible approaches to rendering motion blur and dispersion with VCM.

Per-iteration time and wavelength sampling

A relatively straightforward approach, in terms of implementation effort, to extend our progressive VCM is to draw one random time value and one random wavelength for each rendering iteration and use them for all paths sampled in that iteration. This requires no other modifications to the rendering code, as the Monte Carlo integration over shutter time and light spectrum is entirely delegated to the higher-level secondary pixel estimator (Eq. 7.16). The disadvantage of this approach is that it cannot be easily combined with the memory-efficient VCM implementation described above. This is because the light subpaths sampled at one iteration and the eye subpaths sampled at the subsequent iteration will have mismatching time values and wavelengths, which will prevent the merging of their vertices into full paths.

Vertex merging in time and wavelength

An alternative approach is to sample a different random time value and wavelength for each sub-path and explicitly consider “merging” in the time and spectral domains, in addition to the spatial domain [6]. That is, for vertex merging (VM) we can consider all light vertices within a small neighborhood around the eye query vertex not only in space but also in time and wavelength. The VM path acceptance probability (Eq. 7.10) then needs to include the joint probability for sampling a ‘photon’ inside this higher-dimensional neighborhood:

$$P_{\text{acc}}(\bar{\mathbf{x}}) = \underbrace{\int_{\mathcal{M}_t} p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}) d\mathbf{x}}_{\text{location probability}} \cdot \underbrace{\int_{-r_t}^{r_t} p(t^{\text{eye}} + t) dt}_{\text{time probability}} \cdot \underbrace{\int_{-r_\lambda}^{r_\lambda} p(\lambda^{\text{eye}} + \lambda) d\lambda}_{\text{wavelength probability}} \quad (8.61a)$$

$$\approx \pi r^2 p(\mathbf{x}_{s-1} \rightarrow \mathbf{x}_s^*) \cdot 2r_t p(t^{\text{eye}}) \cdot 2r_\lambda p(\lambda^{\text{eye}}), \quad (8.61b)$$

which is the product of the acceptance probabilities respectively in space, time, and wavelength. Here, r is the spatial merging radius, r_t and r_λ are the time and wavelength merging radii, $p(t)$ and $p(\lambda)$ are the time and wavelength sampling pdfs, and t^{eye} and λ^{eye} are the sampled time and wavelength values of the eye subpath. In Equation 8.61b we have assumed that the time and wavelength sampling pdfs are locally constant around t^{eye} and λ^{eye} respectively, as we did in Equation 7.10b. However, this approximation is rarely necessary as in practice these pdfs have simple forms and so the exact values of the corresponding integrals in Equation 8.61a can be usually computed analytically. We can plug Equation 8.61 into Equation 7.11 to obtain the VM path pdf that we can use to weight the technique in the combined VCM estimator (Eq. 7.14).

Each of the three acceptance probabilities in Equation 8.61 is less than or equal to 1, and so is their product. The final acceptance probability is also at most equal to the one in Equation 7.10. In fact, depending on the values of r_t and r_λ , it can be one or more orders of magnitude smaller, which reflects the fact that narrowing the search range in time and wavelength reduces the efficiency of vertex merging. Aside from the additional blurring bias introduced, this reduced efficiency is the major disadvantage of this approach. Furthermore, as pointed out by Kaplanyan and Dachsbacher [70], increasing the blur dimensionality reduces the error convergence rate of vertex merging. In progressive VCM, the former issue can be ameliorated by keeping around and merging with the light vertices from the last few rendering iterations. This reduces the variance by increasing the number of samples, albeit at the cost of higher memory usage. It can also be shown that, thanks to the MIS combination with unbiased vertex connection techniques, increasing the blur dimensionality of vertex merging does not alter the error convergence rate of the progressive combined VCM estimator.

8.6 Discussion

The main focus of this chapter is the recursive MIS path weight evaluation scheme which avoids redundant computations and improves the MIS efficiency in BPT and VCM. We expect the benefit from this scheme to be particularly noticeable on platforms where random memory access is costly, e.g. graphics processing units (GPUs). Our scheme allows for evaluating the weight of a full path by only using data stored at the two vertices that are connected or merged. Since no other path vertices need to be accessed, we can avoid storing the eye subpath vertices altogether. This effectively makes the second stage of the VCM algorithm an extension to traditional path tracing with next event estimation with the addition of light subpath connection and merging at every eye subpath vertex. Below we discuss other relevant details in the practical implementation of the algorithm.

Parameter choice

We concluded Chapter 7 with a high-level discussion on the parameter choice for the progressive VCM algorithm. In this chapter we showed how to accommodate per-path merging radii in the MIS weights, which allows us to give a more concrete suggestion for this choice. We recommend computing the merging radius at iteration i as

$$r_i = \sqrt{s \cdot A_{\text{pixel}}} \sqrt{i^{\alpha-1}}, \quad (8.62)$$

where A_{pixel} is the area of the pixel footprint at the first merging vertex from the eye along the path. This footprint is optionally scaled by the user parameter $s > 0$ whose default value is 1. Using this radius formula and setting the number of VM light subpaths n_{VM} to the number of pixels makes the variance

of the resulting image independent of its resolution. And while the variance of progressive photon mapping crucially depends on the size of the radius, in VCM we can typically afford using a small, pixel-wide merging radius, thanks to the MIS combination with vertex connection techniques. As in Section 7.6, we recommend setting the radius reduction parameter to $\alpha = 0.75$; however, it is often safe to disable the reduction altogether (by setting $\alpha = 1$) when the size of the merging disk becomes sufficiently smaller than the pixel footprint. Doing this makes the variance converge slightly faster at the cost of a small amount of bias.

Note that Equation 8.62 effectively turns the merging radius r_i into a random variable that is a function of (part of) the path. As a result, it is not anymore guaranteed that the radius for every pixel will decrease strictly monotonically over time. However, the progressive reduction by the factor $\sqrt{i^{\alpha-1}}$ ensures that the asymptotic error behavior of the progressive VCM remains unchanged.

Limitations

One constraint that our recursive path weight evaluation scheme poses is that no terms in the weight can depend on (sub)path lengths. For example, we cannot make the Russian roulette path termination probability or the number of vertex connections at an eye vertex dependent on the eye subpath length. The reason is that these quantities need to be evaluated when accumulating the partial MIS weights during *light* subpath tracing, without any knowledge of the full path length. This limitation has been also pointed out by van Antwerpen [141]. One way to circumvent it is to consider such terms independent of the path lengths when evaluating the MIS weights or to completely exclude them from the weights. Alternatively, the per-vertex storage can be augmented with additional partial weights that are specialized for connections to eye subpaths of different lengths.

Reference implementation

We provide an open-source implementation of the VCM algorithm that uses our efficient path weight evaluation scheme in the SmallVCM renderer [21]. At the time of this writing, the implementation covers all special cases discussed in Section 8.4, but not the extensions from Section 8.5. SmallVCM also implements traditional path tracing, light tracing, bidirectional path tracing, progressive photon mapping, as well as progressive bidirectional photon mapping, which are all special cases of the combined VCM algorithm.

Joint Path Sampling in Participating Media

9

In the previous four chapters we focused on surface light transport and mostly disregarded the potential presence of scattering participating media in the scenes we were interested in rendering. However, media such as clouds, smoke, skin, fog, and liquids are ubiquitous in the world and are responsible for many important visual phenomena. Faithfully simulating light transport in such media is a challenging problem whose applications span many diverse fields beyond realistic image synthesis, including medical imaging and nuclear physics.

Over the last three decades, the computer graphics community has adapted and developed a sizable and diverse arsenal of methods for rendering participating media [7]. Although no single method outperforms all others in every situation, we now have efficient approximations for rendering specific scattering scenarios: (semi-)analytic techniques exist for simulating single scattering [133, 104, 101, 105, 106, 152]; methods based on the diffusion approximation [130] often work well for high-albedo homogeneous media [65, 25, 23, 37] (albeit with strict constraints on the media properties); and, the Feynman path integral approximation [138, 111, 112] holds for highly forward-scattering media with high albedo. Though these methods are each well-suited for specific cases, a common limitation of all is their inefficiency or inaccuracy in handling surface occlusion, heterogeneous media, as well as media with highly anisotropic scattering and moderate albedo. In such cases, one has to resort to general-purpose Monte Carlo integration techniques like the ones we described in Chapter 4. Unfortunately, these techniques can suffer from high variance, especially with anisotropic scattering and even for media with low albedo where it is often not necessary to simulate high-order scattering. In this chapter, we address the efficiency of Monte Carlo techniques for path sampling in such media.

The task of every Monte Carlo rendering method is to construct light transport paths that connect the light sources to the eye. As we discussed in Section 4.2, path vertices should ideally be sampled with a joint pdf proportional to the path's measurement contribution. Unfortunately, the complex shape and high dimensionality of the contribution function makes such joint sampling generally infeasible. Existing surface rendering methods sample paths incrementally via random walks, only locally accounting for certain contribution terms at each vertex. These methods have traditionally been extended to participating media by simply incorporating the propagation distance as an additional independent sampling decision in the random walk. The resulting path probability density is thus a product of the *conditional* densities of each local sampling step, constructed without explicit control over the form of the final *joint* vertex distribution of the complete path, which can lead to significant variance in the final estimator. In fact, though not widely known in computer graphics, it has long been acknowledged in related fields that vertex connection based methods like path tracing with next event estimation and many-light rendering can suffer from *infinite* variance in the presence of scattering media [68].

In this chapter, we analyze why current volumetric path construction schemes often lead to high variance and reveal that such approaches are an unnecessary legacy inherited from traditional surface-

based rendering algorithms. We then propose *joint importance sampling* of path vertices in participating media to account for the product of phase functions and geometry terms along entire sequences of vertices instead of just locally at a single vertex. Although importance sampling the BSDF, the lighting, or their product has been investigated for direct surface illumination (see Section 4.7.1), generalizations to multiple bounces (a sequence of vertices) are challenging due to the curse of dimensionality. We exploit the extra dimension in participating media as a degree of freedom, along with symmetries uniquely present in medium light transport, to make sampling from joint importance distributions practical. For the case of isotropic scattering, we derive analytic formulas for exact importance sampling of the product of geometry terms for a sequence of vertices (three path edges) joining two given subpaths. For anisotropic scattering, we build compact tabulations for importance sampling the product of the geometry and phase function terms. We demonstrate the benefit of the resulting path sampling techniques by integrating them into several rendering algorithms: path tracing, bidirectional path tracing, and many-light methods. We use these techniques to generalize deterministic vertex connections to *connection subpaths*, involving of two or three random sampling decisions, to efficiently simulate higher-order multiple scattering. Our algorithms significantly reduce noise and increase performance in renderings with both isotropic and highly anisotropic, low-order scattering.

The remainder of this chapter is organized as follows. In the next section, we review existing techniques for path sampling in participating media and interpret them as constructing connections between subpath vertices and segments involving 0, 1, and 2 random decisions. We then consider the problem of connecting a given vertex and a given semi-infinite segment via a length-3 connection subpath involving 3 random decisions in Section 9.2. In Section 9.3, we derive analytic formulas for exact importance sampling of the product of geometry terms along such connection subpaths. In Section 9.4 we build compact tabulations of these pdfs to extend them to importance sample the product of geometry and anisotropic phase function terms. We showcase the resulting sampling routines in Section 9.5, describing their integration into several path-based rendering algorithms. Section 9.6 concludes the chapter with a discussion on the limitations and possible extensions of our method.

9.1 Path sampling in media

Central to every Monte Carlo-based rendering algorithm is the task of sampling paths and evaluating estimators for the light transport integral (Eq. 3.27). The integrand, i.e. the path measurement contribution function (Eq. 3.31) can be factorized into the product of four terms:

$$f(\bar{\mathbf{x}}) = \rho(\bar{\mathbf{x}})T_t(\bar{\mathbf{x}})V(\bar{\mathbf{x}})G(\bar{\mathbf{x}}), \quad (9.1)$$

each being the product of all the scattering (BSDFs or phase functions), transmittance, visibility or geometry terms along the path, respectively (see Section 3.5.4). The variance of a path integral estimator depends on the path's sampling probability density $p(\bar{\mathbf{x}}) = p(\mathbf{x}_0, \dots, \mathbf{x}_k)$, which is given by the joint density of the vertices and is fully determined by the sampling technique used to generate the path.

Ideally, the path sampling technique should be chosen such that its corresponding joint pdf is exactly proportional to the measurement contribution function. Since this is generally not feasible in practice, (sub)paths are typically constructed incrementally via random walks, vertex by vertex, importance sampling only certain terms of $f(\bar{\mathbf{x}})$, optionally followed by a connection (Sec. 4.2). The resulting joint path pdf is the product of the conditional pdfs $p(\mathbf{x}_i | \text{vertices sampled before } \mathbf{x}_i)$ for every path vertex \mathbf{x}_i . The variance of the resulting estimator crucially depends on the variability of those path contribution terms that are *not* importance sampled.

9.1.1 Existing techniques

We now analyze the different joint path pdfs induced by existing unbiased participating media rendering methods. We introduce the notion of *connection subpath* and show that equi-angular sampling and virtual ray lights (Sec. 4.5) can be interpreted as general path sampling techniques and employed for connection subpath construction in bidirectional algorithms using both *segments* (lines) and *vertices*. We will motivate our new sampling techniques by identifying the path contribution terms that are *not* importance sampled by each existing technique.

Since we focus on media, and in the interest of keeping the exposition simple, hereinafter we assume that all path vertices correspond to volume scattering events, i.e. that $G(\mathbf{x}, \mathbf{y}) = 1/\|\mathbf{x} - \mathbf{y}\|^2$. We express directional pdfs $p(\omega)$ w.r.t. the solid angle measure, distance pdfs $p(t)$ w.r.t. the Euclidean length on \mathbb{R} , and medium vertex pdfs $p(\mathbf{x})$ w.r.t. the Euclidean volume on \mathbb{R}^3 . Converting from the solid angle \times length product measure to the volume measure requires multiplication by a corresponding geometry term G . We use an example length-3 path **abcd** depicted in Figure 9.1 throughout our discussion.

Unidirectional sampling

The traditional random walk path sampling approach is to first sample a vertex **d** on the eye sensor from a pdf $p(\mathbf{d})$ proportional to the emitted importance and then to incrementally extend the path with new vertices until a light source is hit (Fig. 9.1a). Given vertex **c** in the medium with incident direction ω_{dc} , the next path vertex **b** is sampled as follows: First, a direction ω_{cb} is sampled proportionally to the phase function at **c**, i.e. $p(\omega_{cb} | \omega_{dc}, \mathbf{c}) \propto \rho(\mathbf{c})$. The propagation distance t_{cb} along ω_{cb} is then sampled proportionally to transmittance¹ with pdf $p(t_{cb} | \mathbf{c}, \omega_{cb}) \propto T_r(\mathbf{b}, \mathbf{c})$. The direction ω_{cb} and distance t_{cb} determine the next vertex **b**, with conditional pdf:

$$p(\mathbf{b} | \omega_{dc}, \mathbf{c}) = p(\omega_{cb} | \omega_{dc}, \mathbf{c}) p(t_{cb} | \mathbf{c}, \omega_{cb}) G(\mathbf{b}, \mathbf{c}), \quad (9.2)$$

where $G(\mathbf{b}, \mathbf{c})$ is needed to convert the pdf of **b** to the volume measure.

The path pdf of this unidirectional sampling technique is the product of $p(\mathbf{d})$ and the pdf in Equation 9.2 for every other vertex. This joint pdf is proportional to the product of all terms in the path contribution (Eq. 9.1) except the emitted radiance $\rho(\mathbf{a}) = L_e(\mathbf{a} \rightarrow \omega_{ab})$. Unfortunately, in practice this approach often yields estimators with very high variance due to the negligible probability of hitting small light sources. Alternatively, paths can be constructed unidirectionally starting from light sources, but the resulting light tracing algorithm usually has even higher variance because of the small size of the eye sensor. In practice, both path and light tracing use next event estimation with explicit connections to lights or sensors. These are special cases of the bidirectional path sampling discussed next.

Vertex-vertex connection

To address the high variance of unidirectional sampling, both emitted light source radiance *and* sensor importance must be included in the path pdf. This can be achieved with bidirectional sampling, illustrated in Figure 9.1b. First, one subpath with s vertices is sampled with a random walk from a light source and another one with t vertices from the eye. The endpoints of these independent subpaths

¹Surface scattering occurs if t_{cb} is beyond the nearest surface intersection along the ray $(\mathbf{c}, \omega_{cb})$. We disregard this technicality here for the sake of simplicity.

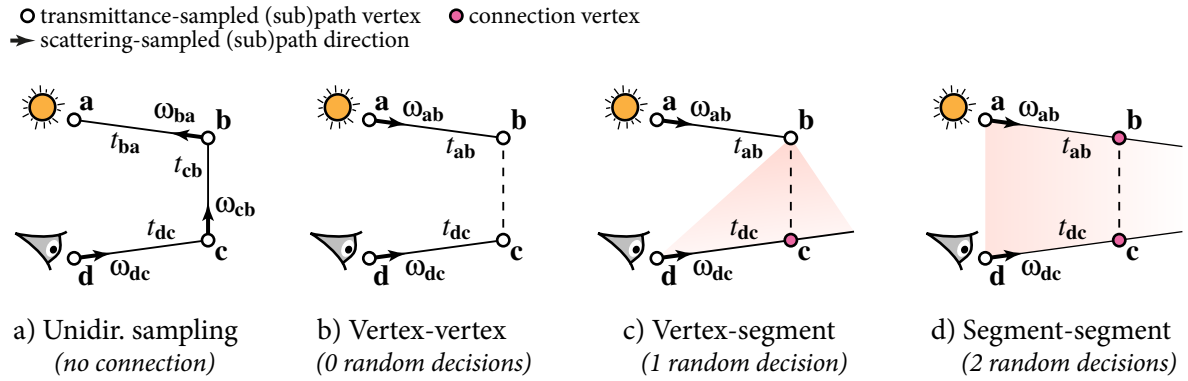


Figure 9.1: A comparison of various existing techniques for sampling light transport paths in a medium. The pink connection vertices are sampled from line distributions that are conditioned on the endpoints and/or end-lines of the light and eye subpaths.

are then deterministically connected with an edge, completing a full path. We already described this *vertex-vertex* connection technique in detail in Section 4.2.3. Next event estimation for path and light tracing corresponds to $s = 1$ and $t = 1$, respectively.

Vertex-vertex connection can also lead to high variance, as it does not importance sample any of the contribution terms that depend on the connection edge. This deterministically constructed path edge can be viewed as a *0-random-decision* subpath connecting the light and eye subpaths (see Figure 9.1b). In fact, the variance is infinite due to the geometry term $G(\mathbf{b}, \mathbf{c})$ diverging as \mathbf{b} and \mathbf{c} approach each other [68]. To resolve this, bidirectional path tracing exploits the fact that any full path can be generated using a number of techniques, each corresponding a unique combination of different lengths for the light and eye subpaths. The multiple importance sampling (MIS) combination of the corresponding estimators yields finite and often relatively low variance, however the resulting combined joint path pdf remains a simple linear combination of the pdfs of the individual techniques, as opposed to the product of all path integrand terms.

Vertex-segment connection

The equi-angular sampling technique, which we introduced in Section 4.5.1, importance samples the geometry term of the connection edge between two subpaths. Kulla and Fajardo [80] used this technique for rendering single scattering, but it can be generalized to a *vertex-segment* connection strategy in bidirectional sampling with a pdf conditioned on both the light and eye subpaths (Fig. 9.1c).

Given the end vertices \mathbf{b} and \mathbf{d} on the light and eye subpaths, along with a direction $\omega_{\mathbf{dc}}$ from \mathbf{d} , the equi-angular technique constructs a *1-random-decision connection subpath* by sampling the propagation distance $t_{\mathbf{dc}}$ along $\omega_{\mathbf{dc}}$ to create vertex \mathbf{c} with a pdf proportional to the inverse squared length of the connection edge \mathbf{bc} : $p(t_{\mathbf{dc}} | \mathbf{b}, \mathbf{d}, \omega_{\mathbf{dc}}) \propto G(\mathbf{b}, \mathbf{c})$. This technique often results in lower variance than transmittance sampling. However, it does not importance sample any other terms in the connection, namely: $T_r(\mathbf{b}, \mathbf{c})$, $V(\mathbf{b}, \mathbf{c})$, $T_r(\mathbf{d}, \mathbf{c})$, $V(\mathbf{d}, \mathbf{c})$, $\rho(\mathbf{b})$, and $\rho(\mathbf{c})$.

Kulla and Fajardo [80] originally proposed equi-angular sampling to “cancel out” the weak singularity in $G(\mathbf{b}, \mathbf{c})$. However, as we will show in Section 9.3.2 below, the true singularity in fact remains “hidden” in the orientation of the ray $(\mathbf{d}, \omega_{\mathbf{dc}})$ w.r.t. the vertex \mathbf{b} . We will eliminate this singularity by importance sampling the ray direction $\omega_{\mathbf{dc}}$.

Segment-segment connection

Recent extensions to photon density estimation have achieved significant improvements by using lines, instead of vertices, as the eye [58] and light [134, 59] path building blocks. Inspired by this work, Novák et al. [96] analyzed many-light rendering in anisotropically scattering media and showed that such line-based path construction can provably reduce estimator singularities. Their virtual ray light (VRL) method, which we summarized in Section 4.5.2, uses a one-sample Monte Carlo estimator to calculate the total energy transfer between a “virtual ray light” and an eye ray. We reinterpret this method as a general bidirectional *segment-segment* connection technique, which we illustrate in Figure 9.1d. Given the end vertex \mathbf{a} of a light subpath, along with direction $\omega_{\mathbf{ab}}$ (i.e. the virtual ray light), and an end vertex \mathbf{d} on an eye subpath, along with direction $\omega_{\mathbf{dc}}$, this *2-random-decision* technique samples the propagation distances $t_{\mathbf{ab}}$ and $t_{\mathbf{dc}}$ from a joint conditional distribution $p(t_{\mathbf{ab}}, t_{\mathbf{dc}} | \mathbf{a}, \omega_{\mathbf{ab}}, \mathbf{d}, \omega_{\mathbf{dc}}) \propto G(\mathbf{b}, \mathbf{c})\rho(\mathbf{b})\rho(\mathbf{c})$. The approximate proportionality stems from various simplifications in the VRL approach, which we will improve upon in Section 9.4.3.

In Figure 9.1 we observe that the only difference between vertex-vertex connection, equi-angular sampling, and VRLs is the pdf for sampling the distances $t_{\mathbf{ab}}$ and $t_{\mathbf{dc}}$. This pdf is either proportional to transmittance or to the scattering and/or geometry term(s) involved in the connection. Each of these techniques can be viewed as creating connections between light and eye subpaths with a varying number of random decisions (0, 1, or 2). In the following sections we go a step further and develop a practical method for creating connection subpaths involving *3 random decisions*, where the extra decision is direction sampling. We then show how to extend unidirectional and bidirectional path tracing with such connections, which will lead to significant variance reduction.

9.1.2 Relationship to neutron transport

Many of the sophisticated participating media rendering techniques used in computer graphics have been adapted (and sometimes unknowingly reinvented) from the neutron transport and radiative transfer literatures, which have a much longer history of investigating these problems [8, 129]. Those fields are often concerned with using Monte Carlo simulations to estimate flux, or dose, at a point or some volume of interest. Density estimation in volumetric photon mapping uses a so-called *collision estimator*, and the path segment improvements [59, 96] share similarity with, and extend upon, *track length estimators* [128]. Volumetric path tracing with next event estimation corresponds to the *uncollided flux* (UCF) estimator used in neutron transport simulations since the early 1960s.

Kalos [68] proved that the UCF estimator has infinite variance due to the $1/d^2$ singularity in the geometry term, and even worse, its use leads to an abysmal $1/\sqrt[3]{N}$ error convergence rate [68, 27]. To remedy this problem, Kalos proposed the *once collided flux* (OCF) estimator, which samples an extra path vertex when forming the connection, and proved that this technique reduces the order of the singularity to $1/d$ and recovers the more favorable $1/\sqrt{N}$ error convergence rate. Since this seminal work, many researchers have suggested improvements or simplifications to the OCF estimator. Some methods focused on further reducing the singularities and bounding the variance [131, 113]. Others simplified the sampling procedure while still retaining a square root convergence rate, making use of the equi-angular approach to sample the extra vertex [67, 115], similarly to Kulla and Fajardo [80]. Much like Kalos’ OCF estimator and its extensions, we sample additional path vertices to reduce singularities; however, we go a step further by considering not just one but two vertices, while also accounting for all geometry and anisotropic scattering terms along the connection subpath.

9.2 Joint path vertex sampling

We now describe our problem setting. We consider a configuration where a light subpath $\bar{\mathbf{x}}_l \mathbf{a}$ and an eye subpath $\mathbf{d} \bar{\mathbf{x}}_e$ are given. Our aim is to construct a full path $\bar{\mathbf{x}}_l \mathbf{a} \mathbf{b} \mathbf{c} \mathbf{d} \bar{\mathbf{x}}_e$ by connecting the endpoints \mathbf{a} and \mathbf{d} with two new vertices \mathbf{b} and \mathbf{c} via a *3-random-decision connection subpath*, as shown in Figure 9.2a. The subpaths $\bar{\mathbf{x}}_l$ and $\bar{\mathbf{x}}_e$ can have arbitrary lengths; zero lengths correspond to the double-scattering case where \mathbf{a} is on a light source and \mathbf{d} is on the eye lens. We also assume that a direction $\omega_{\mathbf{d}\mathbf{c}}$ at vertex \mathbf{d} is given, which is sampled as part of the eye subpath. The input to our sampling techniques is thus a vertex \mathbf{a} with incident direction $\omega_{l\mathbf{a}}$ and a vertex \mathbf{d} with outgoing direction $\omega_{\mathbf{d}\mathbf{c}}$.

Since the new vertex \mathbf{c} must reside on the ray $(\mathbf{d}, \omega_{\mathbf{d}\mathbf{c}})$, our task reduces to sampling a distance $t_{\mathbf{d}\mathbf{c}}$ from \mathbf{d} along the ray $(\mathbf{d}, \omega_{\mathbf{d}\mathbf{c}})$ and another vertex \mathbf{b} in 3D space. The pdf of the resulting full path is then

$$p(\bar{\mathbf{x}}) = p(\bar{\mathbf{x}}_l, \mathbf{a}, \omega_{\mathbf{d}\mathbf{c}}, \mathbf{d}, \bar{\mathbf{x}}_e) p(\mathbf{b}, t_{\mathbf{d}\mathbf{c}} | \omega_{l\mathbf{a}}, \mathbf{a}, \omega_{\mathbf{d}\mathbf{c}}, \mathbf{d}) G(\mathbf{c}, \mathbf{d}), \quad (9.3)$$

where the term $G(\mathbf{c}, \mathbf{d})$ again arises due to measure conversion. Note that the pdf $p(\bar{\mathbf{x}}_l, \mathbf{a}, \omega_{\mathbf{d}\mathbf{c}}, \mathbf{d}, \bar{\mathbf{x}}_e)$ is given by the chosen rendering algorithm, not by our sampling techniques. For example, when using bidirectional path tracing, the light and eye subpaths are sampled independently, in which case this pdf is given by the product of the two subpath pdfs: $p(\bar{\mathbf{x}}_l, \mathbf{a}, \omega_{\mathbf{d}\mathbf{c}}, \mathbf{d}, \bar{\mathbf{x}}_e) = p(\bar{\mathbf{x}}_l, \mathbf{a}) p(\omega_{\mathbf{d}\mathbf{c}}, \mathbf{d}, \bar{\mathbf{x}}_e)$. How the light and eye subpaths are created is however orthogonal to our problem; our sampling techniques only require that the input configuration $(\omega_{l\mathbf{a}}, \mathbf{a}, \omega_{\mathbf{d}\mathbf{c}}, \mathbf{d})$ be provided. Since all our pdfs are conditioned on this input configuration, we will use the following shorthand notation for it:

$$\Xi \equiv \omega_{l\mathbf{a}}, \mathbf{a}, \omega_{\mathbf{d}\mathbf{c}}, \mathbf{d}. \quad (9.4)$$

Ideally, we would like to define the three-dimensional joint pdf $p(\mathbf{b}, t_{\mathbf{d}\mathbf{c}} | \Xi) G(\mathbf{c}, \mathbf{d})$ (i.e., the pdf of the terms that our techniques introduce) to be proportional to the path throughput $f(\mathbf{a} \mathbf{b} \mathbf{c} \mathbf{d})$. Unfortunately, deriving such a pdf is not feasible since its normalization constant is the solution of the path integral. Inspired by prior work [80, 96], we propose to importance sample only the product of the geometry and scattering terms, such that

$$p(\mathbf{b}, t_{\mathbf{d}\mathbf{c}} | \Xi) G(\mathbf{c}, \mathbf{d}) = C_{\Xi} G(\mathbf{a} \mathbf{b} \mathbf{c} \mathbf{d}) \rho(\mathbf{a} \mathbf{b} \mathbf{c}), \quad (9.5)$$

where the normalization factor C_{Ξ} (derived in Section 9.3 below) depends on the input configuration Ξ , and where $G(\mathbf{a} \mathbf{b} \mathbf{c} \mathbf{d}) = G(\mathbf{a}, \mathbf{b}) G(\mathbf{b}, \mathbf{c}) G(\mathbf{c}, \mathbf{d})$ and $\rho(\mathbf{a} \mathbf{b} \mathbf{c}) = \rho(\mathbf{a}) \rho(\mathbf{b}) \rho(\mathbf{c})$. Our contribution is the definition of the joint pdf $p(\mathbf{b}, t_{\mathbf{d}\mathbf{c}} | \Xi)$ along with its corresponding sampling techniques. Note that $\rho(\mathbf{d})$ does not appear in the above equation, since we assume that both the outgoing direction, $\omega_{\mathbf{d}\mathbf{c}}$, and the incident direction at \mathbf{d} are given.

9.2.1 Factorizations of the joint pdf

Given the input configuration Ξ , there are a number of possible ways to obtain the connection subpath vertices. Each approach corresponds to factorizing the joint pdf in Equation 9.5 in a different way, which influences the definition of the conditional pdfs for the individual vertices and their corresponding sampling routines. We consider two such possible factorizations, depicted in Figure 9.2. In this section we only on the factorization of this joint pdf, and we will later explain how to obtain and sample from each factor.

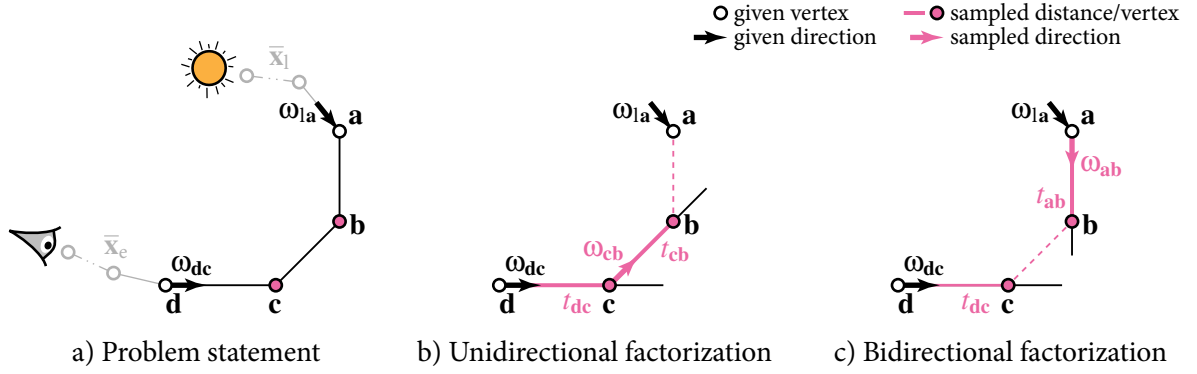


Figure 9.2: Given the vertices \mathbf{a} and \mathbf{d} , with an incident and an outgoing direction, respectively, we aim to sample vertices \mathbf{b} and \mathbf{c} from a joint distribution with density proportional to the product of the geometry and scattering terms of the resulting connection subpath. We derive the sampling techniques corresponding to two possible factorizations of this joint pdf.

Unidirectional factorization

One way to sample \mathbf{b} and \mathbf{c} is to first sample a distance t_{dc} , which effectively samples \mathbf{c} in combination with the given ω_{dc} , then sample a direction ω_{cb} , and finally sample a distance t_{cb} to obtain \mathbf{b} . This process is illustrated in Figure 9.2b. We call this *unidirectional factorization* since \mathbf{c} and \mathbf{b} are sampled from the same direction along the path starting from \mathbf{d} .

Since we sample \mathbf{b} as a distance and direction from \mathbf{c} , we need to change the measure of the joint pdf in Equation 9.5 to

$$p(\mathbf{b}, t_{dc} | \Xi) = p(t_{cb}, \omega_{cb}, t_{dc} | \Xi) G(\mathbf{b}, \mathbf{c}). \quad (9.6)$$

In order to sample the connection subpath, we factorize the joint pdf on the right-hand side above into the product of three conditional pdfs, corresponding to first sampling t_{dc} , then ω_{cb} , and finally t_{cb} :

$$p(t_{dc}, \omega_{cb}, t_{cb} | \Xi) = p(t_{dc} | \Xi) \quad (\mathcal{U}_3) \quad (9.7a)$$

$$p(\omega_{cb} | t_{dc}, \Xi) \quad (\mathcal{U}_2) \quad (9.7b)$$

$$p(t_{cb} | \omega_{cb}, t_{dc}, \Xi). \quad (\mathcal{U}_1) \quad (9.7c)$$

We introduce a shorthand notation \mathcal{U} above, where the subscripts denote the order in which we will derive the conditional pdfs.

Bidirectional factorization

Another way to sample \mathbf{b} and \mathbf{c} is to first sample a direction ω_{ab} from \mathbf{a} , then a distance t_{ab} , which yields \mathbf{b} , and finally a distance t_{dc} to obtain \mathbf{c} . We call this *bidirectional factorization* since \mathbf{b} and \mathbf{c} are sampled from opposite directions.

In this process, illustrated in Figure 9.2c, \mathbf{b} is sampled as a direction and distance from \mathbf{a} , and the change of measure of the target pdf is

$$p(\mathbf{b}, t_{dc} | \Xi) = p(t_{ab}, \omega_{ab}, t_{dc} | \Xi) G(\mathbf{a}, \mathbf{b}). \quad (9.8)$$

The bidirectional factorization samples the connection subpath by factorizing this pdf into the product of three conditional pdfs, corresponding to first sampling $\omega_{\mathbf{ab}}$, then $t_{\mathbf{ab}}$ and $t_{\mathbf{dc}}$:

$$p(\omega_{\mathbf{ab}}, t_{\mathbf{ab}}, t_{\mathbf{dc}} | \Xi) = p(\omega_{\mathbf{ab}} | \Xi) \quad (\mathcal{B}_3) \quad (9.9a)$$

$$p(t_{\mathbf{ab}} | \omega_{\mathbf{ab}}, \Xi) \quad (\mathcal{B}_2) \quad (9.9b)$$

$$p(t_{\mathbf{dc}} | \omega_{\mathbf{ab}}, t_{\mathbf{ab}}, \Xi). \quad (\mathcal{B}_1) \quad (9.9c)$$

As in the unidirectional factorization above, we use a shorthand notation \mathcal{B} for these conditional pdfs.

9.2.2 Sampling from the joint distributions

Given the factorizations defined in Equations 9.7a–9.7c and 9.9a–9.9c, we now need to derive the individual conditional pdfs. We do this via successive marginalization – a process we described in Section 2.4.3. In rendering, marginalization is commonly used for isolated low-dimensional sampling problems, such as drawing a direction proportional to the luminance of an environment map given by a 2D texture [109]. However, marginalization is not typically used for constructing random walk paths.

In the following two sections, we derive the conditional pdfs for both factorizations above using the marginalization relations in Equation 2.38. In Section 9.3, we derive analytic formulas for the pdfs and the sampling routines for the unidirectional factorization for the case of isotropic scattering. In Section 9.4, we will show how to take advantage of the symmetries in the geometry of the sampling problem to construct low-dimensional pdf tabulations for both the unidirectional and bidirectional factorizations for the general case of anisotropic scattering.

9.3 Analytic sampling

We first derive analytic expressions for the conditional pdfs in Equations 9.7a–9.7c for the case of isotropic scattering. Since the phase function is constant, Equation 9.5 simplifies to²

$$p(\mathbf{b}, t_{\mathbf{dc}} | \Xi) G(\mathbf{c}, \mathbf{d}) = C_{\Xi} G(\mathbf{a}, \mathbf{b}) G(\mathbf{b}, \mathbf{c}) G(\mathbf{c}, \mathbf{d}). \quad (9.10)$$

Using Equation 9.6 and canceling out geometry terms, the above further simplifies to

$$p(t_{\mathbf{cb}}, \omega_{\mathbf{cb}}, t_{\mathbf{dc}} | \Xi) = C_{\Xi} G(\mathbf{a}, \mathbf{b}). \quad (9.11)$$

Figure 9.3 illustrates the sampling process. Below we derive \mathcal{U}_1 , \mathcal{U}_2 , and \mathcal{U}_3 in this order via repeated marginalization.

²Note that the constant C_{Ξ} now also incorporates the product of the three isotropic phase functions $(4\pi)^{-3}$.

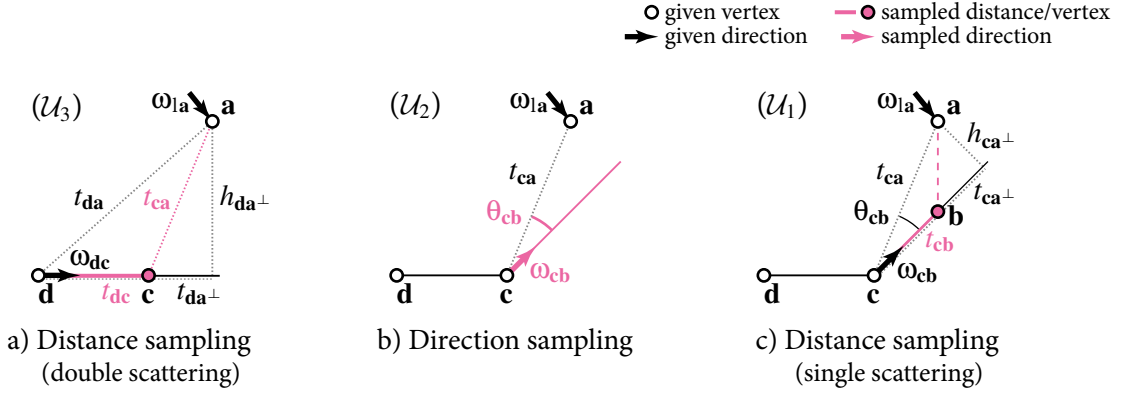


Figure 9.3: Unidirectional factorization of the joint pdf in Equation 9.5. (a) First, a distance t_{dc} is sampled along the ray $(\mathbf{d}, \omega_{dc})$ which yields vertex \mathbf{c} . (b) Then, a direction ω_{cb} is sampled from \mathbf{c} , and finally (c) a distance t_{cb} is sampled along the ray $(\mathbf{c}, \omega_{cb})$, yielding vertex \mathbf{b} .

9.3.1 Derivation of \mathcal{U}_1

Following the standard definition of conditional pdfs in Equation 2.38, we have

$$p(t_{cb} | \omega_{cb}, t_{dc}, \Xi) = \frac{p(t_{cb}, \omega_{cb}, t_{dc} | \Xi)}{p(\omega_{cb}, t_{dc} | \Xi)}. \quad (9.12)$$

From Equations 2.36 and 9.11 it follows that the denominator above is obtained by integrating out t_{cb} :

$$p(\omega_{cb}, t_{dc} | \Xi) = \int_0^\infty p(t_{cb}, \omega_{cb}, t_{dc} | \Xi) dt_{cb} \quad (9.13a)$$

$$= \int_0^\infty C_\Xi G(\mathbf{a}, \mathbf{c} + t_{cb}\omega_{cb}) dt_{cb} = C_\Xi \int_0^\infty \frac{1}{t_{ba}^2} dt_{cb} \quad (9.13b)$$

$$= C_\Xi \int_0^\infty \frac{1}{h_{ca^\perp}^2 + (t_{ca^\perp} - t_{cb})^2} dt_{cb} = C_\Xi \frac{\pi - \theta_{cb}}{t_{ca} \sin \theta_{cb}}, \quad (9.13c)$$

where h_{ca^\perp} is the distance between \mathbf{a} and the ray $(\mathbf{c}, \omega_{cb})$, t_{ca^\perp} is the distance between \mathbf{c} and this projection, and θ_{cb} is the angle between ω_{cb} and the line between \mathbf{c} and \mathbf{a} of length t_{ca} (Fig. 9.3c). By inserting Equations 9.11 and 9.13 into Equation 9.12, we obtain a simple expression for the pdf of t_{cb} :

$$p(t_{cb} | \omega_{cb}, t_{dc}, \Xi) = \frac{t_{ca} \sin \theta_{cb}}{\pi - \theta_{cb}} \frac{1}{h_{ca^\perp}^2 + (t_{ca^\perp} - t_{cb})^2}. \quad (9.14)$$

The denominator of the second term on the right-hand side is simply the squared distance t_{ba}^2 . This pdf is exactly equal to that of equi-angular sampling [67, 115, 80] provided that \mathbf{a} , ω_{cb} , and \mathbf{c} are given.

9.3.2 Derivation of \mathcal{U}_2

The geometric configuration for sampling the direction $\omega_{\mathbf{cb}}$ is illustrated in Figure 9.3b. Again, following the definition of conditional pdfs (Eq. 2.38), we have

$$p(\omega_{\mathbf{cb}} | t_{\mathbf{dc}}, \Xi) = \frac{p(\omega_{\mathbf{cb}}, t_{\mathbf{dc}} | \Xi)}{p(t_{\mathbf{dc}} | \Xi)}. \quad (9.15)$$

Following the marginalization chain, from Equation 9.13 we get

$$p(t_{\mathbf{dc}} | \Xi) = \int_{\mathcal{S}} p(\omega_{\mathbf{cb}}, t_{\mathbf{dc}} | \Xi) d\omega_{\mathbf{cb}} \quad (9.16a)$$

$$= \int_0^{2\pi} \int_0^{\pi} C_{\Xi} \frac{\pi - \theta_{\mathbf{cb}}}{t_{\mathbf{ca}} \sin \theta_{\mathbf{cb}}} \sin \theta_{\mathbf{cb}} d\theta d\phi = \frac{C_{\Xi} \pi^3}{t_{\mathbf{ca}}}. \quad (9.16b)$$

The resulting conditional pdf for sampling $\omega_{\mathbf{cb}}$, obtained by dividing Equation 9.13c by Equation 9.16b, thus depends only on \mathbf{a} and \mathbf{c} since the normalization factors again cancel out:

$$p(\omega_{\mathbf{cb}} | t_{\mathbf{dc}}, \Xi) = \frac{\pi - \theta_{\mathbf{cb}}}{\pi^3 \sin \theta_{\mathbf{cb}}}. \quad (9.17)$$

Note that the above pdf diverges when $\theta_{\mathbf{cb}}$ approaches zero. This same singularity also appears in the path contribution function f (when f is expressed as a function of the angle $\theta_{\mathbf{cb}}$) [115]. By sampling $\omega_{\mathbf{cb}}$ from this pdf, in Section 9.3.5 we will effectively cancel out this singularity, reducing variance. Note also that the equi-angular pdf (Eq. 9.14) has no singularity since $\omega_{\mathbf{cb}}$ is fixed.

9.3.3 Derivation of \mathcal{U}_3

The final marginalization step gives us the pdf for sampling the distance $t_{\mathbf{dc}}$ along the ray $(\mathbf{d}, \omega_{\mathbf{dc}})$. To complete the definition for this pdf, which is already given by Equation 9.16, we only need to derive the normalization factor C_{Ξ} , which we do by enforcing the pdf (Eq. 9.16b) to integrate to 1:

$$\int_0^{\infty} p(t_{\mathbf{dc}} | \Xi) dt_{\mathbf{dc}} = \int_0^{\infty} \frac{C_{\Xi} \pi^3}{\sqrt{h_{\mathbf{da}\perp}^2 + (t_{\mathbf{da}\perp} - t_{\mathbf{dc}})^2}} dt_{\mathbf{dc}} = \int_0^{\infty} \frac{1}{t_{\mathbf{ca}}} dt_{\mathbf{dc}} = 1, \quad (9.18)$$

where $h_{\mathbf{da}\perp}$ is the distance between \mathbf{a} and its projection onto the ray $(\mathbf{d}, \omega_{\mathbf{dc}})$, and $t_{\mathbf{da}\perp}$ is the distance between \mathbf{d} and this projection (Fig. 9.3a). Unfortunately, the above integral diverges at infinity; however, by setting a maximum sampling distance $t_{\mathbf{dc}}^{\max}$ along $(\mathbf{d}, \omega_{\mathbf{dc}})$, we can obtain an analytical expression for it:

$$C_{\Xi} \pi^3 = \frac{1}{C_{t_{\mathbf{dc}}}^{\max}} = \frac{1}{\operatorname{asinh}\left(\frac{t_{\mathbf{dc}}^{\max} - t_{\mathbf{da}\perp}}{h_{\mathbf{da}\perp}}\right) - \operatorname{asinh}\left(\frac{-t_{\mathbf{da}\perp}}{h_{\mathbf{da}\perp}}\right)}. \quad (9.19)$$

Substituting Equation 9.19 into Equation 9.16b yields the pdf for $t_{\mathbf{dc}}$ as simply

$$p(t_{\mathbf{dc}} | \Xi) = \frac{1}{C_{t_{\mathbf{dc}}}^{\max}} \frac{1}{\sqrt{h_{\mathbf{da}\perp}^2 + (t_{\mathbf{da}\perp} - t_{\mathbf{dc}})^2}} = \frac{1}{C_{t_{\mathbf{dc}}}^{\max} t_{\mathbf{ca}}}, \quad (9.20)$$

where $C_{t_{\mathbf{dc}}}^{\max}$ was defined in Equation 9.19 above. Interestingly, this means that for double-scattering the propagation distance should be chosen proportionally to the *inverse distance* to \mathbf{a} , whereas for single-scattering the equi-angular technique samples proportionally to the *inverse squared distance* to \mathbf{a} .

9.3.4 Joint unidirectional pdf

Now that we have completed the marginalization process, we can obtain a closed-form expression for the joint pdf in Equation 9.6 by multiplying the conditional pdfs \mathcal{U}_1 , \mathcal{U}_2 and \mathcal{U}_3 from Equations 9.14, 9.17 and 9.20 respectively:

$$p(\mathbf{b}, t_{\mathbf{dc}} | \Xi) = p(t_{\mathbf{cb}}, \omega_{\mathbf{cb}}, t_{\mathbf{dc}} | \Xi) G(\mathbf{b}, \mathbf{c}) \quad (9.21a)$$

$$= \frac{G(\mathbf{a}, \mathbf{b}) G(\mathbf{b}, \mathbf{c})}{\pi^3 C_{t_{\mathbf{dc}}}^{\max}}. \quad (9.21b)$$

Substituting back into Equation 9.3, we can confirm that the final path pdf indeed includes the geometry term $G(\mathbf{abcd})$.

Note that we subsume Kalos' [68] once collided flux (OCF) approach by sampling two distances and a direction to construct two intermediate vertices between \mathbf{a} and \mathbf{d} . Kalos' OCF corresponds to sampling the vertex \mathbf{b} proportionally to the product of geometry terms to \mathbf{a} and a fixed \mathbf{c} . This has a pdf $p(\mathbf{b} | t_{\mathbf{dc}}, \Xi) = p(\mathbf{b}, t_{\mathbf{dc}} | \Xi) / p(t_{\mathbf{dc}} | \Xi)$, which arises by dividing Equation 9.21 by Equation 9.20.

9.3.5 Unidirectional sampling techniques

In order to use our importance sampling techniques, we not only need the above pdfs, but also the corresponding sampling routines derived from their inverse cumulative distribution functions (CDFs).

Sampling from \mathcal{U}_3

A random distance $t_{\mathbf{dc}}$ along the ray $(\mathbf{d}, \omega_{\mathbf{dc}})$ can be sampled using the inverse CDF of Equation 9.20, derived by integrating the pdf over the ray and solving for $t_{\mathbf{dc}}$:

$$t_{\mathbf{dc}} = t_{\mathbf{da}} \sinh(\xi C_{t_{\mathbf{dc}}}^{\max}) - t_{\mathbf{da}\perp} (1 + \cosh(\xi C_{t_{\mathbf{dc}}}^{\max})), \quad (9.22)$$

where $t_{\mathbf{da}} = \sqrt{h_{\mathbf{da}\perp}^2 + t_{\mathbf{da}\perp}^2}$ is the distance between \mathbf{d} and \mathbf{a} (Fig. 9.3a), and $\xi \in [0, 1)$ is a uniform random number.

Sampling from \mathcal{U}_2

The direction pdf in Equation 9.17 is circularly symmetric, i.e. it depends only on $\theta_{\mathbf{cb}}$. Its corresponding CDF is:

$$P(\omega_{\mathbf{cb}}) = \int_0^{2\pi} \int_0^{\theta_{\mathbf{cb}}} \frac{\pi - \theta}{\pi^3 \sin \theta} \sin \theta d\theta d\phi = \frac{(2\pi - \theta_{\mathbf{cb}})\theta_{\mathbf{cb}}}{\pi^2}. \quad (9.23)$$

We invert it to solve for $\theta_{\mathbf{cb}}$, yielding the simple transformation

$$\theta_{\mathbf{cb}} = \pi(1 - \sqrt{\xi_1}), \quad \text{and} \quad \phi_{\mathbf{cb}} = 2\pi\xi_2, \quad (9.24)$$

where ξ_1 and ξ_2 are uniform random numbers, and $\theta_{\mathbf{cb}}$ and $\phi_{\mathbf{cb}}$ represent standard spherical coordinates with respect to a local frame where the direction from \mathbf{c} to \mathbf{a} is the z -axis (Fig. 9.3b).

Sampling from \mathcal{U}_1

Given a uniform random number ξ , a distance $t_{\mathbf{cb}}$ along $\omega_{\mathbf{cb}}$ is sampled using the inverse CDF of the equi-angular pdf (Eq. 9.14), yielding vertex \mathbf{b} :

$$t_{\mathbf{cb}} = t_{\mathbf{ca}^\perp} + h_{\mathbf{ca}^\perp} \tan\left(\xi(\pi - \theta_{\mathbf{cb}}) + \theta_{\mathbf{cb}} - \frac{\pi}{2}\right). \quad (9.25)$$

9.3.6 Discussion

The end result from the three steps above is a fully analytic method to importance sample the product of the geometry terms along 3-segment paths (double-scattering) in isotropic participating media which generalizes and complements the equi-angular sampling method [80] for 2-segment paths (single-scattering) as well as Kalos' [68] OCF estimator.

Similarly to the unidirectional case, the three conditional pdfs of the bidirectional factorization in Equation 9.9 are defined by the chain of marginalization of the joint pdf in Equation 9.8. The derivation of the first conditional pdf, \mathcal{B}_1 , is the same as for \mathcal{U}_1 . The resulting pdf is again equal to that of equi-angular sampling given \mathbf{b} , $\omega_{\mathbf{dc}}$, and \mathbf{d} . However, in contrast to the unidirectional case, we were unable to obtain analytic expressions for the remaining two conditional pdfs, \mathcal{B}_2 and \mathcal{B}_3 . In the following section, we present a tabulation method that can handle both factorizations, while additionally supporting anisotropic phase functions, without requiring analytic formulations.

9.4 Tabulated sampling

To handle anisotropic scattering, we introduce a method to efficiently tabulate our target pdf in Equation 9.5 using both the unidirectional and the bidirectional factorizations. In constructing our tables, we assume a circularly-symmetric 1D phase function (i.e. depending only on the deflection angle between the incident and outgoing directions). Isotropic scattering is a special case where the phase function is simply $\rho = 1/4\pi$. As we will demonstrate in our results, our tabulations achieve nearly ideal importance sampling of the product of the geometry and scattering terms.

The rest of this section follows the general structure of Section 9.3. We first describe the tabulations for the conditional pdfs in the unidirectional factorization (Sec. 9.4.2) and then follow up with the bidirectional factorization pdfs (Sec. 9.4.3).

9.4.1 General approach

As illustrated in Figure 9.2, both factorizations of our target pdf (Eq. 9.5) consider two types of sampling events: (1) a distance along a given ray and (2) a direction from a given vertex. For a medium with a given phase function, an entire *family* of pdfs exists for the geometric configuration of each sampling event. For instance, given a ray $(\mathbf{c}, \omega_{\mathbf{cb}})$, there is in general a different 1D pdf along that line for every possible relative location of vertex \mathbf{a} and its incident direction $\omega_{\mathbf{1a}}$. Our task is then to construct a table that holds tabulated line pdfs for a discrete set of positions \mathbf{a} and directions $\omega_{\mathbf{1a}}$.

Table parameterization

Even though each conditional pdf is one- or two-dimensional (corresponding to distance or direction sampling), the additional dimensionality of conditional variables makes naïve tabulation intractable. Our key idea to address this problem is to exploit the various symmetries in each geometric configuration of conditional variables by designing a suitable canonical coordinate system. This coordinate system allows us to dramatically reduce the dimensionality of our tables and their construction time.

Pdf parameterization

Depending on the geometric configuration, the pdfs we tabulate may have large variations. This is because we consider the product of the geometry and scattering terms, both of which can have sharp peaks. Such peaks can significantly reduce the accuracy of the tabulation; more importantly, when the geometry term has a singularity, accurate tabulation is not even possible. We address this problem by warping the pdf domains via suitable reparameterizations that analytically eliminate geometric variations.

9.4.2 Unidirectional factorization

For this factorization, illustrated in Figure 9.3, we seek to tabulate the factors given in Equation 9.7 for the joint pdf:

$$p(t_{\mathbf{cb}}, \omega_{\mathbf{cb}}, t_{\mathbf{dc}} | \Xi) = C_{\Xi} G(\mathbf{a}, \mathbf{b}) \rho(\mathbf{a}) \rho(\mathbf{b}) \rho(\mathbf{c}). \quad (9.26)$$

The difference from Equation 9.11 is that now we also take the phase functions into account. For notation simplicity, in this section we will denote the pdfs and the normalization constant C_{Ξ} with the same symbols as in Section 9.3, in spite of them being different.

Tabulation for \mathcal{U}_1

This pdf is the anisotropic generalization of the one defined in Equation 9.12 and illustrated in Figure 9.3c. For the tabulation we can safely ignore the denominator, since – as a function of $t_{\mathbf{cb}}$ – it is merely a normalization constant and we have to normalize our pdfs numerically anyway. We also ignore the terms C_{Ξ} and $\rho(\mathbf{c})$ in the numerator (Eq. 9.26), as both are constant w.r.t. $t_{\mathbf{cb}}$. We seek a pdf

$$p(t_{\mathbf{cb}} | \omega_{\mathbf{cb}}, t_{\mathbf{dc}}, \Xi) \propto G(\mathbf{a}, \mathbf{b}) \rho(\mathbf{a}) \rho(\mathbf{b}). \quad (9.27)$$

Novák et al. [96] proposed an on-the-fly tabulation of this pdf. However, their approach suffers from limited accuracy and overhead from the on-the-fly construction. We avoid these issues by precomputing accurate approximations of a number of such pdfs once and storing them in a compact table, which we can then use to sample a distance along any given ray. Our key observation is that this configuration, in general parameterized by $(\mathbf{a}, \mathbf{c}, \omega_{\mathbf{1a}}, \omega_{\mathbf{cb}})$, can in fact be expressed using just two angles representing $\omega_{\mathbf{1a}}$ in a suitable canonical coordinate system. This allows us to precompute the entire family of pdfs into a compact 2D table of 1D pdfs.

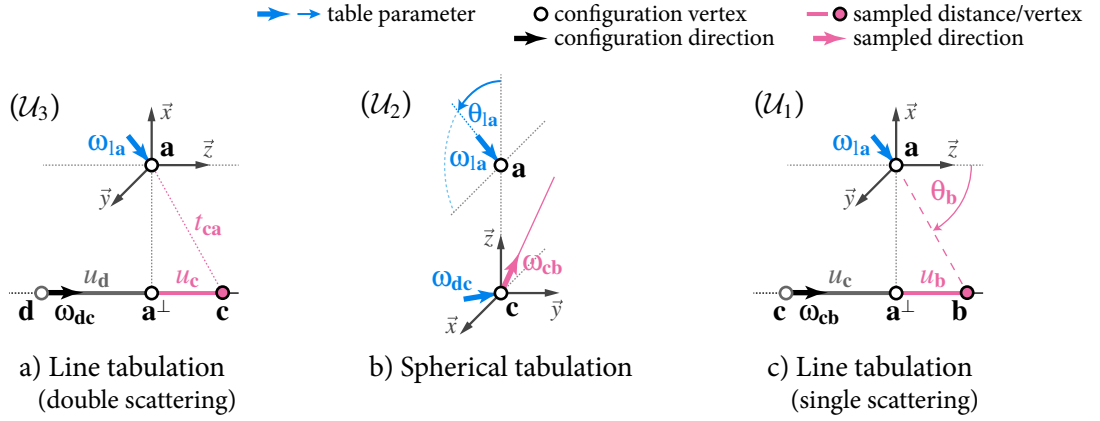


Figure 9.4: Canonical coordinate systems for tabulating the conditional pdfs in the unidirectional factorization (Eq. 9.7) of the joint pdf in Equation 9.5, for the case of anisotropic scattering. For each configuration we construct a table indexed by the entities colored in blue. The table entries are tabulated pdfs that are used for making the corresponding sampling decisions illustrated in Figure 9.3.

Table construction. Figure 9.4c shows the canonical coordinate system used for the tabulation. Given a ray $(\mathbf{c}, \omega_{\mathbf{cb}})$ and a vertex \mathbf{a} , we first rescale the problem such that the distance between the line and the vertex is one. We place the origin at \mathbf{a} , the z -axis is aligned with the ray, and the x -axis lies in the same plane. The table is indexed by the direction vector $\omega_{\mathbf{la}}$. Each entry in the table is a pdf $p(u)$ over the (signed) distance u along the line, measured from the projection \mathbf{a}^\perp of \mathbf{a} onto the line. Similarly to Novák et al. [96], we map the infinite line to a finite angular domain from the x -axis with $\theta = \text{arccot} u$. Each entry in our 2D table is a piecewise-linear approximation of the following pdf:

$$p(\theta(u)) = p(u) \left| \frac{du}{d\theta} \right| = p(u) \frac{1}{G(\mathbf{a}, \mathbf{b}(u))} \propto \rho(\mathbf{a}) \rho(\mathbf{b}(u)). \quad (9.28)$$

We tabulate the entire domain of $\theta \in [0; \pi]$, employing adaptive refinement to improve accuracy. Note that with the above change of variable, our tabulation is constant if the scattering is isotropic. In the special case where scattering is isotropic only at \mathbf{a} (e.g. \mathbf{a} is a point light source), all tabulated pdfs in the 2D table are identical, i.e. the table only has a single entry.

Sampling. For any given ray $(\mathbf{c}, \omega_{\mathbf{cb}})$, vertex \mathbf{a} , and direction $\omega_{\mathbf{la}}$, we first construct the coordinate system as described above. We then index the table with the canonical spherical coordinates of $\omega_{\mathbf{la}}$. We draw an angle $\theta_{\mathbf{b}}$ from the retrieved tabulated pdf, constrained to the interval $\theta_{\mathbf{b}} \in [0; \text{arccot} u_{\mathbf{c}}]$ (see Figure 9.4c), and transform it to a canonical distance $u_{\mathbf{b}} = \cot \theta_{\mathbf{b}}$ to obtain the desired propagation distance along the ray $t_{\mathbf{cb}} = -u_{\mathbf{c}} + u_{\mathbf{b}}$. We finally need to transform the pdf from the angular measure to the Euclidean length measure:

$$p(u_{\mathbf{b}}) = p(\theta_{\mathbf{b}}) \left| \frac{d\theta_{\mathbf{b}}}{du_{\mathbf{b}}} \right| = p(\theta_{\mathbf{b}}) [G(\mathbf{a}, \mathbf{b}) h_{\mathbf{ca}^\perp}], \quad (9.29)$$

where $h_{\mathbf{ca}^\perp}$ is the distance between \mathbf{a} and the ray (Fig. 9.4c).

Tabulation for \mathcal{U}_2

This spherical pdf, used to sample the scattering direction $\omega_{\mathbf{cb}}$ at vertex \mathbf{c} , is the anisotropic variant of the pdf defined in Equation 9.15 and illustrated in Figure 9.3b. Once again, since we will normalize

the pdf after tabulation, we can ignore the division by the normalization factor $p(t_{\mathbf{dc}} | \Xi)$. We therefore seek to tabulate:

$$p(\omega_{\mathbf{cb}} | t_{\mathbf{dc}}, \Xi) \propto \int_0^\infty p(t_{\mathbf{dc}}, \omega_{\mathbf{cb}}, t_{\mathbf{cb}} | \Xi) dt_{\mathbf{cb}} \quad (9.30a)$$

$$\propto \rho(\mathbf{c}) \underbrace{\int_0^\infty G(\mathbf{a}, \mathbf{b}) \rho(\mathbf{a}) \rho(\mathbf{b}) dt_{\mathbf{cb}}}_{P(\omega_{\mathbf{cb}})} \quad (9.30b)$$

where $\rho(\mathbf{c})$ is in front of the integral on the second line, as it does not depend on $t_{\mathbf{cb}}$. This integral $P(\omega_{\mathbf{cb}})$ is the normalization of a pdf in our previous tabulation, \mathcal{U}_1 , and can be readily looked up without recomputation. The directional distribution is proportional to the product $\rho(\mathbf{c})P(\omega_{\mathbf{cb}})$.

Table construction. Figure 9.4b shows the canonical coordinate system for the tabulation. The origin is at \mathbf{c} , the z -axis is the direction from \mathbf{c} to \mathbf{a} , and the x -axis is coplanar with the z -axis and $\omega_{1\mathbf{a}}$. The pdf family in Equation 9.30b then has three degrees of freedom: the direction $\omega_{\mathbf{dc}}$ and the angle $\theta_{1\mathbf{a}}$ of $\omega_{1\mathbf{a}}$ from the z -axis. We would thus need a 3D table of tabulated 2D pdfs, which is unfortunately impractical in terms of both computation and storage. We address this problem by tabulating $\rho(\mathbf{c})$ and $P(\omega_{\mathbf{cb}})$ separately, and sample from their product. We build a 1D table of spherical pdfs for $P(\omega_{\mathbf{cb}})$, indexed by $\theta_{1\mathbf{a}}$, where for every tabulated direction $\omega_{\mathbf{cb}}$ we evaluate $P(\omega_{\mathbf{cb}})$ by looking up the corresponding pdf normalization from the \mathcal{U}_1 table. An intuitive interpretation of this process is that we compute the single-scattered incident radiance field from \mathbf{a} (omitting transmittance) at \mathbf{c} , for a number of lobe orientations at \mathbf{a} . We store the pdfs in the table as fitted mixtures of von Mises Fischer (vMF) distributions. This allows for both compact storage and efficient sampling from the product with the vMF-fitted phase function, which can be easily rotated on-the-fly [140].

As we discussed in Section 9.3, this spherical pdf has a singularity around the pole. A naïve discretization would therefore result in a highly inaccurate approximation, as it cannot capture the infinite variation of the function. To make tabulation possible, we warp the domain using Equation 9.24: $\omega_{u,v} = \omega(\pi(1 - \sqrt{u}), 2\pi v)$. With this change of variables, the pdfs we store are

$$p(\omega_{u,v}) = p(\omega) \left| \frac{d\omega}{d\omega_{u,v}} \right| = p(\omega) \sin \theta_{\mathbf{cb}}. \quad (9.31)$$

We found that for scattering anisotropy of $g = 0.9$, every pdf in the 1D table for $P(\omega_{\mathbf{cb}})$ can be accurately represented with 20 vMF lobes, and the phase function with 3 lobes. We fit a (randomly initialized) vMF mixture to every tabulated spherical pdf using a standard iterative expectation-maximization algorithm. Every iteration first computes the expectation of the directions under the current mixture, followed by an update of the vMF parameters based on that expectation [4].

Sampling. Before sampling, we first transform the given configuration into the canonical coordinate system and index the two tables to retrieve a product vMF distribution (Eq. 9.30b). We then sample (u, v) with pdf $p(u, v)$, which yields $\omega_{\mathbf{cb}} = \omega_{u,v}$ using the above transformation. The solid angle pdf of the sampled direction is

$$p(\omega_{\mathbf{cb}}) = p(u, v) \left| \frac{dudv}{d\omega_{\mathbf{cb}}} \right| = p(u, v) \left[\frac{\pi - \theta_{\mathbf{cb}}}{\pi^3 \sin \theta_{\mathbf{cb}}} \right]. \quad (9.32)$$

Special case for $\rho(\mathbf{a}) = 1/4\pi$. When the scattering at \mathbf{a} is isotropic (e.g. it is a point light), the pdf family (Eq. 9.30b) has only one degree of freedom, the angle $\theta_{\mathbf{dc}}$ between $\omega_{\mathbf{dc}}$ and the z -axis. The whole family can thus be stored in a compact 1D table of 2D pdfs, for a set of angles $\theta_{\mathbf{dc}}$, without the need for product sampling.

Tabulation for \mathcal{U}_3

This last pdf in the marginalization chain is used to sample a distance along the ray $(\mathbf{d}, \omega_{\mathbf{dc}})$. It is the anisotropic variant of the pdf defined in Equations 9.16a and 9.20 and illustrated in Figure 9.3a:

$$p(t_{\mathbf{dc}} | \Xi) \propto \int_S \int_0^\infty p(t_{\mathbf{dc}}, \omega_{\mathbf{cb}}, t_{\mathbf{cb}} | \Xi) dt_{\mathbf{cb}} d\omega_{\mathbf{cb}} \quad (9.33a)$$

$$\propto \int_S p(\omega_{\mathbf{cb}} | t_{\mathbf{dc}}, \Xi) d\omega_{\mathbf{cb}}. \quad (9.33b)$$

The values of this pdf can be readily obtained by looking up in the corresponding normalizations from the \mathcal{U}_2 table described above.

Table construction. The geometric configuration for this family of pdfs, illustrated in Figure 9.4a, is identical to the one for $t_{\mathbf{cb}}$, and we use the same coordinate system as in Figure 9.4c. The only difference is that this time, in order to eliminate variations due to the geometry term, we reparameterize the tabulation domain using a different transformation: $v = a \sinh u$. The stored pdfs are then

$$p(v(u)) = p(u) \left| \frac{du}{dv} \right| = p(u) \sqrt{1+u^2} = p(u)t_{\mathbf{a}}, \quad (9.34)$$

where $t_{\mathbf{a}}$ is the distance between the corresponding point along the ray and \mathbf{a} .

Sampling. Sampling a distance $t_{\mathbf{dc}} = -u_{\mathbf{d}} + u_{\mathbf{c}}$ along a ray $(\mathbf{d}, \omega_{\mathbf{dc}})$ proceeds analogously to the case for $t_{\mathbf{cb}}$. We first construct the canonical coordinate system and retrieve the distribution corresponding to $\omega_{\mathbf{la}}$. We then sample $v_{\mathbf{c}}$ from the corresponding tabulated pdf, constrained to the interval $v_{\mathbf{c}} \in [0; a \sinh u_{\mathbf{d}}]$, and compute $u_{\mathbf{c}} = \sinh v_{\mathbf{c}}$. The line pdf of the sampled distance is:

$$p(t_{\mathbf{dc}}) = p(u_{\mathbf{c}}) = p(v_{\mathbf{c}}) \left| \frac{dv_{\mathbf{c}}}{du_{\mathbf{c}}} \right| = p(v_{\mathbf{c}}) \left[\frac{1}{t_{\mathbf{ca}}} \right]. \quad (9.35)$$

9.4.3 Bidirectional factorization

In the bidirectional factorization, we tabulate the factorized conditional pdfs in Equation 9.9 for the joint pdf:

$$p(\omega_{\mathbf{ab}}, t_{\mathbf{ab}}, t_{\mathbf{dc}} | \Xi) = C_{\Xi} G(\mathbf{b}, \mathbf{c}) \rho(\mathbf{a}) \rho(\mathbf{b}) \rho(\mathbf{c}). \quad (9.36)$$

As in the isotropic case, the conditional pdf \mathcal{B}_1 is the same as \mathcal{U}_1 but with different variable names. We illustrate the coordinate system in Figure 9.5c for reference. We can thus readily reuse the pdf and CDF tables and continue with the next pdf in the marginalization chain.

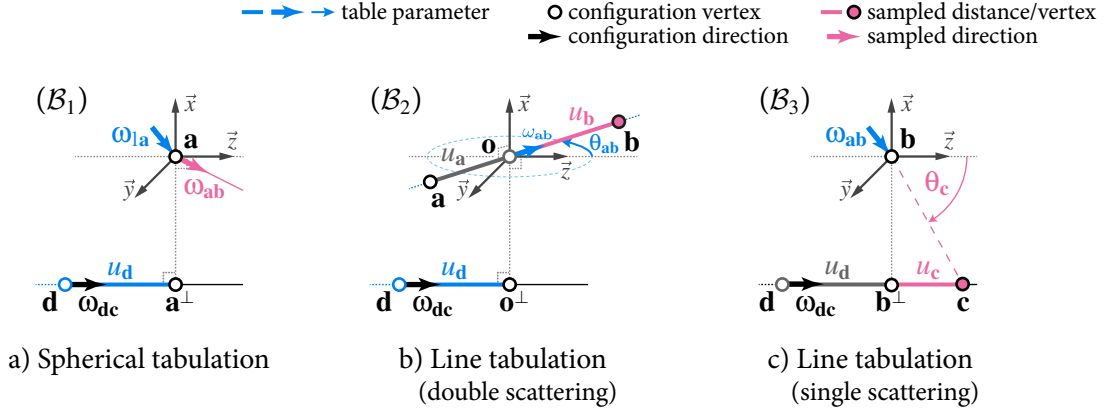


Figure 9.5: Canonical coordinate systems for tabulating the conditional pdfs in the bidirectional factorization (Eq. 9.9) of the joint pdf in Equation 9.5. For each configuration we construct a table indexed by the entities colored in blue. Each table entry is a tabulated pdf used for sampling distances or directions.

Tabulation for \mathcal{B}_2

This pdf is used to sample a distance along a ray $(\mathbf{a}, \omega_{\mathbf{ab}})$, given another ray $(\mathbf{d}, \omega_{\mathbf{dc}})$ (see Figure 9.2c). Following the definition of conditional pdfs in Equation 2.38, we have:

$$p(t_{\mathbf{ab}} | \omega_{\mathbf{ab}}, \Xi) = \frac{p(\omega_{\mathbf{ab}}, t_{\mathbf{ab}}, \Xi)}{p(\omega_{\mathbf{ab}}, \Xi)} \propto p(\omega_{\mathbf{ab}}, t_{\mathbf{ab}}, \Xi) \quad (9.37a)$$

$$= \int G(\mathbf{b}, \mathbf{c}) \rho(\mathbf{a}) \rho(\mathbf{b}) \rho(\mathbf{c}) dt_{\mathbf{dc}} \propto \int G(\mathbf{b}, \mathbf{c}) \rho(\mathbf{b}) \rho(\mathbf{c}) dt_{\mathbf{dc}}. \quad (9.37b)$$

The marginal VRL pdf of Novák et al. [96] roughly corresponds to this pdf, with $(\mathbf{d}, \omega_{\mathbf{dc}})$ being the eye ray and $(\mathbf{a}, \omega_{\mathbf{ab}})$ the VRL. Unlike Novák et al., however, we take into account anisotropic scattering at vertex \mathbf{b} and consider the semi-infinite extent of the line $(\mathbf{d}, \omega_{\mathbf{dc}})$ (which Novák et al. approximate as an infinite line). Our approach dramatically reduces variance, as we show in our results below.

Table construction. Figure 9.5b illustrates the geometric configuration. In the canonical coordinate system, the x -axis is aligned with the shortest connecting line between $(\mathbf{a}, \omega_{\mathbf{ab}})$ and $(\mathbf{d}, \omega_{\mathbf{dc}})$. The origin \mathbf{o} is the end of the connecting line on $(\mathbf{a}, \omega_{\mathbf{ab}})$ and the z -axis is aligned with $\omega_{\mathbf{dc}}$. Similarly to the previous cases, an entire family of these pdfs exists, parameterized by the rays $(\mathbf{a}, \omega_{\mathbf{ab}})$ and $(\mathbf{d}, \omega_{\mathbf{dc}})$. Once again, the symmetries in the geometry configuration and scale invariance allow us to parameterize this family of pdfs by only a few parameters. In fact, with Novák et al.'s assumption that the ray $(\mathbf{d}, \omega_{\mathbf{dc}})$ is infinite, we would only need a single table parameter, the $\theta_{\mathbf{ab}}$ angle between $\omega_{\mathbf{ab}}$ and $\omega_{\mathbf{dc}}$. However, the shape of the pdfs critically depends on the actual location of vertex \mathbf{d} along the line $(\mathbf{d}, \omega_{\mathbf{dc}})$. For that reason, we use the distance $u_{\mathbf{d}}$ from the projection of the origin onto $(\mathbf{d}, \omega_{\mathbf{dc}})$ as an additional parameter, and build a 2D table of 1D pdfs. For every combination of angle $\theta_{\mathbf{ab}}$ and distance $u_{\mathbf{d}}$, the associated pdf assigns probability density to distance u from the origin on the (infinite) line $(\mathbf{a}, \omega_{\mathbf{ab}})$. We build a piecewise-linear approximation of this pdf by looking-up the integral (Eq. 9.37b), to which the constructed pdf is proportional, from the previously constructed \mathcal{B}_1 table.

The pdf $p(u)$ in the canonical coordinate system is defined over an entire infinite $(\mathbf{a}, \omega_{\mathbf{ab}})$ line. To allow tabulation, we map this to a finite domain by parameterizing the position u using $v = \text{asinh } u$. Using this change of variables, the pdfs stored in the table are

$$p(v(u)) = p(u) \left| \frac{du}{dv} \right| = p(u) \sqrt{1+u^2} = p(u) t_{\mathbf{o}\perp}, \quad (9.38)$$

This is the same transformation as in Equation 9.34, with $t_{\mathbf{o}^\perp}$ denoting the distance between the corresponding point and \mathbf{o}^\perp (Fig. 9.5b). Any possible variation in the tabulation is solely due to the anisotropy of the phase function and the position of \mathbf{d} along the line $\omega_{\mathbf{dc}}$.

Sampling. Given two rays $(\mathbf{d}, \omega_{\mathbf{dc}})$ and $(\mathbf{a}, \omega_{\mathbf{ab}})$, we construct the canonical coordinate system and retrieve the distribution corresponding to $\theta_{\mathbf{ab}}$ and $u_{\mathbf{d}}$. We then sample $v_{\mathbf{b}}$ from the tabulated pdf, constraining it to the interval $v_{\mathbf{b}} \in [0; \text{asinh } u_{\mathbf{a}}]$, and compute $u_{\mathbf{b}} = \sinh v_{\mathbf{b}}$. The line pdf of the sampled distance $t_{\mathbf{ab}} = -u_{\mathbf{a}} + u_{\mathbf{b}}$ is

$$p(t_{\mathbf{ab}}) = p(u_{\mathbf{b}}) = p(v_{\mathbf{b}}) \left| \frac{dv_{\mathbf{b}}}{du_{\mathbf{b}}} \right| = p(v_{\mathbf{b}}) \left[\frac{1}{t_{\mathbf{bo}^\perp}} \right]. \quad (9.39)$$

Tabulation for \mathcal{B}_3

The final pdf in the marginalization chain is used to sample a direction $\omega_{\mathbf{ab}}$ at vertex \mathbf{a} (Fig. 9.2c). We seek to tabulate

$$p(\omega_{\mathbf{ab}} | \Xi) = \frac{p(\omega_{\mathbf{ab}}, \Xi)}{p(\Xi)} \propto \int p(\omega_{\mathbf{ab}}, t_{\mathbf{ab}} | \Xi) dt_{\mathbf{ab}} \quad (9.40a)$$

$$\propto \rho(\mathbf{a}) \underbrace{\int G(\mathbf{b}, \mathbf{c}) \rho(\mathbf{b}) \rho(\mathbf{c}) dt_{\mathbf{ab}}}_{P(\omega_{\mathbf{ab}})} \quad (9.40b)$$

where $\rho(\mathbf{a})$ is in front of the integral, as it does not depend on $t_{\mathbf{ab}}$. Similarly to the case for \mathcal{U}_2 , the integral on the second line, denoted $P(\omega_{\mathbf{ab}})$, is the normalization of a corresponding pdf in the \mathcal{B}_2 above, and can be readily looked up. The desired distribution is proportional to the product $\rho(\mathbf{a})P(\omega_{\mathbf{ab}})$.

Table construction. Figure 9.5a shows the canonical coordinate system used for the tabulation. The origin is at \mathbf{a} , the z -axis is aligned with $\omega_{\mathbf{dc}}$, and the x -axis lies in the same plane. The pdf family (Eq. 9.40b) is then spanned by the direction $\omega_{\mathbf{la}}$ and the position of \mathbf{d} along $\omega_{\mathbf{dc}}$ (given by the distance $u_{\mathbf{d}}$ in the figure). Once again, this means that we generally need a 3D table of tabulated spherical pdfs. Unfortunately, in this case the tabulated data cannot be efficiently represented by vMF mixtures, necessitating a different (e.g. wavelet or anisotropic Gaussian based) product sampling approach. We leave this for future work and build our pdf tables assuming isotropic scattering at \mathbf{a} .

Similarly to the special case for \mathcal{U}_2 , the problem reduces to constructing a 1D table of tabulated 2D pdfs, this time indexed by the distance $u_{\mathbf{d}}$. The spherical distributions in this family have a singularity around the z -axis, which is due to the $\sin^{-1} \theta$ factor in analytic inverse CDF of Novák et al. [96]. We eliminate this singularity by warping the spherical domain using the transformation $\omega_{u,v} = \omega(\pi\sqrt{1-u}, 2\pi v)$. With this change of variable, we store

$$p(\omega_{u,v}) = p(\omega) \left| \frac{d\omega}{d\omega_{u,v}} \right| = p(\omega) \sin \theta_{\mathbf{ab}}. \quad (9.41)$$

Sampling. For this sampling decision, we are given a ray $(\mathbf{d}, \omega_{\mathbf{dc}})$ and a point \mathbf{a} . After transforming this input configuration into the canonical coordinate system, we retrieve a pdf by indexing our table with $u_{\mathbf{d}}$. We then sample (u, v) with pdf $p(u, v)$, yielding $\omega_{\mathbf{ab}} = \omega_{u,v}$ using the above transformation. The solid angle pdf of the sampled direction is

$$p(\omega_{\mathbf{ab}}) = p(u, v) \left| \frac{dudv}{d\omega_{\mathbf{ab}}} \right| = p(u, v) \left[\frac{1}{2\pi^2 \sin \theta_{\mathbf{ab}}} \right]. \quad (9.42)$$

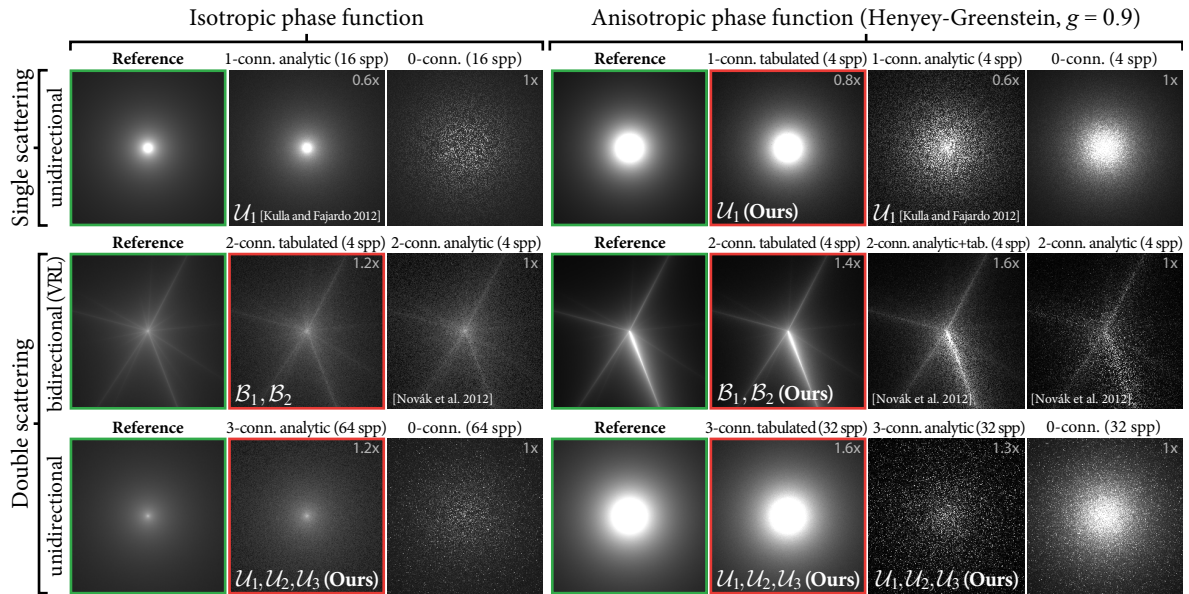


Figure 9.6: Comparison of our analytic and tabulated importance sampling techniques against state-of-the-art techniques in path tracing and virtual ray light (VRL) [96] renderers on a simple scene with an infinite medium and an isotropic point light source. The number in each image indicates its relative render time. Note the poor performance of the analytic techniques with anisotropic phase functions; our tabulations perform particularly well, since in this case transmittance (not importance sampled) has a smaller relative impact on the variability of the light transport.

9.5 Applications and results

The unidirectional and bidirectional sampling techniques described in the previous two sections provide an effective way to construct 3-random-decision connection subpaths. The most direct application of this method is to compute single and double scattering in participating media by connecting eye rays to random locations on light sources. This corresponds to the special case where the prefix and postfix subpaths $\bar{\mathbf{x}}_l$ and $\bar{\mathbf{x}}_e$ introduced in Section 9.2 are both of zero length, i.e. \mathbf{a} is on a light source and \mathbf{d} is on the eye lens.

Though our sampling routines do not currently consider chains of random decisions longer than 3, we can still leverage them to improve higher-order multiple scattering. We do this by generalizing deterministic (0-random-decision) shadow connections used in unidirectional and bidirectional path tracing into 1-, 2-, and 3-random-decision subpaths connecting the endpoints of light and eye subpaths of arbitrary lengths, as shown below.

In this section we describe how to incorporate our sampling routines for practical improvements in rendering. We compare our enhanced algorithms to classical unidirectional and bidirectional path tracing as well as VRL rendering. Figure 9.6 summarizes the connection techniques at our disposal, demonstrating their ideal-case performance. (We omit the images from the bidirectional 3-random-decision subpath technique, which produces similar results to the unidirectional 3-random-decision subpath technique.) Table 9.1 summarizes the resolution, memory footprint, and computation time for the pdf tabulations described in Section 9.4 above. All measurements have been obtained on a mobile quad-core Intel Core i7-2820QM 2.3GHz processor using a CPU ray tracer.

	Table resolution	Pdf resolution	Memory	Time
$\mathcal{U}_1, \mathcal{B}_1$	$\theta_{1a}: 200, \phi_{1a}: 400$	$u_b: 35$ (avg)	22 MB	320 ms
\mathcal{U}_2	$\theta_{1a}: 90$	$\omega_{cb}: 20$ lobes	36 KB	6 s
\mathcal{U}_3	$\theta_{1a}: 90, \phi_{1a}: 180$	$u_c: 100$	13 MB	580 ms
\mathcal{B}_2	$u_d: 300, \theta_{ab}: 100$	$u_b: 200$	24 MB	530 ms
\mathcal{B}_3	$u_d: 100$	$\theta: 200, \phi: 400$	32 MB	680 ms

Table 9.1: Resolution, memory footprint, and construction time statistics for the pdf tabulations illustrated in Figures 9.4 and 9.5.

9.5.1 Unidirectional path tracing

The most common forward path tracer uses unidirectional path sampling of transmittance and phase function starting at the eye, combined with deterministic (0-random-decision) shadow connections (a special case of Figure 9.1b). We use this algorithm as a baseline for our comparisons.

Figure 9.7 show a scene with isotropic (left) and highly anisotropic (right) phase functions and scattering albedo 0.57. Even with explicit (0-random-decision) shadow connections, traditional path tracing suffers from significant spike noise as it cannot importance sample the geometry term or phase function when making the connection.

1 random decision

Our first extension to unidirectional path tracing incorporates 1-random-decision connection subpaths. We generate a random-walk path starting at the eye, just as with traditional unidirectional path tracing. However, instead of connecting each subpath vertex to the light with a deterministic connection, we independently connect each subpath segment to the light using a 1-random-decision subpath, sampled with the \mathcal{U}_1 technique. This corresponds to using (isotropic) equi-angular sampling or Novák et al.’s [96] tabulated anisotropic generalization (Fig. 9.1c) as a vertex-segment connection technique between an eye subpath and a light. As in previous work, we combine \mathcal{U}_1 with transmittance importance sampling via MIS.

This extension corresponds to interpreting previous work as a general vertex-segment connection technique. On top of this, our precomputed tabulation improves render time and affords a more accurate approximation of the anisotropic pdf since we avoid coarse on-the-fly tabulation. The center columns in Figure 9.7 show unidirectional path tracing with such 1-random-decision connections. This reduces RMS error by a factor of 4 to 6 (a $16\times$ to $36\times$ improvement in rendering convergence time) compared to using only deterministic shadow connections.

2 and 3 random decisions

Our main contribution for unidirectional path tracing additionally incorporates our \mathcal{U}_2 and \mathcal{U}_3 sampling techniques for up to 3-random-decision connections (Fig. 9.2b) between each eye subpath segment and a point on a light source. We also sample a 1-random-decision connection for the first eye path segment to account for single scattering. We use Equations 9.22–9.25 for isotropic media and our tabulated equivalents for anisotropic media. For each decision, we randomly choose between our dis-

tance and direction sampling routines and the traditional routines (transmittance and phase function sampling), and use the one-sample MIS balance heuristic to combine the estimates. The resulting algorithm fully subsumes traditional (0-random-decision connection) path tracing, which corresponds to always selecting the traditional techniques. By using our techniques, we can importance sample an additional one, two or three random decisions for longer connections compared to previous work.

The right columns in Figure 9.7 show the result of rendering the DRAGON scene using our enhanced 3-random-decision connection path tracing. This provides a substantial variance reduction compared to using just 1-random-decision connections, resulting in a $15\times$ (isotropic) to $38\times$ (anisotropic) reduction in total RMS error compared to deterministic shadow connections. This corresponds to a $225\text{--}1444\times$ speedup in rendering convergence time.

The LIGHTHOUSE scene in Figure 9.8 has an anisotropic light source and albedo 0.8, and Figure 9.9 shows the two scenes with heterogeneous media; both figures show significant variance reduction using our 3-connection techniques. For heterogeneous media, we do not use the transmittance tabulation of Kulla and Fajardo [80], though a combination with our techniques would be interesting to investigate. In Figure 9.10 we demonstrate longer bounces (path lengths 1–8) with and without our improvements.

9.5.2 Virtual ray lights

We incorporated our tabulated bidirectional sampling techniques into a many-light renderer. This renderer generates random-walk subpaths from the lights with transmittance and phase function importance sampling and stores these subpaths as a collection of VRLs. During rendering, eye rays are connected to the VRLs using a one-sample Monte Carlo estimator. We refer to Novák et al. [96] for the complete description of the VRL algorithm.

Our \mathcal{U}_1 , \mathcal{U}_2 , and \mathcal{U}_3 tabulations provide three concrete improvements over Novák et al.’s [96] approach: (1) our tabulations of the joint VRL pdf (Eq. 4.21) are precomputed, affording higher precision and avoiding expensive on-the-fly tabulation; (2) we eliminate approximations in the marginal distribution (see Section 9.4.3), which significantly reduces noise; and (3) we can importance sample one additional random decision – the emitted direction of the VRL – which allows us to sample VRLs relevant to the viewpoint. We evaluate these improvements compared to a baseline VRL renderer.

In Figure 9.11, we compare Novák et al.’s approach to our improved 2-random-decision (left) and 3-random-decision (right) connections. The medium scattering albedo is 0.89. The right image in particular considers an uncorrected variation of VRLs, where a VRL is generated independently for each eye ray, instead of using a single set of VRLs for the entire image. Our 3-random-decision technique not only importance samples the connection between the eye rays and the VRLs but also the directions of the VRLs themselves when they are sampled from the light sources. In both cases, we see substantial quality improvement in the same rendering time.

9.5.3 Bidirectional path tracing

Bidirectional path tracing (BPT) also traditionally relies on deterministic connections (see Figure 9.1b), but typically converges much faster than unidirectional path tracing, as it can combine many techniques for sampling the same full light transport path. We will show that by incorporating segment-segment 2-random-decision connections into BPT we can obtain significant variance reduction even for this generally more robust algorithm.

Our baseline for comparison is a standard bidirectional path tracer. Both the light $\bar{\mathbf{x}}_l$ and eye $\bar{\mathbf{x}}_e$ subpaths are created using transmittance and phase function importance sampling and then each pair of vertices are coupled with a deterministic 0-random-decision connection to construct a full path from a light to the eye. We combine all possible vertex-vertex connection path sampling techniques using MIS with the balance heuristic.

1 and 2 random decisions

Instead of connecting each pair of vertices, we incorporate our \mathcal{B}_1 and \mathcal{B}_2 techniques for 2-random-decision connections between each pair of segments on the light and eye subpaths, as illustrated in Figure 9.1d. A light subpath segment defines vertex \mathbf{a} and direction $\omega_{\mathbf{ab}}$, and an eye subpath segment provides vertex \mathbf{d} and direction $\omega_{\mathbf{dc}}$. Given this configuration, we use our tabulated bidirectional factorization (Sec. 9.4.3) to sample $t_{\mathbf{ab}}$ and $t_{\mathbf{dc}}$ proportionally to the product of inverse squared distance, $G(\mathbf{b}, \mathbf{c})$, and phase functions, $\rho(\mathbf{b})\rho(\mathbf{c})$. We perform this connection for every eye-light segment pair. This effectively converts VRLs from a many-light algorithm into a bidirectional segment-segment connection technique, leveraging our more accurate and efficient tabulated joint pdfs. As with all our other implementations, we combine transmittance and phase function importance sampling via MIS, thereby subsuming 0- and 1-random-decision connections.

Figure 9.12 shows a scene with medium albedo 0.57 rendered with bidirectional path tracing. Even though BPT is generally much more robust than unidirectional path tracing, incorporating our 2-random-decision connections results in visible noise reduction; RMS error is reduced by roughly a factor of 3.5 and 5 for path lengths of 1–3 and 1–8 respectively. This corresponds to $12\times$ and $25\times$ improvement in rendering convergence speed.

9.6 Discussion

We conclude this chapter by summarizing the limitations of our method and some possible improvements. We also briefly revisit the relations to neutron transport to aid a discussion on the convergence rates of our techniques, based on some prior developments in that field.

Variance due to other terms

Although the geometry and scattering terms are often most responsible for excessive variance in participating media, other terms can introduce high variance in certain scene configurations. For example, transmittance sampling may outperform our techniques in some scenes with highly heterogeneous and dense media. In scenes with uniform light emission (e.g. a constant environment map), traditional unidirectional sampling (Fig. 9.1a) will likely be the best strategy. We try to handle such cases robustly using MIS; however, by splitting samples across multiple techniques we allocate fewer samples to a potentially better technique. Though our method outperforms existing techniques in all our tests, we obtained less benefit in scenes with a combination of caustic paths as well as surface-to-medium transport, e.g. the noise visible near the floor and stools in the `STAGE` scene in Figure 9.12.

Longer connections

While our method is tailored to single and double scattering, in Figures 9.10 and 9.12 we demonstrate practical improvement for path lengths of up to 8 by using our techniques as generalized “shadow connections”. This improvement will likely diminish for even longer paths. However, since longer paths also have smaller contribution, the final image will still benefit from our improvement in lower-order scattering.

Anisotropic light sources

Our current implementation of the bidirectional factorization does not support 3-random-decision connections when the scattering at vertex \mathbf{a} is anisotropic. This is mainly a limitation of our implementation, not our theory. For this particular case we would need a product importance sampling technique that is better suited to the shape of the distributions, e.g. the anisotropic spherical Gaussians framework of Xu et al. [160].

Convergence rates

As discussed in Section 9.1.2, Kalos [68] showed that the UCF estimator (which corresponds to traditional unidirectional path tracing with next event estimation) results in *infinite* variance in the presence of light sources inside participating media. Moreover, in this case the central limit theorem does not hold, leading to a substantially lower asymptotic convergence rate for volumetric path tracing than its surface equivalent ($1/\sqrt[3]{N}$ instead of $1/\sqrt{N}$). By reducing the order of the singularity in the geometry term, Kalos’ [68] OCF estimator provides bounded variance and recovers a $1/\sqrt{N}$ convergence rate. Our techniques inherit this property by subsuming the OCF estimator. Nonetheless, a formal analysis of the exact convergence rates of our methods would be interesting, especially in combination with multiple importance sampling where singularities are further reduced by the balance heuristic. Dubi et al. [27] and Rief et al. [115] also showed that the combination of path tracing and equi-angular sampling retains the $1/\sqrt{N}$ convergence rate, despite the fact that this combination still leaves an angular singularity and has infinite variance. Our techniques also employ equi-angular sampling but additionally remove this angular singularity in a way similar to Raab and Beikert [113] whose “detector emphasis” technique importance samples directions near a point detector located in the medium.

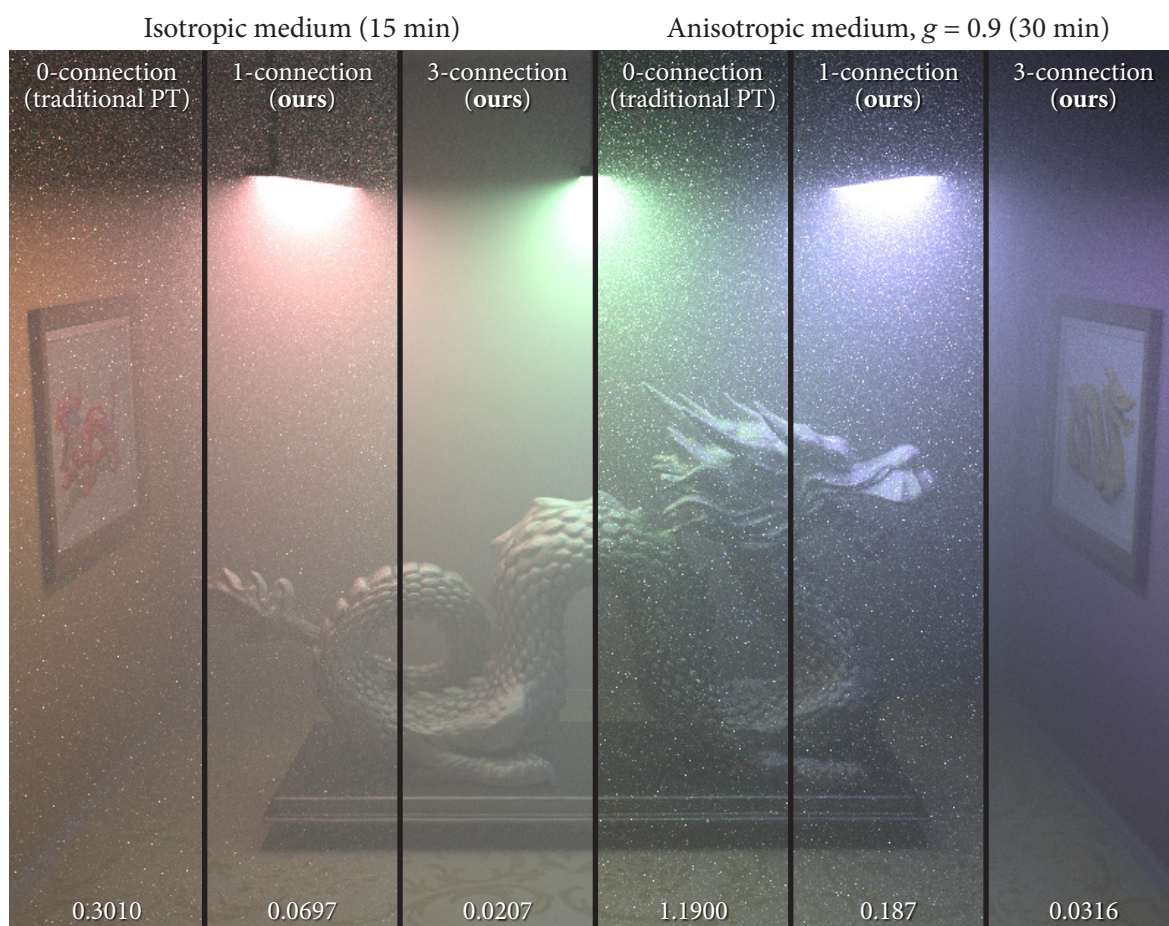


Figure 9.7: Single and double scattering in isotropic and anisotropic media, rendered with unidirectional path tracing using traditional 0-random-decision connections and our extensions incorporating 1- and 3-random-decision connection subpaths. Our techniques achieve $5\times$ to $37\times$ reduction in RMS error (bottom), corresponding to $25\times$ to $1444\times$ reduction in render time.

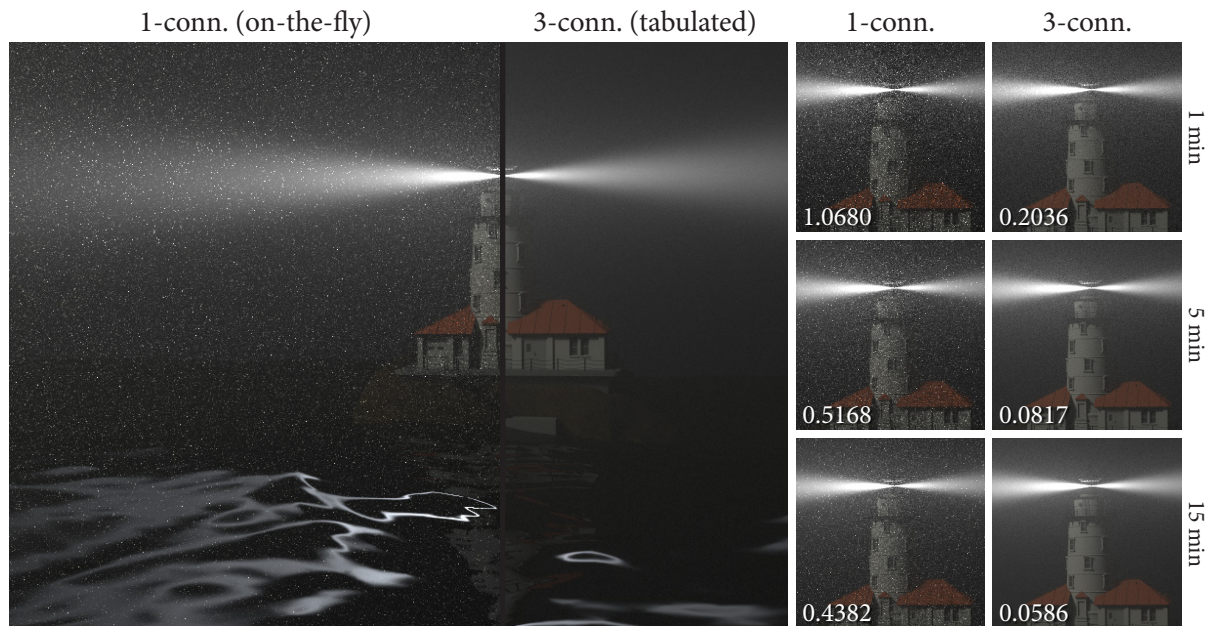


Figure 9.8: A scene containing an anisotropic medium ($g = 0.9$) and an anisotropic point light, rendered with unidirectional path tracing using the on-the-fly tabulated 1-connections of Novák et al. [96] (left) and our improved pre-tabulated 3-connections (right).

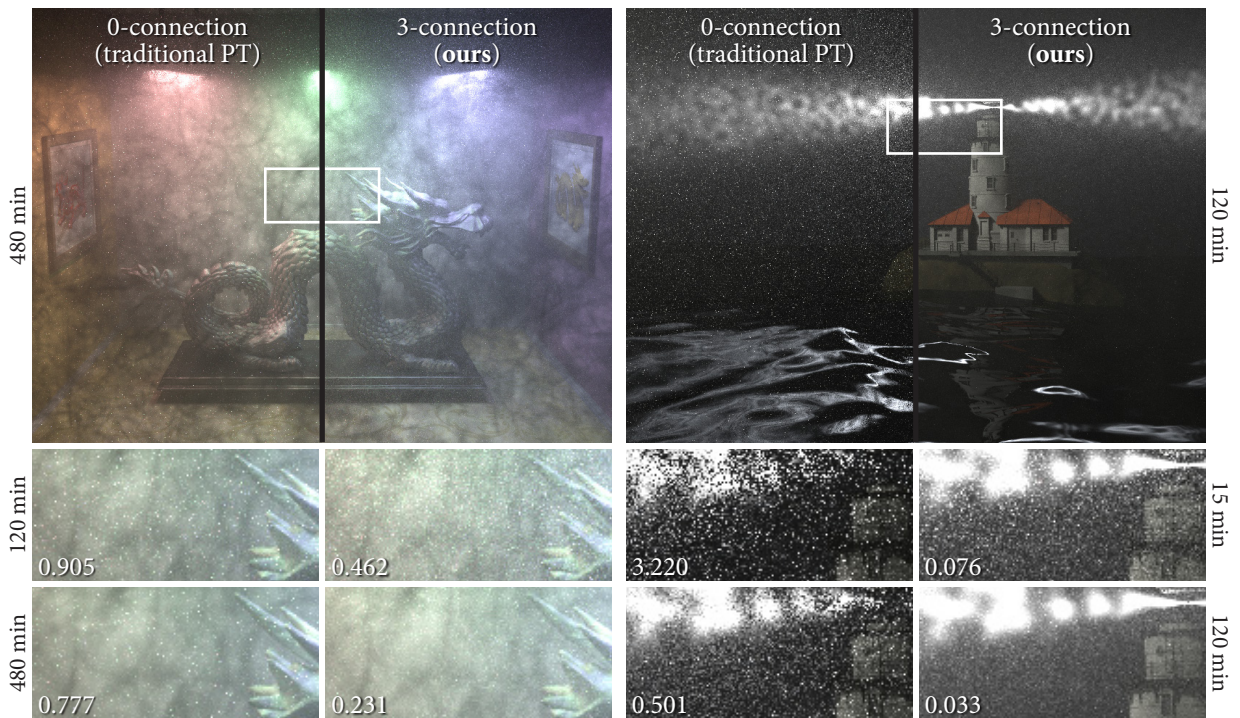


Figure 9.9: Two scenes with anisotropic ($g = 0.9$) heterogeneous media, rendered with single and double scattering using unidirectional path tracing. In contrast to our improved (3-connection) techniques, the traditional (0-connection) technique images miss most of the double-scattered illumination.

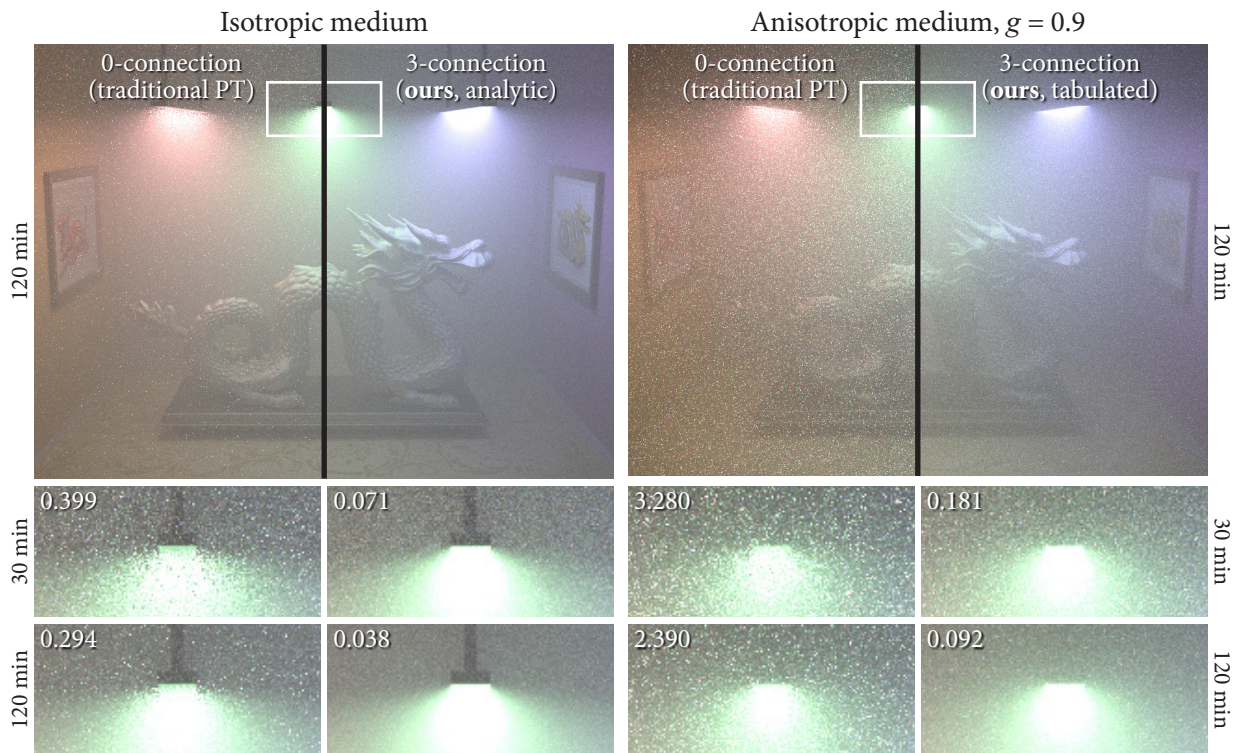


Figure 9.10: The DRAGON scene with an isotropic and anisotropic medium, rendered with unidirectional path tracing. We visualize path lengths 1–8 and compare the traditional (0-connection) sampling techniques to our extended (3-connection) method.

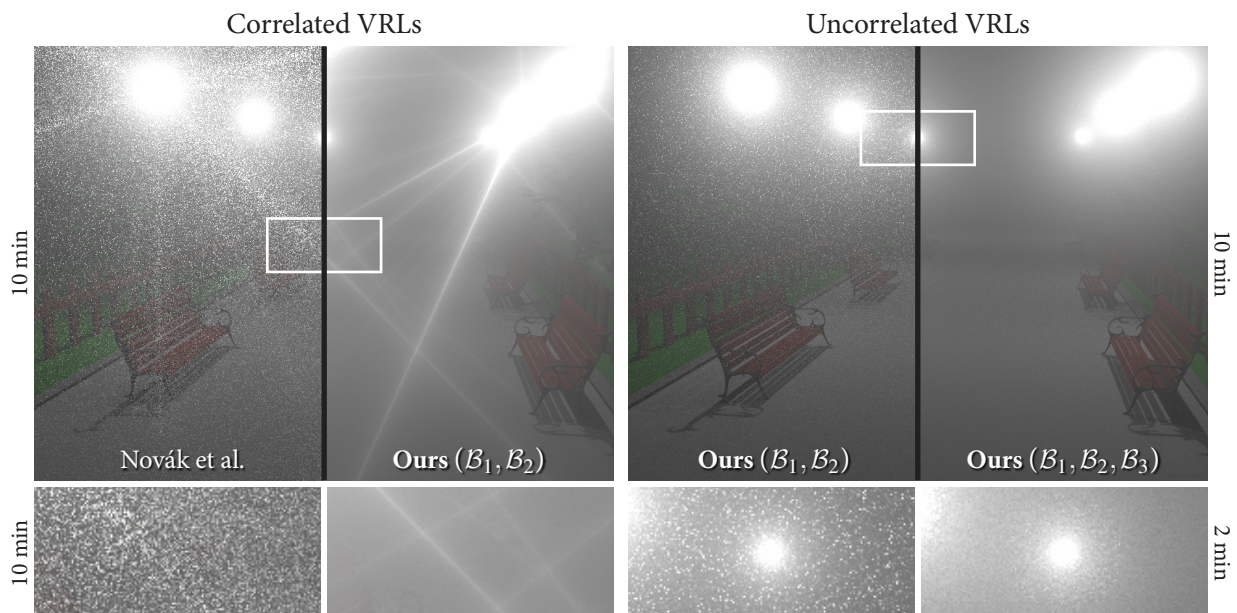


Figure 9.11: Our tabulated pdfs are more accurate than the ones of Novák et al. [96] and dramatically reduce variance when making connections between VRLs and eye rays (left). When sampling an independent VRL for each eye ray (right), our techniques allow us to importance sample the emitted VRL direction to further reduce variance.

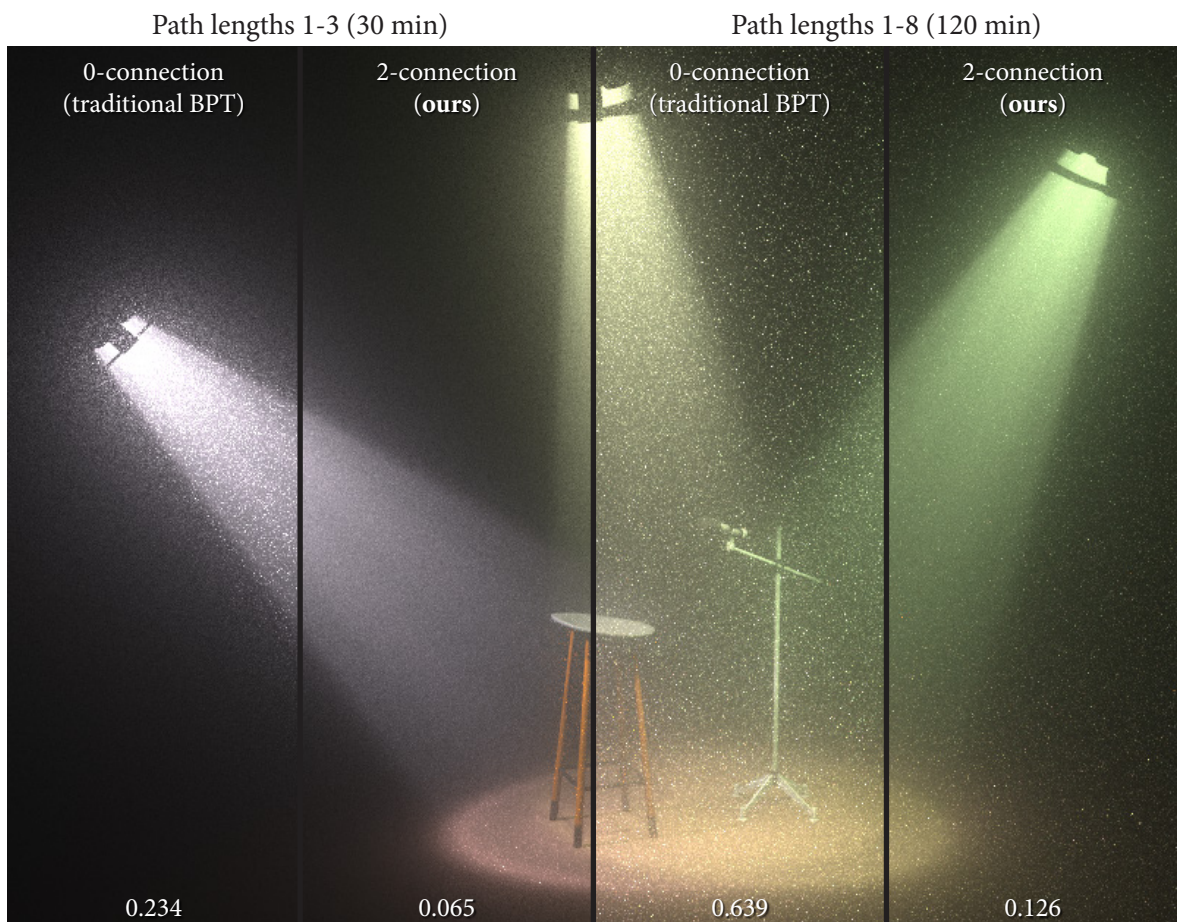


Figure 9.12: An application of our sampling techniques to bidirectional path tracing. Importance sampling more terms in the subpath connections achieves significant reduction in RMS error (bottom), corresponding to $12\times$ (left) to $25\times$ (right) reduction in render time.

Conclusion

10

The ever-growing demand for realistic-looking computer-generated imagery has been a driving force for the research in physically-based rendering for decades. Significant advances have been made over the last 30 years in this field, which is by now widely regarded as mature, yet robust light transport simulation in arbitrary scenes has remained a highly challenging problem.

Various approaches for formalizing and solving the global illumination problem have been proposed. One such approach is the path integral formulation of light transport, which expresses this problem as an integral over all possible trajectories that connect the light sources in the scene to the eye. State-of-the-art rendering algorithms use Monte Carlo integration to compute one such high-dimensional integral for every image pixel. The key to achieving efficiency with Monte Carlo methods is to carefully distribute the costly integration samples, i.e. paths, and the best-known method for doing that is importance sampling.

In this thesis we investigated Monte Carlo path sampling techniques for efficient light transport simulation. We expressed existing methods as sampling techniques in the path integral framework, devised new importance sampling techniques for rendering surfaces and participating media, and combined different techniques to obtain more robust, lower-variance light transport estimators. In this final chapter we summarize the major contributions and results of this work, and discuss follow-up and pertinent future work in addition to what has already been mentioned in the previous chapters.

Importance sampling for many-light rendering

Many-light methods decompose path sampling into a virtual point light (VPL) distribution stage and a subsequent rendering stage that computes the contribution of all VPLs to the surfaces seen from the eye. The fidelity of the resulting image depends on the number of VPLs as well as on the quality of their distribution in the scene. In Chapter 5 we presented an importance-driven VPL resampling algorithm that produces a set of VPLs relevant for the chosen viewpoint, where every VPL brings roughly the same amount of energy to the eye. This is achieved by probabilistically accepting or rejecting each VPL based on an on-the-fly estimation of its image contribution. The algorithm has low memory requirements, a few intuitive parameters, and is easy to implement and integrate into an existing many-light renderer. We demonstrated its ability to find relevant VPLs in scenes with difficult visibility configurations, and saw rendering speed-ups of over an order of magnitude compared to traditional VPL distribution.

Producing high-fidelity images using many-light rendering often requires thousands of VPLs. In such cases the final rendering stage can become prohibitively expensive, even though typically only a small fraction of all VPLs have significant contribution to any point in the scene. In Chapter 6 we presented a method that aims to efficiently find the important VPLs for each point seen from the eye by ex-

exploiting the illumination smoothness in the scene. The main idea is to evaluate and cache the exact contributions of all VPLs at a sparse set of locations and then reuse these evaluations in the form of importance to probabilistically select the few most relevant VPLs at other locations. Using the bootstrap VPL evaluations, we devised several increasingly conservative VPL importance distributions to ensure that sampling remains robust to illumination discontinues. We also proposed a novel multiple importance sampling (MIS) heuristic to combine the many distributions gathered around a point into a low-variance pixel estimator. Our importance caching method is best suited to highly occluded scenes with complex non-uniform lighting, where visibility is often a major source of variance in the many-light estimator. Our implementation can deliver a good initial approximation in a matter of seconds on a commodity CPU, while a simplified version of the method interactively renders accurate and high-quality previews. The algorithm will also likely benefit from a GPU implementation, as all its stages are inherently parallel.

One further potential improvement to our method would be to compute the gradients of the VPL contributions at the importance records and use them to obtain better VPL importance estimates at other locations via extrapolation [156, 50]. The challenge with this approach would be to efficiently normalize the resulting extrapolated VPL probability distributions at each location.

Although many methods exist that exploit illumination smoothness, a unique property of our importance caching approach is that it does so in an unbiased way. This allows us to construct progressive pixel estimators and obtain high-quality images with a bounded memory footprint. However, a current limitation of all many-light rendering methods, including ours, is the bias introduced by clamping the VPL contributions, which makes the progressive pixel estimators inconsistent. We have addressed this problem in a follow-up work [20] by devising a clamping relaxation scheme, similar to the radius reduction scheme in progressive photon mapping, which increases the VPL clamping threshold at every rendering iteration. While a formal analysis of this formulation is still pending, we have experimentally shown that clamping relaxation makes the many-light solution converge to a ground-truth image as the number of rendering iterations increases.

Another interesting direction for extending our importance caching method in future would be to further improve its scalability with the number VPLs. Currently, the time and memory costs of the VPL importance evaluation and caching stage are linear to the number of VPLs. One way to improve this could be to use the lightcuts approach of Walter et al. [150] to cluster all VPLs hierarchically before rendering. Instead of storing the contributions of individual VPLs, each importance record would then store a cut in the light tree, i.e. the (approximate) contributions of entire VPL clusters. Sampling a VPL at a shading point would then proceed in two steps: first selecting a cluster proportionally to its recorded importance, and then choosing a VPL inside the cluster proportionally to its subpath throughput (a.k.a. power) as in lightcuts.

Vertex connection and merging

Many global illumination approaches have been proposed that achieve outstanding rendering efficiency via specially optimized solutions for certain illumination effects, e.g. indirect diffuse lighting or specular caustics, but at the cost of ignoring or inefficiently handling other effects. Robust light transport simulation under a wide variety of lighting configurations has remained an important and challenging problem. In Chapter 7 we addressed this problem by combining two state-of-the-art general-purpose rendering methods: bidirectional path tracing and photon mapping. The key to an automatic and effective combination is our reformulation of photon mapping as a pure Monte Carlo path sampling technique, which we call vertex merging. An important aspect of our vertex merging formulation is the derivation of a corresponding path pdf with an associated product area measure that has the same

unit as that of vertex connection – the family of path sampling techniques used in bidirectional path tracing. This enables the application of multiple importance sampling to combine the two methods into a practical unified rendering algorithm – vertex connection and merging (VCM) – which employs more path sampling techniques than bidirectional path tracing and photon mapping combined. We showed that this algorithm is more robust than either of these two methods alone and demonstrated its efficiency in handling a variety of lighting effects, ranging from direct illumination, diffuse and glossy inter-reflections, to specular-diffuse-specular light transport. We believe that our solution has an immediate practical utility in a wide range of applications, especially in predictive rendering systems, both interactive and offline, where robustness under different lighting setups is critical.

Our path integral formulation of photon mapping also brings new insight into the efficiency of this algorithm which has been praised for its ability to render high-quality caustics. We showed that its vertex merging technique is not intrinsically more robust than the techniques employed by bidirectional path tracing; in fact, most light transport paths have a higher probability of being sampled by vertex connection than by vertex merging. Rather, it is the photon map implementation that allows for efficiently reusing light subpaths to cheaply construct a large number of transport estimators for every pixel, thereby amortizing the light subpath sampling cost over the entire image. This brings brute-force variance reduction that is particularly beneficial for specular-diffuse-specular paths. These paths are notoriously problematic for bidirectional path tracing as none of the vertex connection techniques it employs can sample them with high enough probability. The combined MIS estimator in VCM automatically detects such cases and promotes vertex merging techniques by assigning them proportionally high weights.

An interesting and important property of our progressive VCM algorithm is that, even though it is biased and uses a similar merging-radius reduction scheme as progressive photon mapping (PPM), its error converges at a rate that is higher than that of PPM and equal to that of the unbiased bidirectional path tracing. As we progressively shrink the radius at every rendering iteration, the power MIS heuristic automatically compensates for the reduced efficiency of vertex merging by weighting its estimates inversely proportionally to their expected variance. And while VCM is asymptotically equivalent to bidirectional path tracing, we showed that it greatly benefits from the initial variance reduction brought by vertex merging.

In Chapter 8 we discussed some important aspects of the practical implementation of our VCM algorithm. We devised a recursive MIS path weight evaluation scheme that avoids redundant computations and minimizes random memory access to path vertices by accumulating partial weights during the random walk based subpath sampling. This makes the evaluation of the MIS weight as efficient as the evaluation of the subpath sampling throughput which already benefits from a similar accumulation scheme. We also discussed the handling of special cases such as delta emission and scattering distributions, and presented improvements and extensions to the VCM algorithm to reduce its memory usage, to automatically determine the merging radius for each path, and to support spectral and motion blur rendering.

While our combined VCM algorithm is much more robust to different lighting conditions than each of its ingredients alone, it does not perform better on paths that cannot be efficiently sampled by any of the techniques at its disposal. These techniques are all based on random walks, i.e. local sampling, and are most efficient for constructing paths that contain at least one diffuse (or not too glossy) vertex where two independently sampled subpaths can be joined. Inter-reflections between highly glossy surfaces, or specular caustics falling on a glossy surface, are thus problematic for VCM, as they do not contain any vertices with smooth scattering distributions. Such paths are often better handled by Markov chain Monte Carlo (MCMC) methods [145, 13, 57, 88, 43, 71], which provide approximate global importance sampling of the measurement contribution of entire paths. An interesting avenue for

future work would therefore be to employ MCMC sampling on top of VCM. This would also improve the algorithm's performance on large and highly occluded environments which are notoriously difficult for methods based on sampling subpaths from light sources.

Our exposition in Chapters 7 and 8 was mostly focused on surface rendering, yet extending our combined VCM algorithm to perform vertex connection and merging in participating media is fairly straightforward. However, our preliminary effort in this direction has revealed that vertex merging can be much less beneficial in media than it is on surfaces. The intuitive explanation is that sampling a photon location in a medium requires an additional propagation distance sampling decision. As a result, the probability for a photon landing inside the merging sphere around an eye subpath vertex can be extremely low, causing volumetric caustics to appear much noisier when seen through a reflection or a refraction than when seen directly from the eye. (Recall from our results in Chapter 7 that directly visible caustics are generally more efficiently sampled via vertex connections to the eye.) We have addressed this problem in a recent follow-up work [84], where we have augmented VCM with a variety of more sophisticated volumetric path sampling techniques based on photon points and beams. In this work we have shown that, just like vertex connection and vertex merging, the strengths and weaknesses of these individual techniques complement each other on scenes containing different kinds of media, which makes the case for combining them into a unified rendering algorithm via MIS. The resulting *unified points, beams, and paths* (UPBP) algorithm employs an extension of MIS that combines estimators of different dimensionality in a meaningful and provably good way. This MIS formulation takes the individual estimators in their original form, without the need to express their path pdfs w.r.t. a common measure as was necessary for our VCM formulation in Chapter 7.

Joint path sampling in participating media

For Monte Carlo estimation of the light transport integral, importance sampling theory postulates that the joint path vertex density should be ideally proportional to the measurement contribution function. However, existing methods use local sampling techniques that generate path vertices one by one, without explicit control over their final joint distribution. In Chapter 9, we revealed that this traditional incremental path construction can lead to high variance and slow error convergence, especially in participating media where geometric singularities are even more pronounced than on surfaces and where anisotropic scattering further increases the variation in the path contribution. We then showed that joint path vertex sampling in media is feasible, by devising global importance sampling techniques for the product of geometry and phase function terms along sequences of path vertices. We obtained conditional vertex sampling pdfs via successive marginalization of a prescribed joint path distribution. We derived simple analytic expressions for unidirectional path construction in isotropic media, and introduced a compact tabulation scheme to handle anisotropic scattering and bidirectional sampling. Our method exploits various geometric symmetries in participating media to make the numerical tabulation of such high-dimensional joint distributions practical. We demonstrated the integration of our path sampling techniques into various rendering algorithms, and observed a variance reduction of up to three orders of magnitude compared to currently used techniques. Our recently proposed UPBP algorithm, discussed above, could also benefit from these new sampling techniques.

Recent related work has shown that the traditional point-based subpath connection approach is in fact a legacy of surface-based rendering, and that significant improvements are possible once path segments are considered as fundamental entities of the light transport simulation process [96, 95]. To improve sampling efficiency, these methods perform subpath connections by considering not only subpath vertices but also entire (semi-infinite) segments. In Chapter 9 we provided a unifying view of these methods as vertex-segment and segment-segment connection techniques that construct con-

nection subpaths involving up to two random sampling decisions. The techniques we then developed further improve efficiency by jointly importance sampling longer connection subpaths involving three sequential random decisions.

We believe that our theoretical and practical contributions create a promising new direction of future work on path sampling in media and global path sampling in general. In particular, it could be possible to extend our techniques to jointly importance sample longer connection subpaths with more than three random decisions by successively marginalizing the corresponding higher-dimensional product distributions. The theoretical challenge is to derive either analytic expressions or proper tabulation for the resulting conditional pdfs. It would be interesting to also include the transmittance term in the joint product densities. Our theoretical framework could be used as a stepping stone for such developments.

Final words

In the times when ray tracing methods were too slow to be practical, artists had to rely on rasterization-based rendering technology that could only reproduce a limited set of effects efficiently and required arduous illumination tweaking, complex asset management, and multi-pass rendering. The selling point for physically-based Monte Carlo methods has been their ability to provide a full global illumination solution in a single rendering pass, without excessive tweaking and post-render image compositing. Now that these methods have become the new standard in production rendering, there is a high demand for efficient and practical algorithms that can deliver high-quality results in an automated way.

It is the author's contemplation that a rendering method should fulfill the following general requirements in order to remain widely usable in practice: (1) it should have few parameters and (2) the accuracy of the produced images should only be bounded by computation time and not by the amount of system memory available. The first requirement ensures that the method is easy to control, which is a necessary condition for robustness – a method that relies on laborious scene-dependent parameter tweaking to be efficient can hardly be qualified as robust. The second requirement ensures that the method can deliver results of arbitrarily high quality with enough investment in computation time. It also increases the likelihood that the method will scale well on future commodity hardware whose computational power has been increasing at a higher rate than the size and speed of its memory.

Based on the results of this thesis, the author sees several promising research directions which can lead to the development of more efficient and more robust rendering algorithms that meet the above requirements. First, the potential of global, or joint, path importance sampling through successive marginalization remains largely unexplored, especially for rendering participating media. Such techniques can be used in ordinary Monte Carlo methods, as we have demonstrated, but also in Markov chain methods – another global sampling approach – to provide better seed paths or to serve as a base for improved path mutation strategies. Second, mixing multiple path sampling techniques, each tailored to a different set of lighting effects, is currently the best-known approach to maintaining efficiency on a wide range of input scenes. Devising specialized techniques for effects that remain difficult to sample, such as inter-reflections between highly glossy surfaces, along with novel combination heuristics, is a natural course of advancing the current state of the art. Finally, while consistency with bounded memory footprint is an important requirement for a practical Monte Carlo algorithm, in the author's view, strictly unbiased consistency is not. After all, most existing biased light transport estimators are, by design, often more efficient than their unbiased alternatives. The author therefore envisions future state-of-the-art practical rendering algorithms to rely on intelligent combinations of various light transport estimators that are consistent but not necessarily unbiased.

Bibliography

- [1] James Arvo. Backward ray tracing. In *ACM SIGGRAPH Course Notes, Developments in Ray Tracing*, 1986. 49
- [2] James Arvo and David Kirk. Particle transport and image synthesis. *Computer Graphics (Proceedings of SIGGRAPH)*, 24(4), 1990. 15, 40
- [3] Philippe Bekaert, Philipp Slusallek, Ronald Cools, Vlastimil Havran, and Hans-Peter Seidel. A custom designed density estimator for light transport. Technical report, Max-Planck-Institut für Informatik, 2003. 78, 84
- [4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006. 133
- [5] David Burke, Abhijeet Ghosh, and Wolfgang Heidrich. Bidirectional importance sampling for direct illumination. In *Proceedings of EGSR*, 2005. 52, 62
- [6] Mike Cammarano and Henrik Wann Jensen. Time dependent photon mapping. In *Proceedings of EGSR*, 2002. 115
- [7] Eva Cerezo, Frederic Perez-Cazorla, Xavier Pueyo, Francisco Seron, and Franccois Sillion. A survey on participating media rendering techniques. *The Visual Computer*, 21, 2005. 119
- [8] Subrahmanyan Chandrasekhar. *Radiative Transfer*. Dover Publications, 1960. 30, 123
- [9] Jiating Chen, Bin Wang, and Jun-Hai Yong. Improved stochastic progressive photon mapping with Metropolis sampling. *Computer Graphics Forum (Proceedings of EGSR)*, 30(4), 2011. 54
- [10] Petrik Clarberg and Tomas Akenine-Möller. Practical product importance sampling for direct illumination. *Computer Graphics Forum (Proceedings of Eurographics)*, 27(2), 2008. 52, 53
- [11] Petrik Clarberg and Tomas Akenine-Möller. Exploiting visibility correlation in direct illumination. *Computer Graphics Forum (Proceedings of EGSR)*, 27(4), 2008. 53, 64
- [12] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 24(3), 2005. 52, 62
- [13] David Cline, Justin Talbot, and Parris Egbert. Energy redistribution path tracing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 24(3), 2005. 54, 149
- [14] David Cline, Parris K. Egbert, Justin F. Talbot, and David L. Cardon. Two stage importance sampling for direct lighting. In *Proceedings of EGSR*, 2006. 52
- [15] David Cline, Daniel Adams, and Parris Egbert. Table-driven adaptive importance sampling. *Computer Graphics Forum (Proceedings of EGSR)*, 2008. 53
- [16] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1), 1982. 39

- [17] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *Computer Graphics (Proceedings of SIGGRAPH)*, 18(3), 1984. 1
- [18] Carsten Dachsbacher, Jaroslav Křivánek, Miloš Hašan, Adam Arbree, Bruce Walter, and Jan Novák. Scalable realistic rendering with many-light methods. In *Eurographics 2013 State of the Art Reports*, 2013. 2
- [19] Holger Dammertz, Daniel Sewtz, Johannes Hanika, and Hendrik Lensch. Edge-avoiding a-trous wavelet transform for fast global illumination filtering. In *Proceedings of High Performance Graphics*, 2010. 53
- [20] Tomáš Davidovič, Iliyan Georgiev, and Philipp Slusallek. Progressive lightcuts for GPU. In *SIGGRAPH Talks*, 2012. 148
- [21] Tomáš Davidovič and Iliyan Georgiev. SmallVCM renderer, 2012. <http://www.smallvcm.com> and <http://github.com/SmallVCM>. 117
- [22] Tomáš Davidovič, Jaroslav Křivánek, Miloš Hašan, Philipp Slusallek, and Kavita Bala. Combining global and local virtual lights for detailed glossy illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 29(6), 2010. 75, 77
- [23] Eugene D'Eon and Geoffrey Irving. A quantized-diffusion model for rendering translucent materials. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 30(4), 2011. 119
- [24] Zhao Dong, Thorsten Grosch, Tobias Ritschel, Jan Kautz, and Hans-Peter Seidel. Real-time indirect illumination with clustered visibility. In *Proceedings of Vision, Modeling, and Visualization*, 2009. 61
- [25] Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 24(3), 2005. 119
- [26] Julie Dorsey, Holly Rushmeier, and Francois Sillion. *Digital Modeling of Material Appearance*. Morgan Kaufmann Publishers Inc., 2008. 28
- [27] A. Dubi, T. Elperin, and H. Rief. On confidence limits and statistical convergence of Monte Carlo point-flux estimators with unbounded variance. *Annals of Nuclear Energy*, 9, 1982. 123, 141
- [28] Philip Dutré. Global illumination compendium, 2003. 39
- [29] Philip Dutré, Eric Lafortune, and Yves Willems. Monte Carlo light tracing with direct computation of pixel intensities. In *Proceedings of Compugraphics*, 1993. 43
- [30] Philip Dutré, Kavita Bala, Philippe Bekaert, and Peter Shirley. *Advanced Global Illumination*. AK Peters Ltd., 2006. 21, 25, 31
- [31] Bartosz Fabianowski. *Interactive Manycore Photon Mapping*. PhD thesis, Trinity College Dublin, 2011. 49
- [32] Shaohua Fan. *Sequential Monte Carlo Methods for Physically Based Rendering*. PhD thesis, University of Wisconsin–Madison, 2006. 15
- [33] Shaohua Fan, Stephen Chenney, and Yu-Chi Lai. Metropolis photon sampling with optional user guidance. In *Proceedings of EGSR*, 2005. 54
- [34] Václav Gassenbauer, Jaroslav Křivánek, and Kadi Bouatouch. Spatial directional radiance caching. *Computer Graphics Forum (Proceedings of EGSR)*, 28(4), 2009. 53

- [35] James Gentle. *Random Number Generation and Monte Carlo Methods*. Springer, 2nd edition, 2004. 12
- [36] Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., 1995. 28
- [37] Ralf Habel, Per H. Christensen, and Wojciech Jarosz. Photon beam diffusion: A hybrid Monte Carlo method for subsurface scattering. *Computer Graphics Forum (Proceedings of EGSR)*, 32(4), 2013. 119
- [38] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 28(5), 2009. 50, 51, 92
- [39] Toshiya Hachisuka and Henrik Wann Jensen. Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics*, 30(5), 2011. 51, 54
- [40] Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 27(3), 2008. 15
- [41] Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Transaction on Graphics (Proceedings of SIGGRAPH Asia)*, 30(3), 2008. 50, 77, 82
- [42] Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A path space extension for robust light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 31(6), 2012. 95, 96
- [43] Toshiya Hachisuka, Anton Kaplanyan, and Carsten Dachbacher. Multiplexed Metropolis light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 33(4), 2014. 149
- [44] John Hammersley and David Handscomb. *Monte Carlo Methods*. Chapman and Hall, 1964. 5
- [45] Vlastimil Havran. *Heuristic Ray Shooting Algorithms*. PhD thesis, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, 2000. 39
- [46] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Matrix row-column sampling for the many-light problem. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, volume 26, 2007. 55, 57
- [47] Miloš Hašan, Jaroslav Krivánek, Bruce Walter, and Kavita Bala. Virtual spherical lights for many-light rendering of glossy scenes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 28(5), 2009. 75, 77
- [48] Eugene Hecht and Alfred Zajac. *Optics*. Reading, MA: Addison-Wesley Publishing Company, 1979. 21
- [49] Paul Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics (Proceedings of SIGGRAPH)*, 24(4), 1990. 32
- [50] Robert Herzog, Karol Myszkowski, and Hans-Peter Seidel. Anisotropic radiance-cache splatting for efficiently computing high-quality global illumination with lightcuts. *Computer Graphics Forum (Proceedings of Eurographics)*, 28(2), 2009. 53, 61, 148
- [51] Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2), 1995. 69

- [52] Heinrich Hey and Werner Purgathofer. Importance sampling with hemispherical particle footprints. Technical Report TR-186-2-01-05, Vienna University of Technology, 2001. 52
- [53] Homan Igehy. Tracing ray differentials. In *Proceedings of SIGGRAPH*, 1999. 49, 114
- [54] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics (Proceedings of SIGGRAPH)*, 20(4), 1986. 30
- [55] Wenzel Jakob. Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>. 92
- [56] Wenzel Jakob. *Light Transport on Path-Space Manifolds*. PhD thesis, Cornell University, 2013. 31
- [57] Wenzel Jakob and Steve Marschner. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 31(4), 2012. 28, 54, 149
- [58] Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum (Proceedings of Eurographics)*, 27(2), 2008. 31, 49, 53, 123
- [59] Wojciech Jarosz, Derek Nowrouzezahrai, Iman Sadeghi, and Henrik Wann Jensen. A comprehensive theory of volumetric radiance estimation using photon points and beams. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 30(1), 2011. 54, 123
- [60] Wojciech Jarosz, Derek Nowrouzezahrai, Robert Thomas, Peter-Pike Sloan, and Matthias Zwicker. Progressive photon beams. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 30(6), 2011. 54
- [61] Henrik Wann Jensen. Importance driven path tracing using the photon map. In *Proceedings of EGWR*, 1995. 52
- [62] Henrik Wann Jensen. Global illumination using photon maps. In *Proceedings of EGWR*, 1996. 49
- [63] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., 2001. 2, 50, 77
- [64] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of SIGGRAPH*, 1998. 38, 53, 83
- [65] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of SIGGRAPH*, 2001. 119
- [66] James T. Kajiya. The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH)*, 20(4), 1986. 1, 30, 36, 43, 92
- [67] H.J. Kalli and E.D. Cashwell. Evaluation of three Monte Carlo estimation schemes for flux at a point. Technical report, Los Alamos Scientific Lab, 1977. 47, 123, 127
- [68] Malvin Kalos. On the estimation of flux at a point by Monte Carlo. *Nuclear Science and Engineering*, 16, 1963. 119, 122, 123, 129, 130, 141
- [69] Malvin Kalos and Paula Whitlock. *The Monte Carlo Method, Volume 1: Basics*. John Wiley and Sons, 1986. 5
- [70] Anton Kaplanyan and Carsten Dachsbacher. Path space regularization for holistic and robust light transport. *Computer Graphics Forum (Proceedings of Eurographics)*, 32(2), 2013. 116

- [71] Anton Kaplanyan, Johannes Hanika, and Carsten Dachbacher. The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 33(4), 2014. 149
- [72] Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum (Proceedings of Eurographics)*, 21(3), 2002. 54, 92
- [73] Alexander Keller. Instant radiosity. In *Proceedings of SIGGRAPH*, 1997. 43, 55, 56, 58, 61, 74
- [74] Alexander Keller and Ingo Wald. Efficient importance sampling techniques for the photon map. In *Proceedings of Vision, Modeling, and Visualization*, 2000. 55, 56
- [75] David Kirk and James Arvo. Unbiased sampling techniques for image synthesis. *Computer Graphics (Proceedings of SIGGRAPH)*, 25(4), 1991. 15
- [76] Claude Knaus and Matthias Zwicker. Progressive photon mapping: A probabilistic approach. *ACM Transactions on Graphics*, 30(3), 2011. 50, 51, 53, 54, 77, 82, 88, 89, 92
- [77] Thomas Kollig and Alexander Keller. Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*, 2002. 46
- [78] Thomas Kollig and Alexander Keller. Illumination in the presence of weak singularities. *Monte Carlo and Quasi-Monte Carlo Methods*, 2004. 45, 75, 77
- [79] Janne Kontkanen, Jussi Räsänen, and Alexander Keller. Irradiance filtering for Monte Carlo ray tracing. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*, 2004. 53
- [80] Christopher Kulla and Marcos Fajardo. Importance sampling techniques for path tracing in participating media. *Computer Graphics Forum (Proceedings of EGSR)*, 31(4), 2012. 38, 47, 122, 123, 124, 127, 130, 139
- [81] Jaroslav Křivánek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. Radiance caching for efficient global illumination computation. *IEEE TVCG*, 11(5), 2005. 53
- [82] Jaroslav Křivánek, Kadi Bouatouch, Sumanta Pattanaik, and Jiří Žára. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Proceedings of EGSR*, 2006. 53
- [83] Jaroslav Křivánek, James A. Ferwerda, and Kavita Bala. Effects of global illumination approximations on material appearance. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 29(4), 2010. 77
- [84] Jaroslav Křivánek, Iliyan Georgiev, Toshiya Hachisuka, Petr Vévoda, Martin Šik, Derek Nowrouzezahrai, and Wojciech Jarosz. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 33(4), 2014. 150
- [85] Eric Lafortune and Yves Willems. Bi-directional path tracing. In *Proceedings of Compugraphics*, 1993. 45, 77, 78
- [86] Eric Lafortune and Yves Willems. Using the modified Phong BRDF for physically based rendering. Technical report, Department of Computer Science, Katholieke Universiteit Leuven, 1994. 73
- [87] Eric Lafortune and Yves Willems. Rendering participating media with bidirectional path tracing. In *Proceedings of EGWR*, 1996. 38

- [88] Jaakko Lehtinen, Tero Karras, Samuli Laine, Miika Aittala, Frédo Durand, and Timo Aila. Gradient-domain Metropolis light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 32(4), 2013. 149
- [89] Tzu-Mao Li, Yu-Ting Wu, and Yung-Yu Chuang. SURE-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia)*, 31(6), 2012. 53
- [90] Ivan Lux and Laszlo Koblinger. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press, 1991. 36
- [91] Rafal Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 30(4), 2011. 92
- [92] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1953. 14
- [93] Don P. Mitchell. Consequences of stratified sampling in graphics. In *Proceedings of SIGGRAPH*, 1996. 15
- [94] Bochang Moon, Jong Yun Jun, JongHyeob Lee, Kunho Kim, Toshiya Hachisuka, and Sung-Eui Yoon. Robust image denoising using a virtual flash image for Monte Carlo ray tracing. *Computer Graphics Forum*, 32(1), 2013. 53
- [95] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Progressive virtual beam lights. *Computer Graphics Forum (Proceedings of EGSR)*, 31(4), 2012. 54, 77, 150
- [96] Jan Novák, Derek Nowrouzezahrai, Carsten Dachsbacher, and Wojciech Jarosz. Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 31(4), 2012. 47, 48, 54, 77, 123, 124, 131, 132, 135, 136, 137, 138, 139, 143, 144, 150
- [97] Illuminating Engineering Society of North America Computer Committee. IES recommended standard file format for electronic transfer of photometric data. *IES LM-63-1986*, 1986. 25
- [98] Noboru Ohta and Alan R. Alan Robert Robertson. *Colorimetry: Fundamentals and Applications*. Wiley-IS&T series in imaging science and technology. J. Wiley, 2005. 24
- [99] James Painter and Kenneth Sloan. Antialiased ray tracing by adaptive progressive refinement. *Computer Graphics (Proceedings of SIGGRAPH)*, 23(3), 1989. 15
- [100] Mark Pauly, Thomas Kollig, and Alexander Keller. Metropolis light transport for participating media. In *Proceedings of EGWR*, 2000. 31, 38, 54
- [101] Vincent Pegoraro and Steven Parker. An analytical solution to single scattering in homogeneous participating media. *Computer Graphics Forum (Proceedings of Eurographics)*, 28(2), 2009. 119
- [102] Vincent Pegoraro, Carson Brownlee, Peter Shirley, and Steven Parker. Towards interactive global illumination effects via sequential Monte Carlo adaptation. In *Proceedings of IEEE Symposium on Interactive Ray Tracing*, 2008. 15, 52
- [103] Vincent Pegoraro, Ingo Wald, and Steven Parker. Sequential Monte Carlo adaptation in low-anisotropy participating media. *Computer Graphics Forum (Proceedings of EGSR)*, 27(4), 2008. 52

- [104] Vincent Pegoraro, Mathias Schott, and Steven Parker. An analytical approach to single scattering for anisotropic media and light distributions. In *Proceedings of Graphics Interface*, 2009. 119
- [105] Vincent Pegoraro, Mathias Schott, and Steven Parker. A closed-form solution to single scattering for general phase functions and light distributions. *Computer Graphics Forum (Proceedings of EGSR)*, 29(4), 2010. 119
- [106] Vincent Pegoraro, Mathias Schott, and Philipp Slusallek. A mathematical framework for efficient closed-form single scattering. In *Proceedings of Graphics Interface*, 2011. 119
- [107] Ingmar Peter and Georg Pietrek. Importance driven construction of photon maps. In *Proceedings of EGWR*, 1998. 55
- [108] Matt Pharr. Extended photon map implementation in PBRT, 2005. 52
- [109] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2nd edition, 2010. 13, 21, 24, 28, 39, 49, 111, 126
- [110] Stefan Popov. *Algorithms and Data Structures for Interactive Ray Tracing on Commodity Hardware*. PhD thesis, Saarland University, 2012. 39
- [111] Simon Premože, Michael Ashikhmin, and Peter S. Shirley. Path integration for light transport in volumes. In *Proceedings of EGSR*, 2003. 119
- [112] Simon Premože, Michael Ashikhmin, Ravi Ramamoorthi, and Shree Nayar. Practical rendering of multiple scattering effects in participating media. In *Proceedings of EGSR*, 2004. 119
- [113] A. Raab and G. Beikert. Two new Monte Carlo methods for point flux estimation. *Computer Physics Communications*, 123, 1999. 123, 141
- [114] Matthias Raab, Daniel Seibert, and Alexander Keller. Unbiased global illumination with participating media. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*. 2008. 38
- [115] H. Rief, A. Dubi, and T. Elperin. Track length estimation applied to point detector. *Nuclear Science and Engineering*, 87, 1984. 47, 123, 127, 128, 141
- [116] Tobias Ritschel, Thorsten Grosch, Min H. Kim, Hans-Peter Seidel, Carsten Dachsbacher, and Jan Kautz. Imperfect shadow maps for efficient computation of indirect illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 27(5), 2008. 61
- [117] Tobias Ritschel, Elmar Eisemann, Inwoo Ha, James D.K. Kim, and Hans-Peter Seidel. Making imperfect shadow maps view-adaptive: High-quality global illumination in large dynamic scenes. *Computer Graphics Forum (Proceedings of EGSR)*, 30(8), 2011. 61
- [118] Fabrice Rousselle, Petrik Clarberg, Luc Leblanc, Victor Ostromoukhov, and Pierre Poulin. Efficient product sampling using hierarchical thresholding. *The Visual Computer (Proceedings of CGI)*, 24(7-9), 2008. 52
- [119] Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. Adaptive sampling and reconstruction using greedy error minimization. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 30(6), 2011. 53
- [120] Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 31(6), 2012. 53
- [121] Fabrice Rousselle, Marco Manzi, and Matthias Zwicker. Robust denoising using feature and color information. *Computer Graphics Forum*, 32(7), 2013. 53

- [122] Holly Rushmeier. *Realistic image synthesis for scenes with radiatively participating media*. PhD thesis, Cornell University, 1988. 38
- [123] Benjamin Segovia, Jean Claude Iehl, Richard Mitanchey, and Bernard Péroche. Bidirectional instant radiosity. In *Proceedings of EGSR*, 2006. 55, 57, 59
- [124] Benjamin Segovia, Jean-Claude Iehl, and Bernard Péroche. Metropolis instant radiosity. *Computer Graphics Forum (Proceedings of Eurographics)*, 26(3), 2007. 54, 55, 57, 59
- [125] Pradeep Sen and Soheil Darabi. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Transactions on Graphics*, 31(3), 2012. 53
- [126] Peter Shirley, Bretton Wade, Philip M. Hubbard, David Zareski, Bruce Walter, and Donald P. Greenberg. Global illumination via density estimation. In *Proceedings of EGWR*, 1995. 49
- [127] Peter Shirley, C. Wang, and K. Zimmerman. Monte Carlo techniques for direct lighting calculations. *ACM Transactions on Graphics*, 15(1), 1996. 2
- [128] Jerome Spanier. Two pairs of families of estimators for transport problems. *SIAM Journal on Applied Mathematics*, 14(4), 1966. 123
- [129] Jerome Spanier and Ely Meyer Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, 1969. 5, 36, 123
- [130] Jos Stam. Multiple scattering as a diffusion process. In *Proceedings of EGWR*, 1995. 119
- [131] Herbert Steinberg and Malvin Kalos. Bounded estimators for flux at a point in Monte Carlo. *Nuclear Science and Engineering*, 44, 1971. 123
- [132] Joshua Steinhurst and Anselmo Lastra. Global importance sampling of glossy surfaces using the photon map. In *Proceedings of IEEE Symposium on Interactive Ray Tracing*, 2006. 52
- [133] Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar. A practical analytic single scattering model for real time rendering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 24(3), 2005. 119
- [134] Xin Sun, Kun Zhou, Stephen Lin, and Baining Guo. Line space gathering for single scattering in large scenes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 29(4), 2010. 53, 123
- [135] Frank Suykens and Yves Willems. Adaptive filtering of progressive Monte Carlo image rendering. In *Proceedings of WSCG*, 2000. 53
- [136] László Szirmay-Kalos, Balázs Tóth, and Milán Magdics. Free path sampling in high resolution inhomogeneous participating media. *Computer Graphics Forum*, 30(1), 2011. 38
- [137] Justin Talbot, David Cline, and Parris K. Egbert. Importance resampling for global illumination. In *Proceedings of EGSR*, 2005. 52
- [138] Jerry Tessororf. Radiative transfer as a sum over paths. *Physical Review A*, 35, 1987. 119
- [139] Yusuke Tokuyoshi. Photon density estimation using multiple importance sampling. In *SIGGRAPH Asia Posters*, 2009. 78
- [140] 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. 133
- [141] Dietger van Antwerpen. Recursive MIS computation for streaming BDPT on the GPU. Technical report, Delft University of Technology, 2011. 101, 112, 117

- [142] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 3, 18, 28, 31, 33, 37, 39, 41, 49, 54, 67, 69, 83, 87, 88, 89, 91, 101, 104, 106
- [143] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Proceedings of EGWR*, 1994. 45, 77, 78, 92
- [144] Eric Veach and Leonidas Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of SIGGRAPH*, 1995. 17, 45, 54, 77, 84
- [145] Eric Veach and Leonidas Guibas. Metropolis light transport. In *Proceedings of SIGGRAPH*, 1997. 54, 92, 149
- [146] Jiří Vorba. Bidirectional photon mapping. In *Proceedings of CESC*, 2011. 77, 78, 80, 113
- [147] Ingo Wald. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004. 39
- [148] Ingo Wald, Carsten Benthin, and Philipp Slusallek. Interactive global illumination in complex and highly occluded environments. In *Proceedings of EGWR*, 2003. 55, 57, 60
- [149] Bruce Walter. *Density Estimation Techniques for Global Illumination*. PhD thesis, Cornell University, 1998. 49
- [150] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. Lightcuts: A scalable approach to illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 24(3), 2005. 61, 148
- [151] Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. Multidimensional lightcuts. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 25(3), 2006. 61
- [152] Bruce Walter, Shuang Zhao, Nicolas Holzschuch, and Kavita Bala. Single scattering in refractive media with triangle mesh boundaries. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 28(3), 2009. 119
- [153] Bruce Walter, Pramook Khungurn, and Kavita Bala. Bidirectional lightcuts. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 31(4), 2012. 77
- [154] Rui Wang and Oskar Akerlund. Bidirectional importance sampling for unstructured illumination. *Computer Graphics Forum (Proceedings of Eurographics)*, 28(2), 2009. 52, 61, 62
- [155] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004. 92
- [156] Gregory Ward and Paul Heckbert. Irradiance gradients. In *Proceedings of EGWR*, 1992. 53, 148
- [157] Gregory Ward, Francis Rubinstein, and Robert Clear. A ray tracing solution for diffuse inter-reflection. *Computer Graphics (Proceedings of SIGGRAPH)*, 22(4), 1988. 53, 64
- [158] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6), 1980. 1
- [159] E.R. Woodcock, T. Murphy, P.J. Hemmings, and Longworth T.C. Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Applications of Computing Methods to Reactor Problems*, 1965. 38
- [160] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 32(6), 2013. 141

- [161] Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 29(6), 2010. 38
- [162] Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum (Proceedings of Eurographics)*, 34(2), 2015. 53