

**Robust and Efficient
Monte Carlo Light Transport Simulation
using Regularizations
and the Half Vector Integration Domain**

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften

der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Anton S. Kaplanyan

aus Nevinnomyssk, Rußische Föderation

Tag der mündlichen Prüfung: 23. Januar 2017

Erster Gutachter: Prof. Dr.-Ing. Carsten Dachsbacher

Zweiter Gutachter: Prof. Dr. Thorsten Grosch



This document is licensed under the Creative Commons Attribution –
Share Alike 4.0 International License (CC BY-SA 4.0):
<https://creativecommons.org/licenses/by-sa/4.0/deed.de>

© Copyright 2017 by Anton S. Kaplanyan
All Rights Reserved

Vita and Publications

- 2002 – 2007 Diploma in Applied Mathematics
Moscow Institute of Electronic Technology, Russia
- 2008 – 2011 Sr. Graphics Engineer, Lead Researcher
Crytek GmbH, Germany
- 2011 – 2013 Research Assistant
Karlsruhe Institute of Technology, Germany
-

- Hachisuka, T., Kaplanyan, A. S., and Dachsbacher, C. *Multiplexed Metropolis light transport*. ACM Trans. on Graphics (Proc. SIGGRAPH), 33(4), 2014.
- Hanika, J., Kaplanyan, A., and Dachsbacher, C. *Improved half vector space light transport*. Computer Graphics Forum (Proc. Eurographics Symposium on Rendering), 34(4):65–74, 2015.
- Kaplanyan, A. S. and Dachsbacher, C. *Cascaded light propagation volumes for real-time indirect illumination*. In Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pages 99–107. 2010.
- Kaplanyan, A. S. and Dachsbacher, C. *Adaptive progressive photon mapping*. ACM Trans. on Graphics, 32(2):16:1–16:13, 2013a.
- Kaplanyan, A. S. and Dachsbacher, C. *Path space regularization for holistic and robust light transport*. Computer Graphics Forum (Proc. of Eurographics), 32(2):63–72, 2013b.
- Kaplanyan, A. S., Hanika, J., and Dachsbacher, C. *The natural-constraint representation of the path space for efficient light transport simulation*. ACM Trans. on Graphics (Proc. SIGGRAPH), 33(4), 2014.
- Krivánek, J., Fajardo, M., Christensen, P. H., Tabellion, E., Bunnell, M., Larsson, D., and Kaplanyan, A. S. *Global illumination across industries*. In ACM SIGGRAPH Courses. 2010.
- Prutkin, R., Kaplanyan, A. S., and Dachsbacher, C. *Reflective Shadow Map Clustering for Real-Time Global Illumination*. In Eurographics - Short Papers. 2012.
- Reiner, T., Kaplanyan, A. S., Reinhard, M., and Dachsbacher, C. *Selective inspection and interactive visualization of light transport in virtual scenes*. Computer Graphics Forum (Proc. of Eurographics), 31(2):711–718, 2012.
- Schmidt, T.-W., Novak, J., Meng, J., Kaplanyan, A. S., Reiner, T., Nowrouzezahrai, D., and Dachsbacher, C. *Path-space manipulation of physically-based light transport*. ACM Trans. on Graphics (Proc. SIGGRAPH), 32(4):129:1–129:11, 2013.
- Tatarchuk, N., Chen, H., Evans, A., Kaplanyan, A. S., Moore, J., Jeffries, D., Yang, J., and Engel, W. *Advances in real-time rendering in 3d graphics and games*. In ACM SIGGRAPH Courses. 2009.
- Weber, C., Kaplanyan, A., Stamminger, M., and Dachsbacher, C. *Interactive direct volume rendering with many-light methods and transmittance caching*. In International Symposium on Vision, Modeling and Visualization, pages 195–202. 2013.

Acknowledgments

The present dissertation were only possible due to the help from many people. I would like to thank all of them for supporting, encouraging, and otherwise helping me during my computer graphics research journey at the Karlsruhe Institute of Technology.

My deepest acknowledgment goes to my advisor Carsten Dachsbacher for arousing my interest in the fundamental computer graphics and inspiring me to pursue the academic graphics research. He always encourages me on my research, suggesting interesting directions, while also letting the projects go their own often very interesting way. Last but not least, thanks a lot for teaching and playing Fußball (kicker), as well as organizing the weekly grills in the summer and sending us to HiGraphics in the winter. It was a lot of fun!

I would like to thank my colleagues at KIT Computer Graphics group for inspiring daily in-depth conversations about graphics and rendering, including Jan Novák, Gábor Liptor, Thorsten Schmidt, Thomas Engelhardt, Florian Simon, Johannes Meng, and many others. I am certainly missing our coffee-break discussions about the current problems of rendering. I would gladly do a few more cooking sessions with you guys! It was especially pleasant to have discussions about many rendering-related topics with Johannes Hanika. Special thanks to Johannes Hanika and Christoph Schied for helping with the translation of the abstract and the summary, and to Diana Kheil for helping to cognize German language, for clarifying an infinite number of questions and technicalities, and for organizing our events, especially Christmas parties and Ausflugs!

I am grateful to Toshiya Hachisuka for fruitful and stimulating collaborations on light transport projects, Jaroslav Křivánek and Alexander Keller for getting me involved in many interesting rendering-related events. It was fun talking light transport to you. Next, I would like to thank László Györfi and Harro Walk for a thorough introduction to the state-of-the-art statistical recursive estimators and bandwidth selectors. Peter-Pike Sloan and Peter Shirley, thank you for your early helpful writ-

ing suggestions. I would also like to thank Wenzel Jakob for creating and sharing a wonderful research platform, Mitsuba renderer, as well as for inspiring discussions on half vector space light transport.

I am obliged to Martin Mittring for his support of my early research work, interesting discussions on real-time and offline graphics, as well as useful practical suggestions, and to many people from game and visual effects industries on useful remarks and encouraging insights.

Next, I would like to thank the TDs and Volker Helzle from Filmakademie Baden-Württemberg as well as Christophe Hery from Pixar Animation Studios for their valuable feedback and early support for the artistic light editing tools.

Last but not least, it is hard to overestimate my gratitude to my wife Asia. I am deeply indebted to her for her invaluable and continuous support in my research throughout these years. She was always inspiring and motivating me by her encouragements and illuminations to strive for impressive results and new horizons in my research.

To Asia and Diana

Abstract

Computer-generated imagery has become a central point in large and quickly growing areas such as computer animation, visual effects, architecture, high-quality visualization, and product design. A common requirement to all these areas is the efficient simulation of *light transport* from the light source to the sensor for image synthesis. This is a challenging task, as light can travel along all kinds of paths, and some paths are hard to find when computing the result numerically. The process of light propagation in a virtual scene is described by a path integral equation. This path integral is defined over the *path space*, the space of all possible light paths, on which photons travel from a light source to a virtual sensor. Monte Carlo approaches are employed for computing the path integral, where paths are stochastically generated (sampled). The integration, to compute an image of a virtual scene, in this space is challenging because of the mathematical properties of the integrand. In addition, the demand for both image quality and quantity requires shorter image rendering times, while imposing higher scene complexity at the same time.

We provide several solutions to the problems arising with difficult light paths, especially these involving specular and glossy materials, as well as improve the rendering convergence with difficult illumination effects.

Some physical phenomena, such as perfectly specular reflection, cause singularities in the path integral, and thus are hard or even impossible to sample. We propose an approach to adaptively smooth difficult components of light transport, which makes them samplable while still ensuring convergence to the exact solution. It uses regularization for paths which cannot be sampled in an unbiased way. To introduce as little bias as possible, we selectively regularize individual interactions along paths, and also derive the regularization consistency conditions. Additionally, in order to control the bias introduced by kernel smoothing, which is another means to sample complex light transport, we developed an adaptive bandwidth selection method. We have also analyzed the convergence rate for kernel smoothing methods in light transport.

On the other hand, even not perfectly specular, i.e., glossy, materials can be as difficult to sample. We propose a new way for representing light paths for more efficient exploration of the path space. Based on this, we propose a new mutation strategy, to be used with Markov chain Monte Carlo methods such as Metropolis light transport, which is well-suited for all configurations of surface scattering.

A different yet related problem is the creation of virtual scenes with photorealistic lighting. It can be challenging because it often requires interactive feedback and edits to the scene that are propagated consistently. To alleviate this process, we propose a set of visualization and manipulation tools, which allow the designers to grasp virtual illumination and to alter the physical laws of light transport for convenient illumination setup and analysis, as well as for achieving artistic goals.

Kurzfassung

Computergenerierte Bilder sind ein wichtiger Bestandteil in großen und schnell wachsenden Gebieten wie Computeranimation, visuelle Effekte, Architektur, qualitativ hochwertige Visualisierung, und Produktdesign. Eine gemeinsame Anforderung all dieser Gebiete ist die effiziente Simulation des *Lichttransports*. Dies ist eine herausfordernde Aufgabe, weil Licht entlang unterschiedlichster Pfade transportiert werden kann, und numerische Simulationen Schwierigkeiten haben, alle Pfade gleichermaßen zu berücksichtigen. Der Prozess der Lichtausbreitung in einer virtuellen Szene wird von der Pfadintegralgleichung beschrieben. Dieses Pfadintegral ist über dem *Pfadraum* definiert, dem Raum aller möglichen Lichtpfade entlang welcher Photonen von einer Lichtquelle zu einem virtuellen Sensor gelangen können. Zur Berechnung des Pfadintegrals werden Monte Carlo Ansätze benutzt, welche Pfade stochastisch erzeugen (abtasten). Die Integration in diesem Raum, die ein Bild von einer virtuellen Szene berechnet, ist aufgrund der mathematischen Eigenschaften des Integranden aufwändig. Außerdem macht die wachsende Nachfrage nach Bildqualität und -quantität kürzere Bildberechnungszeiten nötig, während die Szenenkomplexität stetig steigt.

In dieser Arbeit werden mehrere Lösungen vorgestellt, die insbesondere die Herausforderungen schwieriger Lichtpfade, z.B. bei spekularen und glänzenden Materialien, adressieren und die Konvergenz bei der Berechnung verbessern.

Manche physikalischen Lichttransportphänomene, wie perfekte spekulare Reflexionen, verursachen Singularitäten im Pfadintegral und sind deshalb schwierig oder gar unmöglich mit Monte Carlo Methoden abzutasten. Wir schlagen einen Ansatz vor, diese diffizilen Komponenten des Lichttransports adaptiv zu glätten (Regularisieren), was diese abtastbar macht und gleichzeitig die Konvergenz zu der exakten Lösung sicherstellt. Hierfür wird eine Regularisierung für Pfade, die nicht erwartungstreu abgetastet werden können, vorgestellt. Um so wenig wie möglich Bias einzuführen, regularisieren wir selektiv individuelle Interaktionen entlang der Pfa-

de, und leiten außerdem Konsistenzbedingungen für die Regularisierung her. Um den systematischen Fehler der Kernel-Schätzung zu kontrollieren entwickelten wir eine adaptive Methode um die Bandbreite zu selektieren. Wir analysierten die Konvergenzrate von Kernel-Schätzung im Lichttransport.

Andererseits können sogar nicht völlig spekulare (glänzende) Materialien schwierig abzutasten sein. Wir schlagen eine neue Art Lichttransportpfade zu repräsentieren vor, um den Pfadraum effizienter zu explorieren. Basierend darauf schlagen wir eine neue Mutationsstrategie vor, die mit Markoffketten Monte Carlo Methoden wie Metropolis Lichttransport benutzt wird und für alle Konfigurationen von Oberflächenstreuung gut geeignet ist.

Ein anderes und doch verwandtes Problem ist die Erzeugung von virtuellen Szenen mit photorealistischer Szenen und der Entwurf deren Be- bzw. Ausleuchtung. Dies kann herausfordernd sein weil es oft interaktiver Rückmeldung bedarf und die Änderungen sich konsistent auf die Szene auswirken müssen. Um diesen Prozess zu erleichtern schlagen wir Visualisierungs- und Manipulationswerkzeugen vor, welche es einem Designer erlauben, die virtuelle Beleuchtung zu verstehen und die physikalischen Gesetze des Lichttransports selektiv zu verletzen, um künstlerische Vorgaben zu erreichen.

Summary

If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is.

— JOHN LOUIS VON NEUMANN (1903–1957)

High-quality image synthesis is essential for predictive rendering in architecture, product design, and movie productions. An essential part of the photorealistic image synthesis is an accurate global illumination computation, which we target in this thesis. Global illumination provides all parts of illumination in the image and is based on a physically-based computation of light transport for realistic image synthesis. Veach [Veach 1998] introduced the path integral framework for light transport along with important stochastic integration methods based on Monte Carlo and Markov chain Monte Carlo methods for variance reduction and sampling in the space of light transport paths, the *path space*.

Light transport simulation can be extremely challenging depending on the materials and light sources in a scene. Modern virtual production with realistic materials and light sources, such as specular materials and Light-Emitting Diodes (LEDs), complicate the light transport simulation by posing the (near-)singularities in the integrand. Light transport between scene surfaces is traditionally considered separately from light transport in participating media due to inherently different properties. The focus of the present dissertation is on efficient methods for difficult light transport scenarios between surfaces.

Light Transport Problem

The path integral formulation [Veach 1998] expresses the problem of light transport in the form of a single integral over all possible light paths from light sources to the

virtual sensor (see Figure 1).

Since light can interact many times with the scene surfaces, the light paths can become arbitrary long and turn the integration into a high-dimensional problem, which is nowadays solved with Monte Carlo or Markov chain Monte Carlo stochastic integration methods.

The surface scattering is described by a Bidirectional Scattering Distribution Function (BSDF). Depending on the surface roughness, a BSDF can have very narrow directional lobes towards the perfect reflection or refraction direction, or even degenerate to a Dirac delta distribution for specular materials, such as perfect mirror, water, or smooth glass.

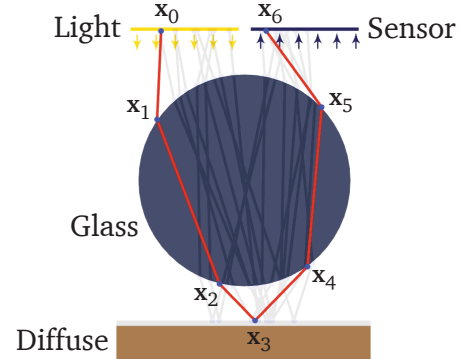


Figure 1: Light transport simulation is based on sampling light paths (light gray piecewise paths) from the light source (top left) going through the virtual scene (glass sphere and diffuse plane on the bottom) and finally reaching the sensor (top right). One such light path is highlighted in red.

Sampling of Rare Light Paths

There are configurations of scene geometry, materials, and lighting that cannot be handled with existing stochastic methods of sampling light paths. In particular, these include specular caustics from point light sources seen through a mirror or multiple consecutive specular interactions (see Figure 2, left). It is impossible to sample such light paths using unbiased stochastic methods [Veach 1998, Theorem 8.2], because they contain singularities at every path edge. We propose a new regularization framework [Kaplanyan and Dachsbacher 2013b] for finding such paths. First, we identify the singularities that prevent a path from being sampled. Then, we propose a selective regularization framework for such paths, and show how the existing methods with regularization can be made consistent, i.e., converging to the correct solution.

The kernel bandwidth controls the balance between the introduced amount of

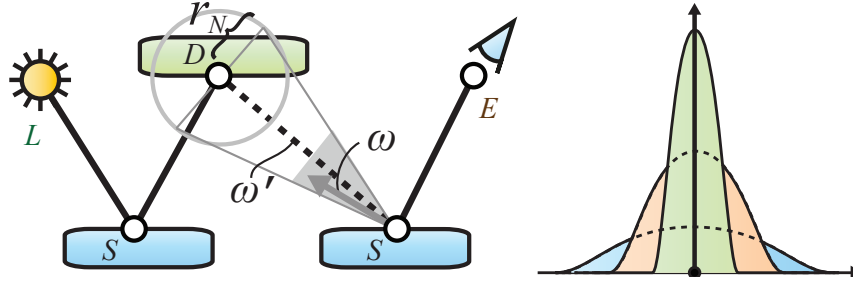


Figure 2: Left: We propose a regularization approach based on mollification in order to sample rare events. Here, the surface with diffuse material (D) is depicted in green, while the specular surfaces (S), described by Dirac delta, are in cyan. Right: A simple mollification example of 1-dimensional Dirac delta distribution with smooth Gaussian distribution. By tending the variance to zero, the Gaussian distribution approaches a Delta distribution in the limit.

smoothing (bias) and the reduction of image noise (variance) in kernel estimation methods, such as photon mapping. We show how this balance can be efficiently maintained by properly selecting the bandwidth [Kaplanyan and Dachsbacher 2013a]. For that, we estimate a local variability of the illumination around every light path. Based on this estimate, we select the bandwidth such that it always keeps bias and variance in an optimal balance, thus achieving faster rendering times with smaller image error. In addition, we also analyze the asymptotic convergence of kernel estimation-based methods and showed that it is slower than that of the unbiased Monte Carlo methods.

Generalized Coordinates for Light Transport

Specular or glossy paths are difficult to sample. However, a dense sampling of such difficult phenomena is required to obtain a converged image. We propose *half vector space light transport* [Kaplanyan et al. 2014] (HSLT), which is a new representation of the path space, where a light transport path is parameterized by its end points and halfway vectors (or microfacets) at every interaction in between (see Figure 3), as opposed to the traditional representation with a set of vertices. This novel representation is motivated by the local observance of Fermat’s principle for reflections and refractions and has multiple beneficial properties, such as the decomposition of the path integral into weakly-dependent two-dimensional in-

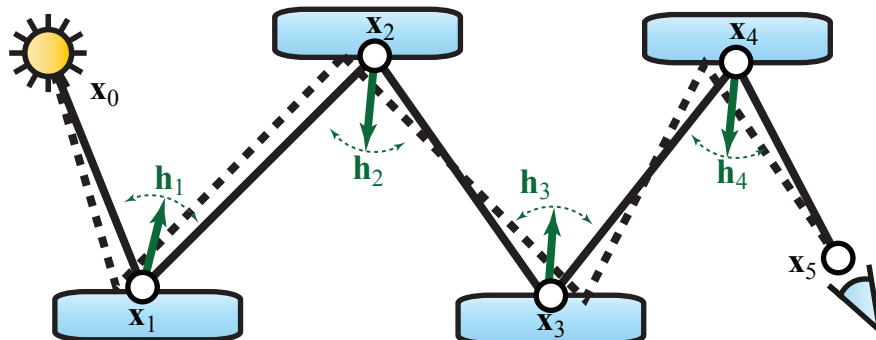


Figure 3: In order to explore difficult paths better, we propose a new generalized coordinate system. It represent a light path as two fixed end points and half-way vectors in between them.

tegrals. These properties make this representation a convenient domain for light paths on surfaces. Finally, due to these properties, it allows to efficiently explore narrow illumination features, such as multiple glossy interactions.

Visualization and Manipulation for Light Transport

Even though light transport methods are designed to efficiently render images, the understanding of the rendered phenomena is also important. However, with many phenomena, such as prominent glossy inter-reflections, the human perception has difficulties comprehending the nature of the observed appearance. We developed multiple visualization approaches for light transport [Reiner et al. 2012] to help grasping the illumination in different situations, ranging from soft diffuse illumination to sharp light fields, such as strong caustics. As the next step, after the parts of the illumination can be well understood, we also provide the artists, e.g., in a movie production, with the freedom of altering the light propagation process in a non-physical manner [Schmidt et al. 2013]. This flexibility is important for achieving certain desired appearance and design goals in rendered images.

Zusammenfassung

If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is.

— JOHN LOUIS VON NEUMANN (1903–1957)

Die Synthese qualitativ hochwertiger und photorealistischer Bilder ist essentiell für die sogenannte Prädiktive Bildsynthese in Architektur, Produktdesign und Filmproduktionen. Eine wichtige Komponente dabei ist die akkurate, physikalisch-basierte Berechnung des Lichttransports, sogenannte globale Beleuchtung, die Gegenstand dieser Arbeit ist. Veach [Veach 1998] führte das Pfadintegralrahmenwerk für Lichttransport ein, zusammen mit wichtigen stochastischen Integrationsmethoden, die auf Monte Carlo und Markoffketten Monte Carlo Methoden für Varianzreduktion und Abtastung im Raum der Lichttransportpfade, dem *Pfadraum*, basieren.

Die Simulation der Lichtausbreitung ist, je nach Materialien, Lichtquellen und Geometrie einer Szene, ein sehr herausforderndes Problem. Heutige Anwendungen, z.B. in Filmproduktionen, Marketing oder in der Produktentwicklung, verwenden realistische Material- und Lichtquellenmodelle (z.B. spekulare Materialien und Licht emittierenden Dioden), was die Simulation aufgrund von (nahezu-) Singularitäten im Pfadintegral weiter erschwert. Der Lichttransport zwischen Oberflächen weist charakteristische Unterschiede zum Transport in partizipierenden Medien auf, daher werden diese beiden Probleme oft separat betrachtet. Die vorliegende Dissertation fokussiert auf effiziente Methoden für Szenarien mit schwierigem Lichttransport zwischen Oberflächen.

Die Lichttransportsimulation

Die Pfadintegralformulierung [Veach 1998] drückt das Problem des Lichttransports in Form eines einzigen Integrals über alle möglichen Lichtpfade zum virtuellen Sensor aus (vgl. Abb. 4). Da die Zahl der Interaktionen von Licht mit Oberflächen einer Szene beliebig sein kann, ist die Länge der Lichtpfade entsprechend hoch bzw. unbegrenzt und die Integration damit ein hochdimensionales Problem. Dieses wird heutzutage in der Computergrafik beinahe ausschließlich mit stochastischen Integrationsmethoden wie Monte Carlo

oder Markoffketten Monte Carlo gelöst. Die Oberflächenstreuung selbst wird mittels der bidirektionalen Reflektanzverteilungsfunktion (BRDF bzw. BSDF) beschrieben. Abhängig von der Oberflächenrauheit kann eine BSDF sehr gerichtet streuen, z.B. in Richtung der perfekten Spiegelungs- oder Brechungsrichtung, oder bei spekularen Materialien wie perfekten Spiegeln, Wasser und glatten Dielektrika sogar zu einer Dirac Deltaverteilung degenerieren.

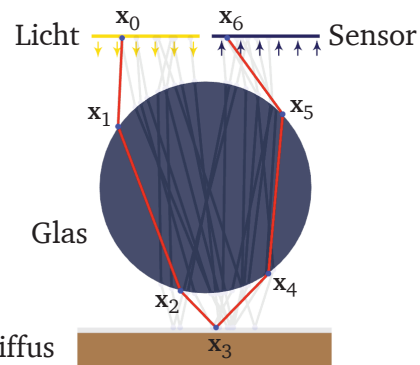


Abbildung 4: Lichttransportsimulation basiert auf Abtastung der Lichtpfade (hellgraue Pfade), die von einer Lichtquelle ausgehend (oben links) durch die virtuelle Szene verlaufen (Glaskugel und diffuse Ebene unten) und letztendlich den Sensor (oben rechts) erreichen. Ein solcher Lichtpfad ist als Beispiel rot hervorgehoben.

Abtastung von seltenen Lichtpfaden

Es gibt Konfigurationen der Szenengeometrie, Materialien und Beleuchtung, die mit existierenden stochastischen Methoden für die Abtastung nur schwer oder gar nicht behandelt werden können. Diese umfassen insbesondere spekulare Kaustiken von Lichtquellen, die durch einen Spiegel oder mehrere aufeinanderfolgende spekulare Interaktionen gesehen werden (vgl. Abb. 5, left).

Es kann gezeigt werden, dass es unmöglich ist, solche Lichtpfade mit erwartungstreuen, stochastischen Methoden abzutasten [Veach 1998, Theorem 8.2], da sie Sin-

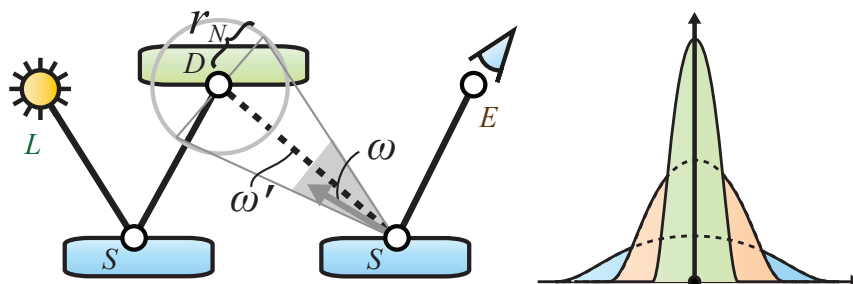


Abbildung 5: Links: Wir schlagen einen Regularisierungsansatz vor, der auf Mollifizierung basiert, um seltene (schwer oder gar nicht abgetastete) Pfade zu erzeugen. Hier ist die Oberfläche mit diffusem Material (D) in grün abgebildet, wohingegen spekulare Oberflächen (S) (Dirac-Delta-Interaktionen) blau sind. Rechts: ein einfaches Mollifizierungsbeispiel einer 1-dimensionalen Dirac-Delta-Verteilung mittels einer Gauß-Verteilung. Indem die Varianz gegen null geht, geht die Gauß-Verteilung als Grenzwert gegen die Deltaverteilung.

gularitäten in aufeinanderfolgenden Dimensionen enthalten. Wir schlagen hierzu ein neues Regularisierungsrahmenwerk vor [Kaplanyan and Dachsbacher 2013b], um solche Pfade bei der numerischen Integration zu finden. Hierzu werden zunächst die Singularitäten, die verhindern, dass der Pfad abgetastet wird, bei der Pfadkonstruktion identifiziert und diese selektiv regularisiert. Wir zeigen weiterhin, wie für diese und andere Methoden mit Regularisierung Konsistenz erreicht werden kann, so dass diese zur korrekten Lösung konvergieren.

Die Kernel-Bandbreite kontrolliert die Balance zwischen der eingeführten Glättung (Bias) und der Reduktion des Bildrauschens (Varianz) in den Kernel-Schätzungsmethoden wie z.B. Photon Mapping.

Wir zeigen wie diese Balance bestmöglich, durch die Wahl der Bandbreite [Kaplanyan and Dachsbacher 2013a], gewählt werden kann. Dafür schätzen wir die lokale Variabilität der Beleuchtung in der Umgebung eines jeden Lichtpfads; durch die entsprechend gewählte Bandbreite werden schnellere Darstellungszeiten bei kleinerem Bildfehler erreicht. Zusätzlich analysieren wir die asymptotische Konvergenz von Kernel-schätzungsbasierten Verfahren und zeigen, dass sie langsamer ist als die erwartungstreue Monte Carlo Methoden.

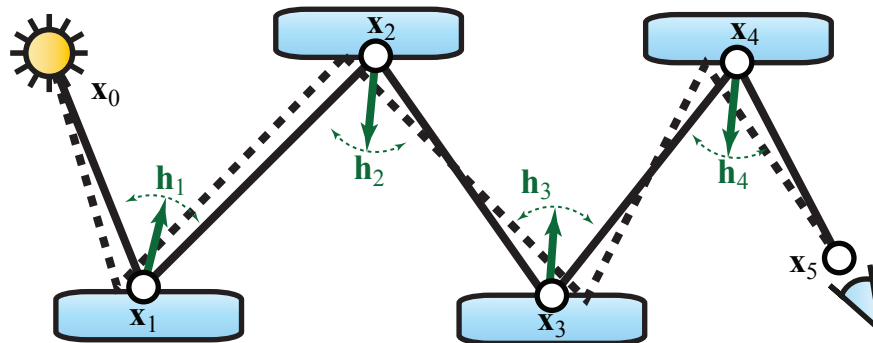


Abbildung 6: Um schwierige Pfade besser zu erkunden, schlagen wir ein neues System von generalisierten Koordinaten vor. Es stellt Lichtpfade als zwei fixe Endpunkte und Halbvektoren an den Interaktionen entlang des Pfads dar.

Generalisierte Koordinaten für Lichttransport

Pfade mit Interaktionen an spekularen oder glänzenden Flächen sind allgemein schwierig abzutasten. Allerdings ist eine dichte Abtastung solcher Phänomene nötig, um ein konvergiertes Bild zu erhalten. Wir schlagen den *Halbvektorraum-Lichttransport* [Kaplanyan et al. 2014] vor, der auf einer neuen Repräsentation des Pfadraumes basiert. Hier ist ein Lichttransportpfad über seine Endpunkte und Halbvektoren an jeder Interaktion dazwischen (vgl. Abb. 6) parameterisiert, im Gegensatz zur traditionellen Repräsentation durch eine Folge von Knotenpunkten. Diese neue Darstellung ist durch die lokale Beobachtung des Fermatschen Prinzips für Reflexionen und Brechungen motiviert und hat mehrere vorteilhafte Eigenschaften, wie beispielsweise die Zerlegung des Pfadintegrals in schwach-abhängige zweidimensionale Integrale. Diese Eigenschaften machen diese Darstellung zu einem sehr gut geeigneten Raum für die Berechnung des Lichttransports zwischen Oberflächen.

Visualisierung und Manipulation von Lichttransport

Obwohl Lichttransportmethoden entwickelt werden, um Bilder effizient darzustellen ist das Verständnis der dargestellten Phänomene ebenso wichtig – sowohl zur Entwicklung neuer oder zur Verbesserung bestehender Simulationen, als

auch beim Entwurf virtueller Szenen. Allerdings hat die menschliche Wahrnehmung bei vielen Phänomenen, wie bei prominenten glänzenden Interreflexionen, Schwierigkeiten die Entstehung der beobachteten Erscheinung zu verstehen. Wir haben hierzu Visualisierungsansätze für Lichttransport entwickelt [Reiner et al. 2012], die helfen, die Beleuchtung in verschiedensten Situationen zu begreifen. Eine darauf aufbauende Anwendung, ist die Anpassung der Lichtausbreitung gewollt-abweichend von der physikalischen Realität, beispielsweise im Kontext von Filmproduktionen, um künstlerischen Vorgaben zu entsprechen [Schmidt et al. 2013].

Contents

Vita and Publications	v
Acknowledgments	vii
1 Introduction	3
1.1 Global Illumination Problem	3
1.1.1 Importance of the Problem	4
1.2 Present Dissertation	5
1.2.1 Original Contributions and Structure	5
1.2.2 List of Additional Publications	7
2 Global Illumination Foundation	11
2.1 Fundamentals of Optics	11
2.1.1 Assumptions and Simplifications in the Thesis	18
2.2 Theory of Light Transport	19
2.2.1 Basic Measures and Radiometric Quantities	20
2.2.2 Thermodynamic Balance Equation for Light Transport	26
2.2.3 Rendering Equation	28
2.2.4 Scattering Distribution Functions	29
2.2.5 Reciprocity and Self-Adjointness	36
2.2.6 Path Integral Formulation and Measurement Equation	37
2.2.7 Path Measurement Domains	41
2.3 Specifics of Image Synthesis with Light Transport	43
2.3.1 Specifics of Human Vision	43

2.3.2	Image Convergence	45
3	Integration Methods	47
3.1	Deterministic Integration Methods	47
3.1.1	Finite Element Methods	47
3.1.2	Ray Transfer Matrices	48
3.2	Monte Carlo Path Integration	49
3.2.1	Monte Carlo Integration	50
3.2.2	Application to Light Transport	52
3.2.3	Variance Reduction	53
3.3	Markov Chain Monte Carlo Integration	57
3.3.1	Metropolis-Hastings Sampling	57
3.3.2	Convergence Criterion for Metropolis-Hastings Methods . . .	60
3.3.3	Application: Metropolis Light Transport	62
3.4	Kernel Estimation within Integration	66
3.4.1	Estimation Error	67
3.4.2	Bandwidth Selection and Convergence	67
3.4.3	A Brief History of Adaptive Estimation	69
4	Path Construction Methods	71
4.1	Fundamentals of Path Construction	72
4.2	Basic Path Construction: Implicit Light Sampling	73
4.3	Connection-based Path Construction	75
4.3.1	Unidirectional Path Construction	76
4.3.2	Bidirectional Path Construction	78
4.4	Estimation-based Path Construction	81

4.5	Path Caching Schemes	84
4.5.1	Photon Maps	84
4.5.2	Light Subpath Pooling and Many-Light Methods	85
4.6	Perturbation-based Path Construction	86
4.6.1	Path Space Perturbations	86
4.6.2	Trajectory Space Perturbation	89
4.6.3	Primary Sample Space Perturbations	90
4.6.4	Further Reading on Perturbation-based Construction	92
I	Efficient Kernel Estimation for Light Transport	95
5	Optimal On-Surface Kernel Estimation	97
5.1	Kernel Estimation in Photon Mapping	98
5.2	Recursive Estimation in PPM	101
5.2.1	Recursive Estimation in Statistics	103
5.3	Asymptotic Convergence Analysis	104
5.3.1	Variance and Expectation of Estimation Error	104
5.3.2	Convergence Rate of Photon Mapping	105
5.4	Adaptive Bandwidth Selection	108
5.4.1	Asymptotic MSE of the Measurement	109
5.4.2	The Plug-In Method for Bandwidth Selection	112
5.4.3	Recursive Estimation of the Laplacian	113
5.4.4	Convergence and Consistency	116
5.5	Implementation Details	117
5.6	Results and Discussion	118

5.7	Limitations and Potential Improvements	119
5.8	Conclusions	121
6	Regularization in Path Space using Kernel Estimation	125
6.1	Background	127
6.1.1	Full-Path Regular Expressions	127
6.1.2	Paths that Cannot Be Sampled	127
6.1.3	Mollification	128
6.2	Path Space Regularization	129
6.2.1	Singularities in the Integrand of the Path Integral	130
6.2.2	Example: Regularization of Specular Singularities	131
6.2.3	Other Types of Regularization	132
6.2.4	Selective Regularization	133
6.2.5	Regularization with Bidirectional Path Construction	134
6.3	Regularization with Integration Methods	135
6.3.1	Consistent Monte Carlo Integration with Regularization	136
6.3.2	Consistent Markov Chain Monte Carlo Methods with Regularization	140
6.4	Implementation Details	145
6.5	Results and Discussion	147
II	Half Vector Space Light Transport	153
7	Light Transport in Half Vector Space	155
7.1	Preliminaries	157
7.1.1	Path Integral in Path Space	157

7.1.2	Generalized Halfway Vector	158
7.2	Light Transport in Half Vector Space	159
7.2.1	Half Vector Domain for Light Paths	159
7.2.2	Analysis of the New Domain	160
7.2.3	Specular Interactions in the New Domain	165
7.2.4	Discussion	166
7.3	Conclusion	167
8	Half Vector Space Path Construction for Efficient Light Transport	171
8.1	Path Construction in Half Vector Space	173
8.1.1	Half Vector Space Mutation	173
8.1.2	Transitioning between Subspaces	177
8.2	Sampling Improvements	180
8.2.1	Optimal Expected Step Size for Interactions	181
8.2.2	Stratification over the Image Plane	184
8.2.3	Sampling Motivated by the Integrand Spectrum	186
8.3	Implementation Details	186
8.4	Results and Discussion	189
8.5	Conclusion	191
III	Additional Applications	195
9	Analyzing and Controlling Light Transport	197
9.1	Interactive Visualization of Light Transport	198
9.1.1	Selection and Classification of Light Transport	198
9.1.2	Light Transport Visualization Tools	200

9.2	Artistic Manipulation of Light Transport	202
9.2.1	Manipulation with Path Retargeting	203
9.2.2	Example Edits	205
9.3	Conclusion	206
10	Conclusion	209
	Bibliography	213
	Index	227

Introduction

1.1 Global Illumination Problem

Ever since generating synthetic images with computers became possible, researchers in computer graphics were fascinated by the reproduction of realistic scenes. The key to photorealism is a physically based *global illumination* computation. Global illumination simulates physically accurate propagation of light in virtual environments. In this thesis, we focus on global illumination between surfaces (in absence of participating media) due to its prominent contribution to the solution in typical scenarios. This problem can be formalized as a *rendering equation* [Kajiya 1986] that describes how the light scatters off a surface point.

Solutions to this problem are nowadays based on stochastic integration methods, such as Monte Carlo or Markov chain Monte Carlo methods. Techniques such as bidirectional path tracing [Lafortune and Willems 1993, Veach and Guibas 1994] or Metropolis light transport [Veach 1998] compute unbiased solutions. However, they converge only slowly to noise-free images. The inherent complexity of light transport also creates challenges that have not yet been fully mastered. For example, computing light transport in complex scenes with highly glossy materials remains nontrivial due to specular-diffuse-specular chains of light-matter interactions that are hard to compute. Moreover, some configurations of scene geometry, materials, and lighting cannot be handled with existing methods.

There are multiple major directions to address the challenges of light transport simulation. In present work, we focus the integration problem of light transport, which remains challenging in scenes with complex lights, geometry, and materials.

1.1.1 Importance of the Problem

Nowadays, global illumination is an essential part of physically correct or photorealistic image synthesis. It is used in many fields where image quality is important, such as predictive rendering in architecture, product concept design, computer-generated movie and visual effects production, as well as computer video games. Global illumination is also an important method for virtual and augmented reality systems, where synthesized images are used to immerse a human into an artificially created environment.

With the recent transition of the major rendering systems to *physically based rendering* (PBR) standard [Pharr and Humphreys 2010], including NVIDIA Iray, Arnold Renderer by Solid Angle Inc. (Autodesk), RenderMan by Pixar Animation Studios, Hyperion by Disney Animation Studios, and others [Keller et al. 2015], having a robust physically accurate solution to global illumination becomes essential. Such systems are widely adopted by multiple industries, however, they are known for being hard to control, especially for large production scenes.

In addition, light transport can provide a physically accurate solution to the global illumination problem, which makes it important for simulation in accuracy-demanding scientific and industrial fields, such as medical imaging, lighting design, optical systems design, fiber optics, medical imaging and reconstruction, astrophysics, and other fields of scientific and industrial visualization.

For all these applications, we outline the following three challenges for a global illumination method:

- *Robustness*: the method should guarantee the convergence to the correct solution for a wide spectrum of possible configurations of input data (scene geometry, materials, light sources, and cameras).
- *Predictability*: the method should have a predictable convergence for a given amount of input data.
- *Performance*: the method should have a practical computational cost.

In this dissertation, we will present several methods that focus on these challenges for difficult regimes of light transport.

1.2 Present Dissertation

In this section, we summarize the contribution of the present work, as well as overview its high level structure.

1.2.1 Original Contributions and Structure

The focus of the present dissertation is efficient light transport methods for physically-accurate solution to global illumination.

Advanced Kernel Estimation Methods. Kernel estimation methods were introduced with photon mapping [Jensen 2001, Shirley et al. 1995] in light transport. They distribute the energy in virtual photons over the scene, and then use a kernel estimation method to gather these photons around the visible points. The main challenge of kernel estimation is the efficient selection of kernel size, called bandwidth. We analyze these methods and provide an efficient on-line kernel bandwidth selection strategy for photon mapping class methods in Chapter 5. In addition, we derived the optimal asymptotic convergence rate for the class of methods based on the kernel estimation, and showed that their convergence rate is generally slower than the convergence rate of the purely stochastic Monte Carlo methods.

The second part of Chapter 5 shows how kernel estimation can be generalized as a path-space regularization, and applied in domains other than the on-surface domain (e.g., in the domain of an outgoing light direction). This reformulation allows the regularization with mollification in solid angle domain to be applied to advanced light transport methods, including Markov chain Monte Carlo methods, such as Metropolis light transport. Moreover, it is also possible to apply regularization selectively only to the hard light paths that require it. This minimizes blurring in the rendered image.

New Domain for Light Transport Integration. Existing domains used for light transport pose integration challenges in many situations, including, e.g., computing scattering through a chain of glossy inter-reflections. In Chapter 7 we present a new integration domain for light transport. In this domain, the integral equation of light transport has important theoretical properties, such as convenient spectral analysis of the integral, as well as useful practical properties.

In Chapter 8 we demonstrate how this new domain can be used in practice to render images with challenging effects, such as specular highlights with multiple highly glossy inter-reflections.

Artistic Control for Light Transport. In many production environments it is important to generate visually appealing images. This goal oftentimes contradicts the physically based nature of light transport. We provide tools for analysis, selection, and artistic manipulation of individual components of the light transport in Chapter 9. We provide a set of tools for artistic analysis that allow to visualize the light flow through the scene in intuitive and direct manner. Once the illumination is analyzed, the digital artist can control the illumination using the light editing tools to achieve the visual goals by altering the light flow in a physically impossible, yet plausible manner with minimal intrusion into the physical process of light transport.

List of Publications. The contributions are based on the results from the following publications:

Kaplanyan, A. S. and Dachsbacher, C. *Adaptive progressive photon mapping*. ACM Trans. on Graphics, 32(2):16:1–16:13, 2013a.

Kaplanyan, A. S. and Dachsbacher, C. *Path space regularization for holistic and robust light transport*. Computer Graphics Forum (Proc. of Eurographics), 32(2):63–72, 2013b.

Kaplanyan, A. S., Hanika, J., and Dachsbacher, C. *The natural-constraint represen-*

tation of the path space for efficient light transport simulation. ACM Trans. on Graphics (Proc. SIGGRAPH), 33(4), 2014.

Reiner, T., Kaplanyan, A. S., Reinhard, M., and Dachsbacher, C. *Selective inspection and interactive visualization of light transport in virtual scenes.* Computer Graphics Forum (Proc. of Eurographics), 31(2):711–718, 2012.

Schmidt, T.-W., Novak, J., Meng, J., Kaplanyan, A. S., Reiner, T., Nowrouzezahrai, D., and Dachsbacher, C. *Path-space manipulation of physically-based light transport.* ACM Trans. on Graphics (Proc. SIGGRAPH), 32(4):129:1–129:11, 2013.

Chapters 5 – 9 of the present dissertation are largely based on these publications.

1.2.2 List of Additional Publications

The following publications were emerged from the dissertation project, but are not covered in the present thesis:

Hachisuka, T., Kaplanyan, A. S., and Dachsbacher, C. *Multiplexed Metropolis light transport.* ACM Trans. on Graphics (Proc. SIGGRAPH), 33(4), 2014.

Kaplanyan, A. S. and Dachsbacher, C. *Cascaded light propagation volumes for real-time indirect illumination.* In Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pages 99–107. 2010.

Global Illumination Foundation

There is no subject so old that something new cannot be said about it.

— FYODOR MIKHAILOVICH DOSTOYEVSKY,
A Diary of a Writer (1876)

In this chapter, we describe the physical foundation for global illumination methods. We start by introducing the fundamentals of physical optics and scoping the phenomena important for light transport. Next, we describe the radiometric measures and quantities used in light transport, followed by introduction of surface scattering and underlying microfacet models

2.1 Fundamentals of Optics

Global illumination studies the distribution of electromagnetic energy in *visible light spectrum*, that is, light visible to bare human eye. Visible light takes a small range of wavelengths from about 380 to 750 nm in the spectrum of electromagnetic radiation (see Figure 2.1).

Common sources of visible light are based on the emission mechanisms of incandescence (Sun, light bulbs), luminescence (light-emitting diodes, fluorescence), and electric discharge (lightning, neon lamps). Sun, the main source of light for humans, is a thermal source with a characteristic spectrum close to the one of an ideal black body, heated to the effective temperature of 5777 degrees Kelvin.¹ The

¹This corresponds to the temperature of the photosphere, the visible surface of the Sun.

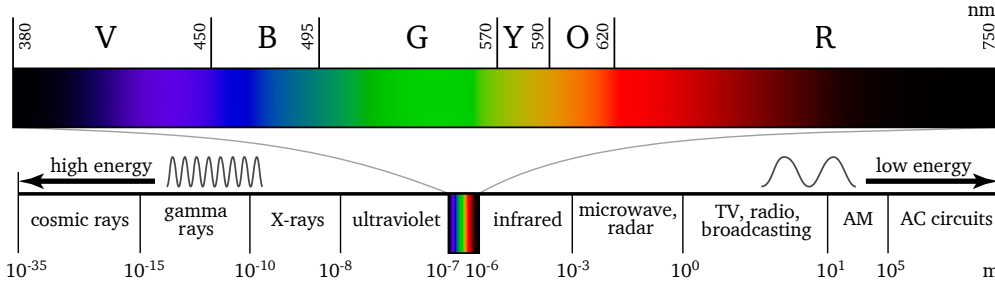


Figure 2.1: Electromagnetic spectrum and a close-up of the spectrum visible for humans with wavelength ticks in nanometers denoting ranges of main rainbow colors (left to right: violet, blue, green, yellow, orange, red).

model of a radiating black body is a convenient idealized physical model of a body that does not reflect any light (has a perfectly black albedo), while isotropically emitting the light due to the thermal emission (incandescence). When heated to a temperature T Kelvin and staying in thermal equilibrium, the black body emits energy according to the Planck's law as

$$B(T, \lambda) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1},$$

where k_B is the Boltzmann constant, h is the Planck constant, and c is the speed of light in the medium that receives the emission. Example spectra for emission with various temperatures are shown in Figure 2.2, top. Due to this shift of the spectral mode, human eye perceives cooler objects with red tint, while warmer objects appear in blue tint. Figure 2.2, bottom shows the perceived color tint of the black body based on its temperature. This correspondence between the color and the black body temperature is called a *color temperature* and is commonly used to characterize the hue of a light source.

Electromagnetic radiation is studied by multiple fields of physics. Optics is an integral part of electromagnetism. However, due to the importance of optical phenomena to humans, historically it has always been considered as a dedicated field of physics. We overview the main subfields of optics, including their subject matter and assumptions of the underlying optical models, in the order from smaller to higher scale.

Quantum optics, as part of *quantum electrodynamics*, is the most advanced field

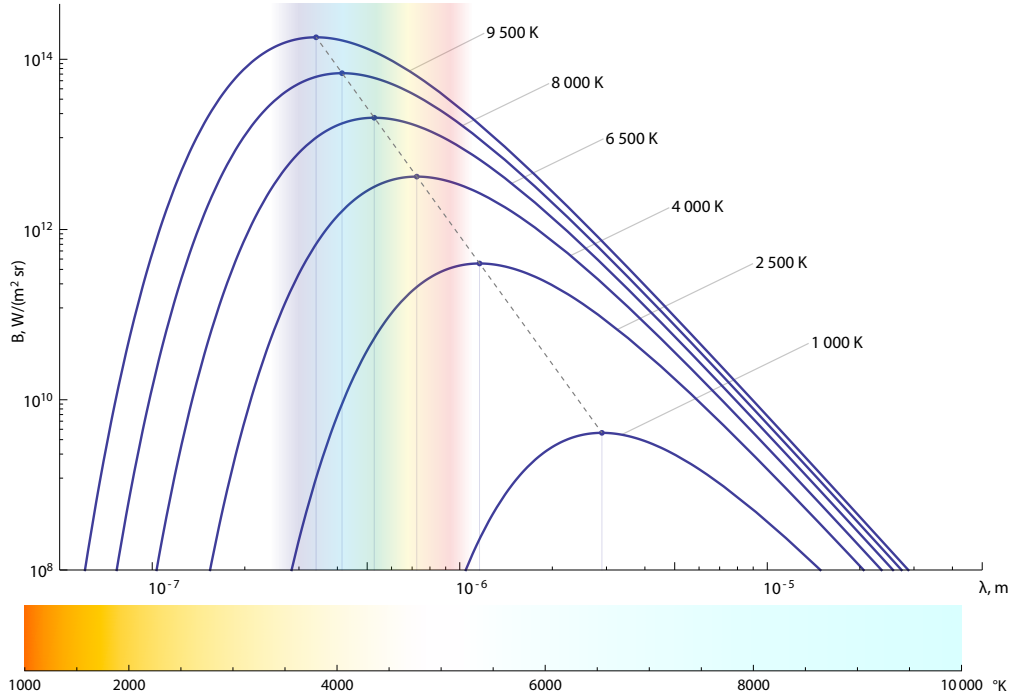


Figure 2.2: Top: distribution of spectral radiance emitted by a black body heated to various temperatures according to the Planck's law. Bottom: reproduction of perceived color of emitter light by a black body at various temperatures.

of optics that describes the interactions of light with matter at nanoscopic scales all the way down to the Planck length ($\approx 1.62 \times 10^{-35}$ m).¹ It studies the interaction of a quantum system of a matter with quantized electromagnetic waves (a photon is considered as both wave and particle). Simulation of quantum optics is computationally intense even for simple systems. If quantum effects are insignificant (which is usually the case on larger scales), simpler models are considered. For a more detailed overview the reader can refer, for example, to the book by Berestetskiy et al. [1982].

Electromagnetic optics include both *wave optics* (scalar approximation of the electromagnetic field) and *geometrical optics* (operates on the macroscale of rays as transport primitives).

¹Even though it is an advanced and solid theory, it is not suitable for describing the processes beyond the Planck scale. Current theoretical developments, such as string theory and M-theory [Becker et al. 2007], are focused on the unified theory beyond this scale.

The electromagnetic field is described by four Maxwell equations [1865]:

Gauss' law:	$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0},$
Gauss' law for magnetism:	$\nabla \cdot \mathbf{B} = 0,$
Ampère's circuital law:	$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t},$
Faraday's law of induction:	$\nabla \times \mathbf{B} = \mu_0 \left(\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right).$

The first two equations describe the properties of the electric amplitude (\mathbf{E}) and the magnetic amplitude (\mathbf{B}) of the electromagnetic field. Gauss' law states that the divergence (change of the density in infinitesimal volume) of the electric field depends on the presence of charges (total charge density ρ) in this region. The second equation is similar but for magnetic fields, yet it also states that the magnetic field is a solenoidal vector field, that is, there are no known magnetic monopoles (pure sources or sinks with only one pole). The second pair of equations describes interdependencies between electric and magnetic fields. The terms ε_0 and μ_0 define electric and magnetic permittivity of the propagation medium, correspondingly. The presence of electric current (represented by its density \mathbf{J}) curls the magnetic field.

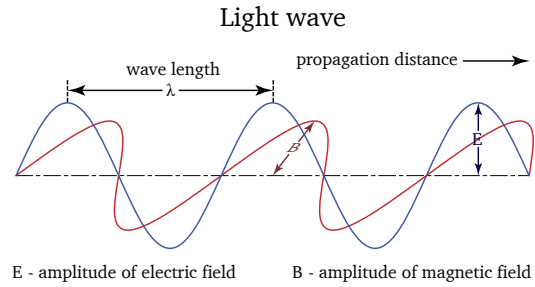
If a planar electromagnetic wave propagates in a uniform medium, the Maxwell equations turn into

$$\begin{aligned} \left(\nabla^2 - \mu\varepsilon \frac{\partial^2}{\partial t^2} \right) \mathbf{E} &= \mathbf{0}, \\ \left(\nabla^2 - \mu\varepsilon \frac{\partial^2}{\partial t^2} \right) \mathbf{B} &= \mathbf{0}, \end{aligned}$$

where $c = 1/\sqrt{\mu\varepsilon}$ is the speed of light in the propagation medium.

The solution to these equation is a pair of oscillation equations

$$\mathbf{E}(r, t) = \mathbf{E}_0 e^{i(kr - \omega t)}, \quad \mathbf{B}(r, t) = \mathbf{B}_0 e^{i(kr - \omega t)}, \quad (2.1)$$



where r is the distance along the propagation direction, k is the wavenumber, and $\omega = kc$ is the angular frequency of the wave. The general solution to the electromagnetic wave equation in homogeneous, linear, time-independent media can be written as a linear superposition of planar waves. Both electric and magnetic fields are linked and behave similarly to planar water waves in this case. This analogy is helpful for discussing the wave optics effects.

A simplified model of light propagation and interaction is considered by *wave optics* (or *physical optics*). Wave optics considers only the wave nature of light and is used at scales of the order of a single wavelength.¹ Wave optics studies the light as a wavefront of continuous waves, including such wave properties as wavelength, phase, and polarization states. It describes the propagation of the wavefront as well as the effects caused by it, such as interference of multiple waves or diffraction of waves after passing through a small obstacle or a slit.

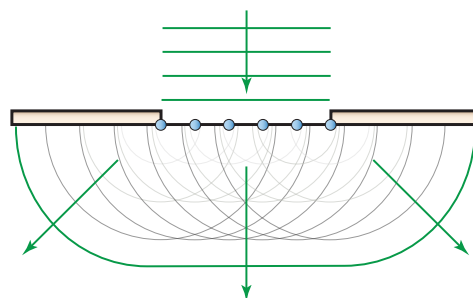


Figure 2.3: According to Huygens-Fresnel principle, wave propagates by inducing secondary waves in every point of the wavefront. Thus, obstacles can change the shape of the wavefront.

Geometrical optics, or *ray optics* studies the macroscopic behavior of visible light, at scales much greater than both the wavelength and the Planck length, where both quantum and wave effects are negligible. Note that since geometrical optics considers only the visible spectrum, the scales of light interaction considered should be at least an order of magnitude greater than 750 nm to be able to neglect the effects of wave optics. This is an appropriate assumption for light transport at humanly observable scales, where the human visual system can resolve spatial details only at a scale of 10 micrometers,² which is significantly greater than the largest wavelength of visible spectrum.

¹Note, that in general the wavelength of electromagnetic radiation can vary in a very wide range from 10^8 to 10^{-35} m, slightly overlapping with quantum scales only in the range of ultrashort wavelength.

²Assuming a young human eye that can focus at 5 cm with the highest angular resolution of 1 arcminute (≈ 0.0003 radians) gives us the smallest recognizable object with a size of 15 micrometers.

This makes the geometric optics an important foundation for solving light transport for image synthesis.

The simplest model of light interacting with surface in geometric optics considers a completely diffuse reflectance of the light off the surface and is called *Lambert cosine law*. This model simplifies reflection off a rough interface or

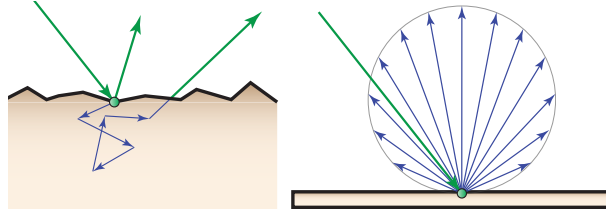
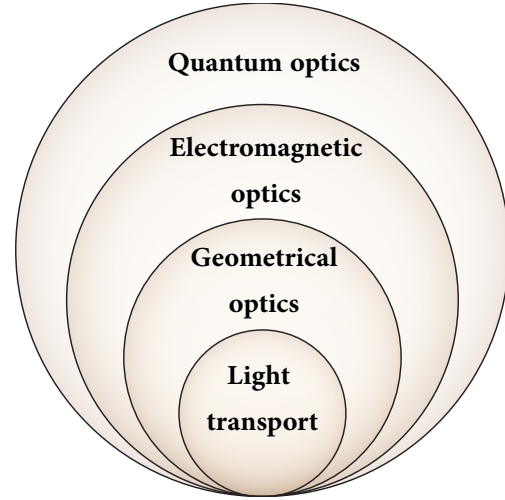


Figure 2.4: Lambert law assumes a microscale rough reflection and/or scattering under the surface (left), which results in a randomly perturbed direction of reflected light (right).

microscale subsurface scattering of light by assuming the reflected direction of the light leaving the surface is uniformly random after one or more interactions. This model is commonly used as a simplifying assumption for light transport, because the apparent brightness of such surface to the observer is constant from all viewing directions. This happens because, although the surface element viewed at a grazing angle emits less energy, at the same time the apparent size of this element is foreshortened for the viewer correspondingly, thus increasing the density of such elements and compensating for the cosine distribution of the reflected energy.

Fermat's principle is a consequence of the Huygens-Fresnel principle from wave optics that is visible on a macroscale of geometric optics, i.e., for the limiting case of negligibly small wavelength $\lambda \rightarrow 0$. In this case, if we consider all possible paths from one point to another, due to infinitesimally small wavelength, various neighboring paths would vary significantly in phase, and thus cancel each other out due to a destructive interference. However, this does not hold for extremal paths (shortest, longest, or stationary), since all neighboring paths in the proximity of such path differ from it by second-order amounts only, and thus interfere with it constructively. Thus, extremal paths contribute to this interference the most. This leads to Fermat's principle: light between two points follows the extremal path, i.e., the path with a shortest, longest, or stationary propagation time.

This principle is valuable for geometric optics and light transport, since it describes how the light behaves on the interface of two media with different speeds of light (i.e., different indices or refraction), as well as how to construct a point-to-point light path with multiple interactions along the path through more complex systems, such as a set of lenses and mirrors. We will also employ this method to construct a light path through multiple arbitrary microfacets in Chapter 7.



Hierarchy of study scopes of various optical theories, where each nested scope covers a subset of the parent scope.

We provide a summary of the aforementioned fields of optics in Table 2.1.

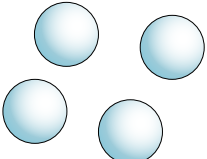

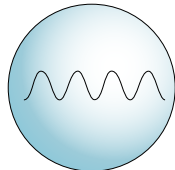
				
Electromagnetic optics		Quantum optics		
Geometrical optics		Wave optics		
Object	Photon as rigid particle	Electromagnetic wave	Quantized waves	
Macro-object	Ray	Monochromatic wavefront	Single photon	
Example study	Radiometric quantities (limit when $\lambda \rightarrow 0$)	Interference, diffraction, dispersion, luminescence, Fresnel effect, polarization	Quantum systems, manipulation of single atoms, photonics	

Table 2.1: Basic radiometric quantities, notation, descriptions and illustrations.

Light transport studies the macroscopic effects of the total equilibrium distribution of light described by the geometrical optics model.

2.1.1 Assumptions and Simplifications in the Thesis

In this dissertation, we focus on the most important computational problems of light transport. Moreover, we consider only the light transport between opaque surfaces in a vacuum due to their the primary importance for image synthesis and challenging computational complexity.

Time Dependency. We assume that the thermodynamic equilibrium is reached instantly after every change in lighting. This is a reasonable assumption for a wide variety of effects. For example, fluorescence, while being omnipresent in the nature, usually delays the equilibrium by not more than a fraction of a microsecond. Phosphorescence can take minutes to achieve equilibrium, however, the effect is usually not prominent and changes slowly. Generally, the assumption that the system is always in a steady state allows us to reduce the dimensionality of light transport problem by omitting the *temporal dimension*.

Wavelength Dependency. Another important assumption that allows us to reduce the dimensionality of light transport is independence on the wavelength. This is a common assumption in transport theories [Case and Zweifel 1967] called *one-velocity transport*. In context of light transport, in media with constant index of refraction¹, the one-velocity condition is equivalent to a one-wavelength condition. In other words, we consider only *elastic scattering*, which means that we assume the wavelength will not change along the path after multiple interactions (as opposed to *inelastic scattering*). Moreover, in many methods we also omit the wavelength dependence all together to further improve practical convergence, however, one can extend all presented methods to *spectral rendering* by solving the light transport for each wavelength independently.

Only Local Effects of Wave Optics. Even though light transport is based on geometrical optics, some local effects of wave optics, which do not break the geometrical assumption of constant radiance along the ray, can be incorporated.

¹We consider only vacuum between surfaces in the present dissertation.

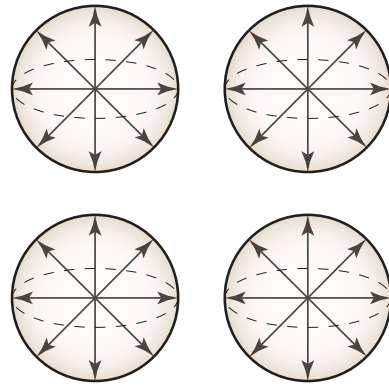
We limit the scope of electromagnetic effects to geometrical optics, assuming that the objects participating in light transport have details significantly larger than the wavelength of visible light. Therefore we do not consider most of the wave/quantum optics effects.

The *Fresnel effect* defines the ratio of reflected to refracted amount of light when moving through the interface between two media. This wave optics effect is observable at microscale, and can thus be incorporated into the properties of a local material or scattering distribution. Note that such macroscale “averaging” is possible only if we are not interested in the polarization of the scattered wave.

2.2 Theory of Light Transport

In this section, we first introduce the basic physical radiometric quantities for light transport, as well as measurement domains and the scattering equation. As a next step, we describe the high-level measurement quantities along the complete photon trajectory connecting a light source to a camera sensor. Based on this, we introduce the path integral formulation for light transport.

We start by describing the fundamental quantities used by the light transport theory. The most comprehensive solution to the light transport problem is the *light field*. The light field also known as *plenoptic illumination function* (or plenoptic function) is used in computer graphics to define the image of a scene from all viewing positions at all viewing angles. In other words, it describes the amount of electromagnetic energy that goes through every point in space in every particular direction in a unit of time. Given such a distribution, it is straightforward to obtain an image by simply evaluating the amount of flux coming into the objective lens of a virtual camera. However, usually computing and storing



Plenoptic function is defined on a five-dimensional space made up by the Cartesian product of Euclidean space \mathbb{R}^3 and the sphere of directions \mathbb{S}^2 .

the complete light field is unnecessary for a single image and computationally prohibitive. Most of the image generation methods evaluate the plenoptic function on demand only at a small set of positions and directions required to synthesize the image. If required, the function can be extended with additional dependencies on time, wavelength, and polarization angle.

As the first step, we will describe the quantities used in optics for measuring the light field. Table 2.2 contains the notation we use throughout the dissertation.

Term	Description
$\mathbf{x}, \mathbf{i}, \mathbf{o}$	Surface point, incident and outgoing direction (3d vectors)
\mathbf{X}	Ordered set of points, such as a light transport path
$(\mathbf{u}, \mathbf{v}, \mathbf{n})$	Tangent frame at a surface point
$\mathbf{v}, \mathbf{v}^\perp$	Vector/point and its planar projection onto tangent frame
$\ \mathbf{v}\ , \bar{\mathbf{v}}$	\mathcal{L}_2 length of a vector, unit (normalized) vector
$d\mathbf{v}, \Delta\mathbf{v}$	Differential of vector, finite difference
$\Sigma, \Sigma $	Matrix Σ , its determinant
$\left \frac{d(\mathbf{b}_0 \cdots \mathbf{b}_n)}{d(\mathbf{a}_0 \cdots \mathbf{a}_n)} \right $	Jacobian determinant of transformation from space \mathbf{A} to \mathbf{B}
$\mathbf{h}, \mathbf{h}^\perp, \mathbf{h}^\parallel$	Halfway vector \mathbf{h} ; \mathbf{h}^\perp in projected solid angle and \mathbf{h}^\parallel in parallel plane domains
$f(\mathbf{X})$	Measurement contribution function of a light path \mathbf{X}
$f_s(\mathbf{i}_i \rightarrow \mathbf{o}_i) = f_s(\mathbf{x}_i)$	Bidirectional scattering distribution function (BSDF) at \mathbf{x}_i

Table 2.2: Mathematical notation used throughout the dissertation.

2.2.1 Basic Measures and Radiometric Quantities

Radiometric quantities are used to measure various densities of electromagnetic radiation, including visible light. In light transport, they are used to define the properties of a light field, scattering processes in micro- and macro-settings on surfaces and in volumes, as well as various quantities used in rendering.

First, we introduce the measures used in light transport. We will derive other radiometric quantities as derivatives of the electromagnetic density with respect to these measures.

Measure Theory. The measure theory operates on functions defined on measurable spaces. A *measurable space* $(\mathbb{X}, \mathcal{X})$ consists of the actual underlying set \mathbb{X} (e.g., real numbers \mathbb{R}) and a non-empty collection \mathcal{X} of subsets of \mathbb{X} (algebra on \mathbb{X}).

Let $(\mathbb{X}, \mathcal{X})$ be a measurable space. A *measure* $\nu : \mathcal{X} \rightarrow \mathbb{R}_+$ is defined on \mathcal{X} as a set function that assigns a non-negative real number to each subset from the collection \mathcal{X} , intuitively interpreted as a size of the subset. Such a measure ν is

- non-negative: $\forall D \in \mathcal{X} : \nu(D) \geq 0$;
- null for the empty set: $\nu(\emptyset) = 0$;
- countably additive: For all countable collections $\{D_i\}_{i \in \mathbb{N}}$ of pairwise disjoint subsets: $\nu(\bigcup_{i \in \mathbb{N}} D_i) = \sum_{i \in \mathbb{N}} \nu(D_i)$.

If a measure ρ is, in addition, normalized, that is, $\rho(\mathbb{X}) = 1$, then ρ is called a *probability density function* (pdf) or a *probability measure*. This is a central measure in statistical integration methods that we will describe later.

Generally, multiple different measures can be defined for the same measurable space. Different measures are useful in integration. Intuitively, a measure can be interpreted as a generalized concept of a differential dx in Riemann integrals, where $dx = d(x)$ is a measure and can be defined differently from regular infinitesimal length, area, or volume measures.

When measuring an event with respect to multiple domains (e.g., both in time and space), it is possible to combine measures from these domains under a single *product measure*. Let ν_1 on $(\mathbb{X}_1, \mathcal{X}_1)$ and ν_2 on $(\mathbb{X}_2, \mathcal{X}_2)$ be two measures from the corresponding measurable spaces. Consider a product space $(\mathbb{X}_1 \times \mathbb{X}_2, \mathcal{X}_1 \otimes \mathcal{X}_2)$ induced by a Cartesian product $\mathcal{X}_1 \otimes \mathcal{X}_2$ of the corresponding algebras. A product measure $\nu_1 \times \nu_2$ can then be defined on this space as

$$\nu_1 \times \nu_2(\mathbf{x}_1 \times \mathbf{x}_2) = \nu_1(\mathbf{x}_1) \nu_2(\mathbf{x}_2),$$

where $\mathbf{x}_1 \in \mathcal{X}_1$ and $\mathbf{x}_2 \in \mathcal{X}_2$.

Consider a function f on a measurable space $(\mathbb{X}, \mathcal{X})$, such that, for any measur-

able set $D \in \mathcal{X}$

$$\nu(D) = \int_D f \, d\mu,$$

where a measure ν is absolutely continuous with respect to the measure μ on $(\mathbb{X}, \mathcal{X})$, that is, $\forall D \in \mathcal{X} : \mu(D) = 0 \implies \nu(D) = 0$. In this case, the function f is called the *Radon-Nikodym derivative* of ν with respect to μ , denoted as

$$f = \frac{d\nu}{d\mu}.$$

This function is unique up to a μ -null set, that is, to any set $N \in \mathcal{X}$ with $\mu(N) = 0$.

Common Measures. One common measure for a two-dimensional element, called the *area measure*, can be defined in two dimensions as as product of two linear measures

$$d\mathbf{x} = dA(\mathbf{x}) = dl(x) \times dl(y),$$

where elements $dl(x)$ and $dl(y)$ are the length measures along the corresponding Cartesian x and y axes. For a two-dimensional manifold in \mathbb{R}^3 , x and y are the axes of the tangent plane at \mathbf{x} . The *projected area measure* $dA_\omega^\perp(\mathbf{x})$ is another useful measure for an area of a surface element projected on a plane along the unit direction vector ω as

$$d\mathbf{x}^\perp = dA_\omega^\perp(\mathbf{x}) = |\omega \cdot (\bar{e}_x \times \bar{e}_y)| dl(x) dl(y) = |\cos \theta| dA(\mathbf{x}),$$

where \bar{e}_x and \bar{e}_y are the orthonormal basis unit vectors along the x and y axes of the tangent plane of the surface where dA is defined, and θ is the angle between the surface normal $\mathbf{n} = \bar{e}_x \times \bar{e}_y$ and the direction of projection ω .

The *solid angle measure* $d\sigma(\omega)$ defines a projected area on a unit sphere subtended by an object observed from the center of this sphere (Figure 2.7). For an infinitesimal area element $d\mathbf{x}$ orthogonal to the viewing direction at the distance r from the center of the sphere, its projection on the unit sphere is $d\sigma(\omega) = d\mathbf{x}/r^2$. After changing $d\mathbf{x}$ to spherical coordinates with origin at the center of the unit sphere as $d\mathbf{x} = r^2 \sin \theta d\theta d\phi$, we can obtain a solid angle measure as

$$d\omega = d\sigma(\omega) = \frac{d\mathbf{x}}{r^2} = \sin \theta d\theta d\phi,$$

where ω is the direction towards the area element, θ and ϕ are the spherical coordinates, the azimuthal and zenith angles correspondingly. The solid angle is measured in a dimensionless unit called a steradian (sr). A solid angle of a fully covered sphere is 4π steradian.

If the viewing point itself is located on a surface point \mathbf{x} with normal \mathbf{n} , then another useful measure, the *projected solid angle* measure $d\sigma_{\mathbf{x}}^{\perp}(\omega)$ can be defined as

$$d\omega^{\perp} = d\sigma_{\mathbf{x}}^{\perp}(\omega) = |\omega \cdot \mathbf{n}| d\sigma(\omega) = |\cos \theta| d\sigma(\omega),$$

where θ is the angle between the surface normal \mathbf{n} and the direction ω ; $|\omega \cdot \mathbf{n}| = |\cos \theta|$ projects the infinitesimal segment of the sphere subtended by the solid angle at ω to the surface of the equatorial disk that is tangential to the surface at \mathbf{x} .

Hereafter, we will use shorthands for the measures wherever possible, i.e., $d\mathbf{x}$ instead of $dA(\mathbf{x})$, $d\omega^{\perp}$ instead of $d\sigma_{\mathbf{x}}^{\perp}(\omega)$, and so on. These measures are widely used in light transport and form a foundation for radiometric quantities.

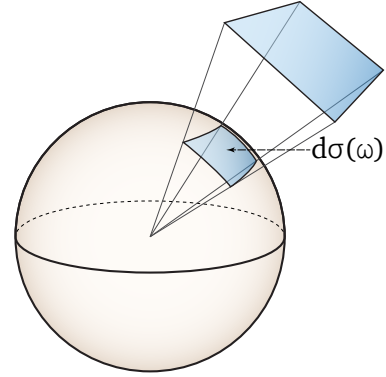
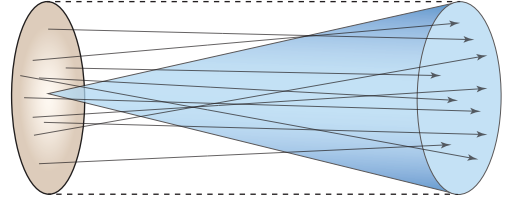


Figure 2.7: Solid angle measure.

Radiometric Quantities. Most of the radiometric quantities can be defined by means of the measure theory as Radon-Nikodym derivatives of the fundamental quantity Q , the energy of the electromagnetic radiation [Veach 1998, p.98]. For example, the central light transport quantity, *radiance* L , can be defined as a derivative of energy Q with respect to the product measure $dt \times d\mathbf{x}^\perp \times d\omega$ as

$$L(\mathbf{x}, \omega) = \frac{dQ}{dt d\mathbf{x}^\perp d\omega} = \frac{dQ}{dt d\mathbf{x} d\omega^\perp}.$$

Radiance is defined as the amount of energy passed in a unit of time through a cross section dA^\perp perpendicular to the differential solid angle of interest $d\omega$.¹ When measured directly on a surface, $d\mathbf{x}^\perp$ can be expressed as an infinitesimal surface patch dS projected onto the measurement direction $d\omega$ as $d\mathbf{x}^\perp = dS \cos \theta$, where θ is the angle between the measurement direction and the surface normal $\mathbf{n}_\mathbf{x}$.



Radiance L can be imagined as the amount of light traveling along straight lines within an infinitesimal tube bounded by a solid angle $d\omega$ and a cross section area $d\mathbf{x}^\perp$.

Radiance stays constant along the ray in a medium with constant index of refraction. Therefore, this quantity is especially convenient to use with transport methods based on raytracing².

Hereafter, when considering cross sections and surfaces, we will denote the values projected perpendicularly to the propagation direction with the perpendicular superscript “ \perp ”. This reflects the fact that only the perpendicular part of the measured quantity is considered. This is a common and convenient consideration in radiometry and electromagnetism, because the parallel part of the quantity usually does not affect its measurement at a particular cross section.

¹ Alternatively, we can define radiance using the *projected solid angle* $d\omega^\perp$.

² We do not consider media with varying indices of refraction $\eta = \eta(\mathbf{x})$, in which the light travels along arbitrary curves, making the raytracing-based methods inefficient. However, in such settings the *basic radiance* $L' = L/\eta^2$ stays constant (Abbe’s law), and thus can be used for transport simulation [Veach 1998, p.210] instead of the radiance L . See Ament et al. [2014] for a formulation of light transport with basic radiance.

We also make another simplification of notation. Even though one should differentiate between the incident radiance $L_i(\omega)$ coming from direction ω and the outgoing radiance $L_o(\omega)$ leaving into the same direction ω , oftentimes the directions are denoted by different variables, therefore, the meaning can be inferred from the arguments. That is, we use $L(\mathbf{i}) \equiv L_i(\mathbf{i})$ and $L(\mathbf{o}) \equiv L_o(\mathbf{o})$ throughout the dissertation for brevity.

In Table 2.3 we provide a set of basic quantities used in light transport with short description as well as a simple illustration of the physical meaning. Spectral quantities are defined the same way by additionally differentiating by the wavelength $d\lambda$, adding $[nm^{-1}]$ to the corresponding measurement unit. Hereafter, we will consider only non-spectral quantities. We explicitly specify when the spectral extension needs special treatment.



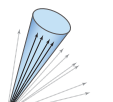
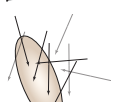
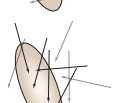
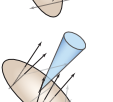
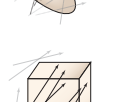
Quantity	Symbol, Unit (SI)	Description	Figure
Radiant energy	Q	$[J]$ Energy (in number of photons).	
Flux	$\Phi = \frac{dQ}{dt}$	$[W]$ Energy passed in a unit of time.	
Radiant intensity	$I = \frac{dQ}{dt d\omega}$	$\left[\frac{W}{sr}\right]$ Energy passed within a differential cone of solid angle $d\omega$ in a unit of time.	
Radiant emittance	$M = \frac{dQ}{dt d\mathbf{x}}$	$\left[\frac{W}{m^2}\right]$ Energy emitted by a surface with differential cross section dA in a unit of time.	
Irradiance	$E = \frac{dQ}{dt d\mathbf{x}}$	$\left[\frac{W}{m^2}\right]$ Energy incident on a surface with differential cross section dA in a unit of time.	
Radiance	$L = \frac{dQ}{dt d\omega d\mathbf{x}^\perp}$	$\left[\frac{W}{sr \cdot m^2}\right]$ Energy passed through a projected differential cross section dA^\perp oriented within a differential cone of solid angle $d\omega$ in a unit of time.	
Radiant energy density	$\omega_e = \frac{dQ}{dt dV(\mathbf{x})}$	$\left[\frac{W}{m^3}\right]$ Energy passed through a differential volume element $dV(\mathbf{x})$ in a unit of time.	

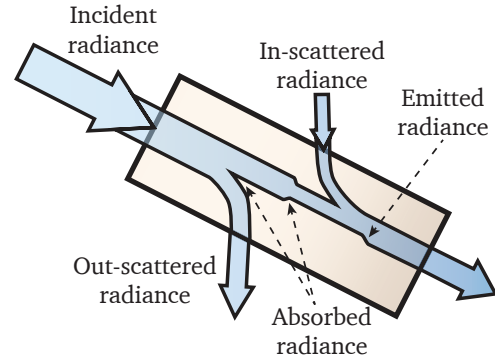
Table 2.3: This table provides basic radiometric quantities, their notation used throughout the thesis, descriptions, and simple illustrations.

We will use these basic quantities to define higher-level light transport quantities, such as the path measurement contribution function, in the following sections.

2.2.2 Thermodynamic Balance Equation for Light Transport

We start by first writing the balance equation for the thermodynamic equilibrium applied to electromagnetic transport, called radiative transfer. We then transform it into a convenient form for light transport on surfaces in the vacuum based on the modern formulation of radiance measurement.

First, we consider the factors that affect the radiance within an infinitesimal volume element $dV(\mathbf{x})$ around a point \mathbf{x} towards an infinitesimal solid angle centered around the direction \mathbf{o} . We can describe the radiance as measuring the number of photons that travel within an infinitesimal cone of directions around \mathbf{o} through a cross-section at the volume element within an infinitesimal amount of time.



Factors that influence the first-order change in radiance in an infinitesimal ray cylinder. Both out-scattered and absorbed radiance is accounted for by extinction coefficient σ_t .

This number first depends on the photons that entered the region and the photons that were emitted within this region. Then, if there is a scattering matter within the volume element, we also need to account for absorption and scattering.

Therefore, we can start with a high-level description

$$\begin{aligned}
 [\text{Change of radiance at } \mathbf{x} \text{ along } \mathbf{o}] = & \\
 & + [\text{Number of photons emitted in the volume element}] \\
 & - [\text{Number of photons absorbed in the volume element}] \\
 & + [\text{Number of photons scattered in the volume element}]
 \end{aligned}$$

First, we expand the first line in this equation. An infinitesimal change in the radi-

ance along the cylinder formed by $d\mathbf{x} \times d\mathbf{o}$ is a directional derivative of L at \mathbf{x} with respect to the direction \mathbf{o} as $\nabla_{\mathbf{o}} = (\mathbf{o} \cdot \nabla)$ leading to

$$[\text{Change of radiance at } \mathbf{x} \text{ along } \mathbf{o}] \equiv (\mathbf{o} \cdot \nabla)L(\mathbf{x}, \mathbf{o}).$$

Next, the amount of emitted energy depends on the content of the volume element (e.g., a flame) and is simply defined by its emitted radiance L_e as

$$[\text{Number of photons emitted in the volume element}] \equiv L_e(\mathbf{x}, \mathbf{o}).$$

Next, the probability of a photon being absorbed by a volume element at \mathbf{x} is defined by the effective cross-section $\sigma_t(\mathbf{x}, \mathbf{o})$ of the absorbing matter in this element, also called an *extinction coefficient*, so the total absorbed radiance becomes

$$[\text{Number of photons absorbed in the volume element}] \equiv \sigma_t(\mathbf{x}, \mathbf{o})L(\mathbf{x}, \mathbf{o}).$$

The extinction coefficient σ_t also relates to the mean free path¹ l between consecutive interactions for a single photon as $l = 1/\sigma_t$.

The last term describes the scattering process within this volume element. Similarly to the absorption, the scattering probability is defined by the scattering cross-section σ_s , which is a relative cross-section of particles that cause scattering. Unlike other effects, scattering can change the direction of a photon, so all photons in-scattered from all directions should be accounted for as

$$[\text{Number of photons scattered in the volume element}] \equiv \sigma_s(\mathbf{x}, \mathbf{o}) \int_{S^2} \rho(\mathbf{x}, \mathbf{i}, \mathbf{o}) L(\mathbf{x}, \mathbf{i}) d\mathbf{i},$$

where $\rho(\mathbf{x}, \mathbf{i}, \mathbf{o})$ is the *phase function*², a distribution that defines what fraction of scattered photons coming from the incident direction \mathbf{i} are scattered into the particular outgoing direction \mathbf{o} . This distribution describes the scattering properties of the medium.

¹The average distance traveled by a photon between interactions with the matter.

²The name originates from astronomy, where it was used as a function of a phase angle (angle between light-object-observer points).

Now, we can put all components together to obtain the *radiative transfer equation* [Chandrasekhar 1960] as

$$(\mathbf{o} \cdot \nabla)L(\mathbf{x}, \mathbf{o}) = L_e(\mathbf{x}, \mathbf{o}) - \sigma_t(\mathbf{x}, \mathbf{o})L(\mathbf{x}, \mathbf{o}) + \sigma_s(\mathbf{x}, \mathbf{o}) \int_{S^2} \rho(\mathbf{x}, \mathbf{i}, \mathbf{o})L(\mathbf{x}, \mathbf{i}) d\mathbf{i}. \quad (2.2)$$

Here, we assume that there is no *inelastic scattering* and the dependence on the wavelength is omitted as per simplifications discussed in Section 2.1.1. Note that both σ_t and σ_s can generally depend on \mathbf{o} [Jakob et al. 2010].

One important special solution to the radiative transfer for absorption only is called the *Beer-Lambert law*. In this case, the ratio of the transmitted to the incident flux, called the *transmittance* is

$$T(\mathbf{x}) = \frac{\Phi_t}{\Phi_i} = \exp \left(- \int_0^l \sigma_t(z) dz \right).$$

This law shows that the light energy decreases exponentially in absorbing materials.

2.2.3 Rendering Equation

An important role in rendering is devoted to surface scattering, as in the real world it is a common and prominent effect. Moreover, accurately modeling surface reflection is important for high-fidelity image appearance in many scenarios encountered in the real world. Therefore, many important applications of light transport for image synthesis were historically with opaque surfaces defined in the vacuum (assuming no volumetric scattering).

An equation describing this process is called the *rendering equation* (also known as scattering equation) and represents a special case of Eq. 2.2 for opaque surfaces in the vacuum. It can be derived as a limiting case for two regimes. First, vacuum has no volumetric scattering ($\sigma_s \equiv 0$) and causes no absorption ($\sigma_t \equiv 0$), and thus preserves radiance, i.e., $(\mathbf{o} \cdot \nabla)L(\mathbf{x}, \mathbf{o}) \equiv 0$. Second, once hit a surface, light gets unconditionally rescattered with $\sigma_t = \sigma_s = 1$ on a surface. A more detailed derivation along the same lines can be found in [Hege et al. 1996].

This equation was simultaneously introduced into computer graphics by David Immel et al. [1986] and James Kajiya et al. [1986] and describes the outscattered radiance at a surface point \mathbf{x} in direction \mathbf{o} for surfaces in the vacuum as

$$L(\mathbf{x}, \mathbf{o}) = \int_{S^2} f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) L(\mathbf{i}) |\langle \mathbf{n}, \mathbf{i} \rangle| d\mathbf{i} = \int_{S^2} f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) L(\mathbf{i}) d\mathbf{i}^\perp, \quad (2.3)$$

where f_s is a *bidirectional scattering distribution function* (BSDF) that describes the scattering properties of the material and is introduced in Section 2.2.4. The measure $d\mathbf{i}^\perp$ and the corresponding cosine term $|\langle \mathbf{n}, \mathbf{i} \rangle|$ in the integral are the only differences to the integral in the volumetric scattering term of Eq. 2.2. This cosine projection accounts for the change in the effective area of the interface when measured from the incident direction \mathbf{i} . It can be shown that this projection is a naturally appearing pullback of the differential form onto the lower-dimensional manifold of the surface [Lessig et al. 2012].

We also assume only elastic scattering and omit the dependence on the wavelength λ and time moment t , as they are straightforward to add and complicate the notation.

2.2.4 Scattering Distribution Functions

Both the radiative transfer equation Eq. 2.2 and the rendering equation Eq. 2.3 require bidirectional scattering distributions (phase function and BSDF) to be provided. In this section, we introduce several common scattering distributions for both media and surface scattering scenarios.

Volumetric Scattering. The *phase function* is the angular distribution of radiance scattered by a particle of a medium. It is defined as a fraction of the in-scattered radiance along the direction \mathbf{i} that is outscattered towards the direction \mathbf{o} as

$$\rho(\mathbf{x}, \mathbf{i}, \mathbf{o}) = \frac{dL(\mathbf{o})}{L(\mathbf{i}) d\mathbf{i}}. \quad (2.4)$$

This is a dimensionless function with units of sr^{-1} . The scattering distribution in the medium is considered to be constant throughout the medium (no dependence on \mathbf{x}) and circularly symmetric around the axis of incident direction due to the underlying central symmetry of a particle the light interacts with. Therefore, we can define a phase function being solely a function of the angle θ between in-scattering direction \mathbf{i} and out-scattering direction \mathbf{o} as

$$\rho(\theta) = \frac{L(\theta)}{\int_0^\pi L(\theta) \sin \theta \, d\theta}.$$

Due to the energy conservation, the phase function should be normalized

$$\frac{1}{4\pi} \int_0^{2\pi} \int_0^\pi \rho(\theta) \sin \theta \, d\theta \, d\phi = 1.$$

A common phase function for volumetric scattering is the *Heney-Greenstein phase function* [Heney and Greenstein 1941]

$$\rho(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos \theta)^{3/2}},$$

where $g \in (-1; 1)$ defines the type of scattering, ranging from backwards-scattering ($g < 0$) through isotropic scattering ($g = 0$) to forward scattering ($g > 0$).

It is also possible to define a phase function $\rho(\mathbf{x}, \mathbf{i}, \mathbf{o})$ that depends on both incident and outgoing directions. This can be required for rendering materials with oriented volumetric structure [Jakob et al. 2010].

Surface Scattering. Similarly to the phase function, the *bidirectional scattering distribution function* (BSDF) defines the amount of light scattered from the incident direction \mathbf{i} to the outgoing direction \mathbf{o} as

$$f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) = \frac{dL(\mathbf{o})}{L(\mathbf{i}) \, d\mathbf{i}^\perp} = \frac{dL(\mathbf{o})}{L(\mathbf{i}) \cos \theta \, d\mathbf{i}} = \frac{dL(\mathbf{o})}{dE(\mathbf{i})}, \quad (2.5)$$

which can be considered analogous to the definition of a phase function and defines a ratio of reflected radiance to the incident radiance onto the surface. Note the

difference in the measure between Eq. 2.5 and Eq. 2.4: BSDF uses projected solid angle measure due to the foreshortening of projected area of the planar interface, while a phase function is defined in free space and thus is measured with respect to a regular solid angle measure. This projection is the same pullback of the differential form in presence of a surface [Lessig et al. 2012].

Similar to the phase function, a BSDF should be normalized due to energy conservation as

$$\frac{1}{4\pi} \int_{S^2} f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) d\mathbf{i}^\perp = 1. \quad (2.6)$$

BSDF defines the appearance of a surface. A tremendous body of work has been done for developing BSDF models in the past. Two major classes of BSDFs are based on

- *Measured data.* Surface scattering is measured in laboratories (e.g., using a goniometric measurement stand) and stored in tabular form parameterized with some convenient parameterization, e.g., using a half way vector \mathbf{h} and a retro-reflective direction ω_d [Rusinkiewicz 1998]. It is also common to fit the measured data into some parametric models (e.g., [Lafortune et al. 1997]) for more convenient manipulations and reduction of the measurement noise.
- *Parametric models.* Based on the assumption about the microscopic structure of the surface, it is possible to develop an approximate or closed-form scattering model. For example, a class of *microfacet models* assumes that the underlying surface can be represented as a height field at microscale.

There are multiple other classes of BSDFs, including phenomenological models (e.g., [Phong 1975]), as well as models simulating wave optics phenomena (e.g., [Cuypers et al. 2012]). In the present thesis we will mostly use parametric models due to their practicality, compactness, and robustness.

One simple BSDF model is the *Lambertian reflectance model* for an ideal diffusely

reflecting surface

$$f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) = \rho_D \frac{1}{\pi},$$

where ρ_D is the view-independent albedo of the surface. The Lambertian BSDF is constant and models only reflection, taking zeros if incident or outgoing direction is below the surface's horizon. The term $1/\pi$ comes from the normalization requirement in Eq. 2.6 with the integration over the hemispherical support.

Another simple BSDF model is that of *perfect specular reflection*. Perfect and almost perfect mirrors cause delta responses or arbitrarily narrow modes. A perfect specular reflection is formulated as an angular delta distribution in the domain of halfway vectors [Veach 1998, Walter et al. 2007] as

$$f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) = F(\mathbf{i}, \mathbf{o}) \delta_{\mathbf{h}}(\mathbf{n} - \mathbf{h}) \left| \frac{d\mathbf{h}}{d\mathbf{o}^\perp} \right| = F(\mathbf{i}, \mathbf{o}) \frac{\delta_{\mathbf{h}}(\mathbf{n} - \mathbf{h})}{|\langle \mathbf{o} \cdot \mathbf{n} \rangle|}, \quad (2.7)$$

where $\mathbf{h} = \frac{\mathbf{i} + \mathbf{o}}{\|\mathbf{i} + \mathbf{o}\|}$ is the halfway vector that also defines the slope of a microfacet at the interaction¹; \mathbf{n} is the surface normal at the point \mathbf{x} . The term $\delta(\mathbf{n} - \mathbf{h})$ is a two-dimensional delta distribution on the sphere of directions \mathbb{S}^2 and is equal to zero everywhere, except where the slope of the surface \mathbf{n} coincides with the halfway vector \mathbf{h} of the interaction. This is an explicit constraint that manifests the law of reflection: the reflection angle is equal to the angle of incidence.

The term $F(\mathbf{i}, \mathbf{o})$ models the *Fresnel effect*, the partial reflection of light when moving through an interface between two media with different indices of refraction. The index of refraction for a material can be defined by a single real value $n = c/v$ that describes how much faster light propagates in the vacuum compared to this medium. For highly absorbing media, such as conductors, the complex refractive index has a non-zero imaginary part $n = c/v + i \cdot \kappa$. It is convenient to express the imaginary part as an extinction coefficient $\kappa = \sigma_t$. If we plug this index into the solution for a planar wave in Eq. 2.1, κ would turn into a decaying real-valued

¹The halfway vector can be generalized for both reflection and refraction as $\mathbf{h} = \frac{n_1 \mathbf{i} + n_2 \mathbf{o}}{\|n_1 \mathbf{i} + n_2 \mathbf{o}\|}$ [Sommerfeld and Runge 1911, Walter et al. 2007], where n_1 and n_2 are the corresponding indices of refraction. Both indices are set to one for reflection.

part of the exponent. The real part of the refractive index n is also related to the imaginary part κ via the *Kramers-Kronig relations*, so that if one part is known for all n , then another can be derived from this property of a complex function.

The two basic BSDF models, Lambertian and perfect specular, represent two extremes of the scattering, an ideal-diffuse and a perfect specular case. However, almost all real-world materials are glossy, forming the whole spectrum in between. We use models from *microfacet theory* to describe a wide variety of scattering materials. Microfacet theory originates from radiophysics, where it was initially used to simulate radar reflections from rough objects [Beckmann and Spizzichino 1963, Smith 1967]. The surface is modeled by so-called microfacets, tiny planar elements with slope at a certain height above the tangent plane to the surface. Each microfacet is usually considered to be perfectly specularly reflecting or refracting [Walter et al. 2007]. One common approach is to assume these microfacets are induced by a continuous and planar height field [Smith 1967].

The BSDF for specular interaction in Eq. 2.7 is a special case of such a microfacet surface with specular microfacets that are all aligned with the tangent plane. For a rough surface, the delta distribution $\delta_{\mathbf{h}}$ is replaced by a stochastic *normal distribution function*¹ (NDF) $D(\mathbf{h})$, which defines the probability density of encountering a slope with a normal aligned with \mathbf{h} as

$$D(\mathbf{h}) = \frac{dA_{\mathbf{h}}}{dA d\mathbf{h}},$$

where $d\mathbf{h}$ is an infinitesimal solid angle around the desired slope \mathbf{h} , $dA_{\mathbf{h}}$ is the area of microfacet slopes with normals within the infinitesimal cone of normals $d\mathbf{h}$ centered around the desired slope normal \mathbf{h} , and dA is the total area of the considered infinitesimal patch of a surface. By construction, a normal distribution function should obey multiple properties, such as non-negativity, normalization, and area preservation conditions [Walter et al. 2007].

If the surface is rough, that means it has microscale bumps and cavities that besides increasing scattering spread also cause self-occlusion and self-shadowing at

¹The term “normal distribution” is widely used in statistics, however, it has nothing in common with the normal distribution function in graphics that represents distribution of surface normals.

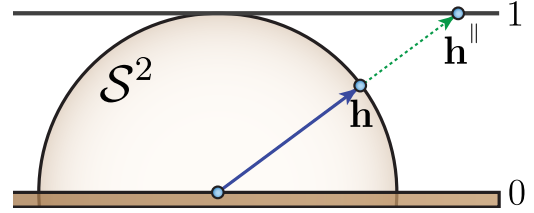
the microscale. The bidirectional *shadowing-masking term* $G(\mathbf{i}, \mathbf{o})$ is introduced to account for these effects. Similarly to the NDF, it describes what portion of microfacets is visible from both directions \mathbf{i} (self-occlusion, or masking effect) and \mathbf{o} (self-shadowing of microfacets). Shadowing and masking effects dominate more for rougher surfaces and at grazing angles of incidence. This term depends on the underlying distribution of microfacets, therefore, is tightly coupled with the NDF $D(\mathbf{h})$.

Next, we replace the term $\delta_{\mathbf{h}}$ with a more generic NDF $D(\mathbf{h})$ and account for the shadowing-masking term G . This way, we obtain a Cook-Torrance BSDF model, a widely used component microfacet BSDF model developed by Torrance and Sparrow [1967] and introduced to graphics [Cook and Torrance 1982] as

$$f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) = \frac{F(\mathbf{i}, \mathbf{o})G(\mathbf{i}, \mathbf{o})D(\mathbf{h})}{4|\langle \mathbf{i}, \mathbf{n} \rangle||\langle \mathbf{o}, \mathbf{n} \rangle|},$$

where $F(\mathbf{i}, \mathbf{o})$ is the Fresnel term, $G(\mathbf{i}, \mathbf{o})$ is the shadowing-masking term, and $D(\mathbf{h})$ is the normal distribution function. The denominator is a Jacobian of the density change required to transform the radiance and irradiance from their corresponding projected solid angle domains to the tangent plane (domain of microfacets) [Walter et al. 2007].

When working with the halfway vectors, it is convenient to represent them in another domain, a *parallel-plane domain*, or a so-called *slope domain* (see illustration). In this domain, a halfway vector $\mathbf{h}^{\parallel} = \mathbf{h}_{u,v}/\langle \mathbf{h}, \mathbf{n} \rangle$ is a point on a two-dimensional plane that is lifted by one unit above the tangent plane of



A halfway unit vector $\mathbf{h} \in S^2$ and a halfway vector $\mathbf{h}^{\parallel} \in \mathbb{R}^2$ on a plane parallel to the tangent plane lifted up by one unit (parallel-plane domain).

the surface. In this domain, the vector \mathbf{h}^{\parallel} is comprised of two slopes (partial derivatives) of the microfacet heightfield along the u and v axes of the tangent frame. Also, the slope amplitude $\tan \theta_{\mathbf{h}} = |\mathbf{h}^{\parallel}|$ equals the planar distance to the origin. This is a convenient space, as the derivatives of the height field become points on this plane. Therefore, we will use this as the primary space for defining

common NDFs $D(\mathbf{h}^\parallel)$.

Two commonly used models of rough microfacet surface are the Trowbridge-Reitz model [Trowbridge and Reitz 1975] (also known as GGX in graphics [Walter et al. 2007]) and the Beckmann [Beckmann and Spizzichino 1963] model.

The *Beckmann distribution* [Beckmann and Spizzichino 1963] is the first model of rough microfacet surfaces. First, the microfacet heights are assumed to be distributed normally, leading to a closed-form solution for the shadowing-masking term G [Smith 1967]. Second, the slopes are also assumed to be distributed normally. That is, $D(\mathbf{h}^\parallel)$ is a bivariate Gaussian distribution with a diagonal 2×2 covariance matrix $\Sigma = \text{diag}(\alpha_u^2, \alpha_v^2) = \text{diag}(2\sigma_u^2, 2\sigma_v^2)$

$$D(\mathbf{h}) = 2\mathcal{N}(0, \Sigma) \left| \frac{d\mathbf{h}^\parallel}{d\mathbf{h}} \right| = \frac{2}{\pi|\Sigma|^{1/2} \cos^4 \theta_{\mathbf{h}}} e^{-\mathbf{h}^\parallel \Sigma^{-1} \mathbf{h}^\parallel}.$$

Note that the coefficient of two in the numerator originates from the normalization of the NDF with respect to its required properties, thus it is more convenient to use $\alpha^2 = 2\sigma^2$ as a direct roughness parameter.

Note that the Beckmann distribution being a bivariate Gaussian in the domain of slopes, has multiple convenient properties, such as closed-form convolution and multiplication with another Beckmann distribution as well as an easily computable spectrum in the Fourier space.

The *Trowbridge-Reitz distribution* assumes the normals being distributed according to the visible normals of an ellipsoid. It is defined as a distribution of normals of a quadric $\mathbf{h}^\top \Sigma^{-1} \mathbf{h}$ [Neyret 1998, Walter et al. 2007] as

$$D_{\text{TR}}(\mathbf{h}) = \frac{1}{\pi|\Sigma|^{1/2} (\mathbf{h}^\top \Sigma^{-1} \mathbf{h})^2} \left| \frac{d\mathbf{h}^\parallel}{d\mathbf{h}} \right| = \frac{\alpha_u^2 \alpha_v^2}{\pi \left(\alpha_u^2 \mathbf{h}_u^2 + \alpha_v^2 \mathbf{h}_v^2 + \alpha_u^2 \alpha_v^2 \right)^2 \langle \mathbf{h}, \mathbf{n} \rangle^4},$$

where the 3×3 diagonal matrix $\Sigma = \text{diag}(\alpha_u^2, \alpha_v^2, 1)$ is the descriptor of the ellipsoid's principal axes $(\alpha_u, \alpha_v, 1)$ in the tangent frame of the surface. Such an ellipsoid has always a unit vertical axis due to the NDF normalization requirement. This distribution has a heavy tail, i.e., it falls off slower than exponentially. This means

that given the same roughness parameters $\alpha_{u,v}$, the Trowbridge-Reitz distribution produces smoother and wider highlights compared to the Beckmann distribution. Since the shadowing-masking term is derived from a particular microfacet distribution, one can also use a corresponding shadowing-masking term G_{TR} [Walter et al. 2007] in the Cook-Torrance BSDF model together with the Trowbridge-Reitz distribution.

2.2.5 Reciprocity and Self-Adjointness

Some light transport scenarios are more convenient to solve by simulating direct light propagation from the light source to the sensor, while other scenarios are more efficient to solve in a reverse manner, by propagating importance, a virtual quantity, from the camera to the light source. Therefore, it is essential to have reciprocal scattering distributions and self-adjoint operators in order to be able to freely change the propagation direction on demand. It enables direction-independent light transport simulation.

The physical process of light transport is reciprocal by nature, in a sense that one can exchange the light source with the observer and the observed energy flow will remain the same. In order to use this physical property, one needs to carefully define the scattering process and scattering distributions such that:

- Scattering operators, such as the integral in Eq. 2.3, should be self-adjoint (symmetric with respect to the propagation direction).
- Scattering distributions should also be symmetric with respect to the propagation direction.

Self-adjointness of scattering operators breaks on a refractive interface, but can be achieved by propagating *basic radiance* $L' = L/n^2$ instead of radiance and changing the corresponding measures [Veach 1998, p.203]. In practice, instead of propagating the basic radiance, the implementation usually updates the ratio of refractive indices every time a path passes an interface with different indices of refractions.

The Symmetry of scattering distributions

$$f_s(\mathbf{i}, \mathbf{o})/n_o^2 = f_s(\mathbf{o}, \mathbf{i})/n_i^2$$

is achieved by making sure they are self-adjoint at the design phase or during measurement. All distributions considered in the present thesis are symmetric.

There are more sources of non-symmetric scattering, such as shading normal ¹. We refer to Chapter 5 in the thesis of Eric Veach [1998] for an exhaustive description as well as solutions for these problems.

2.2.6 Path Integral Formulation and Measurement Equation

In this section we focus on the rendering equation (Eq. 2.3) and show how to reformulate it from the recursive integral form into an explicit integration problem [Veach 1998, Chapter 8] based on Feynman's path integral framework [Feynman and Hibbs 1965].

2.2.6.1 Chain Formulation of Rendering Equation

We start with the rendering equation that describes scattering at a single surface interaction as

$$L(\mathbf{x}, \mathbf{o}) = \int_{S^2} f_s(\mathbf{x}, \mathbf{i}, \mathbf{o}) L(\mathbf{i}) d\mathbf{i}^\perp.$$

We expand the incident radiance as the outgoing radiance at the point it originates from as

$$L_i(\mathbf{x}, \mathbf{i}) = L_o(\mathbf{y} \equiv \chi_{\mathcal{M}}(\mathbf{x}, \mathbf{i}), \mathbf{o}' \equiv -\mathbf{i}),$$

where $\chi_{\mathcal{M}}(\mathbf{x}, \mathbf{i})$ is the ray tracing operator that returns the closest surface along the ray (\mathbf{x}, \mathbf{i}) . This way, the rendering equation can be expanded recursively all the way to the light source.

Since we are eventually interested in measuring the outgoing radiance in order

¹Shading normals are used to define a fictitious tangent bundle on a surface. This bundle is used for shading, while the underlying geometry can have a different, coarser, tangent bundle.

to obtain an image, we also add the measurement equation at the sensor end of the path as

$$I^j = \int_{\mathcal{M} \times \mathcal{S}^2} W^j(\mathbf{x}, \mathbf{i}) L(\mathbf{x}, \mathbf{i}) d\mathbf{x} d\mathbf{i}^\perp, \quad (2.8)$$

where I^j is the total incident flux onto the j th sensor element (e.g., a pixel of an image), \mathcal{M} is the manifold of all surfaces in the scene, $W(\mathbf{x}, \omega)$ is the sensor responsivity at the point \mathbf{x} towards the direction ω , and the integration happens over all scene surfaces¹ and all directions in order to measure flux at the sensor element. This is a special case of rendering equation where the last interaction is a terminating interaction (energy sink) and which also requires integrating over the area of the sensor element in order to obtain total incident flux.

Next, we change the measurement domain from projected solid angle $d\mathbf{i}^\perp$ of an incident direction \mathbf{i} at a surface point \mathbf{x} to surface area measure $d\mathbf{y}$ at another surface point along the direction \mathbf{i} .

The transformation Jacobian from the projected solid angle measure at \mathbf{x} to the area measure at \mathbf{y} is called a *geometric term*²

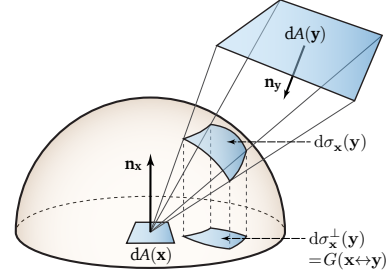
$$G(\mathbf{x} \leftrightarrow \mathbf{y}) = \left| \frac{d\mathbf{i}_\mathbf{x}^\perp}{d\mathbf{y}} \right| = \frac{|\langle \mathbf{i}, \mathbf{n}_\mathbf{x} \rangle| |\langle \mathbf{i}, \mathbf{n}_\mathbf{y} \rangle|}{\|\mathbf{x} - \mathbf{y}\|^2} V(\mathbf{x}, \mathbf{y}). \quad (2.9)$$

If we switch from solid angle domain, where we trace paths, into a pure on-surface integration domain, we also have to account for the binary *visibility term* $V(\mathbf{x}, \mathbf{y})$ that takes values of one whenever points \mathbf{x} and \mathbf{y} are mutually visible and zero otherwise. In presence of participating media, this term can take non-binary values due to volumetric transmittance; for the limiting case surfaces in a vacuum, visibility takes only binary values.

¹Note that an arbitrary region of the space can be a sensor. For the rest of the space, the sensor responsivity $W \equiv 0$.

²Also called *geometry term* in the literature.

Another intuitive explanation of the geometric form factor between $dA(\mathbf{x})$ and $dA(\mathbf{y})$ is provided by the *Nusselt analog*, the visible solid angle of the patch $dA(\mathbf{y})$ from point \mathbf{x} , which is projected from unit sphere onto the tangent plane at \mathbf{x} . The projection of $d\mathbf{y}$ onto the unit sphere incorporates the second dot product and the reciprocal squared distance term, while the projection onto the disk corresponds to the first dot product.



The Nusselt analog is a popular way of illustrating the geometric interpretation of the transformation performed by geometric term.

After changing the integration domain, the rendering equation is described by three consequent interaction points $\mathbf{x}, \mathbf{y}, \mathbf{z}$ (\mathbf{y} being the central point) instead of a single central point \mathbf{x} and two directions \mathbf{i} and \mathbf{o} , and it can be written as

$$L(\mathbf{y} \rightarrow \mathbf{z}) = \int_{\mathcal{M}} f_s(\mathbf{x} \rightarrow \mathbf{y} \rightarrow \mathbf{z}) L(\mathbf{x} \rightarrow \mathbf{y}) G(\mathbf{x} \leftrightarrow \mathbf{y}) d\mathbf{x},$$

where integration now happens over all scene surfaces \mathcal{M} instead of a sphere of directions; and the BSDF takes three points instead of a point and two directions (directions can be trivially computed). Similarly, Eq. 2.8 can be transformed into a three-point integral.

Now that we have everything transformed into the domain of surface points, we can combine multiple interactions into a single long integral. For example, we can write an expression for the first bounce of indirect illumination, where the path starts from the light, then after two surface interactions it hits the sensor

$$I_4 = \int_{\Omega_4(\mathcal{M})} L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1) G(\mathbf{x}_0, \mathbf{x}_1) f_s(\mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \mathbf{x}_2) G(\mathbf{x}_1, \mathbf{x}_2) \times \\ f_s(\mathbf{x}_1 \rightarrow \mathbf{x}_2 \rightarrow \mathbf{x}_3) G(\mathbf{x}_2, \mathbf{x}_3) W(\mathbf{x}_2, \mathbf{x}_3) d\mathbf{x}_0 d\mathbf{x}_1 d\mathbf{x}_2 d\mathbf{x}_3,$$

where I_4 denotes the flux caused only by the paths of length four (first indirect bounce), i.e., from all paths with four interactions $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$. The integration is performed over the space $\Omega_4(\mathcal{M}) = \mathcal{M} \times \mathcal{M} \times \mathcal{M} \times \mathcal{M}$ of all paths of length four, which is a quadruple Cartesian product of scene manifold \mathcal{M} with itself.

2.2.6.2 Path Integral Formulation

We can combine the measurements I_k for different path lengths k into a single *path integral* that expresses the function of camera response measurement for a pixel I in the form of the integral that is defined over the *measurement contribution function* $f(\mathbf{X})$ on the product surface area measure $d\mathbf{X}$ [Veach 1998, p.223] as

$$I = \int_{\Omega} f(\mathbf{X}) d\mathbf{X}, \quad \text{with } d\mathbf{X} = \prod_{i=0}^{|\mathbf{X}|} d\mathbf{x}_i, \quad (2.10)$$

where $\Omega = \Omega(\mathcal{M}) = \bigcup \Omega_k(\mathcal{M})$ is the unified space of all possible light paths of all lengths; $\mathbf{X} \in \Omega$ is a complete path from light to camera in path space, which is represented as a vector of points on the scene manifold \mathcal{M} , i.e. $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k)_{k=1 \dots \infty}$; I is a camera sensor measurement for a single sensor element (e.g., image pixel); $f(\mathbf{X})$ is the measurement contribution function, which includes a product of geometric terms and BSDFs at all points of an argument path \mathbf{X} of length k and has a form

$$f(\mathbf{X}) = L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1) G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1) \left(\prod_{j=1}^{k-1} f_s(\mathbf{x}_{j-1} \rightarrow \mathbf{x}_j \rightarrow \mathbf{x}_{j+1}) G(\mathbf{x}_j \leftrightarrow \mathbf{x}_{j+1}) \right) W(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k), \quad (2.11)$$

where $G(\mathbf{x}_j \leftrightarrow \mathbf{x}_{j+1})$ is the geometric factor between the points \mathbf{x}_j and \mathbf{x}_{j+1} ; $f_s(\mathbf{x}_{j-1} \rightarrow \mathbf{x}_j \rightarrow \mathbf{x}_{j+1})$ is the BSDF at the point \mathbf{x}_j ; $L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1)$ is the emitted radiance at the point \mathbf{x}_0 on the light source and $W(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k)$ the sensor responsivity at the point \mathbf{x}_k on the sensor. Please refer to [Veach 1998] for more details on the path integral formulation. The measurement contribution function $f(\mathbf{X})$ measures the amount of differential flux Φ flowing through infinitesimal areas at every surface interaction

$$f(\mathbf{X}) = \frac{d\Phi}{d\mathbf{x}_0 \dots d\mathbf{x}_k} \cdot \left[\frac{W}{m^{2(k+1)}} \right]$$

The resulting path integral is convenient for numerical integration, as, unlike the

rendering equation, it is not a recursive equation, therefore, has an explicit form that is suitable for conventional integration methods. Importantly, this integral also unifies paths of different length into a single integral using a shared *physical* framework. This formulation enables direct comparison of two paths \mathbf{X}_1 and \mathbf{X}_2 of different lengths ($|\mathbf{X}_1| = k_1 \neq k_2 = |\mathbf{X}_2|$) in terms of the amount of carried differential flux using the differential patches $d\mathbf{x}_0 \dots d\mathbf{x}_k$ and the corresponding measurement contribution functions $f(\mathbf{X}_1)$ and $f(\mathbf{X}_2)$ for these paths.

2.2.7 Path Measurement Domains

We have considered a change of variables from projected solid angle measure $d\mathbf{i}^\perp$ to surface area measure $d\mathbf{x}$ in order to transform the rendering equation into the three-point integral over the scene surfaces. This is one of many possible changes of variables. However, before introducing other possible domains, we first formalize the change of variables in a path integral of variable dimensionality in Eq. 2.10.

Let us consider the aforementioned change of variables as an example. It is possible to write the path integral in the solid angle domain¹

$$\int_{\Psi} f^\omega(\mathbf{X}^\omega) d\mathbf{X}^\omega, \quad (2.12)$$

where $\mathbf{X}^\omega = (\mathbf{x}_0, \mathbf{o}_0, \mathbf{o}_1, \dots, \mathbf{o}_{k-1})_{k=1 \dots \infty}$ is a path of length k that conventionally starts at the light source point \mathbf{x}_0 and emits in the direction \mathbf{o}_0 and so on until it hits the sensor; $d\mathbf{X}^\omega = d\mathbf{x}_0 \times d\mathbf{o}_0 \times \dots \times d\mathbf{o}_{k-1}$ is the corresponding product measure; the integration is performed over the unified domain $\Psi = \bigcup \Psi_k$ and $\Psi_k = \mathcal{M} \times \prod_{i=0}^{k-1} \mathcal{S}^2$ of spheres for every interaction. Note that every new interaction point is determined by the previous directions.

Now, we can change the integration domain from this product solid angle measure to product area measure. The path integral for light transport is always finite-dimensional with varying number of dimensions. Therefore, a regular change of

¹We will not provide a full expansion of the path integral in this domain due to its bulkiness, however, the main derivation steps include using the ray tracing operator, moving the cosine out of the $d\mathbf{i}^\perp$ measure and writing a recursive integral.

variables applies

$$\int_{\Psi} f^{\omega}(\mathbf{X}^{\omega}) d\mathbf{X}^{\omega} = \int_{\Omega} f^{\omega}(\mathbf{X}^{\omega}) |J_{\mathbf{X}^{\omega} \rightarrow \mathbf{X}}| d\mathbf{X},$$

where we changed the integration domain and the integration measure, but not the integrand itself. The corresponding Jacobian for this change of variables is trans-dimensional and for a given path of length k takes a form

$$J_{\mathbf{X}^{\omega} \rightarrow \mathbf{X}} = \begin{matrix} & \mathbf{o}_0^{\perp} & \mathbf{o}_1^{\perp} & \cdots & \mathbf{o}_{k-1}^{\perp} \\ \mathbf{x}_1 & \left(\frac{d\mathbf{o}_0^{\perp}}{d\mathbf{x}_1} \right) & \left(\frac{d\mathbf{o}_1^{\perp}}{d\mathbf{x}_1} \right) & \cdots & \left(\frac{d\mathbf{o}_{k-1}^{\perp}}{d\mathbf{x}_1} \right) \\ \mathbf{x}_2 & \vdots & \vdots & \ddots & \left(\frac{d\mathbf{o}_{k-1}^{\perp}}{d\mathbf{x}_2} \right) \\ \vdots & \vdots & \vdots & \cdots & \cdots \\ \mathbf{x}_k & \left(\frac{d\mathbf{o}_0^{\perp}}{d\mathbf{x}_k} \right) & \left(\frac{d\mathbf{o}_1^{\perp}}{d\mathbf{x}_k} \right) & \cdots & \left(\frac{d\mathbf{o}_{k-1}^{\perp}}{d\mathbf{x}_k} \right) \end{matrix}.$$

By construction of the integral, all previous interactions are independent from the next interactions, that is, $d\mathbf{o}_i^{\perp}/d\mathbf{x}_j = 0$ for all $i < j$. Therefore, this Jacobian matrix $J_{\mathbf{X}^{\omega} \rightarrow \mathbf{X}}$ is of an upper triangular form, and thus the determinant is the product of diagonal terms

$$|J_{\mathbf{X}^{\omega} \rightarrow \mathbf{X}}| = \left| \prod_{i=0}^{k-1} \frac{d\mathbf{o}_i^{\perp}}{d\mathbf{x}_{k+1}} \right| = \prod_{i=0}^{k-1} G_{\mathbf{x}_i \rightarrow \mathbf{x}_{k+1}}.$$

This leads to the measurement contribution function $f(\mathbf{X})$ we derived earlier

$$f^{\omega}(\mathbf{X}^{\omega}) |J_{\mathbf{X}^{\omega} \rightarrow \mathbf{X}}| = f(\mathbf{X}).$$

We can also write a form of Eq. 2.12 starting from the sensor instead. Then the Jacobian would be lower triangular (as opposed to the upper triangular). However, this would not change the last two equalities and would lead us to the same result.

For many possible changes of variables, the Jacobian matrix can be of a full form. This involves a prohibitive quadratic computational cost for a path length k for computing the full Jacobian. There are also domains, like a domain of halfway vectors described in Section 7.2.1, where this Jacobian matrix is sparse and the integrand has interesting properties.

2.3 Specifics of Image Synthesis with Light Transport

Prior to this section, we mostly discussed generic methods of measuring flux incident to arbitrary sensors. When applying light transport to image synthesis, one usually uses a camera model in order to synthesize an image. The process essentially simulates virtual photography by computing light transport process. The simplest and most popular camera model is the *pinhole camera*, where the outside world is seen through a pinhole¹ in a planar curtain that is parallel to the image plane. The image plane is a virtual film backplate separated into a usually regular grid of *pixels*, tiny square sensors that capture the incident flux throughout the shutter time (a short period of measurement time when the camera shutter is open).

Every pixel j requires to conduct a separate measurement I^j according to Eq. 2.8 with its own associated responsivity function $W^j(\mathbf{x}, \omega)$. This is a challenging problem of image synthesis with light transport, as it requires to simulate enough paths through every pixel of the sensor in order to obtain representative measurements for all pixels of the image.

More advanced physically based camera models also model other parts of a camera [Kolb et al. 1995], including lens body. These models can simulate advanced camera effects, such as depth of field and anamorphic lenses. We refer an interested reader to the recent advances in camera models [Hanika and Dachsbacher 2014].

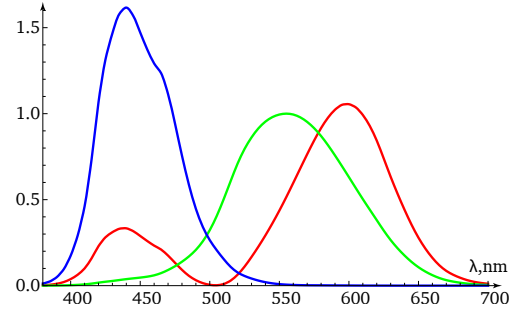
2.3.1 Specifics of Human Vision

The human visual system is an advanced hybrid image capturing, processing, and analysis system to the date, consisting of a precise optical system, single-photon sensitive sensor (retina) with tristimulus-based color detection, on-sensor signal processing and compression, as well as high-level image analysis system in the brain, such as relational spatial recognition and object recognition [Wandell 1995]. Here we focus only on rendering-related specifics of human perception.

¹Mathematically, it is defined as an infinitesimal hole in a planar aperture of the camera.

Color Perception. One important property is color perception. The human retina comprises three types of cones, the responsivity $W(\lambda)$ of the eye depends on the wavelength.

This sensitivity to different wavelengths vs. energy carried by these wavelengths is measured for XYZ primary colors in CIE standard and depicted on the right. Based on this data, image synthesis is oftentimes simplified from a full-fledged spectral light transport to transporting the packets of (red, green, blue) triplets of spectra that correspond to spectral responses of three types of retinal cones. This simplification can lead to rendering errors in some cases and might require additional corrections [Meng et al. 2015].



Retinal tristimulus (red, green, and blue) spectral sensitivity curves in XYZ color space as seen by the CIE Colorimetric Observer [1931].

Intensity Perception. Perceived luminance is computed by weighting the incident spectrum with the responsivity curves of all three cones. Perceived luminance is useful in light transport, e.g., for comparing the relative importance of light paths carrying different spectra according to their perceptual importance.

The human visual system perceives the luminance non-linearly, according to a *Weber-Fechner law*, a power curve $I_{\text{perceived}} = I^\gamma$. This curve relates the actual intensity I to the perceived intensity, which follows the Stevens' power law between the physical stimulus and its perceived intensity. Typical values for γ vary from one display standard to another and are usually in the range of 0.45–0.55. Multiple image compression and rendering techniques use this law to store and present the image in a more perceptually efficient way.

Contrast. The human eye is sensitive to local contrast changes in the image, as well as to silhouettes of objects. We refer an interested reader to a detailed description of human perception [Williamson and Cummins 1983].

A more involved characteristic of human visual system is the ability to discern between luminance levels in a static picture. The *contrast sensitivity function*, relates the frequency of the changing luminance to its contrast (relative difference between luminance levels) as depicted in Figure 2.13. An interesting corollary is, for a low-resolution or low-contrast image one can increase its local contrast in order to improve visual acuity.

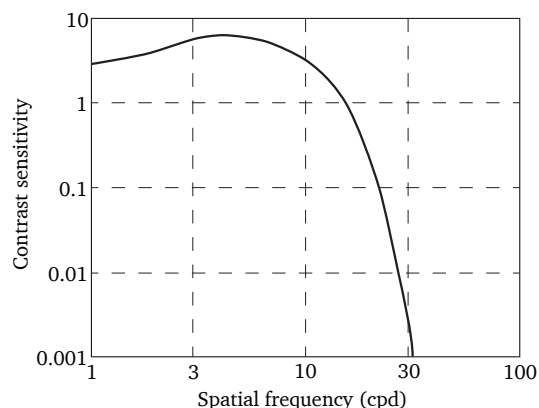


Figure 2.13: The ability to detect contrast differences (vertical) as a function of the angular frequency (horizontal, cycles/degree) [Yellott et al. 1980].

One recent work that employs perceptual sensitivity of edges in rendering is gradient-domain Metropolis light transport [Lehtinen et al. 2013], which focuses more computational power on image gradients. The final image is then reconstructed by solving a Poisson equation.

Another practical approach to local contrast enhancement of an image, called unsharp masking, where a blurred (unsharp) version of the image is masked onto the original image, was applied in rendering [Ritschel et al. 2008].

The *tone mapping* techniques, such as [Reinhard et al. 2002], are used to compress the high dynamic range of intensities to an image displayable on devices with lower dynamic range. These techniques rely on contrast sensitivity function and are usually designed to preserve image contrast.

2.3.2 Image Convergence

A modern image conveys millions of pixels and the demand for increasing the number of independent measurements is ever growing with increasing resolution and advances in displays. Moreover, if a pixel measurement has high error, human perception can interpret it as a high contrast region, which is unacceptable for image

quality. In fact, human perception is more sensitive to measurement error than many mathematical error metrics (such as the L_2 metric) commonly used in radiative and neutron transport.

On the other hand, if the error is distributed uniformly across all measurements and is significantly lower than image signal, then human perception is more forgiving to such types of image error, even if the average mathematical error (such as a mean squared error) is larger.

Therefore, it is important to have a uniform convergence across the image. In order to achieve uniform error distribution across the image plane it is essential to collect enough representative paths for every pixel measurement. Practical rendering methods should account for this requirement at the early stages of design by handling the error across the image and amortizing the computational cost by exploring the coherence of the light field within local regions of the image.

One approach is to use uniform, stratified and low-discrepancy distributions of samples across image plane [Pharr and Humphreys 2010, Chapter 7]. Another approach used in stochastic integration methods, such as Monte Carlo, is called *adaptive sampling*. It addresses the maximum pixel error across the image by adaptively refining the measurements with the highest error to achieve uniform convergence. We will also pay attention to the requirement of keeping the image error uniform and low in the proposed methods.

Integration Methods

Formulating the light transport measurement problem as a conventional integration problem opens the door to the existing integration methods. In this chapter, we classify existing light transport integration methods.

3.1 Deterministic Integration Methods

This section includes several non-stochastic integration methods. Generally, these methods are more challenging to apply because of the *curse of dimensionality*¹ for the high-dimensional light transport integration problem.

3.1.1 Finite Element Methods

The class of *finite element* methods [Hrennikoff 1941] approximates the integration problem in Eq. 2.10 by partitioning the integration domain into a finite number of subdomains and applying Riemann sums as an approximate solution, such that the solution for each subdomain (element) is easily computable yet the overall approximation error stays reasonably low. In the context of light transport integration, the most popular finite element method is called *radiosity* [Goral et al. 1984, Heckbert and Winget 1991], where the scene geometry \mathcal{M} is partitioned into a finite set of N disjoint simplified elements $\mathcal{M} \approx \bigcup_i^N \mathcal{E}_i$. The idea is to choose planar patches as the approximate elements such that the irradiance caused by the element \mathcal{E}_i onto

¹The volume of the problem space grows much faster than the effective volume that contains the important signal, thus decreasing the efficiency and complicating the localization of the signal.

\mathcal{E}_j can be easily computed as

$$E_{\mathcal{E}_i \rightarrow \mathcal{E}_j} = \int_{\mathcal{E}_i} f_s(\mathbf{x}_h \rightarrow \mathbf{x}_i \rightarrow \mathbf{x}_j) G(\mathbf{x}_i, \mathbf{x}_j) L(\mathbf{x}_h \rightarrow \mathbf{x}_i) d\mathbf{x}_i. \quad (3.1)$$

Next, one assumes that all elements are Lambertian reflectors, therefore the dependence on \mathbf{x}_h is removed. The method then consists of the following steps:

- All patches are initialized with the direct irradiance from the light sources.
- A matrix system with all elements $E_{\mathcal{E}_i \rightarrow \mathcal{E}_j}$ is solved for each bounce to advance the light propagation.
- After a finite number of iterations, the resulting irradiance is reconstructed for every directly observable point.

Therefore, radiosity can be considered a finite Riemann sum for the path integral with the aforementioned assumptions and approximations. Multiple extensions were also developed, including hierarchical approaches, extension to glossy materials, and Galerkin methods.

Lattice-based methods. The idea behind *Discrete Ordinates Methods* (DOM) [Chandrasekhar 1960] is to discretize the radiance in the radiative transfer equation (RTE) in space and/or in orientation (DOMs are typically used for computing radiative transfer in participating media). These discrete values are used to approximate the RTE: the radiance distribution is stored in a 3D grid, and light is propagated in a finite set of directions by the exchange between neighboring volume elements, reducing the computation to local neighborhood only.

3.1.2 Ray Transfer Matrices

Another important method used for computing specular light transport, especially through optical systems, is called the *ABCD matrix ray transfer matrix* method [Gerard and Burch 1975]. Specular transport (no scattering) makes the system deterministic in a sense that for every ray in such system there is only a small finite

number of outgoing rays. This method relies on the assumption of *paraxial optics*, where the angle θ between the ray and the optical axis of the system is assumed to be small ($\theta \approx \sin \theta \approx \tan \theta$), that is, the line along which there is a rotational symmetry of the system (e.g., center of a lens).

Therefore, the idea is to treat the optical system as a black box and parameterize the outgoing ray as a first-order expansion of the input ray as

$$\begin{pmatrix} \mathbf{x}_o \\ \mathbf{o}^\parallel \end{pmatrix} \approx \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \mathbf{x}_i \\ \mathbf{i}^\parallel \end{pmatrix}. \quad (3.2)$$

This is a linear relation between a ray $(\mathbf{x}_i, \mathbf{i}^\parallel)$ that enters the optical system and the ray $(\mathbf{x}_o, \mathbf{o}^\parallel)$ that leaves it. The incident \mathbf{i}^\parallel and outgoing \mathbf{o}^\parallel directions are parameterized in parallel plane domain (see Figure 2.10). This domain is more convenient because many operators have an exact linear form in it. For example, the operator that propagates an incident ray by a distance d has an exact matrix form

$$S = \begin{pmatrix} \mathbf{1} & d \\ \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

The top left matrix (A) in Eq. 3.2 tracks the compression of a differential patch orthogonal to the ray direction. The determinant $|A|$ provides the first-order change (compression/decompression) of the radiance along the ray.

This approach was employed in graphics, e.g., in pencil tracing [Shinya et al. 1987], for rendering various specular effects, such as reflection and refraction. We will discuss the generalization of ray transfer matrices to light transport with arbitrary scattering along a path in Chapter 7.

3.2 Monte Carlo Path Integration

The deterministic integration methods have higher computational cost in a high-dimensional space due to their design. For example, given an integrand of d variables, with deterministic methods we need to somehow subdivide each dimension

into N pieces and evaluate the function at N^d points to bound the integration error. We can see that the computational complexity grows exponentially with number of dimensions, which is referred to as the *curse of dimensionality*. The Monte Carlo integration method was first proposed by Metropolis and Ulam [Metropolis and Ulam 1949] and transforms the integration into a stochastic process. The convergence of this integration, as we will see, does not depend on the number of dimensions.

3.2.1 Monte Carlo Integration

Monte Carlo integration can be derived from a regular integration with the following steps. First, we interpret the integration problem as a problem of finding an expected value

$$\int_{\Omega} f(\mathbf{X}) d\mathbf{X} = \int_{\Omega} \frac{f(\mathbf{X})}{p(\mathbf{X})} p(\mathbf{X}) d\mathbf{X} = E \left[\frac{f(\mathbf{X})}{p(\mathbf{X})} \right],$$

where a random variable $\mathbf{X} \propto p(\mathbf{X})$ is distributed on the integration domain Ω with respect to some probability density function p^1 , which takes positive values on the entire support of the integrand.

In order to sample the probability density, we can use its *Cumulative Distribution Function* (CDF) $F(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{y}) d\mathbf{y}$. This integration is performed in the high-dimensional domain Ω . Assuming we can generate a high-dimensional uniform random vector $\Xi = (\xi_1, \dots, \xi_k) \in \mathcal{U}^k(0; 1)$, we can use the inverse CDF of p to transform this vector into a random vector \mathbf{X} distributed according to p as follows

$$\mathbf{X}_i = F^{-1}(\Xi) \propto p(\mathbf{X}). \quad (3.3)$$

Then, we can estimate this expected value by drawing a sample $(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$ of size N with the density p to evaluate the empirical average of the expectation

$$\bar{f}_N = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{X}_i)}{p(\mathbf{X}_i)}. \quad (3.4)$$

¹As we defined it before, the probability density function is a normalized measure on Ω .

This sum is called an *estimator* of f and converges almost surely to the actual expectation E as $N \rightarrow \infty$ due to the Law of Large Numbers.

The ratio $\frac{f(\mathbf{X})}{p(\mathbf{X})}$ is called a *throughput* and is an important value that is practically computed during integration, as well as used for efficiency analysis of Monte Carlo methods. In the context of light transport, this ratio is called a *path throughput*.

Integration Error and Convergence Rate. Given that $f^2(\mathbf{X})$ has finite expectation under p , we can write the variance of the estimator \tilde{f}_N in Eq. 3.4 as

$$\text{Var}[\tilde{f}_N] = \frac{1}{N^2} \sum_{i=0}^N \text{Var} \left[\frac{f}{p} \right] = \frac{\sigma_{f/p}^2}{N},$$

where $\sigma_{f/p}^2 = \text{Var}[f/p]$ is the variance of the throughput f/p . By employing the Law of Large Numbers again, for sufficiently large N the integration error of the original estimator in Eq. 3.4 is bounded by¹

$$|E[f] - \tilde{f}_N|^2 \leq \sigma[\tilde{f}_N] = \sqrt{\text{Var}[\tilde{f}_N]} = \frac{\sigma_{f/p}}{\sqrt{N}}.$$

This fundamental result shows that the estimation error depends on the variance of the ratio f/p being estimated as well as the number of samples N and vanishes as $O(1/\sqrt{N})$, that is, the error vanishes with a square root of number of samples N (i.e., in order to reduce the error twice, one should quadruple the number of sample). Secondly, the asymptotic convergence of the Monte Carlo integration method is independent of the number of integration dimensions, thus providing good practical properties for high-dimensional integration and alleviating the curse of dimensionality.

On the other hand, if the distribution p is disproportional to the target function f , the variance $\sigma_{f/p}$ of the estimated ratio can take high values, which can cause the actual error being large, requiring impractically many samples for convergence. In other words, the Monte Carlo method reformulates integration in a form that is more suitable for complex integrands and high dimensions by providing the con-

¹An expanded derivation can be found, e.g., in [Robert and Casella 2005].

trol over the new integration measure p . The goal is then to make this measure approximately proportional to the target function in order to reduce the variance $\sigma_{f/p}^2$. We discuss methods for reducing the variance of the estimated value $\sigma_{f/p}$, e.g., using importance sampling, in Section 3.2.3.

3.2.2 Application to Light Transport

We have deliberately used the same notation throughout the introduction to Monte Carlo methods as in the path integral formulation of light transport to emphasize that we can immediately apply the Monte Carlo method for light transport problem.

As a canonical example, we can sample N paths $(\mathbf{X}_1, \dots, \mathbf{X}_N)$ from a uniform distribution p on the path space Ω . First, in order to select the subspace Ω_k of a constant dimensionality, we sample the path length k . The physical process of light transport does not contain infinite paths due to the unavoidable energy loss during the physical process. Therefore, the path length k can be practically limited to some k_{\max} , thus we can draw k from a uniform distribution $k \propto p_k = 1/k_{\max}$. Once the path subspace Ω_k is chosen, we need to sample $k + 1$ vertices uniformly on the scene manifold \mathcal{M} according to the integration measure $d\mathbf{X}_k = d\mathbf{X}_0 \dots d\mathbf{X}_k$. That is, each vertex \mathbf{x}_i is selected on the scene manifold according to the probability $p(\mathbf{x}_i) = 1/dA(\mathcal{M})$, where both the sensor and the light sources are also scene surfaces. Once this is done, we have constructed a path that is uniformly sampled according to the integration measure $d\mathbf{X}$, so we can simply evaluate the path measurement contribution at every sampled path \mathbf{X} and use the Monte Carlo estimator in Eq. 3.4 to numerically integrate the desired measurement.

This is an example of a Monte Carlo light transport method. Even though it is a valid method, it is rarely used in practice due to the high variance $\text{Var}[f/p]$ in this space. There are multiple sources of variance, e.g., one source is the chain product of geometric terms G : its visibility subterm V can often turn into zero, especially for highly occluded scenes, because we sample every vertex independently without taking into account the visibility to its predecessor and successor vertices; the inverse squared distance term can also cause numerical instabilities [Keller 1997]

whenever two adjacent vertices are spatially too close to each other. Another source of potentially high variance stems from the BSDF terms f_s at every vertex. For example, in a scene with many highly glossy surfaces, it is important to sample path vertices in a way all incident and outgoing directions are close to the angles of a perfect specular interaction.

In the following section we discuss mathematical techniques of reducing the variance for a generic integration, while in Chapter 4 we discuss how to reduce variance specific to light transport using various path construction strategies.

3.2.3 Variance Reduction

The majority of the integration error reduction techniques in Monte Carlo are focused on distributing the samples such that the variance $\sigma_{f/p}^2$ is reduced.

Importance Sampling. The variance $\sigma_{f/p}^2 = \text{Var}[f/p]$ of the estimated function f/p is

$$\sigma_{f/p}^2 = \text{Var} \left[\frac{f(\mathbf{X})}{p(\mathbf{X})} \right] = \int_{\Omega} \left(\frac{f(\mathbf{X})}{p(\mathbf{X})} - E \left[\frac{f(\mathbf{X})}{p(\mathbf{X})} \right] \right)^2 p(\mathbf{X}) d\mathbf{X}, \quad (3.5)$$

and is subject to minimization by p .

The most popular way of reducing the variance $\sigma_{f/p}^2$, called *importance sampling*, consists of finding a suitable *proposal probability density* function p . It should be easy to sample from, while it should mimic the behavior of the target function as close as possible, i.e., cover the important regions of the target function f .

One idealized example of importance sampling, when the proposal is set to $p(\mathbf{X}) = f(\mathbf{X})/c$, is called *perfect importance sampling* or *zero-variance importance sampling*. Here $c = \int_{\Omega} f(\mathbf{X}) d\mathbf{X}$ is the normalization constant for this ideal probability density. The variance $\sigma_{f/p}^2$, and thus the integration error, becomes zero in this case, which is easy to see from Eq. 3.5. However, the catch here is that the constant c requires to compute the solution to the original problem of integration.

Moreover, in order to sample from p one needs to invert the high-dimensional cumulative distribution function of p according to Eq. 3.3, which is usually impractical due to the underlying complexity of the function f . Although, the ideal sampling is not practical, it provides a valuable insight into how a good density should be constructed.

Another example of importance sampling commonly used in the context of light transport is called *joint importance sampling*. The idea is to importance-sample only some subterms of the target function. First, we assume the function f can be separated on its support into

$$f(\mathbf{X}) = g(\mathbf{Y}) * h(\mathbf{Z}),$$

where functions g and h are lower-dimensional subparts of the function f and $\mathbf{X} = \mathbf{Y} \parallel \mathbf{Z}$ is obtained by concatenating the lower-dimensional vectors \mathbf{Y} and \mathbf{Z} .

Then, given that the normalizations of g and h are known or computable, we can attempt to construct importance functions $q(\mathbf{Y}) = g(\mathbf{Y})/\|g(\mathbf{Y})\|$ and $r(\mathbf{Z}) = h(\mathbf{Z})/\|h(\mathbf{Z})\|$, so that the joint importance function can be constructed as $p(\mathbf{X}) = q(\mathbf{Y})r(\mathbf{Z})$. The overall variance $\sigma_{f/p}^2$ is then reduced by the corresponding variances of the importance-sampled subterms of f . This technique is also useful in case the target function f is complex, and thus the efficient importance sampling of the full function is not feasible. We show later that most of the existing path construction methods sample some terms of the path measurement contribution function, but not others.

Multiple Importance Sampling. As follows from the name, the idea is to use multiple proposal densities for importance sampling. This is useful if we have a set of J *alternative* proposal densities ($p_1, ..p_J$) to sample the same element \mathbf{X} , where each density is similar to some important places of the target function f , but not the others. *Multiple importance sampling* (MIS) allows to efficiently construct a combined proposal out of the set of multiple complementary proposals. Having multiple proposals is a common situation in light transport, because every light path can be constructed in multiple alternative ways.

The first and the simplest interpretation of MIS [Veach and Guibas 1995] is to statistically model it as a *deterministic mixture model*. In this model we consider n_j samples from every proposal p_j , resulting in a virtual pool of $n = \sum_{j=1}^J n_j$ samples. This pool is a mixture proposal p , which is used for resampling. In practice, this virtual pool is never allocated, instead a proposal p_j is selected according to its selection probability $q_j = n_j / \sum_j n_j$ and then the final sample is drawn from it, leading to a *sampling* single-sample model. Moreover, the number of samples n_j influences the efficiency insignificantly [Veach and Guibas 1995], so in practice a uniform mixture model is used by setting all $n_j \equiv 1$.

We can then compute a *combined sample density* p for a sample \mathbf{X} as a joint probability density

$$p(\mathbf{X}) = \sum_{j=1}^J q(j) p_j(\mathbf{X}) = \frac{1}{J} \sum_{j=1}^J p_j(\mathbf{X}),$$

where $q(j) = 1/J$ is a uniform probability mass function of resampling the technique j from the virtual pool. Every density in this mixture is weighted by the factor $1/J$, which is a small price to pay for combining multiple good complementary proposals, so that the combined mixture can still take high values in important regions of all proposals p_j .

The Monte Carlo estimator can then be written as

$$\bar{f} = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{n_i} \frac{f(\mathbf{X}_i)}{\frac{1}{J} \sum_{j=1}^J p_j(\mathbf{X}_i)} = \frac{1}{J} \frac{f(\mathbf{X}_i)}{\frac{1}{J} \sum_{j=1}^J p_j(\mathbf{X}_i)} = \frac{f(\mathbf{X}_i)}{\sum_{j=1}^J p_j(\mathbf{X}_i)},$$

where the estimation is performed for a single mixture pool of $n = J$ samples total. This is known as the *balance heuristic*, which is the optimal heuristic in a sense of overall variance reduction [Veach and Guibas 1995].

In practice, multiple proposals can have high overlap of their importance regions, so the mixture density is weakened by averaging with lower-valued proposals. This is a common situation in graphics. In this case, it is useful to reformulate the mix-

ture model as follows

$$\bar{f} = \frac{f(\mathbf{X}_{i,j})}{\sum_{j=1}^J p_j(\mathbf{X}_{i,j})} = \frac{f(\mathbf{X}_{i,j})/p_j(\mathbf{X}_{i,j})}{\left(\sum_{j=1}^J p_j(\mathbf{X}_{i,j})\right)/p_j(\mathbf{X}_{i,j})} = w_j \frac{f(\mathbf{X}_{i,j})}{p_j(\mathbf{X}_{i,j})},$$

where $\mathbf{X}_{i,j}$ is the i th sample that was generated with the j th proposal, and $w_j = p_j(\mathbf{X}_i) / \sum_{j=1}^J p_j(\mathbf{X}_i)$ is the weighting function. Note that it turned into a convex combination of individual estimators $\frac{f(\mathbf{X}_{i,j})}{p_j(\mathbf{X}_{i,j})}$ for each proposal p_j . This way, we can combine these estimators in an unbiased way as long as the weights stay normalized $\sum_{j=1}^J w_j = 1$. This provides more freedom in combining multiple proposals in an unbiased way. One useful choice of weights for light transport is called the *power heuristic*, where the weights are

$$w_j^{(\beta)} = \frac{p_j^\beta(\mathbf{X})}{\sum_{j=1}^J p_j^\beta(\mathbf{X})}, \quad (3.6)$$

which essentially raises the existing proposals to the power β before weighting them. It is easy to check that such weights also sum up to one, leading to an unbiased combined estimator. This approach tries to contrast the winning proposals. It showed to be practically better with the power $\beta = 2$ by emphasizing the locally best proposals [Veach and Guibas 1995].

Multiple follow up techniques have been developed on top of MIS. For example, combining MIS with adaptive sampling, where the allocation of samples n_j in the pool and the adjustment of weights w_j is done adaptively [Cornuet et al. 2012], leads to an evolving population of samples, which provides the foundation for sequential Monte Carlo approaches.

3.3 Markov Chain Monte Carlo Integration

Markov chain Monte Carlo is another class of integration methods that uses correlated sampling with Markov chains in order to stochastically explore the integrand function by performing a random walk according to the integrand shape. Specific to this method, new sample is generated based on the previous sample or a history of samples, which creates a stochastic process with memory. This class of methods alleviates the curse of dimensionality even further than ordinary Monte Carlo methods, because they always store a valid previous sample for both worst-case fall back and resampling. Such memory-reliant sampling approaches provide more robustness and flexibility in exploring sparse high-dimensional functions.

3.3.1 Metropolis-Hastings Sampling

The first and the most popular Markov chain Monte Carlo method is the *Metropolis-Hastings algorithm*. The method can be explained as follows. Assume we have an unknown function $f(\mathbf{X})$ in a high-dimensional *state space* Ω that can be evaluated point-wise. Direct sampling according to f can be difficult, however, we can sample proportionally to f . We can construct a Markov chain, whose posterior distribution converges to the function of interest f accurate to the normalization factor (since f is generally not normalized). This method was proposed by Nicolas Metropolis et al. [1953] for the Boltzmann distribution and then generalized by Wilfred Keith Hastings [1970] for arbitrary target functions.

This method was inspired by a real-world physical thermodynamic process of a molecular movement during heat transfer, which describes a stochastic thermal movement of molecules in a closed system in thermal *equilibrium*, which is an essential condition that guarantees that the Markov chain that walks according to this process has a unique *stationary distribution*, i.e., a unique posterior distribution that it will form in the limit. Indeed, if the system were thermally rebalancing over time, the Markov chain would walk on a changing target function, leading to an evolving posterior distribution.

The equilibrium condition entails the principle of *detailed balance*. This principle states that the system can be in equilibrium only if the thermal flux from each point \mathbf{X}_i to each point \mathbf{X}_j in the system is balanced out by the same amount of flux flowing back from \mathbf{X}_j to \mathbf{X}_i . This can be formalized in terms of Markov chain's transition probabilities as

$$P_{i \rightarrow j} = P_{j \rightarrow i}. \quad (3.7)$$

In this case, such a Markov chain is called *reversible*. Note that the current transition probability $P_{i \rightarrow j} = T_{i \rightarrow j} a_{i \rightarrow j}$ is a joint probability that consists of two steps: the probability $T(\mathbf{X}_j | \mathbf{X}_i) \equiv T_{i \rightarrow j}$ of proposing (sampling) state j from the state i ; and a is the *acceptance probability* that we describe below.

The idea of Metropolis-Hastings method is to run a reversible Markov chain with a sample \mathbf{X}_i as its current state on the state space Ω , and at every step adjust the transition probability between Markov chain states using a conditional rejection probability a based on the desired target distribution. This probability is similar to the ordinary rejection sampling probability used in Monte Carlo methods. However, the key difference is that it is conditional on the current state as $a(\mathbf{X}_j | \mathbf{X}_i) \equiv a_{i \rightarrow j}$, i.e., it is a probability of conditionally accepting the new proposal state \mathbf{X}_j given the current state \mathbf{X}_i . It is therefore called an *acceptance probability*, because at each move i of the Markov chain we either accept the proposal state \mathbf{X}_j with probability $a_{i \rightarrow j}$ or otherwise reject it and keep the current state \mathbf{X}_i as the next state (with probability $1 - a_{i \rightarrow j}$ accordingly).

Given this acceptance probability, the posterior distribution of the constructed Markov chain converges to the desired target function f up to a normalization constant. Note that the detailed balance equation is affected by this acceptance probability.

The *proposal distribution* $T_{i \rightarrow j}$ plays a role in Markov chain Monte Carlo methods analogous to distributions used for importance sampling in ordinary Monte Carlo. The goal is to construct a good proposal distribution such that the important regions of the target f are visited more often. This framework is more flexible than the one of Monte Carlo, as each proposal distribution, in addition to the regular

independent sampling, can also rely on the current valid state of a Markov chain when drawing a new sample.

The proposal distribution also changes the detailed balance, so that the final detailed balance condition becomes

$$f(\mathbf{X}_i)T_{i \rightarrow j}a_{i \rightarrow j} = f(\mathbf{X}_j)T_{j \rightarrow i}a_{j \rightarrow i}.$$

In order to quickly reach the equilibrium during the walk, the best strategy is to maximize both $a_{i \rightarrow j}$ and $a_{j \rightarrow i}$. Therefore, we can express $a_{i \rightarrow j}$ from the previous equation and obtain the final Metropolis-Hastings acceptance probability by assuming that the reverse walk is always accepted ($a_{j \rightarrow i} = 1$) [Peskun 1973] as

$$a_{i \rightarrow j} = \min \left(1, \frac{f(\mathbf{X}_j)/T_{i \rightarrow j}}{f(\mathbf{X}_i)/T_{j \rightarrow i}} \right) = \min \left(1, \frac{f(\mathbf{X}_j)T_{j \rightarrow i}}{f(\mathbf{X}_i)T_{i \rightarrow j}} \right), \quad (3.8)$$

where the target function is evaluated and divided by the proposal probability, similarly to importance sampling, at both current and proposal samples.

The Metropolis-Hastings algorithm always constructs a posterior distribution as a histogram up to a scale factor, while the scale (normalization constant) of the original target function is usually unknown. Therefore, another important aspect is to find the normalization constant $b = \int_{\Omega} f(\mathbf{X}) d\mathbf{X}$ of the function f , i.e., the total mass of f over the whole state space Ω . This can be as hard as the original problem. For the particular needs of light transport it can be easily estimated using one of the alternative methods discussed later.

Start-up Bias. If the walk was started in a low-density region of the target f , it can take many steps for a Markov chain to move to high-density regions. The posterior histogram gets skewed by these initial moves, biasing the resulting posterior distribution into the low-density region of the target function. This bias is called the *start-up bias*, since it occurs due to an unfortunate initial state \mathbf{X}_0 of the Markov chain. It can require many more moves to compensate for such bias.

3.3.2 Convergence Criterion for Metropolis-Hastings Methods

We describe a formal weak ergodicity (convergence) condition for a Metropolis-Hastings method. First, we introduce some terminology related to the transition properties of a Markov chain, and then define the criterion for the transition probability and the target function that ensures that the Metropolis-Hastings algorithm converges to the desired target distribution.

The transition (Markov) kernel $K(\cdot, \cdot)$ of a Markov chain is a mapping $\Omega \mapsto \Omega$ with a source $\mathbf{X} \in \Omega$ and a target $\mathbf{Y} \in \Omega$ states, which is defined as

$$K(\mathbf{X}, \mathbf{Y}) = p(\mathbf{Y}|\mathbf{X}),$$

where $p(\mathbf{Y}|\mathbf{X})$ is the conditional probability density of moving to the new state \mathbf{Y} given that the current state of the Markov chain is \mathbf{X} .

The transition kernel is also defined for a set $A \subset \Omega$ as

$$K(\mathbf{X}, A) = \int_A K(\mathbf{X}, \mathbf{Y}) d\mathbf{Y},$$

denoting the probability of moving to the set A from the current state \mathbf{X} .

A transition kernel $K(\cdot, \cdot)$ is called *Harris-recurrent* if, given a Markov chain $\{\mathbf{X}_n\}$, sets A and B from Ω along with a positive number ε and a probability measure ρ with $\rho(B) = 1$, the following holds

1. If $\tau_A = \inf \{n \geq 0 : \mathbf{X}_n \in A\}$, then $\forall \mathbf{Z} \in \Omega : P(\tau_A < \infty | \mathbf{X}_0 = \mathbf{Z}) > 0$.
2. If $\mathbf{X} \in A$ and $C \subset B$, then $K(\mathbf{X}, C) \geq \varepsilon \rho(C)$.

In other words, given two states \mathbf{X}_1 and \mathbf{X}_2 in A , there is at least an ε chance that they can be moved together to the same point at the next step.

A transition kernel $K^n(\cdot, \cdot)$ is called an *n-step transition kernel* and is defined re-

cursively as a convolution

$$K^n(\cdot, \cdot) = \int_{\Omega} K(\mathbf{Y}, \cdot) K^{n-1}(\cdot, \mathbf{Y}) d\mathbf{Y},$$

where $K^0 \equiv K(\cdot, \cdot)$. The term $K^n(\mathbf{X}_0, \mathbf{X}_n)$ denotes the probability density of landing into the state \mathbf{X}_n after n moves of the Markov chain initialized with the state \mathbf{X}_0 .

Note that generally the kernel $K = K^n$ depends on the move n . In this case the Markov chain is called a *non-homogeneous Markov chain*. The central limit theorems, which provide the ergodicity conditions for non-homogeneous Markov chains, were developed by Dobrushin [1956]. Before introducing the criterion of weak ergodicity, we first define the *Dobrushin's ergodic coefficient* $\delta(K)$ as

$$\delta(K^{(m,k)}) = \frac{1}{2} \sup_{\mathbf{X}, \mathbf{Y} \in \Omega} \int_{\Omega} |K^m(\mathbf{X}, \mathbf{Z}) - K^{m+k}(\mathbf{Y}, \mathbf{Z})| d\mathbf{Z},$$

which is bounded in the range $0 \leq \delta(K) \leq 1$ and “assesses” the changes in the kernel $K(\cdot, \cdot)$ between the steps m and $m + k$.

Now we can write the necessary and sufficient criterion for the *weak ergodicity* of a non-homogeneous Markov chain [Dobrushin 1956] as

$$\forall k \geq 0 : \lim_{m \rightarrow \infty} \delta(K^{(m,k)}) \rightarrow 0, \quad (3.9)$$

which intuitively means that the changes in the transition kernel at every move (iteration) should be small enough to provide a weak mixing of the Markov chain. Note that $\delta(K^{(m,k)}) \leq \prod_{n=m}^k \delta(K^{(n,n+1)})$ for all m and k . We denote the ergodic coefficient δ for a single step, with a slight and innocuous ambiguity, as $\delta(K^{(n,n+1)}) = \delta(K^n)$. Therefore Eq. 3.9 can be written in an equivalent way as

$$\prod_{n=0}^{\infty} \delta(K^n) \rightarrow 0. \quad (3.10)$$

This is a convenient ergodicity condition for Markov chain convergence, which we will use later in the second part of Chapter 5. It states that the target function

can actually change with time as long as these changes vanish in the limit. We refer the interested reader to the corresponding literature on Markov chains for the further reading, e.g., the book of Pierre Brémaud [Bremaud 1999].

3.3.3 Application: Metropolis Light Transport

A prominent example application of Metropolis-Hastings algorithm to light transport is *Metropolis light transport* (MLT) method [Veach and Guibas 1997]. It employs Markov chain Monte Carlo (MCMC) sampling using the Metropolis-Hastings algorithm for computing path integrals by iteratively sampling a new path \mathbf{X}_{t+1} based on the current path \mathbf{X}_t .

For the Metropolis-Hastings algorithm we need to compute the proposal pdf $T(\mathbf{X}_t \rightarrow \mathbf{X}_{t+1})$ of sampling the light path \mathbf{X}_{t+1} given \mathbf{X}_t to perform a step. To ensure the Markov chain converges to the intended stationary distribution, a newly proposed mutation is accepted according to the acceptance probability in Eq. 3.8, where in the context of light transport \mathbf{X} is a full path from the path space Ω , $f(\mathbf{X})$ is a path measurement contribution function from path integral in Eq. 2.10.

The proposal distribution $T(\mathbf{X}_t \rightarrow \mathbf{X}_{t+1})$ proposes a full path \mathbf{X}_{t+1} given a current valid path \mathbf{X}_t . In MLT, a proposal distribution is called the *mutation strategy*, because it mutates the existing light path in some manner. It is possible and desirable to have multiple such proposals that can perform specialized path mutations.

Note that the measurement I^j in Eq. 2.8 is defined for a single j th sensor element (e.g., image pixel). However, during the mutation, the path can land on another pixel of the image. For the random walk to be more efficient, we can instead explore the whole image at once. Therefore, we replace the responsivity function W^j of each individual sensor element j by the overall *image responsivity* function

$$W(\mathbf{X}) = \sum_j = 1^J W^j(\mathbf{X}),$$

where J is the total number of sensor elements (e.g., pixels), and this function denotes the total contribution of a current path \mathbf{X} to the image plane. By using

this function in the path measurement contribution function f , we can guide the random walk across the image without focusing on individual sensor elements. This way, MLT can share path samples \mathbf{X}_i among all pixels of a sensor, i.e., estimating all pixel integrals at once, thus effectively computing an image histogram. For this, each sample is projected to the image plane and its contribution is accumulated accordingly.

The main steps of the MLT algorithm are as follows:

1. Generate an initial state \mathbf{X}_0 (full path) of a Markov chain using one of the ordinary Monte Carlo sampling methods (e.g., path tracing or bidirectional path tracing discussed in Chapter 4).
2. Start the mutation process. Mutate the current path \mathbf{X}_t into a new proposed path \mathbf{X}' using one of the available mutation strategies, compute the proposal density $T(\mathbf{X}_t \rightarrow \mathbf{X}')$.
3. Compute the acceptance probability from Eq. 3.8; based on the acceptance probability, either accept the new proposed path \mathbf{X}' as \mathbf{X}_{t+1} , or reject it and keep $\mathbf{X}_{t+1} = \mathbf{X}_t$.
4. Accumulate the contribution of the resulting path \mathbf{X}_{t+1} to the image plane, for every affected pixel j apply the pixel filter $W^j(\mathbf{X}_{t+1})$ to deposit the contribution of the path to the image histogram formed by the posterior distribution of the walk.
5. Proceed to step 2 to cycle the rendering process.

MLT is more robust in the complex illumination scenarios, such as strong indirect lighting, difficult visibility (e.g., small slits or windows), caustics, and specular inter-reflections. It explores the path space locally by “remembering” the last successful path, because the current state of a Markov chain is always a correct full path from light source to the sensor. In addition, the Markov chain explores similar surrounding paths easier by using mutations that slightly perturb the current path, thus exploring the complete illumination features at low cost. However, this can also cause some unwanted correlations of samples, slowing down the overall

rendering convergence. Therefore one needs to carefully construct proposal densities that efficiently explore the path space. Moreover, it is important to take the desired stratification of samples on the image plane into account, as discussed in Section 2.3. The MLT framework provides us with the flexibility of designing path generators for difficult special situations, as will be discussed in Chapter 5.

Mutation Strategies. Let us outline the criteria that an ideal mutation strategy should fulfill. The mutation should be as lightweight as possible, that is, it should try to introduce minimal changes to the path, requiring as few vertex updates as possible. It should also produce a sequence of samples with low correlation, proposing sufficiently large steps in path space, while exploring the important local features, such as glossy highlights. Also, specific to the image rendering process, the mutation should try to sample the image plane as uniform as possible. This is usually hard to control in the context of MCMC, therefore, the best practice often is to reseed the chain with paths stratified over the image plane. However, we will show an alternative stratification approach in Chapter 8.

Finally, it is fine if the mutation can efficiently explore only some particular small subset of the path space, e.g., only caustics, leaving other features to other specialized mutations. This is one of the advantages of the MLT framework. We will discuss existing proposals for generating light paths in Chapter 4.

Normalization. The key difference of MLT to the usual MCMC situation is that we have alternative Monte Carlo methods that can generate an image. So, many regular MCMC problems like the estimation of the normalization constant and start-up bias can be solved easier. For example, in order to compute the normalization constant, which is an average flux received by the image plane, it is practically sufficient to sample a few hundred thousand paths with an ordinary Monte Carlo method, which is a negligible cost compared to the actual image rendering.

Bootstrapped Initialization. In order to reduce the start-up bias, we can bootstrap the initial walk by initializing the Markov chain directly within the high-

contribution regions of the target distribution. The common practice is to collect many samples from an ordinary Monte Carlo rendering method (such as path tracing or bidirectional path tracing, see Section 4) and then seed the Markov chain with one sample importance-resampled according to the path contribution. This way, the process is bootstrapped with an initial path \mathbf{X}_0 obtained from a good ordinary Monte Carlo method and subsequently the path space is explored using MCMC with specifically designed mutation strategies. It is also possible to completely eliminate the start-up bias by assigning the Monte Carlo path throughput $f(\mathbf{X})/p(\mathbf{X})$ as a weight to all samples [Veach 1998]. This weight compensates for the fact that the distribution p used to importance-sample \mathbf{X}_0 does not do ideal sampling proportional to f .

In case of many chains, their initial states can be also stratified with respect to the path contribution to have a good initial coverage of the path space as well as the image plane. The method also scales well with tens of thousands of Markov chains for massively parallel computations.

3.4 Kernel Estimation within Integration

Kernel estimation is a statistical framework for estimating the value of a function f at point \mathbf{x} based on the statistical distribution of observed values of the function around this point. Even though this method might look unconnected to the integration problems at first, we show how it can be artificially introduced into the integration. The integrand is split into an observable and a probing part, in order to break down the dimensionality and consequently reduce the variance of the final estimate.

Let $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ be an independent sample drawn from some unknown distribution p of points on a surface \mathcal{M} , and $(f(\mathbf{x}_1), \dots, f(\mathbf{x}_N))$ are evaluated values of the function f at these points. Then the function value at a point \mathbf{x} can be computed using a kernel estimator as

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N k_r(\mathbf{x} - \mathbf{x}_i) = \frac{1}{Nr^d} \sum_{i=1}^N k\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{r}\right),$$

where $k(\cdot)$ is a *smoothing kernel*, a non-negative function with zero mean, preferably compact support, and vanishing moments [Silverman 1986]. The kernel k should be continuous, bounded, and normalized [Parzen 1962]. If the argument of $k(\cdot)$ is a vector, we assume its Euclidean length is used to evaluate the kernel. The term $k_r(t) = 1/r^d k(t/r)$ is a scaled and renormalized version of the canonical kernel $k(t)$, where $1/r^d$ is the renormalization factor for a d -dimensional space. Kernel estimation is used for variance reduction in light transport. For the particular case of the kernel estimation on a surface, the dimensionality d equals two.

This estimator can be also reformulated as a convolution $\langle k_r * f \rangle$ evaluated at the point \mathbf{x} as

$$f'_r(\mathbf{x}) = \int_{\mathcal{M}} \frac{1}{r^d} k\left(\frac{\|\mathbf{x} - \mathbf{y}\|}{r}\right) f(\mathbf{y}) d\mathbf{y}.$$

The parameter r is called a *kernel bandwidth* and controls the smoothing window. This parameter controls the trade-off between the bias introduced by kernel

smoothing (when the bandwidth is large) and the variance of the estimated value (when the bandwidth is small), and therefore is an important control parameter of the kernel estimation. In kernel estimation methods the optimal choice of the kernel bandwidth is a challenging problem. As we will show in Chapter 5, a properly selected bandwidth can significantly minimize the overall estimation error and improve the asymptotic convergence rate.

3.4.1 Estimation Error

We can write the error of the kernel estimation using the *mean squared error* (MSE) as

$$\begin{aligned} MSE_r(\hat{f}) &= E \left[\hat{f}(\mathbf{x}) - f(\mathbf{x}) \right]^2 = \left(E \left[\hat{f}(\mathbf{x}) \right] - f(\mathbf{x}) \right)^2 + \text{Var} \left[\hat{f}(\mathbf{x}) \right] \\ &= \mathcal{B} \left[\hat{f}(\mathbf{x}) \right]^2 + \text{Var} \left[\hat{f}(\mathbf{x}) \right], \end{aligned}$$

where $\mathcal{B} \left[\hat{f}(\mathbf{x}) \right]^2$ is the *bias*, the systematic error of the kernel estimator caused by smoothing, and $\text{Var} \left[\hat{f}(\mathbf{x}) \right]$ is the variance of the estimator that comes from the variance of the observed function f itself plus the variance of the distribution of observed points \mathbf{x}_i . The mean squared error is a widely used loss function and well-suited for analyzing the performance of estimators as it enables partitioning of variance and bias. The structure of the error reveals that one can trade more systematic error for more noise and vice versa, which is called the *bias-variance trade-off*.

3.4.2 Bandwidth Selection and Convergence

In order to find the balance in the bias-variance trade-off, we further expand the bias term of the MSE as

$$\mathcal{B} \left[\hat{f}(\mathbf{x}) \right] = E \left[\hat{f}(\mathbf{x}) \right] - f(\mathbf{x}) = \int_{\mathcal{M}} \frac{1}{r^d} k \left(\frac{\|\mathbf{x} - \mathbf{y}\|}{r} \right) f(\mathbf{y}) d\mathbf{y} - f(\mathbf{x}),$$

which, after expanding $f(\mathbf{y})$ in a Taylor series around point \mathbf{x} [Silverman 1986, p.39], can be written as

$$\mathcal{B}[\hat{f}(\mathbf{x})] \approx \frac{1}{2} r^2 k_2 |\Delta f(\mathbf{x})| \propto O(r^2),$$

where the constant $k_2 = \int_{\mathcal{M}} t^2 k(\mathbf{x}) d\mathbf{x}$ is the second moment of the estimation kernel and $\Delta \equiv \nabla^2$ is a Laplacian operator of the function f evaluated at point \mathbf{x} , which represents a local “curvature” of the function f .

The estimation variance, the second term of the MSE, can be expanded as

$$\begin{aligned} \text{Var}[\hat{f}(\mathbf{x})] &= \frac{1}{N} \int_{\mathcal{M}} \frac{1}{r^d} k^2\left(\frac{\|\mathbf{x}-\mathbf{y}\|}{r}\right) f(\mathbf{y}) d\mathbf{y} - \frac{1}{N} f^2(\mathbf{x}) \\ &= \frac{1}{N r^d} \int_{\mathcal{M}} (f(\mathbf{x}) - r \mathbf{z} \nabla f(\mathbf{x}) + \dots) k^2(\|\mathbf{z}\|) d\mathbf{z} + O\left(\frac{1}{N}\right) \approx \frac{1}{N r^d} k_3 f(\mathbf{x}) \propto O\left(\frac{1}{r^d}\right), \end{aligned}$$

where we performed the change of variable $\mathbf{y} = \mathbf{x} - \mathbf{z}r$, and the constant $k_3 = \int_{\mathcal{M}} k^2(\mathbf{x}) d\mathbf{x}$ is the squared L_2 norm of the estimation kernel.

Now we can write a first-order approximation of the MSE as

$$MSE_r(\hat{f}) \approx \frac{1}{4} r^4 k_2^2 |\Delta f(\mathbf{x})|^2 + \frac{1}{N r^d} k_3 f(\mathbf{x}), \quad (3.11)$$

which depends on both the kernel bandwidth and the inverse of it, and thus is subject to a minimization with respect to r .

After a simple minimization procedure, one can find that the first-order optimal bandwidth r in the sense of minimizing MSE is

$$r_{\text{opt}} = \left(\frac{d}{N} \frac{k_3}{k_2^2} |\Delta f(\mathbf{x})|^{-2} \right)^{\frac{1}{d+4}} \propto O\left(N^{-\frac{1}{d+4}}\right).$$

Now we can plug this optimal bandwidth back into the MSE in Eq. 3.11, which leads to the optimal MSE as

$$MSE_{\text{opt}} \propto O\left(r^{-\frac{4}{d+4}}\right).$$

This is an important asymptotic result showing the optimal convergence rate of the mean squared error of the kernel estimator. Note that this rate depends on the dimensionality d of the kernel estimation, which shows that generally this method is subject to the *curse of dimensionality* and is thus not suitable for high-dimensional estimation.

We will use this framework in Chapter 7 to derive the optimal convergence rate of the photon mapping method that is based on kernel estimation.

3.4.3 A Brief History of Adaptive Estimation

In addition to the fundamental analysis of kernel estimation, this section provides a brief overview of the existing adaptive methods for kernel estimation. The development of *adaptive kernel estimators* started in the 1960s, when Parzen [1962] showed the optimal rate of convergence and asymptotically optimal bandwidth reduction rate for the kernel estimator. Rudemo [1982] suggested the first *cross-validation method*. Silverman [1986] suggested a *rule of thumb* for bandwidth selection, which assumes that the underlying function can be well approximated by a normal distribution. Quickly evolving fields, such as computational finance, artificial intelligence, robotics, machine learning, and computer vision, created much activity in this field in the 1990s. Popular methods include *wavelet shrinkage* [Donoho et al. 1993], an estimation technique based on the idea of thresholding the wavelet coefficients and the *Intersection of Confidence Intervals* (ICI) [Katkovnik and Shmulevich 2002], which is based on the calculation of estimates of the function and their variances. Neural networks and support vector machines [Cortes and Vapnik 1995] can also be used to learn an unknown function.

The *plug-in bandwidth selection* [Park and Marron 1990, Sheather and Jones 1991] selects the optimal bandwidth by “plugging in” the estimate for higher-order derivatives of a function into the formula for the asymptotically optimal bandwidth, as was shown in Eq. 3.11. We apply this approach to develop the adaptive progressive photon mapping in Chapter 5.

Path Construction Methods

Nature uses only the longest threads to weave her patterns, so each small piece of her fabric reveals the organization of the entire tapestry.

— RICHARD FEYNMAN,
The Character of Physical Law (1965)

In this chapter we connect two previous chapters even further by discussing a large body of existing path construction methods. We discuss how the light transport integration problem introduced in Chapter 2 can be solved using numerical integration methods from Chapter 3. Most of the practical integration methods, such as Monte Carlo and Markov chain Monte Carlo methods, rely on the construction of a complete light path from a light source to a sensor as a building block. Therefore, their efficiency directly depends on the efficiency and robustness of the underlying framework for constructing such light paths.

The central task of the light transport simulation is to solve the path integral from Eq. 2.10

$$I = \int_{\Omega} f(\mathbf{X}) d\mathbf{X},$$

in order to obtain measurements of the incident flux at all elements of a sensor. This requires constructing a full *light path* $\mathbf{X} = (\mathbf{x}_0 \dots \mathbf{x}_k)$, which is an ordered list of vertices on the manifold \mathcal{M} of all surfaces in the virtual scene. By convention, the first vertex \mathbf{x}_0 is always located on a light source, and the end vertex \mathbf{x}_k is always on an element of the measuring sensor (e.g., a camera).

4.1 Fundamentals of Path Construction

There are two major classes of path construction methods: local path construction and global path construction. Both types use *ray tracing* in order to obtain each inner vertex of a light path. Ray tracing can be defined as a function that takes a ray (\mathbf{x}, \mathbf{o}) and returns the closest intersection point \mathbf{x} with the scene geometry \mathcal{M} along this ray

$$\mu_{\mathcal{M}}(\mathbf{x}, \mathbf{o}) = \mathbf{x} + \mathbf{o}t, \quad \text{where } t = \inf(t > 0 | \mathbf{x} + \mathbf{o}t \in \mathcal{M}).$$

It deterministically finds the next point along the ray (\mathbf{x}, \mathbf{o}) after \mathbf{x} that belongs to a scene surface. In order to accelerate ray tracing, usually some form of an *acceleration structure* is built around the scene geometry and then traversed for every ray. Accelerating ray tracing is a large and active research field by itself. For a more detailed overview of the acceleration methods, we refer to Pharr et al. [2010].

The first path construction class, *local path construction* is a sequential process of constructing a path (or a subpath) vertex by vertex. First a start vertex is selected; it can be either a vertex on the sensor, or on the light source, or, in some cases, a vertex in the middle of an existing path. Then the following two steps are performed in alternation until some path termination criterion is met:

1. Stochastically sample new ray direction \mathbf{o}_i at the current path vertex \mathbf{x}_i , e.g., using local importance sampling of scattering distribution at \mathbf{x}_i .
2. Find the next path vertex \mathbf{x}_{i+1} along the ray $(\mathbf{x}_i, \mathbf{o}_i)$ using ray tracing.

In order to use local construction methods with stochastic integration, one needs to compute the probability density of constructing a path \mathbf{X} in on-surface area measure, since this is the integration measure of the light transport path integral (see Eq. 2.10). We call such a probability density $p(\mathbf{X})$ a *path probability density*, or shortly a *path pdf*. We start the overview of existing construction methods with this local path construction class of methods.

The second class, *global path construction*, tries to construct the whole path \mathbf{X} at

once. This is a challenging task, as it usually requires knowing all possible combinations of interactions and surface shapes, which combinatorially increases the complexity even in very simple scene configuration [Reif et al. 1994]. We will show that in some cases the global knowledge of such configurations is not required, and one can rely on some local knowledge of geometry shapes and materials around a guiding path. We introduce a method for global path construction in Chapter 8.

4.2 Basic Path Construction: Implicit Light Sampling

We are ultimately interested in the amount of flux on the sensor elements when computing a solution to light transport. Only a small fraction of all emanated photons ends up landing on the camera sensor. For example, a simple 100-watt incandescent light bulb emits around 10^{32} photons/second, and only about 10^{20} photons would reach the sensor element in a camera in a bright room with average illumination conditions. Therefore, simulation of light transport in this physical way would likely require an impractical amount of computation.

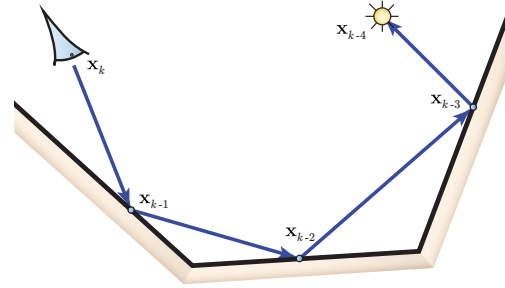


Figure 4.1: An example of path construction with implicit light sampling from the sensor for a path of length k . The path is traced until it eventually hits the light source. We deliberately denoted the light source vertex as \mathbf{x}_{k-4} , as the path length remain unknown until after the path is terminated at the light source.

Path tracing, the first light transport simulation method with Monte Carlo integration for image rendering was proposed by Jim Kajiya [1986]. Path tracing uses the rendering equation in Eq. 2.3. Contrary to the physical process, the path construction starts with a vertex \mathbf{x}_k at the end of a light path, i.e., at the sensor element as shown in Figure 4.1. It should be noted that we can reverse the physical light transport process only if all interactions and processes are reciprocal, as discussed in Section 2.2.5. Paths are initiated at every pixel of an image by stochastically sampling the area within a pixel, thus obtaining the initial path vertex \mathbf{x}_k . The outgoing direction at the first vertex is importance-sampled according to the directional

distribution of the responsivity W at the selected sensor element. All subsequent vertices are constructed using local path sampling by consecutively selecting the outgoing direction at every vertex. The path is terminated when the newly sampled vertex lands on a light source (Figure 4.1). This path construction technique is called the *implicit light sampling*, because the end point on the light source is not sampled based on its emission profile, but it is obtained as a hit point after a series of tracing steps.

Path Probability Density. Path construction pdf depends on the construction direction. For the convenience of notation, in this section we will reindex path vertices according to the construction process, i.e., the first sampled vertex is \mathbf{x}_0 and it starts on the sensor since we construct a path starting from the camera.

Sampling probability of each new vertex \mathbf{x}_i depends only on the position of the previous vertex \mathbf{x}_{i-1} and the previous outgoing direction \mathbf{o}_{i-1} , which can be written as $p(\mathbf{x}_i) = p(\mathbf{x}_i | (\mathbf{x}_{i-1}, \mathbf{o}_{i-1}))$. The process of sampling \mathbf{o}_{i-1} depends only locally on \mathbf{x}_{i-1} . Once both \mathbf{x}_{i-1} and \mathbf{o}_{i-1} are sampled, the ray $(\mathbf{x}_{i-1}, \mathbf{o}_{i-1})$ deterministically defines the next path vertex \mathbf{x}_i via ray tracing, and thus there is no stochastic decision involved. However, the probability density for the path integration should be defined in the on-surface domain $d\mathbf{x}$, while the ray sampling pdf is defined in the outgoing solid angle domain $d\mathbf{o}$ as a joint pdf $p(\mathbf{x}_{i-1}, \mathbf{o}_{i-1}) = p(\mathbf{x}_{i-1})p(\mathbf{o}_{i-1} | \mathbf{x}_{i-1})$. This domain change as well as the corresponding density change is handled by the Jacobian

$$G'(\mathbf{x}_{i-1}, \mathbf{x}_i) = \left| \frac{d\mathbf{o}_{i-1}}{d\mathbf{x}_i} \right| = \frac{|\langle \mathbf{n}_i, \mathbf{o}_{i-1} \rangle|}{\|\mathbf{x}_i - \mathbf{x}_{i-1}\|^2} = \frac{G(\mathbf{x}_{i-1}, \mathbf{x}_i)}{|\langle \mathbf{n}_{i-1}, \mathbf{o}_{i-1} \rangle|}.$$

This Jacobian, called a *one-sided geometric term*, is similar to the geometric term defined in Eq. 2.9, except that it transforms from solid angle $d\mathbf{o}$ as opposed to the projected solid angle $d\mathbf{o}^\perp$. Therefore, the source projection term $|\langle \mathbf{n}_{i-1}, \mathbf{o}_{i-1} \rangle|$ is removed from the original geometric term. This difference makes it non-symmetric, and thus dependent on the path construction direction.

We can transform the joint probability density to the on-surface area measure as

$$p(\mathbf{x}_i) = p(\mathbf{x}_{i-1})p(\mathbf{o}_{i-1})G'(\mathbf{x}_{i-1}, \mathbf{x}_i).$$

This is a recursive formula and we can expand it into the total path probability $p(\mathbf{X})$ for local construction methods as a joint probability

$$\begin{aligned} p(\mathbf{X}) &= p(\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0) \times \dots \times p(\mathbf{x}_k|\mathbf{x}_0\mathbf{x}_1\dots\mathbf{x}_{k-1}) \\ &= p(\mathbf{x}_0) p(\mathbf{o}_0)G'(\mathbf{x}_0, \mathbf{x}_1) p(\mathbf{o}_1)G'(\mathbf{x}_1, \mathbf{x}_2) \times \dots \times p(\mathbf{o}_{k-1})G'(\mathbf{x}_{k-1}, \mathbf{x}_k). \end{aligned} \quad (4.1)$$

In both Monte Carlo and Markov chain Monte Carlo estimators, the integrand is divided by this path probability, thus all one-sided geometric terms cancel out with the corresponding full geometric terms in the path measurement contribution function $f(\mathbf{X})$ in Eq. 2.11, leaving only the outgoing projective terms. This can be formulated as a general rule: whenever the next vertex is obtained from the outgoing direction, the ray tracing essentially performs a perfect importance sampling of the corresponding geometric term in the measurement equation. Note that it is not always possible to obtain all path vertices by performing ray tracing, therefore, some geometric terms usually do not cancel out.

4.3 Connection-based Path Construction

Path tracing with implicit light sampling allows us to conveniently sample many complete light paths, e.g., by tracing until a large local area light or a distant emitting environment map is hit. However, the success of hitting a light source when sampling an outgoing direction from a path vertex highly depends on the subtended visible solid angle of the light source from this vertex. Therefore, path tracing with implicit light sampling can be highly inefficient for small or distant area lights, and would not work for point lights with zero area. Alternatively, a path can be constructed by trying to directly connect two subpaths from both sides.

4.3.1 Unidirectional Path Construction

In addition to the implicit light sampling technique, in his seminal paper, Jim Kajiya [1986] also proposed sampling a point on the light source and then connecting to this point. This process is shown in Figure 4.2 and is called a *next event estimation* or an *explicit light sampling* technique.

The idea is that along with tracing a path starting from the sensor, at every vertex \mathbf{x}_i of such a path, another vertex \mathbf{x}_0 is sampled on a light source, and the current vertex \mathbf{x}_i is connected to this vertex on the light source (depicted by dashed red lines) by tracing a ray to check the mutual visibility. In case of successful connection (i.e., when the orientation of normals allows and there are no

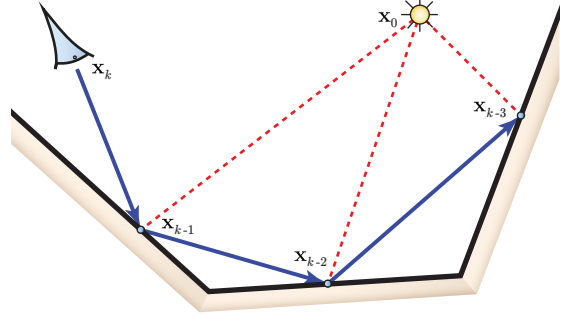


Figure 4.2: An example of path construction with explicit connection to the light (next event estimation). At every newly sampled vertex of the path we attempt to connect to a sampled point \mathbf{x}_0 on the light source.

occluding objects in between), such a path is considered to be a complete light path and is accumulated to the sensor measurement. Due to the connection at every vertex, this method can construct multiple complete paths during a single tracing process. These paths share one or more segments starting from the sensor, which can make them correlated in path space, leading to correlation artifacts. However, these paths have different dimensionality and represent different illumination effects (e.g., direct illumination and then single bounce indirect illumination), and thus the correlation is insignificant [Veach 1998]. The path pdf is the same as in Eq. 4.1, except that the last term $p(\mathbf{o}_{k-1})G'(\mathbf{x}_{k-1}, \mathbf{x}_k)$ is replaced by an on-surface area probability density $p(\mathbf{x}_0)$ of sampling a point on the light source. This is due to the fact that we do not determine the end vertex on the light source by tracing to it, but instead perform a deterministic connection to a stochastically sampled point on the light source. This is reflected by the term $p(\mathbf{x}_0)$ in the joint path pdf due to the fact that the point on the light source is sampled independently from the sensor subpath. Note that after divide by this pdf in the estimator, the

last geometric term in the path throughput does not cancel out anymore and remains in the path measurement contribution function. This indicates that we do not importance-sample this term, but instead we try to connect two preselected vertices.

Along with constructing a path from the sensor, it is also possible to perform construction starting from a light source. This process is called *light tracing* [Arvo 1986]. A complete path is constructed exactly the same way as in path tracing with next event estimation, however, sensor and light are swapped (see Figure 4.3). That is, we first sample a point \mathbf{x}_0 on a

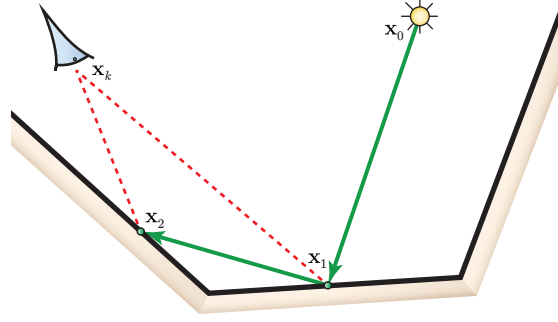


Figure 4.3: Light tracing, starts constructing the path from the light source and tries to connect every vertex to the point sampled on the sensor.

light source and then trace, while connecting each newly traced vertex \mathbf{x}_i to the point on the sensor pupil or aperture, followed by depositing the result into the corresponding sensor element. This process is very close to the real-world physical propagation of light, except the last edge of the path is constructed by connecting the last on-surface interaction to the camera instead of relying on hitting the camera pupil incidentally.

Path pdf $p(\mathbf{X})$ is computed similarly to the one in path tracing with next event estimation, in fact using exactly the same joint pdf, except that indices are aligned with the opposite construction direction due to the convention. That is, the indexing of path vertices starts on the light source, i.e., \mathbf{x}_0 means the vertex on the light source.

While light tracing can be generally inefficient, it handles specific types of paths, such as caustics through refractive objects, better than the path constructors that start from the sensor, because reaching the sensor from the light source is easier.

4.3.2 Bidirectional Path Construction

Eric LaFortune and Yves D. Willems [1993] and Eric Veach and Leonidas J. Guibas [1994] independently proposed a unified framework called *bidirectional path tracing* (BPT or BDPT) for constructing a path starting from both the sensor and the light source ends.

The idea is to construct a path by deterministically connecting two unfinished subpaths. The path construction process iterates through the following steps:

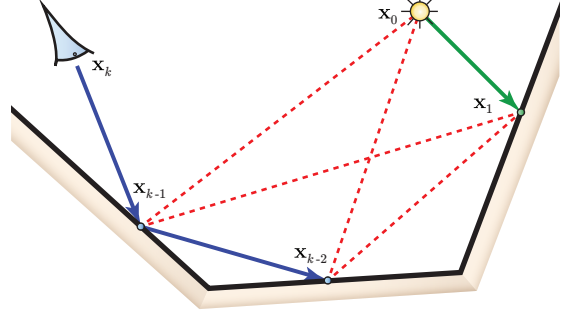


Figure 4.4: After tracing two subpaths, one from the sensor and one from emitter, we deterministically connect each vertex of one subpath to each vertex of another.

1. Sample a subpath $\mathbf{X}_E = (\mathbf{x}_k \mathbf{x}_{k-1} \dots)$ from the sensor (similarly to path tracing), called *eye subpath* or *camera subpath* (blue subpath in Figure 4.4).
2. Sample a subpath $\mathbf{X}_L = (\mathbf{x}_0 \mathbf{x}_1 \dots)$ from the light source (similarly to light tracing), called *light subpath* (green subpath in Figure 4.4).
3. Perform connections for every possible pair of vertices (red dotted lines in Figure 4.4): \mathbf{x}_i from \mathbf{X}_E and \mathbf{x}_j from \mathbf{X}_L (Figure 4.4). Check visibility between \mathbf{x}_i and \mathbf{x}_j and if the connection is successful, compute measurement contribution of the resulting path $f(\mathbf{X})$.
4. For each successfully connected path, compute its construction pdf $p(\mathbf{X})$ and normalization weight w_{ij} and accumulate the weighted path throughput to the corresponding elements on the sensor (e.g., image pixels).

Note that every iteration of this process constructs a group of up to $K = |\mathbf{X}_E| \cdot |\mathbf{X}_L|$ complete paths as an output. Therefore, the sampled vertices from both the camera subpath and the light subpath are reused for many paths.

Bidirectional methods are based on deterministic connections of path vertices, which implies that the bidirectional scattering distribution functions (BSDF) at two connected vertices cannot be importance-sampled.

Full Path Pdf. The joint probability of constructing a general path of length k out of a camera subpath with s vertices and a light subpath with t vertices is a combination of the corresponding joint pdfs from light tracing and path tracing written as:

$$\begin{aligned}
 p(\mathbf{X}) &= p(\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0) \times \dots \times p(\mathbf{x}_{t-1}|\mathbf{x}_0 \dots \mathbf{x}_{t-2}) \\
 &\quad \times p(\mathbf{x}_{s+t})p(\mathbf{x}_{s+t-1}|\mathbf{x}_{s+t}) \times \dots \times p(\mathbf{x}_t|\mathbf{x}_{s+t} \dots \mathbf{x}_{t+1}) \\
 &= p(\mathbf{x}_0)p(\mathbf{o}_0)G'(\mathbf{x}_0, \mathbf{x}_1) \times \dots \times p(\mathbf{o}_{t-2})G'(\mathbf{x}_{t-2}, \mathbf{x}_{t-1}) \\
 &\quad \times p(\mathbf{x}_{s+t})p(\mathbf{o}_{s+t})G'(\mathbf{x}_{s+t}, \mathbf{x}_{s+t-1}) \times \dots \times p(\mathbf{o}_{t+1})G'(\mathbf{x}_{t+1}, \mathbf{x}_t). \quad (4.2)
 \end{aligned}$$

Note that the pdf between two connection vertices is absent. The reason is once the two subpaths are constructed, the connection is a deterministic operation.

Multiple Alternative Techniques. Every path within this framework can be constructed in multiple independent ways, called *techniques*. Figure 4.5 shows all four possible ways of constructing a direct lighting path (a path with two edges).

The set of techniques for a single path are independent from each other, since they construct the path by sampling different subpaths. Therefore, every technique is an independent estimator for this path. Moreover, paths of different lengths have a different number of construction techniques. Therefore, every path constructed by a group of techniques has to be weighted to normalize its contribution to the image.

A trivial weighting is to divide the contribution by the number of construction techniques. However, this weighting scheme has one disadvantage. Imagine a path with two close-by vertices, e.g., an inter-reflection in the corner. In this case, the geometric term from Eq. 2.9 between these two vertices is unbounded. Therefore, paths constructed by connecting these two vertices will have this geometric term in

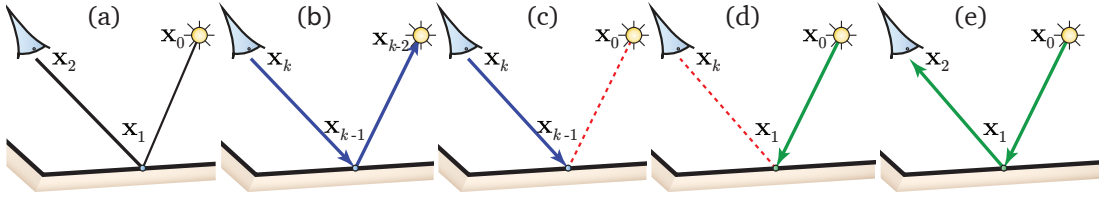


Figure 4.5: Bidirectional path tracing provides multiple techniques for constructing the same path (a). From left to right: (b) path is traced from sensor all the way to the light source; (c) path tracing with next event estimation; (d) light tracing with connection to the sensor; (e) path is traced from the light until it reaches the sensor (rarely used in practice). Generally, for a path of length k there are $k + 2$ techniques to construct it within bidirectional path tracing.

their path throughput, potentially contributing bright outliers to the image, called *fireflies*. Another weighting method based on a bundling heuristic was proposed by Eric LaFortune and Yves D. Willems [1993] to reduce this effect.

Multiple Importance Sampling. An efficient weighting solution was proposed by Eric Veach and Leonidas J. Guibas [1994] based on multiple importance sampling (Section 3.2.3). The idea is to interpret multiple construction techniques for constructing the same path as multiple importance functions for this path. In this case, a weight for a path, constructed with a technique using subpaths of length s and t , can be computed using multiple importance sampling from Eq. 3.6, with $p_j = p_{s,t}$ being the pdf of constructing a path with a particular s, t technique from Eq. 4.2. In his further studies, Eric Veach suggested to use a power heuristic with a power $\beta = 2$ for the best results, because it emphasizes efficient techniques more, thus reducing the perceived image noise. Weighting according to multiple importance sampling suppresses the singularities that appear in geometric terms by giving such techniques correspondingly low weights.

Even though bidirectional path tracing is a robust method, Veach [1998] observed that there exist configurations of geometry, light sources, and sensors, which cannot be sampled by any unbiased method.

4.4 Estimation-based Path Construction

In this section, we introduce *photon mapping* [Jensen 1996], one use case of kernel estimation for light transport.

Photon mapping artificially splits each path in the path integral into two subpaths and introduces an on-surface kernel estimation that can connect subpaths from the light side to subpaths from the sensor side.

In order to estimate the value of the measurement contribution efficiently with a kernel of an on-surface bandwidth r , photon mapping first distributes a high number N of light subpaths stochastically, called virtual photons, and stores the path vertices on scene surfaces in some acceleration structure (e.g., a kd-tree) as a *photon map*. In the second phase, eye subpaths are traced from the sensor, and the photons are gathered around every

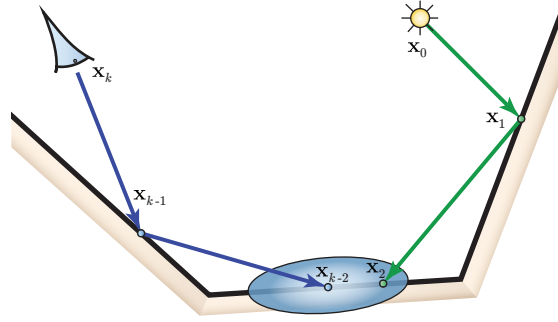


Figure 4.6: After tracing two subpaths, one from sensor and one from emitter, we merge the end vertex of one subpath to the end vertex of another based on the distance between them. If the distance is within the kernel bandwidth, the resulting constructed path is weighted by kernel value at this point.

hit point, giving a kernel-estimated value of the total outgoing radiance. Kernel estimation trades variance (visible as noise in the images) for bias (usually visible as oversmoothing of the illumination), which is a well-known “artifact” in the field of statistical kernel estimation [Silverman 1986]. The benefit of photon mapping is that the kernel estimation makes it more robust for complex illumination, such as specular-diffuse-specular paths, which are difficult (or sometimes even impossible) to sample with pure Monte Carlo-based approaches as we will see in Chapter 5. The noise of an image generated with photon mapping can vanish faster than with unbiased methods [Hachisuka et al. 2012], which is achieved by trading variance for bias.

We show how photon mapping can be derived from the measurement equation

Eq. 2.8, which can be written as

$$I = \int_{\mathcal{M} \times \mathcal{S}^2} W(\mathbf{x}, \mathbf{o}) L(\mu_{\mathcal{M}}(\mathbf{x}, -\mathbf{o}), \mathbf{o}) d\mathbf{x} d\mathbf{o}^\perp,$$

where \mathcal{S}^2 is the unit sphere and \mathcal{M} is the manifold of all scene surfaces in \mathbb{R}^3 ; $W(\mathbf{x}, \mathbf{o})$ is the importance of the eye subpath along the ray (\mathbf{x}, \mathbf{o}) ; $L(\mu_{\mathcal{M}}(\mathbf{x}, -\mathbf{o}), \mathbf{o})$ is the total outgoing radiance (carried by light subpaths of all lengths) at the point $\mathbf{x}_o = \mu_{\mathcal{M}}(\mathbf{x}, -\mathbf{o})$, where $\mu_{\mathcal{M}}(\mathbf{x}, -\mathbf{o})$ is the ray-casting function. The outgoing radiance at a surface point \mathbf{x}_o in direction \mathbf{o} can be estimated [Pharr and Humphreys 2010] with

$$\begin{aligned} L_o(\mathbf{x}_o, \mathbf{o}) &= \int_{\mathcal{S}^2} f(\mathbf{x}_o, \mathbf{o}, \mathbf{i}) L(\mathbf{x}_o, \mathbf{i}) d\mathbf{i}^\perp = \int_{\mathcal{S}^2} f(\mathbf{x}_o, \mathbf{o}, \mathbf{i}) \int_{\mathcal{M}} \delta(\mathbf{x}_o - \mathbf{x}_i) L(\mathbf{x}_i, \mathbf{i}) d\mathbf{x}_i d\mathbf{i}^\perp \\ &\approx \int_{\mathcal{S}^2} f(\mathbf{x}_o, \mathbf{o}, \mathbf{i}) \int_{\mathcal{M}} k_r(\mathbf{x}_o - \mathbf{x}_i) L(\mathbf{x}_i, \mathbf{i}) d\mathbf{x}_i d\mathbf{i}^\perp, \end{aligned} \quad (4.3)$$

where δ is an artificially introduced Dirac delta function, which we then replace by the smoothing kernel $k_r(t)$. Here we use \mathbf{x}_o and \mathbf{x}_i to denote end points of eye subpath and light subpath correspondingly.

After inserting Eq. 4.3 into the measurement equation, we apply Fubini's theorem multiple times to obtain

$$\begin{aligned} I &\approx \int_{\mathcal{M} \times \mathcal{S}^2} W(\mu_{\mathcal{M}}(\mathbf{x}_o, -\mathbf{o}), \mathbf{o}) \int_{\mathcal{S}^2} f(\mathbf{x}_o, \mathbf{o}, \mathbf{i}) \int_{\mathcal{M}} k_r(\mathbf{x}_o - \mathbf{x}_i) L(\mathbf{x}_i, \mathbf{i}) d\mathbf{x}_i d\mathbf{i}^\perp d\mathbf{x}_o d\mathbf{o}^\perp \\ &= \int_{\mathcal{M} \times \mathcal{M}} k_r(\mathbf{x}_o - \mathbf{x}_i) \int_{\mathcal{S}^2} W(\mu_{\mathcal{M}}(\mathbf{x}_o, -\mathbf{o}), \mathbf{o}) \int_{\mathcal{S}^2} f(\mathbf{x}_o, \mathbf{o}, \mathbf{i}) L(\mathbf{x}_i, \mathbf{i}) d\mathbf{i}^\perp d\mathbf{o}^\perp d\mathbf{x}_i d\mathbf{x}_o \\ &= \int_{\mathcal{M} \times \mathcal{M}} k_r(\mathbf{x}_o - \mathbf{x}_i) \Psi(\mathbf{x}_o, \mathbf{x}_i) d\mathbf{x}_i d\mathbf{x}_o, \end{aligned} \quad (4.4)$$

where $\Psi(\mathbf{x}_o, \mathbf{x}_i)$ is the contribution of all possible full paths constructed by connecting the endpoint \mathbf{x}_o of all eye subpaths to the endpoint \mathbf{x}_i of all light subpaths. Such full paths are weighted by $k(\mathbf{x}_i - \mathbf{x}_o)$.

Consider the example shown in Figure 4.7: path vertices from eye subpath are denoted as \mathbf{e}_i , light subpath vertices are \mathbf{l}_i ; the path $\mathbf{e}_0(\text{eye})-\mathbf{e}_1-\mathbf{e}_2-\mathbf{l}_2-\mathbf{l}_1-\mathbf{l}_0(\text{light})$

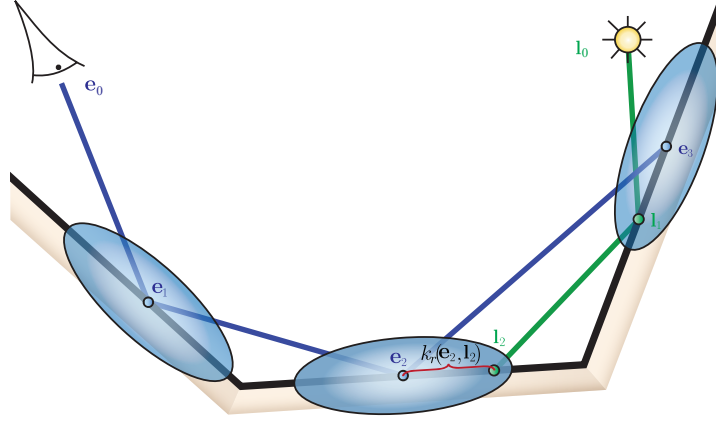


Figure 4.7: An example of path path construction with kernel estimation: an eye subpath $\mathbf{e}_0 - \mathbf{e}_1 - \mathbf{e}_2$ is connected to a light subpath $\mathbf{l}_0 - \mathbf{l}_1 - \mathbf{l}_2$ using the kernel k_r .

is one possible path constructed in $\Psi(\mathbf{e}_2, \mathbf{l}_2)$ and weighted by $k(\mathbf{e}_2 - \mathbf{l}_2)$. Let us now assume that we draw N eye subpaths from the importance distribution and J random light subpaths for each eye subpath from the radiance distribution. Then we can estimate the measurement with

$$\hat{I}_N = \frac{1}{NJ} \sum_{i=1}^N \sum_{j=1}^J k_{r_i}(\mathbf{e}_i - \mathbf{l}_j^i) \psi_{i,j}, \quad (4.5)$$

where $\psi_{i,j}$ is the contribution of the full path constructed using a single eye subpath ending at \mathbf{e}_i and a single light subpath ending at \mathbf{l}_j^i (see Figure 4.7). The superscript i denotes that a new sample set (a photon map) $\{\mathbf{l}^i\}_J$ is used for every new eye subpath i . This new formulation enables us to express the optimal bandwidth selection using a generalized kernel smoothing operator on top of the pixel measurement as will be shown later in Chapter 5.

The probability for two path vertices to land near each other (within a small radius r) can be extremely low [Georgiev et al. 2012], which can be caused, e.g., by a large scene extent or a high number of light sources. Therefore, in order to be a viable path construction technique, any kernel-based path construction requires efficient subpath caching techniques.

Sources of Bias. In addition to the variance coming from Monte Carlo sampling of subpaths, kernel-based methods have another source of error, called *bias*. Bias is a systematic estimation error that, unlike variance, might vanish slower or not vanish at all even with an infinite number of samples. The only source of biased estimation is the imprecise connection of two subpaths with proximity-based kernel merging of the two vertices. This bias can be classified into multiple subtypes, see Ingo Wald’s master thesis [Wald 1999] for more details. We will discuss finite-sample bias of kernel-based path generators, as well as the asymptotic behavior of bias in Chapter 5 and establish the link to the Monte Carlo variance.

4.5 Path Caching Schemes

All bidirectional path construction methods that work with subpaths from both camera and light direction at some point face a problem of efficiently selecting these subpaths. The problem of predicting the efficiency of constructing and then connecting two subpaths can be potentially as hard as the original problem of light transport. It is also complicated by the reach variety of scene and lighting complexity. Therefore, many prevalent solutions to this problem use caching of subpaths in some form.

4.5.1 Photon Maps

In photon mapping [Jensen 1996, Shirley et al. 1995], a large number of light subpaths is generated and the vertices are stored as photons. Light subpaths are then connected to camera subpaths using kernel estimation. These methods recently received more attention since the advent of progressive methods [Hachisuka et al. 2008, Knaus and Zwicker 2011] and unified frameworks coupling photon mapping and unbiased Monte Carlo methods [Georgiev et al. 2012, Hachisuka et al. 2012]. Compared to pure Monte Carlo methods, these approaches can introduce visible bias and still need significant amounts of memory for storing photons, even though the progressive methods partly alleviate this problem.

We show in Section 6 how photon mapping can be extended to a more general selective regularization in path space.

Progressive Photon Mapping. Hachisuka et al. [2008] introduced *progressive photon mapping* (PPM), which does not require storing all light subpaths for radiance estimation at the same time. In the progressive variant, the photon map is regenerated and the kernel estimation is updated recursively. They showed that photon mapping can be made a consistent method, where the systematic error vanishes in the limit. It has been further generalized to handle distributed ray-tracing effects [Hachisuka and Jensen 2009], and the photon tracing has been improved by using adaptive mutations of visible photon paths [Hachisuka and Jensen 2011]. Belcour and Soler [2011] proposed to enrich emitted photons with the spectrum of the carried radiance to adaptively choose the radiance estimation radius. Knaus and Zwicker [2011] developed a probabilistic framework, studied the asymptotic behavior of bias and variance, proposed a memoryless estimator, and generalized PPM to participating media. Jarosz et al. [2011] combined PPM with photon beams for improved rendering of participating media. Hachisuka et al. [2010] describe an error estimation framework for bias and variance estimation and early rendering termination. Georgiev et al. [2012] propose the combination of PPM and unbiased Monte Carlo estimators. Independently and parallel with their work, Hachisuka et al. [2012] also propose to combine PPM with unbiased techniques and derive optimal values of some PPM parameters for the case of multiple importance sampling. We will discuss these improvements and further improve the efficiency of kernel-based methods by building a theory of the optimal convergence rate in Chapter 5.

4.5.2 Light Subpath Pooling and Many-Light Methods

Instant radiosity (one of the *many-light methods*) can be considered as a caching scheme for bidirectional path tracing, which caches light subpath vertices. In this method, the entire lighting in a scene is represented by a set of *virtual point lights* (VPLs) [Keller 1997]. Multiple techniques based on this idea have gained attention in recent years. Often these methods employ clustering and importance sampling

(e.g. [Walter et al. 2005]), as well as multiple importance sampling [Walter et al. 2012]. We refer the reader to recent surveys on this topic [Dachsbacher et al. 2013, Krivánek et al. 2013].

4.6 Perturbation-based Path Construction

The core idea of all perturbation-based path generators is to construct a new light path based on an existing (previously constructed) path. One important trait of this class of generators is that they generate a set of *correlated paths*, that is, the newly perturbed path is similar to its predecessor. Therefore, one desired property of a perturbation-based path generator is to be able to explore the path space evenly by being able to perform large changes to the path.

All of the perturbation-based path generators require an initial *seeding path*. Therefore, they rely on other path construction methods, such as bidirectional path tracing (see Section 4.3.2) for constructing these initial seeding paths.

Since all of these generators rely on a previously constructed paths, one natural integration method for them is Markov chain Monte Carlo path integration.

4.6.1 Path Space Perturbations

Path space perturbations operate in the original domain of the path integral, the path space

$$\Omega = \Omega(\mathcal{M}) = \bigcup \Omega_k(\mathcal{M}),$$

which is the unified space of all possible light paths of all lengths. A path $\mathbf{X} \in \Omega$ is generated as a sequence of vertices on the scene manifold \mathcal{M} as $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k)_{k=1 \dots \infty}$ by perturbing path vertices.

Specialized Perturbations. As a part of Metropolis light transport, there is a set of basic path space perturbations. The first three perturbations in this set are crafted to

cover particular illumination effects, such as glossy reflections and caustics. These perturbations take the existing path and try to construct a new path of the same length by perturbing one of the outgoing directions at one or more vertices of the original path.

Given a path \mathbf{X} of length k , the first generator, *caustic perturbation*, selects a subpath of random length $l < k$ starting from the light source and perturbs it (see Figure 4.8, left). The perturbation finds the first vertex with a non-specular interaction starting from the light source and then perturbs its outgoing direction and generates the rest of the subpath using unidirectional light tracing. If the generation is successful, the new light subpath is then connected to the remaining camera subpath that remains unchanged. If the connection also succeeds, the new proposal path is generated. This perturbation is especially useful if the light source is hard to hit from the sensor and the perturbed subpath is constrained by specular or highly-glossy interactions. In this case, the caustic perturbation tries to perturb the light subpath through the constrained interactions until it reaches an easily connectible vertex with high amount of scattering¹.

A similar perturbation to the caustic perturbation, which instead perturbs from the sensor is called *lens perturbation* (see Figure 4.8, middle). The idea is symmetric to the caustic perturbation, except it starts from the sensor. In addition, the vertex at the camera is considered to be always mutable. This allows this perturbation to perturb the outgoing direction from the sensor, and therefore, travel across pixels on the image plane. This is an important strategy for image synthesis, as, unlike the caustic perturbation, it allows for simultaneous light transport integration at multiple pixels by walking on the image plane.

The last small-step perturbation, called *multichain perturbation*, is a combination of the lens and caustic perturbations. The idea is to start at either sensor or light source and select multiple subpaths to perturb (see Figure 4.8, right). For example, it can select a sensor subpath of length l , and after successfully perturbing, it can select another subpath of length l' starting at the end point of the just perturbed subpath. This is especially helpful when both light source and camera are hard to

¹Vertices with highly scattering interactions, such as diffuse, are easier to connect to.

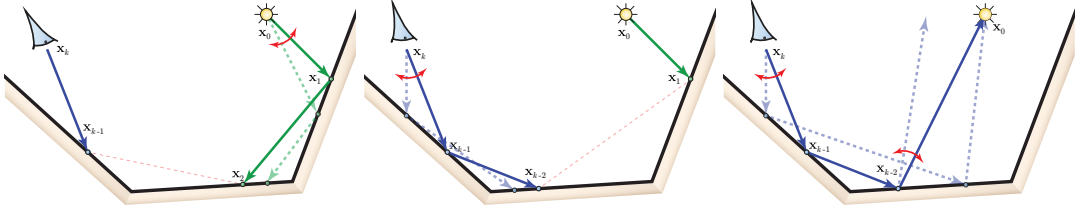


Figure 4.8: Path-space perturbations used in the original Metropolis light transport. From left to right: (a) caustic perturbation, (b) lens perturbation, (c) multichain perturbation.

connect to. One example is the caustics of an area light on the bottom of a swimming pool seen by a camera through the water interface. Both lens perturbation and caustic perturbation would fail to perturb such a path, because both of them would fail to connect to the water interface after perturbing one subpath from either side. A multichain perturbation can perturb two chains in this case, one from the camera to the bottom, and the second one through the water interface to the light source.

The sampling pdfs for these three perturbations are computed in the on-surface area measure analogously to the corresponding connection-based generators from Section 4.3. The full path pdf should also include the probability of selecting a subpath(s) to perturb on the original path.

Transdimensional Perturbation. All aforementioned mutation strategies have one important limitation: They do not change the path length. However, in order to explore all illumination features with various interactions, it is essential to have a mutation that can generate paths of different lengths from different subspaces of the path space Ω . Moreover, if we use a Markov chain Monte Carlo method, the exploration of the complete path space is a necessary condition for Markov chain to converge to a unique stationary distribution. Indeed, if a chain would not be able to reach all subspaces, it would explore only a particular subspace based on the initial seeding path.

In order to explore the complete path space, Veach and Guibas [1997] proposed to use a *bidirectional mutation*. This mutation randomly selects a subpath located anywhere on the current path, then deletes it, selects the length of a new sub-

path, and finally generates a new subpath of a pre-selected length using one of the connection-based path construction techniques (e.g., path tracing). The end vertices of a deleted subpath are carefully chosen to have non-specular interactions, otherwise, the connection-based methods would fail to connect to the end vertices.

The sampling probability density for the bidirectional mutation is a joint pdf of the pdf from the connection-based generator and the *probability mass function*¹ (pmf) of all decisions regarding the old and new subpaths (deletion start and length, as well as the length of the new subpath).

4.6.2 Trajectory Space Perturbation

Jakob and Marschner [2012] focused difficult light transport through specular manifolds and introduced another perturbation-based path construction method, called *manifold exploration*, that is specifically designed to improve the sampling of specular and highly glossy paths. They noticed that the integrand becomes ill-posed due to the manifold constraints of specular interactions. The method is based on the observation that a specular vertex entails a Dirac delta in the integrand (due to the BSDF), which effectively removes this dimension from the integration. This reduces the domain of integration to a lower-dimensional manifold embedded in path space. The path construction is a combination of a local linear solver and ray tracing for walking along such manifolds. This mutation strategy can replace the set of Veach’s caustic, lens, and multichain perturbations. The idea is to perturb the path in a specialized space, where it preserves hard constraints, like specular reflections, by utilizing the local knowledge about the geometry around the current path. In order to construct a new subpath, the method first constructs a local on-surface parameterization for every vertex of the current path and iteratively constructs the new path in the space of this local tangent frame parameterization. This perturbation performs a deterministic *manifold walk* procedure, which enforces Fermat’s principle as a constraint on specular vertices. For example, to render a caustic the algorithm first performs a lens perturbation by tracing from the sensor towards a perturbed direction until reaching the caustic. Then the rest of the path up to the

¹A probability mass function is a discrete analog of the probability density function.

light source is updated in a way that all half vectors of the inner vertices on the subpath are fixed.

By preserving the half vectors at these vertices, manifold exploration can construct a subpath from one fixed point to another through a chain of specular or highly-glossy interactions. In fact, this strategy attempts to eliminate some of the integration dimensions (at the specular/glossy interactions), while sampling other dimensions.

We discuss this method in more detail in Section 8.1.1, where we will improve it for glossy interactions. Note that unbiased (MC)MC methods, including manifold exploration, cannot find all possible transport paths, which can be alleviated by the regularization we propose in Section 6.

4.6.3 Primary Sample Space Perturbations

So far we have discussed path generators that operate on path vertices and outgoing directions. Kelemen et al. [2002] observed that in all Monte Carlo based light transport methods a light path of length k is generated stochastically with importance sampling from a vector of pseudorandom *primary samples* (ξ_0, \dots, ξ_n) drawn from a unit hypercube \mathcal{U}^n , where n is usually proportional to k . *Primary Sample Space Metropolis Light Transport* (PSSMLT, or Kelemen MLT) is a variation of Metropolis light transport algorithm, where perturbation of a path is performed in this space. Instead of perturbing vertex positions or outgoing directions in the path space Ω , PSSMLT stores the set of primary samples associated with the current path and perturbs these samples followed by the generation of a new perturbed path. The path itself can then be generated using one of the standard connection-based path generators.

This method is also relatively easy to implement, as it can be built as a superstructure on top of an existing Monte Carlo method, such as path tracing. In this case, the underlying pseudorandom number generator is replaced by a state vector of primary samples and the Monte Carlo method merely uses the perturbed primary samples as a new set of random numbers in order to generate a perturbed path. In

the end, the Metropolis move from Eq. 3.8 should be performed to either accept or reject the new perturbed vector of primary samples.

Small Step Perturbation. A *small step* perturbation takes a vector of random numbers used to generate the current path and perturbs this vector using a symmetric probability, usually a multidimensional Gaussian distribution [Kelemen et al. 2002] with a small step in the unit hypercube. A new path is then constructed using this new vector of perturbed random numbers. If the vector is too short, e.g., due to the new vertices or selection of another material layer, the new random numbers are generated independently and the vector is extended for such a path. One assumption behind these perturbations is that the importance sampling functions make the integrand flat enough in the primary space, so that it is possible walk over this space using a simple uniform random walk.

Large Step Perturbation. In order to discover new features quicker, we also need to occasionally do larger steps in the primary space. Therefore, a *large step* mutation was proposed, which regenerates the complete random vector from scratch and tries to construct a new path. That is equivalent to just generating a random path with the underlying Monte Carlo algorithm (e.g., path tracing). This mutation is selected with a small probability p , which needs to be accounted for during the computation of the acceptance probability. Note that the perturbed path does not depend on the current path of the Markov chain, because it was regenerated from scratch using independent sampling.

Another attractive property of the PSSMLT method is the elegance of computing an acceptance probability. Due to the fact that all perturbations in primary sample space are symmetric, they cancel out in the ratio of acceptance probability in Eq. 3.8. The remaining terms, the ratio of the path measurement contribution functions to the pdf coming from importance sampling, form the exact path throughput (see Section 3.2.1) that is used in ordinary Monte Carlo methods, such as path tracing and bidirectional path tracing.

4.6.4 Further Reading on Perturbation-based Construction

Several flavors of MLT and applications of MCMC sampling techniques have been proposed in computer graphics literature. Kitaoka et al. [2009] applied replica exchange to global illumination, where different path types or light transport phenomena (e.g., caustics, indirect diffuse illumination, etc.) are sampled each with a dedicated Markov chain using a tailored target distribution. MCMC sampling has also been used with photon mapping where light subpaths are mutated [Chen et al. 2011, Fan et al. 2005, Hachisuka and Jensen 2011].

Energy redistribution path tracing (ERPT) [Cline et al. 2005] uses the original set of MLT mutations [Veach and Guibas 1997] with a large number of short Markov chains to stratify the image plane sampling. ERPT is a variation of MLT that utilizes the fact that independent samplers, such as bidirectional path tracing, already provide a very good path generation. The idea is to try to redistribute the amount of energy carried by each initial path using short random walks. In order to do that, multiple Markov chains are started with the same seed path, where the number of chains is computed adaptively based on the path throughput. This scheme is very similar to the *lens mutation* [Veach and Guibas 1997] (not to be confused with lens perturbation), except that the number of mutations between reseeds of the chain is very low. The efficiency of ERPT is sensitive to the efficiency of the seeding path generator. The distribution region is manually set by the user, making it sensitive to the parameter tweaking for achieving the best trade-off between redistribution and stratification.

Also, extensions of MLT, such as Population Monte Carlo [Cappé et al. 2004] and *gradient-domain Metropolis light transport* [Lehtinen et al. 2013], do not directly change the path construction and can thus be used together with any existing perturbation. Gradient domain MLT computes image gradients plus a coarse approximation of an image. The final image is then obtained by solving a Poisson equation. The Population Monte Carlo (PMC) technique [Cappé et al. 2004] does not exactly belong to the class of Markov chain Monte Carlo methods, although it uses samples from the past. PMC iterates on populations of importance samples with kernels depending on previous samples, which approximate the ideal importance function.

I

Efficient Kernel Estimation for Light Transport

Optimal On-Surface Kernel Estimation

In this chapter, we analyze the properties of on-surface light transport methods based on kernel estimation. Particularly, we focus on the efficient convergence of photon mapping. Our work carries on the asymptotic analysis started by Knaus and Zwicker [2011] and the error estimation of Hachisuka et al. [2010]. We make use of adaptive recursive estimation methods [Hall and Patil 1994, Jones and Sheather 1991, Wand and Jones 1994] to develop a novel data-driven bandwidth selection for photon mapping.

Hachisuka et al. [2008] introduced a Progressive Photon Mapping (PPM), which uses an iterative kernel estimation to merge the radiance estimates from a series of photon maps. Similar to original photon mapping, it has low variance due to the reuse of light subpaths. In this chapter, we focus on the important question: what is the optimal reuse strategy for light subpaths? To this end, we make the following contributions.

- We establish a connection between PPM and the field of recursive estimation in statistics.
- Based on an asymptotic analysis of the class of PPM methods, we derive the optimal asymptotic convergence rate and the optimal values for previously user-specified parameters in PPM.
- We use a kernel estimation formulation of PPM as a basis for an adaptive data-driven bandwidth (gather radius) selection, which automatically and locally balances bias and noise in order to minimize the total error of a rendered image. The results of our method show that it converges faster and has less noise and bias than existing PPM methods.

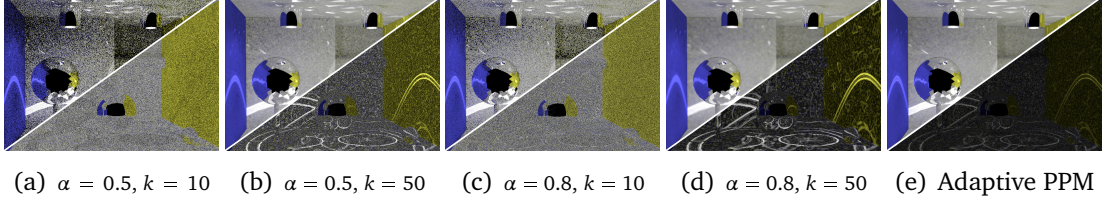


Figure 5.1: The BOX scene rendered with standard PPM with different values of the user-specified parameters α and k (for the k -NN search to determine the initial bandwidth r_1). Result images (top left part) and differences to the converged images (bottom right part) are shown after emitting 10 million photons. Here our adaptive method (e) shows the benefits of minimizing the overall estimation error.

The chapter is structured as follows: Section 5.1 introduces the background theory of progressive photon mapping. In Section 5.2 we establish a relation to existing recursive estimators in statistics and present an asymptotic analysis of PPM, leading to the optimal asymptotic convergence rate. We derive the novel adaptive bandwidth selection method in Section 5.4. Section 5.5 provides implementation details of our method. We present results in Section 5.6 which demonstrate the benefits of our adaptive method.

5.1 Kernel Estimation in Photon Mapping

This chapter is inspired by progressive photon mapping [Hachisuka et al. 2008], its extension stochastic PPM [Hachisuka and Jensen 2009], and the probabilistic analysis of PPM [Knaus and Zwicker 2011]. These techniques are able to robustly sample difficult paths in light transport, for example, reflected caustics, without suffering from the high memory requirements of the original photon mapping method. In this section, we will briefly recapitulate the most relevant aspects for our work.

The main challenge in conventional photon mapping is to balance the trade-off between the variance, or noise, and the expected error, or bias, in the radiance estimate. A photon map is used to obtain an approximation of the on-surface radiance distribution and, given a finite number of photons, one can either achieve low variance or low expected error, but not both. The main insight of progressive photon mapping is that we can obtain a solution with vanishing variance and expected

Table 5.1: Notations used throughout this chapter.

Symbol	Description
N	Number of estimation iterations (eye subpaths)
J	Photon map size (emitted photons per iteration)
$M \equiv NJ$	Total number of emitted photons
\mathbf{x}, \mathbf{o}	Surface point and outgoing direction
$\text{Var}[X]$	Variance of expression X
$E[X]$	Expected value of expression X
$\epsilon_N, \bar{\epsilon}_N$	One-step and average radiance estimation error at step N
\mathbf{e}, \mathbf{l}	Vertex of an eye subpath and light subpath
I, \hat{I}_N	Exact and estimated measurement of a pixel value
W	Importance (contribution of eye subpaths)
L	Outgoing radiance
γ_j	Contribution of the light subpath (i.e., photon) j
i	Eye subpath index
j	Light subpath index, photon index
r_N	Kernel radius (bandwidth) at step N
k, k_r	Canonical and normalized kernel with bandwidth r
Ψ	Contribution of all full paths through the estimation points
$\psi_{i,j}$	Contribution of eye subpath i with light subpath j

error by averaging the results over many photon maps [Hachisuka et al. 2008]. Therefore, progressive photon mapping is a multipass method that solves the rendering equation by accumulating statistics of photons from multiple photon maps. The initial eye pass traces rays from the camera and stores the non-specular hit points. The following photon passes trace photons from light sources and update the statistics on the hit points using these photons. The photon statistics include position of a hit point \mathbf{x} , the incident radiance times the BSDF $f(\mathbf{x}; \mathbf{o})$, and the shrinking sequence of search radii $\{r_N\}$. Here, the subindex N is the number of the photon tracing passes as well as the number of eye samples. At the N -th pass, the radiance value at the position \mathbf{x} towards the direction \mathbf{o} is estimated as

$$\hat{L}(\mathbf{x}, \mathbf{o}) = \frac{1}{J} \sum_{j=1}^J k_r(\mathbf{x}_j - \mathbf{x}) \gamma_j, \quad (5.1)$$

where the notation of variables is listed in Table 5.1. This estimation is based on the contributions γ_j of all light subpaths around the point \mathbf{x} applying a smoothing filter with kernel k_r .

Knaus and Zwicker [2011] rewrote this approximation as $\hat{L}(\mathbf{x}, \mathbf{o}) = L(\mathbf{x}, \mathbf{o}) + \epsilon$,

where ϵ represents the error introduced by the radiance estimation. Then they obtain a complete pixel measurement estimate \hat{I}_N as

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^N \frac{W_i}{p_{e,i}} \hat{L}(\mathbf{x}_i, \mathbf{o}_i) \quad (5.2)$$

for which the expected value $E[\hat{I}_N]$ converges to the true pixel value I and the variance $\text{Var}[\hat{I}_N]$ approaches zero as the number of eye samples N increases. Here, $W_i \equiv W(\mathbf{x}_i, \mathbf{o}_i)$ is the importance of the i -th eye subpath ending at the point \mathbf{x}_i and $p_{e,i}$ is the pdf of sampling this eye subpath.

Knaus and Zwicker re-derived PPM based on a more general probabilistic framework which allowed them to analyze the variance and the expected error of the radiance estimation in photon mapping from a probabilistic perspective. They showed that if the variance and the expected value of the average error $\bar{\epsilon}_N$ both approach zero as $N \rightarrow \infty$ then the estimator is consistent if

$$\begin{aligned} \text{Var}[\bar{\epsilon}_N] \rightarrow 0 &\implies \text{Var}[\hat{I}_N] \rightarrow 0, \\ E[\bar{\epsilon}_N] \rightarrow 0 &\implies E[\hat{I}_N] \rightarrow I. \end{aligned}$$

As a consequence, with each new sample the variance of the error ϵ_N at every iteration is allowed to increase by a factor

$$\frac{\text{Var}[\epsilon_{N+1}]}{\text{Var}[\epsilon_N]} = \frac{N+1}{N+\alpha}$$

for some constant α with $0 < \alpha < 1$.¹ This parameter α controls how quickly the variance is allowed to increase in each iteration. Knaus and Zwicker showed that α effectively determines a trade-off between vanishing variance and expected value of the average error. In summary, the variance and the expected value of the average error vanish as desired if $0 < \alpha < 1$, where α controls the relative rate of decrease. In addition, their asymptotic analysis showed that: (1) the variance vanishes to the order of $O(1/N^\alpha)$, and (2) the expected value of the average error

¹This is asymptotically equivalent to the initial radius reduction scheme by Hachisuka et al. [2008], but easier to analyze.

is proportional to the square radius and vanishes to the order of $O(1/N^{1-\alpha})$. That is, for $0 < \alpha < 1$ the squared radius decreases slower than the inverse number of photons. As a consequence, more and more photons are collected with increasing N , which guarantees $L(\mathbf{x}, \mathbf{o}) = \lim_{N \rightarrow \infty} \hat{L}_N(\mathbf{x}, \mathbf{o})$. Note that this equation neither converges for $\alpha = 0$, because the initial radius will not be decreased, nor for $\alpha = 1$ as the noise level increases.

As we will see, the choice of the parameter α is crucial and directly affects the asymptotic convergence rate of rendering methods based on kernel estimation. Moreover, we will also show that the choice of the initial radius r_1 affects progressive rendering as well as the absolute error of an image.

5.2 Recursive Estimation in PPM

In this section we analyze the radiance estimation in PPM and its asymptotic behavior. PPM methods comprise two user-specified parameters:

- the constant $\alpha \in (0; 1)$ controls the bandwidth reduction, effectively balancing between variance and bias reduction, and
- the initial per-pixel bandwidth r_1 which defines the initial “smoothing” of the radiance estimate.

Figure 5.1 shows the influence of differently chosen values on the rendering convergence. The choice of an optimal value for the parameter α seemed to be unintuitive, which can be observed from varying values given in previous works, for example, see Hachisuka et al. [2008], Knaus and Zwicker [2011]. In this section, we analyze the progressive radiance estimate and show that a unique asymptotically optimal α value exists. This is the first step towards our new method, as we will see that for progressive rendering, that is, when displaying results prior to convergence, the selection of the second parameter, the bandwidth r_1 , is of utmost importance (Section 5.4).

We begin by rewriting the radiance estimator used by Knaus and Zwicker [2011], Eq. 5.1 and Eq. 5.2, as a recursive equation using the notation that we will use throughout this chapter. We have

$$\begin{aligned}\hat{I}_N &= \frac{1}{N} \sum_{i=1}^N \frac{W_i}{p_{e,i}} \hat{L}(\mathbf{e}_i, \mathbf{o}_i) = \frac{1}{N} \sum_{i=1}^N \frac{W_i}{p_{e,i}} \left(\frac{1}{J} \sum_{j=1}^J k_{r_i}(\mathbf{e}_i - \mathbf{l}_j^i) \gamma_j^i \right) \\ &= \frac{N-1}{N} \hat{I}_{N-1} + \frac{1}{N} \frac{W_N}{p_{e,N}} \frac{1}{J} \sum_{j=1}^J k_{r_N}(\mathbf{e}_N - \mathbf{l}_j^N) \gamma_j^N,\end{aligned}\tag{5.3}$$

where $\hat{L}(\mathbf{e}_i, \omega_i)$ is the reflected radiance at the point \mathbf{e}_i and is approximated using kernel estimation (second equality); γ_j^i is the contribution of the j -th light subpath from the i -th photon map, landing at the point \mathbf{l}_j^i ; all other variables are listed in Table 5.1. The second line is obtained by regrouping and rearranging the terms, where the last summand is the contribution of the new photon map obtained from the step N .

The term $k_{r_N}(t) = \mu(r_N)^{-1} k(t/r_N)$ is the scaled and renormalized kernel $k(t)$, where $\mu(r_N) = \pi r_N^2$ is the normalization factor for an isotropic, circular on-surface estimation. For the sake of consistency with the existing work in statistics, we call the radius r_N *kernel bandwidth*. In PPM it shrinks with every iteration and converges to zero as $N \rightarrow \infty$. Similarly to Hachisuka et al. [2010], we use the Perlin kernel $k(t) = 1 + t^3(-6t^2 + 15t - 10)$ [Perlin 2002]. Note that the choice of the bandwidth r_N , but not the particular shape of the kernel, is important for the efficiency of the estimator [Silverman 1986].

PPM combines the results of photon maps using a shrinking bandwidth, controlled by the parameter α . Intuitively, the bandwidth trades both sources of error: bias and variance. In a first step, we will derive the asymptotically optimal value for α . However, as we will see, it only balances bias and variance in the limit. Thus our goal is to select the bandwidth for every pixel measurement and every iteration more precisely such that we can minimize the error on finite samples as well.

5.2.1 Recursive Estimation in Statistics

Recursive estimators, such as the radiance estimator in Eq. 5.3, have been proposed by Wolverton and Wagner [1969] and Yamato [1971]. Such an estimator is *consistent*, that is, it yields unbiased results in the limit, if, for some $\lambda \geq 0$ and a kernel with at least two existing non-zero derivatives, the sequence of bandwidths $\{r_N\}$ obeys the following conditions in d dimensions [Hall and Patil 1994, Yamato 1971]:

$$\lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{i=1}^N r_i^{-(d+\lambda)} \rightarrow 0, \quad \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N r_i^2 \rightarrow 0. \quad (5.4)$$

These two conditions guarantee that variance and bias, respectively, vanish in the limit. The parameter λ should be zero for estimating the function value [Hall and Patil 1994]. Later in Section 5.4.3 we will use the values $\lambda > 0$ for the estimation of the function derivatives. An equivalent result to Eq. 5.4 for $\lambda = 0$, that is, only for the estimation of the function value but not its derivatives, can be obtained from Eqs. 3, 5, 9, and 10 in Knaus and Zwicker's work [2011].

Both original PPM methods [Hachisuka et al. 2008, Knaus and Zwicker 2011] use an asymptotically equivalent formula for bandwidth shrinkage

$$r_{N+1}^2 = r_N^2 \frac{N + \alpha}{N + 1} = r_N^2 \left(1 + \frac{\alpha - 1}{N + 1} \right). \quad (5.5)$$

When updating r_N as in Knaus and Zwicker [2011] using Eq. 5.5, the conditions in Eq. 5.4 are met only for $\alpha \in (0; 1)$, which is in accordance with previous PPM methods.

This formula can be replaced with a simplified yet asymptotically equivalent explicit formula which allows for an easier analysis. To this end, we set $t_N \equiv r_N^2$ and solve the asymptotic equation as $N \rightarrow \infty$

$$t_{N+1} - t_N = t_N \frac{\alpha - 1}{N + 1} \Leftrightarrow t'(N) = t(N) \frac{\alpha - 1}{N + 1} \Rightarrow t(N) = c(N + 1)^{\alpha - 1}$$

for some constant c . Thus the sequence $\{r_N\}$ ($r_N = \sqrt{t_N}$) can be written as an explicit expression of r_1 as

$$r_N = r_1 N^{\frac{\alpha-1}{2}} \propto O(N^{\frac{\alpha-1}{2}}). \quad (5.6)$$

It simplifies the asymptotic analysis and is directly computable for any iteration N .

5.3 Asymptotic Convergence Analysis

5.3.1 Variance and Expectation of Estimation Error

We use the same notation as in Knaus and Zwicker [2011] and the initial equations used for the asymptotic analysis are taken from this article. Note that all approximations are done in the scope of an *asymptotic analysis*.

We can simplify the asymptotic variance of the estimation error (based on two last equations in Appendix E of Knaus and Zwicker [2011]). We have

$$\begin{aligned} \text{Var}[\bar{\epsilon}_N] &\approx \frac{\text{Var}[\epsilon_1]}{N^2} \left(1 + \sum_{i=2}^N i \alpha B(\alpha, i) \right) \\ &= \frac{\text{Var}[\epsilon_1]}{N^2} \left(1 + \frac{2\Gamma(N + \alpha) - \alpha\Gamma(\alpha)\Gamma(N + 2)}{(\alpha - 2)\Gamma(N + \alpha)} \right) \\ &\approx \frac{\text{Var}[\epsilon_1]}{N^2} \frac{\alpha\Gamma(\alpha)\Gamma(N)(N + 1)(N + 2)}{(2 - \alpha)\Gamma(N + \alpha)} \\ &\approx \frac{\text{Var}[\epsilon_1]}{N^2} \frac{\alpha B(\alpha, N)}{2 - \alpha} \approx \text{Var}[\epsilon_1] \frac{\alpha\Gamma(\alpha)N^{-\alpha}}{2 - \alpha} \approx \frac{\text{Var}[\epsilon_1]}{(2 - \alpha)N^\alpha}, \end{aligned}$$

where $B(\cdot)$ and $\Gamma(\cdot)$ are the Beta and the Gamma functions, respectively. The expected estimation error can also be simplified (based on last equation, second line in Appendix F of Knaus and Zwicker [2011]):

$$\begin{aligned} \text{E}[\bar{\epsilon}_N] &= \frac{\text{E}[\epsilon_1]}{N} \left(\frac{\alpha + N}{B(\alpha, N)\alpha^2 N} - \frac{1}{\alpha} \right) \\ &\approx \frac{\text{E}[\epsilon_1]}{N} \frac{\alpha + N}{\Gamma(\alpha)N^{-\alpha}\alpha^2 N} \approx \frac{\text{E}[\epsilon_1]}{\alpha^2 \Gamma(\alpha)N^{1-\alpha}} \approx \frac{\text{E}[\epsilon_1]}{\alpha N^{1-\alpha}}. \end{aligned}$$

5.3.2 Convergence Rate of Photon Mapping

Throughout the chapter we base our analysis on the minimization of the *Asymptotic Mean Squared Error* (AMSE). The *asymptotic* mean squared error of progressive photon mapping can be written using Eq. 5.3 and the previous derivations and results for variance and expectation of the measurement as

$$\begin{aligned} \text{AMSE}[\hat{I}_N] &= \text{Var}[\hat{I}_N] + \mathcal{B}[\hat{I}_N]^2 \approx \frac{1}{N} \text{Var} \left[\frac{W}{p_e} L \right] + \\ &\quad \mathbb{E} \left[\frac{W}{p_e} \right]^2 \left(\frac{\text{Var}[\epsilon_1]}{(2-\alpha)N^\alpha} + \frac{E[\epsilon_1]^2}{\alpha^2 N^{2-2\alpha}} \right). \end{aligned} \quad (5.7)$$

Now we can derive the asymptotic convergence rate for PPM and the asymptotically optimal value of the parameter α . After equating the first derivative of the Eq. 5.7 to zero and performing asymptotic simplifications we obtain

$$N^{3\alpha-2} = \frac{\alpha^2 \text{Var}[\epsilon_1]}{2(2-\alpha) \mathbb{E}[\epsilon_1]^2}.$$

Assuming that the right part is finite, the only local minimum with respect to α is

$$\boxed{\alpha_{\text{opt}} = 2/3.} \quad (5.8)$$

Therefore, in terms of the MSE, this is the asymptotically optimal value of α for shrinking the bandwidth.

From Eq. 5.6 and Eq. 5.8 we can obtain the asymptotically optimal shrinking rate of the bandwidth

$$\boxed{r_N = r_1 N^{-1/6} \propto O(N^{-1/6}).} \quad (5.9)$$

This result is important, as it shows that using α_{opt} for the bandwidth update yields the same asymptotically optimal rate that we find in statistics literature [Wand and Jones 1994].

Note that Knaus and Zwicker [2011] thoroughly studied the dependence of bias

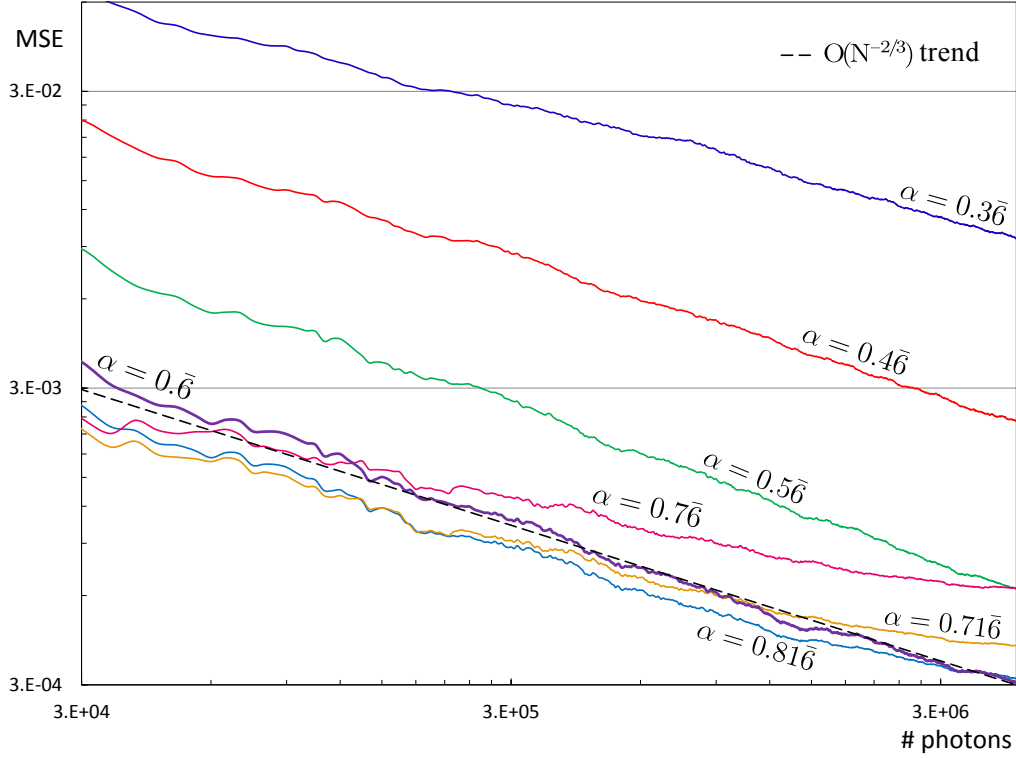


Figure 5.2: Mean Squared Error (plotted in log-log scale) for the image shown in Figure 5.4(a) rendered with PPM with different values of α . The initial bandwidth is selected with a k-NN search with $k = 10$ and different values for α : $0.3\bar{6}/0.4\bar{6}/0.5\bar{6}/0.6$ (optimal)/ $0.71\bar{6}/0.7\bar{6}/0.81\bar{6}$. Note that the optimal convergence rate can be outperformed by suboptimal α -values depending on the choice of the initial bandwidth.

and variance on the parameter α *independently* of each other. They show that different α -values trade the vanishing rate between variance and bias. Yet α_{opt} balances these rates such that the MSE is asymptotically minimized, that is, variance and (squared) bias vanish with the same rate. Note that any non-optimal α unbalances asymptotic bias and variance and slows the convergence rate.

It is also noteworthy that the optimal value of α and the asymptotic behavior of the sequence $\{r_N\}$ remain the same with sparse updates of the bandwidth, that is, when the bandwidth is updated once per J samples, as the conditions in Eq. 5.4 are still met. Consequently, the *asymptotic* convergence of PPM does not depend on the size of the photon map. However, in order to obtain consistent results with finite number of samples, one can employ the compensation procedure proposed by Jarosz et al. [2011].

Convergence Rate. Substituting α_{opt} back into the AMSE leads to

$$\text{AMSE}[\hat{I}_N] \approx \frac{1}{N} \text{Var} \left[\frac{W}{p_e} L \right] + \frac{3}{4N^{\frac{2}{3}}} \text{E} \left[\frac{W}{p_e} \right]^2 \left(3 \text{E} [\epsilon_1]^2 + \text{Var} [\epsilon_1] \right). \quad (5.10)$$

This shows that the optimal convergence rate for PPM is the convergence rate of the slowest term and therefore is

$$\boxed{\text{AMSE}[\hat{I}_N] \propto O(N^{-2/3})}. \quad (5.11)$$

This result also aligns with convergence rates found in existing literature for two-dimensional estimators [Silverman 1986, Wand and Jones 1994]. Note that this is the *optimal asymptotic convergence rate* for photon mapping techniques using a density estimation on surfaces. An experimental measurement of the convergence rate with different values of α is shown in Figure 5.2.

It is interesting to compare this result to the asymptotic convergence rate of unbiased Monte Carlo techniques such as path tracing. The AMSE of these techniques (consisting of variance only) is $\text{AMSE}[MC] = \text{Var}[MC] = O(M^{-1})$, where M is the number of sampled full paths for a pixel. The maximum asymptotic convergence rate of PPM is $O(N^{-2/3})$ and thus slower than that of unbiased Monte Carlo techniques. This is because the probability of constructing a full path in PPM is also proportional to the kernel footprint πr_N^2 , which shrinks as $O(N^{-1/3})$, thus reducing the probability of constructing a full path.

However, this does not mean that unbiased Monte Carlo estimators should always be preferred due to their superior rate of convergence. Photon mapping can be more efficient at constructing some types of paths, leading to a better convergence in finite amount of time. Moreover, there exist *undecidable paths* in path space, such that no algorithm can construct all of them. One example is pure specular-diffuse-specular paths, for example, a reflected caustic seen from a pinhole camera where the light is coming from a point light source. In all unbiased Monte Carlo techniques such paths require finding the path from one given point to another one. It is possible to find such paths numerically in simple cases [Mitchell and

Hanrahan 1992], but this introduces bias and cannot construct these paths exactly. Note that generally there exist configurations of perfect specular surfaces where this was proven undecidable [Reif et al. 1994]. And this is not just a theoretical issue: almost all presented example scenes contain such paths (for example those shown in Figure 5.6).

Finite Number of Samples. Obviously we are interested in an optimal result for a finite number of iterations and photons when rendering images. Most importantly, α_{opt} only defines the asymptotic shrinkage rate of the bandwidth, but not the optimal value for a particular (finite) number of iterations, as the initial bandwidth r_1 is an unknown in the bandwidth update in Eq. 5.9. With finite numbers of samples it is important to directly select the optimal bandwidth at every iteration in addition to the optimal asymptotic shrinkage rate. This can be observed from the results shown in Figure 5.2: due to a suboptimal initial bandwidth r_1 the optimal α -parameter outperforms others only when many samples are taken. In order to select the bandwidth more precisely, we reformulate PPM in the next section. In Section 5.4 we derive an adaptive method which is asymptotically optimal as well, but also yields near-optimal results significantly faster. As we will see, selecting the bandwidth is of utmost importance for rendering images with finite numbers of photons.

5.4 Adaptive Bandwidth Selection

The main goal of the preceding derivations is to analyze this trade-off and to obtain the optimal strategy for balancing variance and bias at every step of the recursive estimation. In statistics, this problem is known as *bandwidth selection* [Jones and Sheather 1991, Silverman 1986], where the optimal bandwidth r is selected in order to minimize the estimation error.

Knaus and Zwicker [2011] studied several methods for determining the *initial* bandwidth r_1 (starting condition for Eq. 5.5): (1) globally constant bandwidth, (2) local bandwidth based on k -Nearest-Neighbors (k -NN) search, and (3) local band-

width based on ray differentials—and none of them leads to the optimal choice (see Knaus and Zwicker [2011] for discussion). Figure 5.6 shows an example where the result is still oversmoothed even after emitting 10^{11} photons. The reason is that with an improper initial bandwidth, the recursive estimator accumulates suboptimal results, while subsequent iterations (with smaller bandwidth) are weighted less and less; thus the error remains large even for large N .

Bandwidth selection should be also locally adaptive in order to smooth the signal in sparse regions as well as preserve crisp radiance discontinuities. In the following, we derive the optimal bandwidth to improve the efficiency of the measurement estimation with a finite number of samples.

5.4.1 Asymptotic MSE of the Measurement

Our goal is a bandwidth selection procedure which minimizes the AMSE. We start with the AMSE from Eq. 5.7, which contains both the variance coming from eye subpaths and the kernel estimation MSE (variance and squared bias) as

$$\text{AMSE}[\hat{I}_N] \approx \text{Var} \left[\frac{1}{N} \sum_{i=1}^N \frac{W}{p_e} L \right] + \text{E} \left[\frac{W}{p_e} \right]^2 \left(\text{E} [\bar{\epsilon}_N]^2 + \text{Var} [\bar{\epsilon}_N] \right).$$

The first step for bandwidth selection is to analyze the relations between the Monte Carlo variance and the variance coming from the kernel estimation. The term L in the first summand is the radiance at a single iteration i . If we denote as S_i the average number of full paths for the Monte Carlo integration of the pixel measurement constructed at the i th iteration, then the first summand of the AMSE becomes

$$\text{Var} \left[\frac{1}{N} \sum_{i=1}^N \frac{W}{p_e} L \right] = \text{Var} \left[\frac{1}{N} \sum_{i=1}^N \frac{W}{p_e} \frac{1}{S_i} \sum_{j=1}^{S_i} \gamma_j^i \right] = \text{Var} \left[\frac{1}{N} \sum_{i=1}^N \frac{1}{S_i} \sum_{j=1}^{S_i} \psi_{i,j} \right].$$

The samples $\psi_{i,j}$ are independent and identically distributed and thus we apply the Bienaymé formula. Also the number of samples S_i at the i -th iteration can be approximated as $S_i \approx \pi r_i^2 J p_l$ where p_l is the mean density of light subpaths

(photons) around all estimation points. By this we obtain:

$$\text{Var} \left[\frac{1}{N} \sum_{i=1}^N \frac{1}{S_i} \sum_{j=1}^{S_i} \psi_{i,j} \right] \approx \frac{1}{\pi p_l J} \text{Var}[\psi] \frac{1}{N^2} \sum_{i=1}^N r_i^{-2}.$$

The variance $\text{Var}[\bar{\epsilon}_N]$ can be expressed using Appendix B from Knaus and Zwicker [2011], and by using Hall and Patil [1994] we obtain

$$\text{Var}[\bar{\epsilon}_N] \approx (\text{Var}[\gamma] + \text{E}[\gamma]^2) \frac{p_l k_3}{J} \frac{1}{N^2} \sum_{i=1}^N r_i^{-2},$$

where $k_3 = \int_{\mathbb{R}} k(t)^2 dt$. Thus the variance of the AMSE becomes

$$\left(\aleph p_l k_3 + \frac{1}{\pi p_l} \text{Var}[\psi] \right) \frac{1}{J N^2} \sum_{i=1}^N r_i^{-2} \approx \frac{1}{\pi p_l J} \text{Var}[\psi] \frac{1}{N^2} \sum_{i=1}^N r_i^{-2},$$

where $\aleph = \text{E}[W/p_e] (\text{Var}[\gamma] + \text{E}[\gamma]^2)$. The last step is based on the assumption (in spirit of Knaus and Zwicker [2011]) that $1/p_l \gg p_l$. Using the equality $\text{E}[\frac{W}{p_e}] \text{E}[\bar{\epsilon}_N] = \text{E}[\frac{W}{p_e} \bar{\epsilon}_N] = \mathcal{B}[\hat{I}_N]$, we obtain

$$\text{AMSE}[\hat{I}_N] \approx \frac{1}{\pi p_l J} \text{Var}[\psi] \frac{1}{N^2} \sum_{i=1}^N r_i^{-2} + \mathcal{B}[\hat{I}_N]^2. \quad (5.12)$$

where $\mathcal{B}[\hat{I}] = \text{E}[\hat{I}] - I$ is the bias of the estimator; p_l is the average density of light subpaths in the proximity of all estimation points in the footprint of the pixel measurement. Therefore, the variance of the AMSE is dominated by the variance term of the Monte Carlo integration of the measurement.

The goal of local bandwidth selection is to minimize the AMSE by balancing between the measurement variance and the bias of the kernel estimation. Note that this is possible as the bandwidth also influences the Monte Carlo variance of the path integral: the larger the bandwidth, the more full paths are constructed.

Using existing approximations (e.g., Hall and Patil [1994], Hachisuka et al. [2010], and Knaus and Zwicker [2011]) the asymptotic approximation

of the bias is expressed as

$$\mathcal{B}[\hat{I}_N] \approx \frac{1}{2}k_2\Delta I \frac{1}{N} \sum_{i=1}^N r_i^2, \quad (5.13)$$

where ΔI is the Laplacian of the measurement with respect to the kernel estimation, which we discuss in detail in Section 5.4.3; and $k_2 = \int_{\mathbb{R}} t^2 k(t) dt$ is the kernel-dependent constant [Silverman 1986]. The exact value for the Perlin kernel can be found in Hachisuka et al. [2010]. Using Eq. 5.12 and Eq. 5.13 we obtain

$$\text{AMSE}[\hat{I}_N] \approx \frac{1}{\pi p_l J} \text{Var}[\psi] \frac{1}{N^2} \sum_{i=1}^N r_i^{-2} + \frac{1}{4} k_2^2 (\Delta I)^2 \left(\frac{1}{N} \sum_{i=1}^N r_i^2 \right)^2.$$

One can minimize this equation with respect to r_N , while letting all other r_i to be constant. However, the solution is computationally expensive and we need to make further simplifying assumptions.

Simplified Bandwidth Selection. We assume that the bandwidth r_N approaches the optimal value \bar{r}_N quickly enough such that we can let $\bar{r}_N \approx r_i \approx 1/N \sum_{i=1}^N r_i$. In other words, we assume that r_i quickly stabilizes during a few initial iterations and does not change significantly during the subsequent iterations (and thus the difference in error introduced by using \bar{r}_N instead of r_i is negligible). Therefore, we replace all r_i in the previous equation by \bar{r}_N and obtain the new simplified AMSE'

$$\text{AMSE}'[\hat{I}_N] \approx \frac{1}{\pi p_l J N \bar{r}_N^2} \text{Var}[\psi] + \frac{1}{4} k_2^2 (\Delta I)^2 \bar{r}_N^4. \quad (5.14)$$

Minimizing AMSE' means that we always minimize the current error by using the parameters estimated during previous iterations. After minimizing this equation with respect to \bar{r}_N , we obtain an asymptotically optimal bandwidth as

$$\boxed{\bar{r}_N = \left(\frac{2 \text{Var}[\psi]}{\pi J p_l k_2^2 (\Delta I)^2} \right)^{1/6} N^{-1/6}}, \quad (5.15)$$

where the density p_l is estimated using a recursive estimator identical to Eq. 5.3), but without the actual photon values γ_j^i .

It is the same asymptotic convergence rate as the one we obtained for kernel estimation earlier in Section 5.3.2. However, Eq. 5.15 provides an explicit expression for bandwidth selection, which will allow us to obtain faster convergence and better results in finite amount of time.

The Laplacian ΔI in Eq. 5.15 is unknown, yet essential for the bandwidth selection. To this end, we discuss various bandwidth selection approaches and show how to estimate ΔI in Section 5.4.3.

5.4.2 The Plug-In Method for Bandwidth Selection

Many adaptive estimation techniques are known in literature (see Jones et al. [1996] for a survey), however, most of them are not suitable for our purposes as they either require to store a huge amount of data (e.g., Donoho et al. [1993] and Katkovnik and Shmulevich [2002]), or require access to all previous samples, for example, (biased) cross-validation and expectation-maximization methods. We use the recursive modification [Hall and Patil 1994] of the *plug-in bandwidth selection* method [Sheather and Jones 1991, Wand and Jones 1994]. It can be computed fully recursively and efficiently, has low variance, and good behavior for finite samples.

The main idea of the plug-in method is to plug an estimate $\Delta \hat{I}_N$ of the unknown Laplacian ΔI into the original estimator Eq. 5.15. This requires another kernel estimation for the derivatives of on-surface radiance. However, as noted by Jones and Sheather [1991], this requires selecting yet another bandwidth r' for the estimation of the derivatives, which is different from the optimal bandwidth r used for estimating the measurement itself. The necessity of selecting another bandwidth seemingly leads to a nested hierarchy of such bandwidths for higher derivatives. However, as pointed out in previous work [Jones et al. 1996, Wand and Jones 1994], the second bandwidth r' is less critical. In fact, we can estimate it using a normal distribution fit to the estimated derivatives (the so-called rule of thumb),

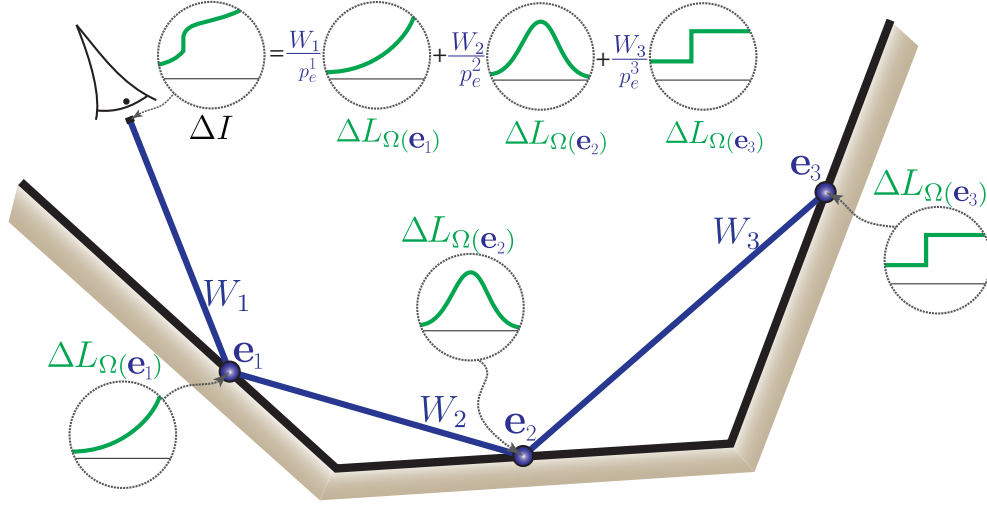


Figure 5.3: The Laplacian of a pixel measurement is computed as the sum of the Laplacians of the on-surface incident radiance at the path vertices, $L_{\Omega(e_i)}$, weighted by the respective importance W_i .

without causing a significant loss in convergence performance. In the nomenclature of Wand and Jones [1994], we use a simplified $\mathcal{S} - \mathcal{N}$ scheme: in the \mathcal{N} -step, we use a normal distribution fit to select the bandwidth r' for the estimation of the Laplacian. And the \mathcal{S} -step uses the plug-in method to select the final bandwidth r based on the value of estimated Laplacian.

5.4.3 Recursive Estimation of the Laplacian

In order to estimate the bias introduced by kernel smoothing, it is necessary to estimate the curvature of the underlying function. In our case, the estimation is done by constructing full paths using a spatial kernel smoothing to connect eye subpaths to light subpaths. This means that smoothing happens around every vertex of every possible path of the measurement equation I . To this end, we use a two-step approach: first, we estimate the Laplacian at the surface at every path vertex of the measurement (denoted as *vertex-Laplacian*, $L_{\Omega(e_i)}$, Figure 5.3). Next, we compute a sum of all estimated vertex-Laplacians weighted by the respective importance function. We denote this as the *Laplacian of the pixel measurement* with respect to the on-surface incident radiance (ΔI in Figure 5.3). It determines the average curva-

ture of the incident radiance at all path vertices of paths going through the pixel weighted by the respective importance. Thus, intuitively, it determines the flatness of the incident radiance for the measured pixel.

We compute the vertex-Laplacian $\Delta \hat{L}_{\mathbf{e}_N}$ by approximating the second derivatives along two orthogonal directions $\Delta \hat{L}_{\mathbf{e}_N} = \hat{L}_{\mathbf{e}_N, \mathbf{u}}^{(2)} + \hat{L}_{\mathbf{e}_N, \mathbf{v}}^{(2)}$, where \mathbf{e}_N is the estimation point at the step N ; \mathbf{u} and \mathbf{v} are two orthonormal basis vectors defined on the surface at the estimation point; and $\hat{L}_{\mathbf{e}_N, \mathbf{u}}^{(2)}$, $\hat{L}_{\mathbf{e}_N, \mathbf{v}}^{(2)}$ are the estimators for the second derivative along \mathbf{u} and \mathbf{v} , respectively. Note that the vertex-Laplacian is invariant to the rotation of the basis (\mathbf{u}, \mathbf{v}) .

Each derivative is estimated recursively using shrinking finite differences as proposed by Ngerng [2011]. Furthermore, we replace \hat{L}_N with \hat{I}_N and $\hat{L}_{\mathbf{e}_N}^{(2)}$ with $\hat{I}_{\mathbf{e}_N}^{(2)}$ after weighting by the importance, and denote the value of the measurement with radiance estimated at the point $\mathbf{e}_i + \mathbf{u}r'_i$ as $\hat{I}_{+\mathbf{u}r'_i}$. The equation is provided for $\hat{I}_{\mathbf{u}, N}^{(2)}$ (and is analogous for $\hat{I}_{\mathbf{v}, N}^{(2)}$)

$$\begin{aligned} \hat{I}_{\mathbf{u}, N}^{(2)} &\approx \frac{1}{N} \sum_{i=1}^N \frac{\hat{I}_{+\mathbf{u}r'_i} + \hat{I}_{-\mathbf{u}r'_i} - 2\hat{I}}{r_i'^2} \\ &= \frac{1}{NJ} \sum_{i=1}^N \sum_{j=1}^J \psi_{i,j} \frac{1}{r_i'^2} \left(k_{r'_i}(\mathbf{e}_i - \mathbf{l}_j^i + \mathbf{u}r'_i) + k_{r'_i}(\mathbf{e}_i - \mathbf{l}_j^i - \mathbf{u}r'_i) - 2k_{r'_i}(\mathbf{e}_i - \mathbf{l}_j^i) \right). \end{aligned} \quad (5.16)$$

Consequently, we compute the Laplacian of the measurement as

$$\Delta \hat{I}_N = \hat{I}_{N, \mathbf{u}}^{(2)} + \hat{I}_{N, \mathbf{v}}^{(2)}. \quad (5.17)$$

Radiance Steps (Discontinuities). The asymptotic behavior of the AMSE at discontinuities has a higher convergence rate. Van Eeden [1985] showed that for a point on the discontinuity, the optimal bandwidth shrinkage rate is $O(N^{-1/2d})$, where d is the dimensionality of the estimator. Due to the robustness of the Laplacian estimator the discontinuities are handled automatically: the finite difference in the numerator of Eq. 5.16 becomes constant and the optimal shrinkage rate of r'_N leads to the growth rate of the Laplacian $O(N^{1/4})$. After substituting this rate

into Eq. 5.15, one can validate that the asymptotic shrinkage rate of r_N becomes indeed $O(N^{-1/4})$.

Consistency of the Derivative Estimation. In general, for a consistent recursive estimation of the s -th-order derivative, the asymptotic bandwidth reduction rate of r' should obey Eq. 5.4 with $\lambda = 2s$ [Hall and Patil 1994]. This restricts the valid range for the power of the shrinkage rate of r' in $d = 2$ dimensions to

$$O(N^{-1/(2s+2)}) < r'_N < O(N^0).$$

Consequently, for a consistent estimation of the second derivative the rate must be slower than $O(N^{-\frac{1}{6}})$ (which is by coincidence equal to the optimal shrinking rate for the estimation bandwidth r). This means that we cannot use the same bandwidth for the estimation of both the function and its second derivative, otherwise the estimation of the latter diverges.¹

Analogous to Section 5.3.2, we obtain the asymptotically optimal values by minimizing the AMSE of the second derivative estimator as in Wand and Jones [1994] (where r'_1 is some constant)

$$r'_{N,\text{opt}} = r'_1 N^{-1/8}.$$

Selecting the Optimal Bandwidth for the Derivative Estimation. In the \mathcal{N} -step of the $\mathcal{S} - \mathcal{N}$ scheme, we estimate the bandwidth r'_N by applying Silverman's [1986] rule of thumb to the second derivative. That is, we assume that the actual function of the Laplacian ΔI can be approximated by a normal distribution and select the optimal bandwidth r' based on this assumption. For the 2D Perlin kernel the bandwidth is

$$r'_N \approx 1.9635 \hat{\sigma}_N N^{-1/8}, \quad (5.18)$$

¹This also shows that when using Eq. 5.5 for updating r' for second derivative estimation (as in Hachisuka et al. [2010]), the valid range for the α parameter is $(2/3; 1)$, otherwise the estimation diverges. The optimal value for the second derivative is $\alpha_{\text{opt}} = 3/4$.

where $\hat{\sigma}_N = \sqrt{\text{Var}[\Delta \hat{I}_N]}$ is the standard deviation of the Laplacian estimate; the variance is then estimated recursively by updating the first two moments of the derivative estimator.

5.4.4 Convergence and Consistency

The adaptive bandwidth selection scheme does not change the asymptotic convergence rate of PPM acquired in Eq. 5.11. However, it provides better results with finite samples (see Figure 5.5) compared to the bandwidth selection used in previous work. Moreover, in contrast to our method, the bias in the previous PPM methods is still noticeable even in relatively converged images after emitting 10^{11} photons (see Figure 5.6). The reason is inherent to the recursive estimator: the contribution of the N -th iteration contributes to the solution with a weight $1/N$, that is, an initially unlucky choice of the bandwidth can be hardly corrected later on. In contrast, our method recalibrates the bandwidth based on the input data at every iteration.

In order to ensure a robust behavior of our adaptive method, we bound the optimal bandwidth r_N in Eq. 5.15 within the interval $[r_{\min}N^{-1/6}, r_{\max}N^{-1/6}]$, where r_{\min} and r_{\max} are user-defined minimum and maximum initial bandwidths. This guarantees convergence in regions where no reliable statistics are available, for example, in completely black areas of the image. These bounds are not crucial for the method and are reached rarely, usually in some dark places during initial iterations when the estimates are noisy. We found that setting r_{\min} and r_{\max} to $10^{-6}R_{\text{scene}}$ and $10^{-1}R_{\text{scene}}$, respectively, where R_{scene} is the bounding radius of the scene, works well for all scenes we have tested. In principle, if the bounds are too narrow, the benefits of the adaptive method might be smaller, as there is less freedom to adjust the bandwidth; if the upper bound is too high, the performance might drop during the first iterations due to extensive queries to the photon map.

Algorithm 1 Adaptive bandwidth selection for a single pixel

Initialization stage (before the first frame)

$$\{r'_1, r_1\} \leftarrow r_{\max} \quad \triangleright \text{Both bandwidths start with maximum value}$$

$$\{E[\psi], E[\psi^2], E[\Delta \hat{I}_0], E[\Delta \hat{I}_0^2], \hat{p}_{l,0}\} \leftarrow 0 \quad \triangleright \text{Clear other values}$$
Estimation pass N

Generate photon map

for each pixel **do**

Trace eye subpath

 $\{E[\Delta \hat{I}_N], E[\Delta \hat{I}_N^2]\} \leftarrow \text{Eq. 5.17} \quad \triangleright \text{Update Laplacian's moments}$ $r'_N \leftarrow \text{Eq. 5.18} \quad \triangleright \text{Update Laplacian's bandwidth } r'_N$ $E[\psi], E[\psi^2] \leftarrow \text{Eq. 5.3} \quad \triangleright \text{Update moments of contribution } \psi$ $\hat{p}_{l,N} \leftarrow \frac{1}{NJ} \sum \sum k_{r_i}(\mathbf{e}_i - \mathbf{l}_j^i) \quad \triangleright \text{Update density } p_l \text{ (Eq. 5.3 w/o } \gamma)$ $r_N \leftarrow \text{Eq. 5.15} \quad \triangleright \text{Update PPM bandwidth } r_N$ Obtain radiance estimate using bandwidth r_N

Update pixel value

end for

5.5 Implementation Details

In this section, we detail the differences of an APPM implementation compared to standard PPM. APPM requires to store 6 additional single precision floating point values per pixel $E[\psi]$, $E[\psi^2]$, $\Delta \hat{I}_N$, $E[\Delta \hat{I}_N]$, $E[\Delta \hat{I}_N^2]$, and $\hat{p}_{l,N}$. At the first iteration, we initialize the values of r_1 and r'_1 with the maximum bandwidth r_{\max} from Section 5.4.4. In the subsequent iterations, we update the estimates for these values and recompute the bandwidth r_N using Eq. 5.15. We use Eq. 5.16 for the estimation of the Laplacian ΔI , where the bandwidth r'_N is selected according to Silverman's rule of thumb from Eq. 5.18.

The pseudocode shown in Algorithm 1 summarizes our bandwidth selection process. All variances are computed as a difference of the corresponding moments, that is, as $\text{Var}[X] = E[X^2] - E[X]^2$.

In our experiments, the additional computation for our method in all presented scenes added an overhead of 2% to 7% (see Table 5.2). Knaus and Zwicker [2011] also pointed out that local per-pixel statistics have only minimal impact on performance. The measured performance loss and memory cost of storing the statistics per pixel proved to be dominated by ray tracing costs even with simple scenes.

Scene	Time, sec		MSE, $\times 10^2$	
	PPM	Our method	PPM	Our method
BOX	10.14	10.18	1.91	0.79
CLOCKS	12.13	12.76	6.17	1.76
CORNELL	13.49	13.57	0.032	0.004
ROOM	20.73	21.34	4.61	1.12
TORUS	6.32	6.76	0.073	0.016

Table 5.2: Performance and error of PPM and our method after 5 million emitted photons. The MSE is computed against the reference solution, which is obtained from the original PPM [2009] with large number of photon passes and the global bandwidth smaller than the pixel footprint.

5.6 Results and Discussion

In our implementation we use NVIDIA OptiX [Parker et al. 2010] as a GPU-accelerated ray tracing platform and all measurements were taken on an Nvidia GTX 570 GPU. All result images were rendered at a resolution of 768^2 . In all tests we used a photon map of size $J = 256^2$ (the total number of photons shot per iteration). Table 5.2 shows performance measurements and MSE for our test scenes.

We observe the selected bandwidth being quite noisy in some scenes (e.g., in the BOX and TORUS scenes in Figure 5.5), however, this does not lead to a noisier image. This noise mostly appears in uniformly lit regions and is caused by noisy statistical data. Therefore, the bandwidth might fluctuate around the optimal value, but is increased to capture more photons, which also result in smooth estimates.

Figure 5.6 shows a comparison of the bias in quasi-converged images. We can observe that the initial radius selection results in visible bias in PPM even after 100 billion photons. This is due to the fact that the initial bias vanishes very slowly. Thus the convergence slows down significantly when drawing a large number of samples. Although the bias should be eliminated in the limit, practically this does not happen due to the finite number of rendering iterations and the finite precision of the rendering pipeline.

The close-ups in Figure 5.5 show more noise for PPM compared to APPM. How-

ever, in some regions smaller kernel bandwidths are selected (e.g., close to the caustic edges in the BOX scene) and hence APPM exhibits more high-frequency noise along radiance discontinuities as fewer photons are considered for the estimation. This is a consequence of selecting the bandwidth such that it minimizes the AMSE. Because this is not a perceptual metric, two images with the same error computed with PPM and APPM cannot be considered as equally visually good. Adaptive photon shooting [Hachisuka and Jensen 2011] can augment APPM here to make an image converge more uniformly.

The benefit of APPM for progressive results can also be observed in Figure 5.5. APPM achieves results with significantly lower noise and reduced blur compared to standard PPM. This is because the bandwidth is chosen per pixel such that bias and variance are balanced and thus the MSE is minimized at every iteration. This leads to faster convergence in regions with flat or very smooth lighting where the Laplacian of the pixel measurement is low.

5.7 Limitations and Potential Improvements

One potential limitation of the method is the assumption that there is a single optimal bandwidth for a pixel. The measurement can be represented as a weighted sum of multiple modes (weighted by the importance of the eye subpaths). As a potentially difficult case, imagine some dim caustics on top of some brightly and uniformly lit diffuse surface: the optimal bandwidth would mostly be selected to ensure a good convergence of the diffuse illumination, thus making the caustics blurrier. However, according to the experience of many practitioners of the plug-in method, it can gracefully handle complex and even discontinuous multimodal distributions. Our experience with the adaptive bandwidth selector (based on the plug-in method) is similar and can also be observed from the results, for example, the illumination on the alarm clock glass in Figure 5.5.

Another limitation is that regions with discontinuities might appear noisier than with the original PPM method (see Figure 5.5, Box-scene for APPM with 5 million photons). Even though such regions converge quickly after a moderate number of

photons (Figure 5.5, APPM with 20 million photons), we believe that photons can be shot adaptively, for example, using the work of Hachisuka et al. [2011] in order to provide better convergence; combining both algorithms would be an interesting future work.

The provided APPM derivation is trivial to extend for volumetric radiance estimation [Knaus and Zwicker 2011]. The optimal bandwidth in this case is $r_{N,\text{vol}} \propto O(N^{-1/7})$ and $AMSE_{\text{vol}} \propto O(N^{-4/7})$. For progressive photon beams [Jarosz et al. 2011], the optimal shrinkage rate can be similarly obtained as $r_{N,\text{ppb}} \propto O(N^{-1/5})$ and the convergence rate is $AMSE_{\text{ppb}} \propto O(N^{-4/5})$. Interestingly, the convergence rate for progressive photon beams is faster than the one of progressive photon mapping, which means that the participating media rendering with such transport has faster convergence (given the same number of samples) than the surface rendering with PPM.

Even though the adaptive method can be similarly derived for volumetric phenomena, the *curse of dimensionality* comes into play and influences the adaptivity more than for on-surface estimation. We believe that a thorough analysis of every such method is required, which we leave for the future work.

Potential Improvements. Similar to other PPM methods, we currently assume to have only one eye subpath for a single photon map. An interesting question is to what degree we can reuse a single photon map with different eye subpaths, as dropping the photon map after performing an estimation for a single eye subpath is not always the most efficient strategy. This is especially interesting for scenes with glossy surfaces or out-of-focus regions. Both PPM and APPM can be trivially modified to reuse a photon map for several eye subpaths, however, this ratio should be adaptively estimated by means of statistical methods.

We believe that our method opens a valley for new research in the field of path-space kernel estimation. As future work, it would be interesting to use more elaborate bandwidth selection procedures, for example, the $\mathcal{D} - \mathcal{S} - \mathcal{N}$ scheme [Wand and Jones 1994], which would lead to an anisotropic bandwidth selection with a matrix of bandwidths. Such an improvement can significantly help in the regions

with highly anisotropic signal in image space, such as elongated caustics and heavily motion-blurred objects.

Alternative estimation techniques, such as memory-optimized wavelet estimation or neural networks, are also an interesting direction. The use of perceptual metrics instead of the MSE would be another important and promising step towards more efficient bandwidth selection criterion. Lastly, our estimation technique can also be used for early rendering termination in spirit of Hachisuka et al. [2010], as well as for adaptive sampling of eye and light subpaths.

5.8 Conclusions

In this chapter, we analyzed the optimal convergence rate and consistency conditions for progressive photon mapping techniques. We presented a data-driven adaptive local bandwidth selection method, which requires no crucial user-specified parameters. Moreover, we showed how, compared to previously existing techniques, the convergence can be sped up on finite numbers of photons, and how quasi-converged images of better quality can be obtained.

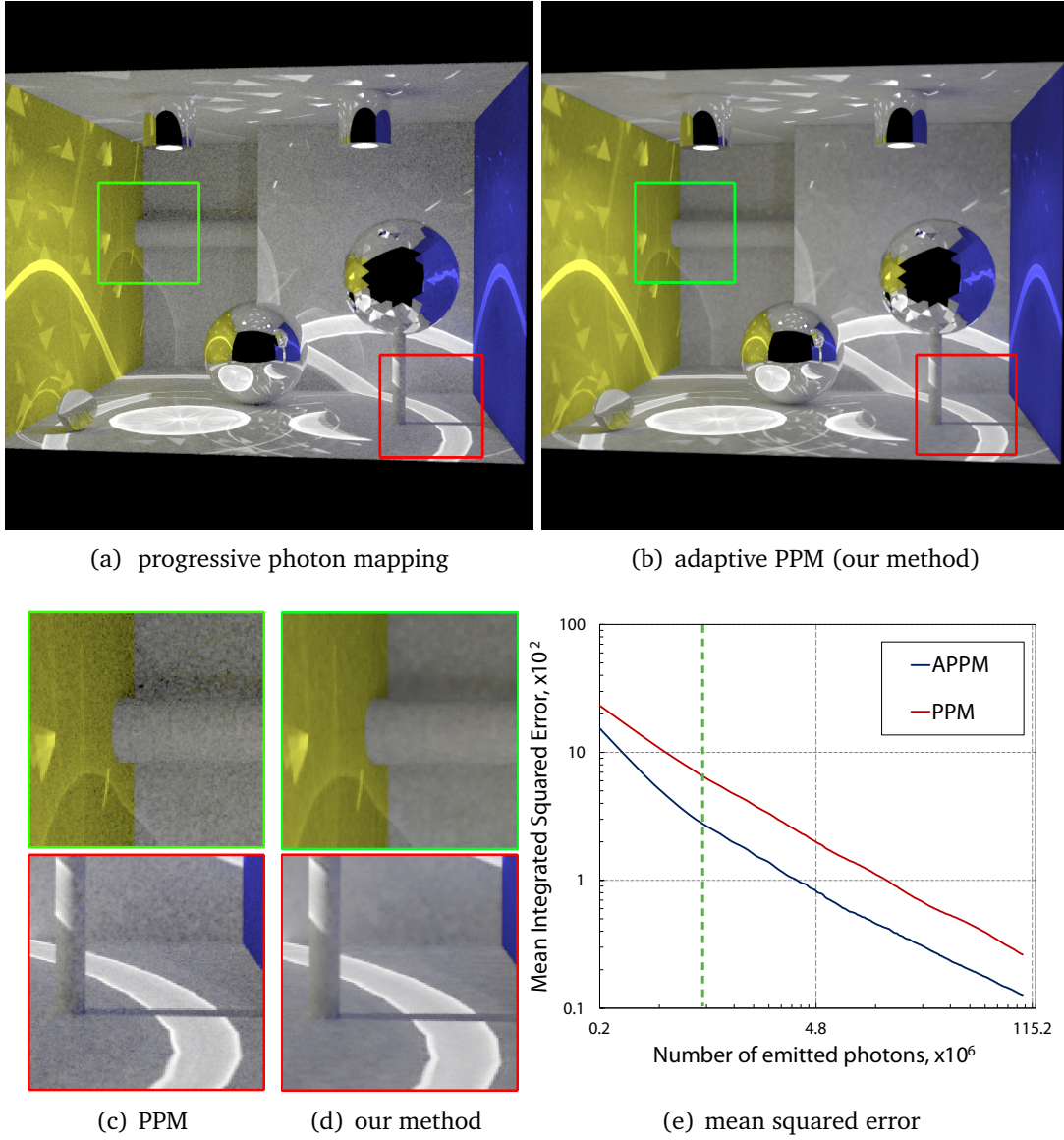


Figure 5.4: The Box scene is challenging for global illumination algorithms because of its specular-diffuse-specular paths. These images show renderings with depth-of-field using progressive photon mapping and adaptive progressive photon mapping (our method) using 20 million photons and 300 iterations. From left to right: (a) PPM with $\alpha = 2/3$ and $k_{\text{NN}} = 10$; (b) adaptive PPM; (c) and (d) show close-ups; (e) plots of the L_2 loss (mean squared error) of (a) and (b). The dashed green line roughly separates two phases: (1) where the bandwidth adapts to the lighting in the scene, and (2) the equalization of the convergence rate.

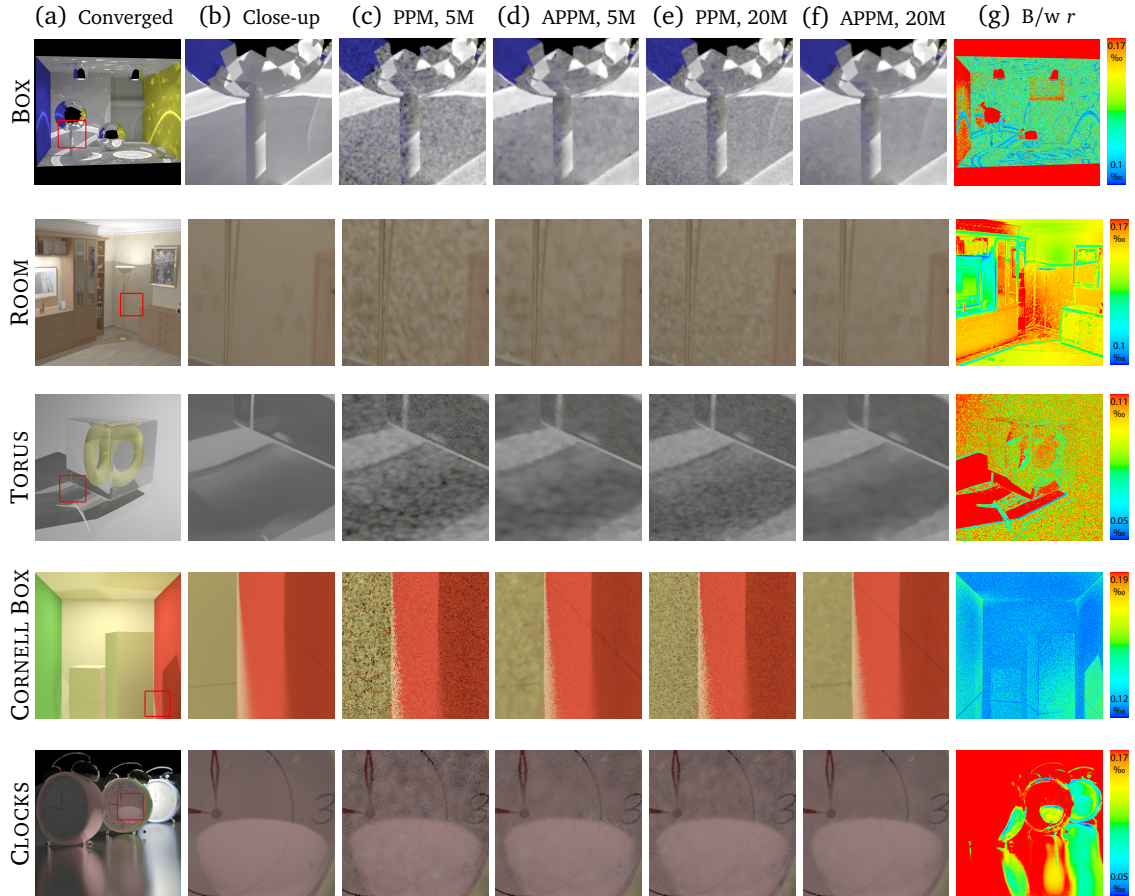


Figure 5.5: We show a comparison of the progressive convergence for different scenes using the original PPM and our adaptive PPM method 5 million and 20 million emitted photons, respectively. The initial bandwidth for PPM has been selected using a k -NN search with $k = 10$ and the parameter $\alpha = \alpha_{\text{opt}} = 2/3$. The last column shows the selected bandwidth using our method. All scenes are illuminated by point light sources and have perfectly reflective or refractive materials which makes it impossible to render them with existing unbiased techniques. The bandwidth range is specified in per mille with respect to the bounding radius of the corresponding scene.

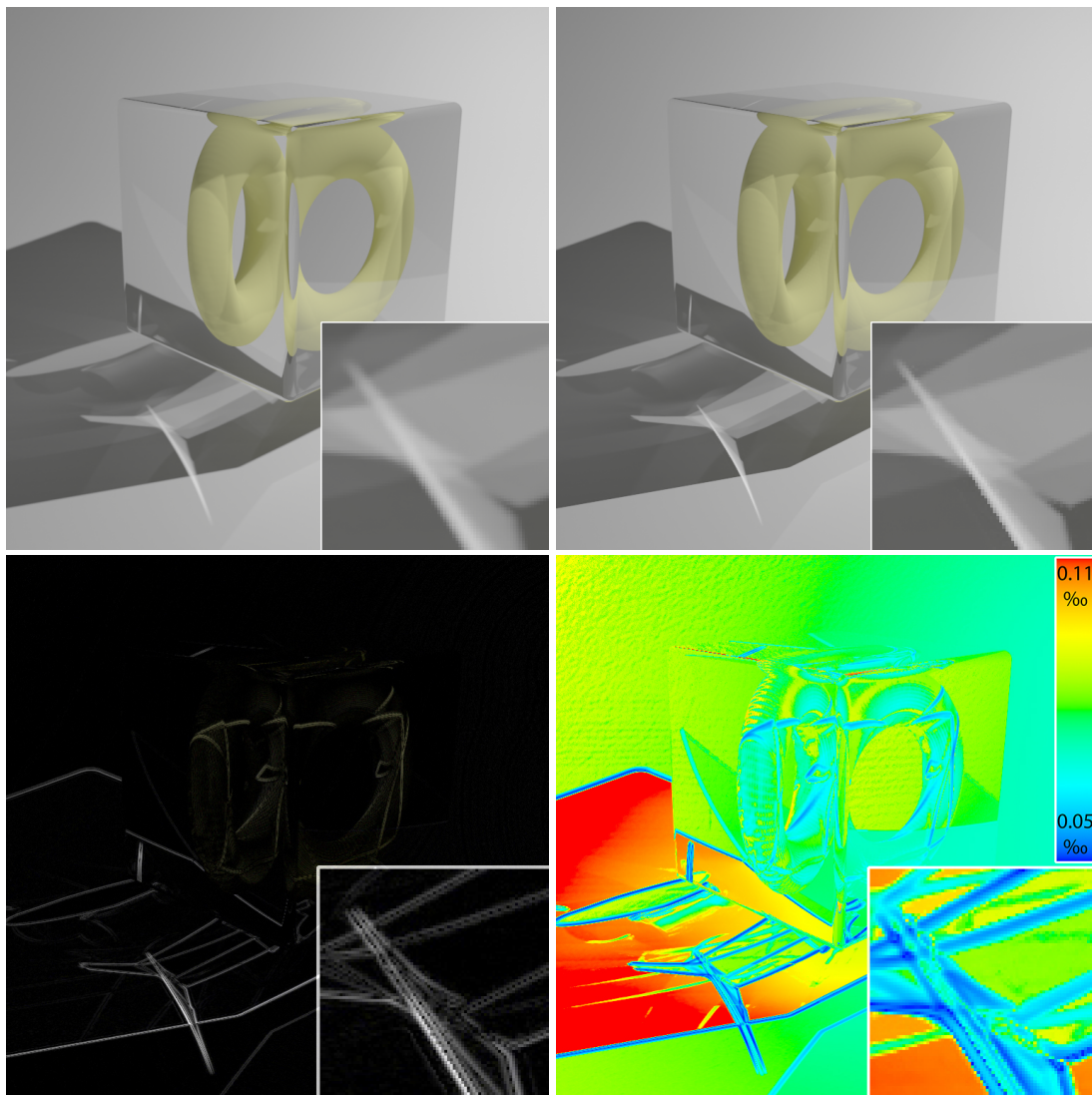


Figure 5.6: The TORUS scene illuminated with a single point light source, rendered with 100 billion emitted photons. For standard PPM (top-left) the initial bandwidth was chosen using the k -NN search with $k = 10$, and $\alpha = 2/3$. Top-right: adaptive PPM; bottom-left: the absolute difference to PPM, scaled by a factor of 20—note that the bias is still clearly visible for standard PPM; bottom-right: the selected bandwidth using APPM.

Regularization in Path Space using Kernel Estimation

It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience.

— ALBERT EINSTEIN

On the Method of Theoretical Physics (Oxford, 10 June 1933)

In this chapter we propose a new regularization framework for handling hard-to-sample paths in light transport in a consistent (biased) way using kernel smoothing, while the remaining transport can still be computed in an unbiased way.

Modern light transport simulation is required to handle challenging scenes, e.g., these including complex materials or lighting from light-emitting diodes (LEDs) used in cars and interior lighting. An emitting surface of a typical LED has a very small area (ranging from a few square millimeters to a few square micrometers), therefore, such luminaires are often treated as point or almost point light sources. The presence of point lights makes the light transport simulation with existing unbiased methods problematic. Moreover, as pointed out by Hachisuka et al. [2008], if a light source is enclosed in a glass fixture, the majority of unbiased methods fail to simulate the light transport carried by such paths in presence of mirrors. These problems stem from singularities that appear in the integrand of the light transport path integral.

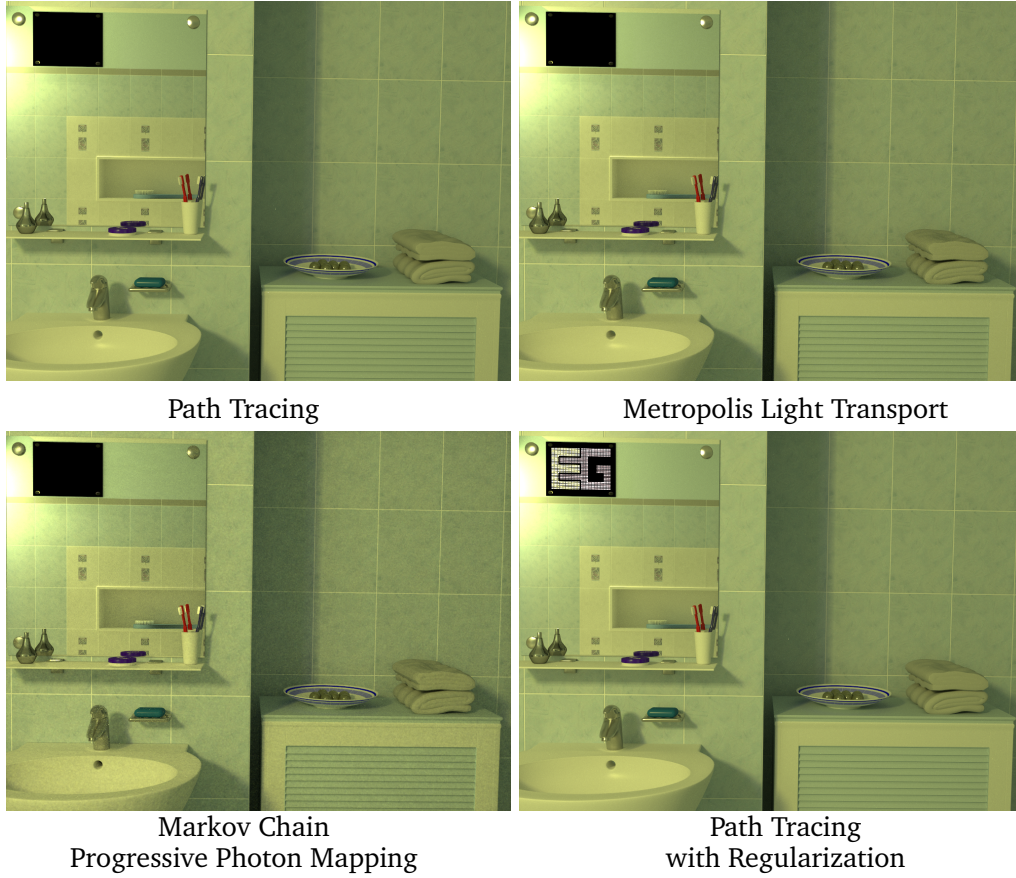


Figure 6.1: Equal-time rendering (2 hours) of the BATHROOM scene with a pinhole camera and indoor illumination from an LED panel. The LED panel showing the EG logo is modeled of 4194 point light sources (not covered by a fixture). The reflection of these light sources in the mirror is missing with previous methods as neither path tracing, nor Metropolis light transport, nor progressive photon mapping can handle all light transport paths. With our regularization, applicable to all unbiased methods, even path tracing can sample such difficult paths in a consistent way – without introducing bias to the rest of the image.

The core idea of our method is to identify paths which cannot be sampled because of singularities of the integrand during path construction. We then use regularization at a minimum number of path vertices to turn it into a samplable path with as little bias as possible.

In particular, in this chapter we introduce:

- A regularization framework that is easy to integrate into existing renderers;
- Robust light transport simulation by selective regularization for hard paths;
- Consistent estimation for paths not sampled using unbiased methods.

6.1 Background

6.1.1 Full-Path Regular Expressions

We use Heckbert's notation [Heckbert 1990] for light paths, where a path is classified by the types of interactions (D for diffuse and S for specular) at every vertex from the light source to the sensor. Classes of interactions are described using regular expression notation, e.g., $(D)^*S$ describes a path with zero or more diffuse interactions, ended with a vertex at specular interaction. Hereafter we denote *any* of D or S types with the letter $A \equiv (S|D)$. We also use an extension [Veach 1998] to describe the properties of light sources and sensors: a light path begins with $L(S|D)(S|D)$, where the first letter after L (indicating the path origin on the light source) denotes a finite-area source D or source S with zero area, and the second letter denotes emission over a finite solid angle D or a set of directions with measure zero S . Analogously, the path ends with $(S|D)(S|D)E$ where the first letter denotes the directional sensitivity distribution of the sensor and the second whether the sensor has a finite area or is a point. Letters between this prefix and suffix correspond to the inner path vertices as in Heckbert's original notation. We use highlighting to more easily distinguish the vertices corresponding to light sources (**green**), interactions (**black**), and camera (**brown**). For example, $LSD(S)^+DDSE$ describes paths producing caustics cast by a point light source, going through one or more mirrors to a pinhole camera.

6.1.2 Paths that Cannot Be Sampled

In his thesis, Veach defines the problem of samplable paths with the following theorem [Veach 1998, Theorem 8.2]:

Theorem 1. *Let \mathbf{X} be a path generated by a local path construction algorithm for which the measurement contribution function is non-zero. Then a path \mathbf{X} necessarily has the form LA^*DDA^*E , i.e., it must contain the substring DD . Furthermore, it is possible to generate any path of this form using local path construction strategies.*

In other words, the probability of sampling paths different from Theorem 1 is zero for unbiased methods with local path construction, including path tracing (PT) and bidirectional path tracing (BPT), which are also required to initialize Metropolis light transport (MLT). Note that path tracing can only sample a subset of the paths which are possible with BPT, e.g., $LS\textcolor{green}{D}SDD^+\textcolor{red}{SDE}$ can only be sampled with a complete set of bidirectional estimators. The mutation strategies for MLT proposed by Veach [1998] are also based on local path construction, therefore, obey this theorem. The recent manifold exploration mutation strategy [Jakob and Marschner 2012] can theoretically explore such non-samplable paths. However, the algorithm cannot find such paths on its own, i.e., finding these paths remains an unsolved problem (see Figure 6.6 for example).

In contrast, consistent methods, such as photon mapping and progressive photon mapping [Hachisuka and Jensen 2009, Hachisuka et al. 2008], relax the condition of the Theorem 1, enabling the sampling of complex paths of the form $L\textcolor{green}{AAA}^*D\textcolor{red}{AAE}$. This is achieved by generating light subpaths $L\textcolor{green}{AAA}^*D$ and camera subpaths $D\textcolor{red}{AAE}$ independently, and then merging the D -vertices using kernel estimation based on the spatial proximity condition. We will show that is equivalent to regularizing one specular connection vertex. However, these methods apply regularization for each path they sample and yet cannot construct all potential paths, e.g., these of a form $L\textcolor{green}{SA}(S)^+\textcolor{red}{ASE}$, which correspond to a path going from a point light source to a pinhole camera through a chain of pure specular interaction (see Figure 6.1).

6.1.3 Mollification

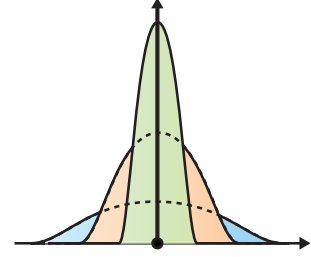
In this section we discuss the approximation of delta distributions by integrable functions. This process will be used to enable the sampling of difficult parts of the path space. The Dirac delta distribution, which introduces the singularities, is zero everywhere except at $x = 0$, where $\delta(0) = +\infty$, and obeys $\int_{\mathbb{R}} f(x)\delta(x)dx = f(0)$ for any absolutely integrable function $f \in L_1$. We distinguish between delta functions in spatial and angular domain by the argument. Typically, in order to approximate a d -dimensional delta distribution δ by integrable functions, one constructs

a sequence of smooth positive functions $\{\varphi_r\} \subset L_1$, such that

$$\left. \begin{array}{l} \|\text{supp}(\varphi_r)\| \propto O(r^d) \\ \int_{\mathbb{R}^d} \varphi_r(\mathbf{x}) d\mathbf{x} = 1 \end{array} \right\} \Rightarrow \lim_{r \rightarrow 0} \varphi_r(\mathbf{x}) = \delta(\mathbf{x}), \quad (6.1)$$

where $\|\text{supp}(\varphi_r)\|$ is a mass of the function support. In other words, a normalized sequence of smooth functions, whose supports are vanishing as $O(r^d)$, implies that it converges to a d -dimensional delta function. The sequence $\{\varphi_r\}$ is called a *mollifier* of the delta function and provides a sequence of integrable functions that approximates $\delta(\mathbf{x})$ in the limit.

It is usually expressed in a form of $\varphi_r(\mathbf{x}) = 1/r^d \varphi(\mathbf{x}/r)$, where $\varphi(\mathbf{x})$ is a smooth normalized canonic kernel also used in kernel estimation. The parameter r is called the *mollification bandwidth* (which is a kernel bandwidth) and has to be gradually reduced to zero in the limit. We discuss the reduction rate for the bandwidth for different integration methods in Section 6.3. Hegland and Anderssen [1996] provide further information on mollifiers and their properties.



An example of mollification for a specular reflection is to replace the delta distribution in the specular BSDF by a function with finite support (see Section 6.2.2).

6.2 Path Space Regularization

In this section, we introduce our concept of selective regularization of the path space, which enables a consistent sampling of arbitrary paths with local path construction methods. The core idea is to turn a path which cannot be sampled into a path of the type from Theorem 1 by selectively mollifying one or more interactions, i.e., by turning them from S into D . There are two major reasons for S vertices to occur in a path: (1) irregular sensors and light sources; and (2) pure specular materials causing delta responses in the BSDF.

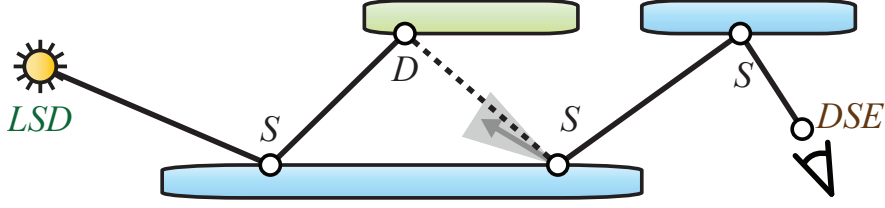


Figure 6.2: A path that cannot be sampled in an unbiased way. Note that a regularization of either of the S -vertices on the lower blue surface (here illustrated for the right vertex) enables us to sample the path (see Section 6.2).

6.2.1 Singularities in the Integrand of the Path Integral

Light transport can exhibit very narrow modes and singularities due to delta, or near-delta, distributions in the integrand of the path integral in Eq. 2.10. Some singularities cannot be explicitly sampled with unbiased integrators (e.g., the double-reflection in Figure 6.2) and thus some parts of the light transport are not accounted for in the global illumination solution. In the following we discuss when such modes and singularities appear during the path construction process.

Singular BSDFs. Singularities in a BSDF can occur at a smooth or near-smooth surface. One example is a perfect specular reflection defined in Eq. 2.7 as

$$f_s(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k \rightarrow \mathbf{x}_{k+1}) = \delta(\mathbf{o} - \mathbf{o}_i) / \langle \mathbf{o}_i \cdot \mathbf{n} \rangle. \quad (6.2)$$

The BSDF of a specular refraction can be defined analogously by taking into account Snell's law and the indices of refraction.

Singularities in Sensors and Emitters. In addition to material models, some other models used in computer graphics also cause singularities in the path integrand. For example, a pinhole camera and a point light source have zero area, while directional light sources or orthographic cameras have a discrete direction. Configurations of such singularities lead to light paths that do not fulfill the Theorem 1, therefore, cannot be sampled using local path construction methods. A

pinhole camera and a point light source models can be correspondingly defined as:

$$\begin{aligned} W(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k) &= \delta(\mathbf{x}_k - \mathbf{x}_{camera})W_0(\mathbf{o}); \\ L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1) &= \delta(\mathbf{x}_0 - \mathbf{x}_{light})L_0(\mathbf{o}), \end{aligned}$$

where $W(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k)$ is the sensor response function of a pinhole camera; $L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1)$ is an emitted radiance distribution for the point light source; and $W_0(\mathbf{o})$ and $L_0(\mathbf{o})$ are the angular sensitivity and radiant intensity correspondingly. Both models have a spatial delta distribution that prevents the path construction methods from implicitly hitting these elements, because the probability of hitting a point (area of a point light or pinhole of a camera) by randomly shooting rays is zero. Another example is a thin laser beam that can be modeled as both spatial and angular delta distributions.

6.2.2 Example: Regularization of Specular Singularities

If a path contains a delta distribution in the BSDF, as in Eq. 6.2, a path construction method, such as next event estimation in path tracing cannot connect such specular surface to a light source, therefore, prohibiting the construction. However, we can regularize such specular interaction by applying mollification in the angular domain, and therefore, enable the path construction. The simplest mollifier that meets the conditions of Eq. 6.1 for $\delta(\mathbf{o} - \mathbf{o}_i)$ is a constant angular mollifier

$$\varphi_\epsilon(\mathbf{o}, \mathbf{o}') = \frac{1}{2\pi(1 - \cos \epsilon)} \mathbb{1}_{\angle(\mathbf{o}, \mathbf{o}') < \epsilon},$$

where $\mathbb{1}$ is an indicator function, i.e., it takes values of one for the angle between \mathbf{o} and \mathbf{o}' is less than ϵ and zero elsewhere. Since the mollifier has to be a normalized kernel, the normalization in the solid angle domain is achieved using the factor $1/(2\pi(1 - \cos \epsilon))$. We used a constant kernel with the indicator function as a canonical example. Normalized smooth kernels with a vanishing first moment are more preferred for improved convergence.

The mollification angle ϵ is a bandwidth parameter of the mollifier. Instead of

letting the user to manually choose the value for initial bandwidth, we select the bandwidth similar to photon mapping. We compute ϵ based on the distance l between the two path vertices on the path edge with mollification and a user-specified spatial radius r as $\epsilon = \arctan(r/l)$ (see Figure 6.3, left). Note that for $r \ll 1$ the term $\arctan(r/l) \approx r/l$. This allows us to use r instead of ϵ in the asymptotic convergence conditions in Section 6.3.

6.2.3 Other Types of Regularization

Spatial Regularization. is introduced, e.g., using a spatial spherical mollifier $\varphi_r(\mathbf{x}, \mathbf{x}_0) = \frac{3}{4\pi r^3} \mathbb{1}_{\|\mathbf{x}-\mathbf{x}_0\| < r}$. This mollifier obeys the conditions in Eq. 6.1. Spatial mollification can be useful in participating media rendering, e.g., in spirit of volumetric photon mapping [Jensen and Christensen 1998]. However, as will be described in Section 6.3, the dimensionality of the delta distribution directly affects the convergence of the integration. Therefore, we recommend to avoid spatial mollification if possible, as the spatial delta distribution is 3-dimensional compared to the 2-dimensional angular delta distribution in the specular BSDF.

Near-Singular Interactions. Many of the described non-samplable paths do not exist in the real world, where no perfect point light sources or pinhole cameras exist. However, in the context of path construction, near-singular functions, e.g., highly-glossy BSDFs, are almost as problematic as singularities, and the path construction probability can be arbitrarily close to zero. In general, mollification is applicable to any near-singular function leading to consistent integration.

Regularization of BSDFs depends on their representation: measured BSDFs can be numerically convolved with the mollifier kernel. For the regularization of narrow lobes of analytic BSDFs, the following two scenarios can be applied:

- Importance-sample the BSDF, then connect only if the target connection vertex lies within the mollification cone of the BSDF sample. This is a statistical equivalent of the convolution with the mollifier.

- Extract a dominant lobe of the analytic model (e.g., a percentile in solid angle [Jakob and Marschner 2012]) as a function of the roughness parameters. When regularization is required, this term allows to control the roughness of the model based on the mollification bandwidth.

The decision on when to regularize near-specular paths depends on multiple factors, including performance-accuracy trade-offs, as well as the curvature of the local light field.

Global path construction methods, such as manifold exploration [Jakob and Marschner 2012] and half-vector space mutation that will be introduced in Chapter 8, help exploiting near-singular paths (caused by narrow BSDF lobes and complex illumination). The present regularization framework focuses on an orthogonal problem: finding such *non-samplable* light paths.

6.2.4 Selective Regularization

An ad-hoc regularization approach – to mollify all delta distributions in the integrand – would lead to a large amount of bias in the image rendered with a finite number of samples. Minimize the bias, we apply mollification *selectively* to as few interactions as possible (e.g., to only one *S*-vertex in Figure 6.2). For a selective regularization we require all interactions along the path to be known first. Otherwise, it is not possible to determine whether the path can be constructed in an unbiased way or not (i.e., the fulfillment of Theorem 1).

Example: Regularization within Path Tracing. If the paths are generated using a unidirectional path tracer with next event estimation from Section 4.3.1, we decide about the regularization only at the next event estimation, because the path type is known completely only at this stage. Consequently, the only vertex we can mollify at this stage, without regenerating the path, is either the vertex of the last interaction or the position on the light source (or both if necessary). For a unidirectional path tracer, every path that ends with "S" is non-samplable.

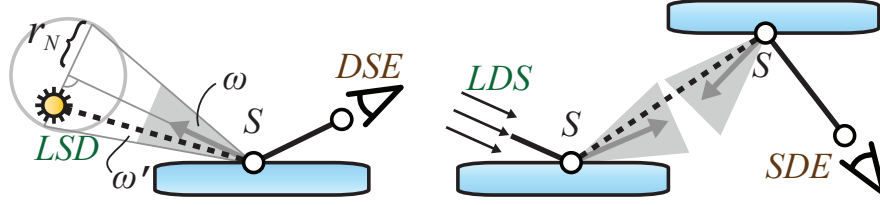


Figure 6.3: Left: regularization with mollification at the next event estimation in unidirectional sampling. Right: regularization at two interactions with bidirectional sampling.

Consider an example where a subpath from a pinhole camera to a specular surface $SDSE$, which is to be connected to a point light source LSD , can be turned into a samplable path by mollifying the last-generated interaction type yielding $LSDDSE$ (Figure 6.3, left). Alternatively, the point on the light source can be also mollified to enable path construction.

6.2.5 Regularization with Bidirectional Path Construction

With bidirectional path construction, the decision can be made only at the last stage of connecting the two subpaths, as the type of the full path is not known before. At this point, we can regularize either of the two connection vertices (or both of them) to turn it into a path of the type in Theorem 1. Figure 6.3 (right) shows an example where the regularization of interactions at both connection vertices is required.

In bidirectional methods, e.g., BPT described in Section 4.3.2, we deal with a set of alternative techniques for constructing the same path. If the resulting path can be constructed, there is no need to apply regularization, therefore, we handle it in an unbiased way as in original BPT, weighting the respective contributions of the bidirectional estimators using multiple importance sampling (MIS) [Veach and Guibas 1994]. However, if the path cannot be sampled in an unbiased way, we regularize it. Similarly, to BPT, this also leads to a set of multiple alternative techniques. This set of techniques for such regularized path always consists solely of biased estimators. We propose one weighting strategy for regularized paths, which can be seen as a flavor of MIS for a set of biased techniques.

Weighting Strategy for Biased Techniques. Consider an example of a path *LSDSDSE* (as in Figure 6.2) which can be turned into a samplable path by mollifying the BSDF at the first or second specular vertex in the middle part, yielding two different techniques E_1 and E_2 . In order to achieve consistent results, we have to choose the weights for combining these techniques, w_1 and w_2 , such that $w_1 + w_2 = 1$, analogously to MIS. However, in contrast to MIS, the goal of the combining strategy should be bias minimization given that both estimators with regularization are biased. Hegland and Anderssen [1996] show that the mollification error (bias) directly depends on the bandwidth. To minimize bias, we propose to use a *sampling* maximum distance heuristic which assigns $w_i = 1$ if the technique E_i corresponds to the path with the connection edge with longest distance l among the techniques with the lowest possible total number of dimensions of the mollifiers at all regularized vertices. Otherwise we assign $w_i = 0$. This is equivalent to a minimization of the mollification bandwidth r .

Discussion. When using regularization for near-singular interactions in Section 6.2.3, we can face the problem of mixing consistent with unbiased estimators of the same path. In this case, it is suitable to use the modification of MIS with the bias considerations as in [Georgiev et al. 2012].

6.3 Regularization with Integration Methods

It is possible to apply regularization to both unidirectional and bidirectional path construction. This enables the regularization framework for all existing unbiased methods, including bidirectional path tracing and Metropolis light transport.

The regularization with mollification is orthogonal to the integration method: the integration method does not require changes, instead the regularization selectively smooths singular modes in the underlying integrand. Therefore, all existing variance reduction techniques, path construction methods, and mutation strategies can employ this type of regularization without any additional changes.

We describe how to gradually reduce the mollification bandwidth with each in-

tegration iteration to achieve consistent integration, where bias vanishes in the limit. We derive the conditions for bandwidth shrinkage rate, i.e., how one should compute the bandwidth r_n for the n -th iteration from the user-specified initial mollification bandwidth r_0 .

6.3.1 Consistent Monte Carlo Integration with Regularization

In order to achieve a consistent estimation of the path integral in Eq. 2.10 with Monte Carlo (MC) integration methods, such as PT and BPT, the mollification bandwidth has to be gradually reduced after each step of the integration. Here we show that the regularization framework, including arbitrary (e.g., spatial) mollifications, is consistent under general conditions.

First, we prove that it is possible to construct a consistent Monte Carlo estimator with the canonical case of a d -dimensional delta distribution $\delta(\cdot)$ located at the origin of the d -dimensional space $\Lambda \cong \mathbb{R}^d$.

Lemma 1. *Given a d -dimensional integrand of a form $f(x) = \delta(x)g(x)$, let $f_r(x) = \varphi_r(x)g(x)$ be a mollification of $f(x)$, where $g(x)$ is Lebesgue-integrable on Λ ; and $\varphi_r(x)$ is a mollifier function with support on a ball $\mathbb{B}_r \subset \Lambda$, such that $||\mathbb{B}_r|| \propto r^d$. Then, given that the random variable X is i.i.d., $X \sim \pi(\Lambda)$, where $\pi(\cdot)$ is some importance sampling distribution of $g(x)$, such that $\forall X_n \in \Lambda \mid g(X_n) > 0 : p(X_n) > 0$, then the expectation of the Monte Carlo estimate $\hat{F}_N = \mathbb{E} \left[\frac{1}{N} \sum_{n=1}^N \frac{f_{r_n}(X_n)}{p(X_n)} \right]$ is*

$$\text{plim}_{N \rightarrow \infty} \left[\hat{F}_N \right] = \int_{\Lambda} f(x) dx = g(0) = F, \quad (6.3)$$

for bandwidth sequences $\{r_n\}$ decreasing within the bounds $\mathcal{O}(n^{-1/d}) < r_n < \mathcal{O}(1)$.

Proof. Our proof is based on the results of Tornberg [2002]. In order to prove Eq. 6.3, we show that the mean squared error (MSE) of the left-side estimate vanishes to zero. We split the MSE into an analytical part $\mathcal{B}[\hat{F}]^2$ and a stochastic part

$\text{Var}[\hat{F}]$:

$$MSE[\hat{F}] = E[\hat{F} - F]^2 = \mathcal{B}[\hat{F}]^2 + \text{Var}[\hat{F}].$$

Analytical Error. This mollification error (bias) $\mathcal{B}[\hat{F}]$ can be written as follows for a given constant bandwidth r

$$\mathcal{B}[\hat{F}_r] = \int_{\Lambda} \varphi_r(x) g(x) dx - \int_{\Lambda} \delta(x) g(x) dx = \int_{\Lambda} \varphi_r(x) g(x) - g(0).$$

We denote the k -th moment of the mollifier function φ_r as $\mathcal{M}_k(\varphi_r) = \int_{\Lambda} x^k \varphi_r(x) dx$. Next, by expanding $g(x)$ into a d -dimensional Taylor series around zero in the first term, we obtain:

$$\begin{aligned} \mathcal{B}[\hat{F}_r] &= \sum_{k=0}^{\infty} \left(g^{(k)}(0) \int_{\Lambda} \frac{x^k}{k!} \varphi_r(x) dx \right) - g(0) \\ &= g(0) \mathcal{M}_0(\varphi_r) + g'(0) \mathcal{M}_1(\varphi_r) + \frac{1}{2} g''(0) \mathcal{M}_2(\varphi_r) + \mathcal{O}(\mathcal{M}_3(\varphi_r)) - g(0) \\ &= \frac{1}{2} g''(0) \mathcal{M}_2(\varphi_r) + \mathcal{O}(\mathcal{M}_4(\varphi_r)), \end{aligned}$$

where the multidimensional derivative $g^{(k)}$ is expressed as $g^{(k)}(x) = \nabla^k g(x)$. The last equality is due to the vanishing odd moments of φ_r and the normalization property of the mollifier.

Next, we insert the mollifier of a form $\varphi(x) = r^{-d} \varphi_0(x/r)$:

$$\begin{aligned} \mathcal{B}[\hat{F}_r] &= \frac{1}{2} g''(0) \int_{\mathbb{B}_r} x^2 r^{-d} \varphi_0(x/r) dx + \mathcal{O} \left(\int_{\mathbb{B}_r} x^4 \varphi_r(x) dx \right) \\ &= \frac{1}{2} g''(0) r^2 \mathcal{M}_2(\varphi_0) + \mathcal{O}(r^4), \end{aligned}$$

where the constant $\mathcal{M}_2(\varphi_0) = \int_{\Lambda} x^2 \varphi_0(x) dx$ is the second moment of the initial mollifier function φ_0 .

Now we consider the upper bound of the collected average error \mathcal{B}_N during Monte

Carlo simulation at step N , where r_n is varying at each step:

$$\mathcal{B}[\hat{F}]_N = \frac{1}{N} \sum_{n=1}^N \mathcal{B}[\hat{F}_{r_n}] = \frac{1}{N} \sum_{n=1}^N \left(\frac{1}{2} g''(0) r_n^2 \mathcal{M}_2(\varphi_0) + \mathcal{O}(r_n^4) \right).$$

We need to enforce $\mathcal{B}[\hat{F}]_N \rightarrow 0$ as $N \rightarrow \infty$ for a consistent estimation. Thus we have the first boundary condition for the bandwidth shrinkage rate:

$$r_n < \mathcal{O}(n^0) = \mathcal{O}(1). \quad (6.4)$$

Stochastic Error. Here we make sure that the stochastic term of the MSE – the variance of the Monte Carlo integration – also vanishes. The variance of the estimator \hat{F} for a single sample with mollification bandwidth r is

$$\begin{aligned} \text{Var}[\hat{F}_r] &= \mathbb{E}[\hat{F}_r^2] - \mathbb{E}[\hat{F}_r]^2 = \int_{\Lambda} r^{-2d} \varphi_0(x/r)^2 g(x) dx - \left(g(0) + \mathcal{B}[\hat{F}_r] \right)^2 \\ &= \int_{\Lambda} r^{-2d} \varphi_0(x/r)^2 \sum_{k=0}^{\infty} \frac{x^k}{k!} g^{(k)}(0) dx - \left(g(0) + \mathcal{B}[\hat{F}_r] \right)^2 \\ &= r^{-d} \left(\int_{\Lambda} \varphi_0(y)^2 dy \right) g(0)^2 + \mathcal{O}(1). \end{aligned}$$

The equation in the last line is obtained after the change of variable $y = x/r$. Also according to Eq. 6.4 the bandwidth r is asymptotically $r \ll 1$. We then express the average variance of the Monte Carlo integration as

$$\text{Var}[\hat{F}]_N = \frac{1}{N^2} \sum_{n=1}^N \text{Var}[\hat{F}_{r_n}] = g(0) \left(\int_{\Lambda} \varphi_0(x)^2 dx \right) \frac{1}{N^2} \sum_{n=1}^N r_n^{-d} + \frac{1}{N^2} \sum_{n=1}^N \mathcal{O}(1).$$

The first equality is due to the assumption that all samples are i.i.d. Again, we enforce $\text{Var}[\hat{F}]_N \rightarrow 0$ as $N \rightarrow \infty$ in order to eventually achieve consistency. This yields the second boundary condition for the bandwidth shrinkage rate

$$r_n > \mathcal{O}(n^{1/d}).$$

□

Theorem 2. *Given the unified space of all light transport paths Ω , the Monte Carlo estimation of the path integral as in Eq. 2.10 with a measurement function $f(\mathbf{X})$ as an integrand, converges consistently under the conditions of Lemma 1 with selective regularization of $f_r(\mathbf{X})$, i.e.,*

$$\text{plim}_{N \rightarrow \infty} [\hat{I}_N] = \mathbb{E} \left[\frac{1}{N} \sum_{n=1}^N \frac{f_{r_n}(\mathbf{X}_n)}{p(\mathbf{X}_n)} \right] = I,$$

only if the sequence of mollification bandwidths $\{r_n\}$ decreases within the bounds $\mathcal{O}(n^{-1/d}) < r_n < \mathcal{O}(1)$.

Proof. The integrand $f(\mathbf{X})$ can be uniquely separated into two integrands: an integrand which consists solely of samplable (regular) paths $g(\mathbf{X})$ and an integrand consisting only of non-samplable (irregular) paths $h(\mathbf{X})$ (due to the determinism of the path classification). Then the path integral can also be decomposed into a regular and an irregular part $I = I^0 + I^1 = \int_{\Omega} g(\mathbf{X}) d\mu(\mathbf{X}) + \int_{\Omega} h(\bar{x}) d\mu(\mathbf{X})$, where f_r is a selectively regularized measurement function. The conventional Monte Carlo proof passes for the regular integral $\int_{\Omega} g(\mathbf{X}) d\mu(\mathbf{X})$. Thus we need to prove that the Monte Carlo estimate of the last integral converges to I^1 .

Given an i.i.d. random variable on Ω , and the fact that the space Ω is translation invariant, we apply Lemma 1 to the Monte Carlo estimator of the regularized integral $\hat{I}_r^1 = \int_{\Omega} h_r(\mathbf{X}) d\mu(\mathbf{X})$, yielding

$$\text{plim}_{N \rightarrow \infty} [\hat{I}_N^1] = I^1.$$

This result implies the bounds for the mollification bandwidth shrinkage rate from Lemma 1. □

Recall that one or both interactions might require regularization. For the latter case, the two delta distributions can be combined into one using a product measure. In this case the dimensionality of the new-formed delta distribution is the sum of dimensions of the two original ones.

Therefore, if the mollification is applied to a d -dimensional delta distribution, the

reduction rate has to be in the following boundaries:

$$O\left(n^{-1/d}\right) < r_n < O(1), \quad (6.5)$$

where n is the index of the sample taken for the integration. One simple sequence is $r_n = r_0 n^{-\lambda}$, where r_0 is the user-specified initial mollification radius and $\lambda \in (0; 1/d)$ for a d -dimensional mollification. This rate for a 2D mollification matches the one we have seen in Chapter 5 for PPM for $r_n = r_0 n^{(\alpha-1)/2}$ with $\alpha \in (0; 1)$. Similarly to selecting the optimal α parameter for PPM, the optimal parameter $\lambda = 1/6$ for a single-vertex specular mollification and $\lambda = 1/12$ for two vertices. This mollification parameter λ is defined globally and the bandwidth shrinking causes the regularized parts of the integrand to become sharper with every integration step, yielding a consistent estimation in the limit.

Note that a simple path tracer, augmented with regularization of specular interactions, is already able to handle all light transport paths consistently. However, both PT and BPT can be inefficient at sampling regularized paths compared to the Markov chain Monte Carlo methods.

6.3.2 Consistent Markov Chain Monte Carlo Methods with Regularization

Markov chain Monte Carlo (MCMC) methods, such as Metropolis light transport (MLT), can improve on the convergence of MC methods: once a rare path is found, it is exploited thoroughly by the Markov process. Regularization can help finding such rare paths in the first place. Moreover, regularization used with MCMC methods enables a consistent estimation of paths that cannot be found otherwise.

We start by generating initial paths using unbiased local path construction (BPT in our implementation). While mutating a new path, we apply the selective regularization to non-samplable paths by mollifying the interactions, as in Section 6.2, and using the decreasing mollification bandwidth r_n at mutation n .

By applying regularization to MLT, we essentially combine a simulated annealing

optimization method with MCMC integration. This way, singularities and sharp peaks of the integrand are regularized (smoothed out) in the beginning of the integration such that a Markov chain can easily find them. In order to achieve consistency, the regularization has to vanish throughout the integration. An important condition is to “sharpen” these peaks *slower* than the integrand converges. Otherwise, a Markov chain might get stuck in a sharp peak for a long time, jeopardizing the convergence. Note that with regularization, a Markov chain can find hard paths during the integration even if it is not initialized with any of them, i.e., there is no need to initialize it with non-samplable paths.

The theorem states that given an ergodic set of mutation strategies, selective regularization leads to a consistent estimation of the integral in Eq. 2.10.

Theorem 3. *Given a space of all light transport paths Ω as a state space, and a Harris-recurrent transition kernel $K(\cdot, \cdot)$ on it, Markov chain Monte Carlo estimation of an integral in Eq. 2.10 with selective regularization of the integrand $f_r(\mathbf{X})$ (as in Section 6.2), converges almost surely if the bandwidth shrinkage rate r_n is such that $O(\gamma^n) \leq r_n < O(1)$ for some $\gamma \in (0; 1)$.*

Proof. Similarly to the proof of Theorem 2, we decompose Ω into a subspace of regular paths Ω_{reg} and a subspace of irregular (non-samplable) paths Ω_{irreg} .

The outline of our proof is as follows: we first use the lower bound of the original Harris-recurrent kernel $K(\cdot, \cdot)$ (initially defined for regular paths only) and then show that by adding the regularization for the ill-posed paths, the Markov chain, which runs on both regular and (regularized) irregular paths, becomes weakly ergodic.

Since the original kernel K is Harris-recurrent (see Section 3.3.2), it passes the original Metropolis proof [Metropolis et al. 1953] and there exists a lower bound $0 < \kappa < 1$ for it, such that for all regular paths on Ω_{reg} the following holds [Bremaud 1999]:

$$\delta(K^n) \leq \mathcal{L}\kappa^n,$$

for some positive finite constant $0 < \mathcal{L} < \infty$.

The Markov chain running on regularized paths is similar to the simulated annealing process: the target distribution is slowly “cooled down” throughout the integration. Therefore, following Brémaud [1999], we first find a lower bound of the regularized kernel K'_{r_n} for the step n . For that, we first assume that the mollification happens in $0 < d < \infty$ dimensions. By adding the mollification, we inflate the transition kernel also to be non-zero on the target subspace of all irregular paths Ω_{irreg} and its regularized neighborhood. It is important to notice that the set of regularized paths, which represents one irregular subpath, should have a smoothly vanishing mass due to the gradual reduction of the mollification bandwidth (and, correspondingly, the support of the regularized set).

Then we recall the property of a d -dimensional mollifier: there is a constant $0 < \mathcal{M}_\varphi < \infty$ (usually equal one) that $\varphi_0 < \mathcal{M}_\varphi$ everywhere, and thus $\varphi_n < \mathcal{M}_\varphi r_n^{-d}$. The transition probability for escaping such set of (regularized) irregular paths is proportional to contribution of regularized paths, which grows proportionally to the lower bound of the mollifier. Using this property, along with the lower bound κ for the non-mollified part of the original transition kernel (we use the regular mutations from the original kernel K for non-mollified dimensions), we obtain a lower bound for the Dobrushin’s ergodicity coefficient for the transition kernel K'_{r_n} at move n with bandwidth r_n (see e.g. [Brémaud 1999, Chapter 6] for a more detailed proof of the simulated annealing):

$$\delta(K'^n_{r_n}) \leq \delta(K^n) \mathcal{M}_\varphi \mathcal{L} r_n^{-d} \leq \mathcal{M}_\varphi \mathcal{L} \kappa^n r_n^{-d}. \quad (6.6)$$

See also Mitra et al. [1986] for similar bounds on simulated annealing with slowly mixing non-homogeneous Markov chains.

Now we can use Eq. 3.10 (Dobrushin’s inequality for non-homogeneous Markov chains)

$$\prod_{n=1}^{\infty} \delta(K'^n_{r_n}) \rightarrow 0,$$

which guarantees the weak ergodicity of the constructed Markov chain.

By inserting the lower bound from Eq. 6.6 into this equation

$$\prod_{n=1}^{\infty} \delta(K'_{r_n}) \leq \prod_{n=1}^{\infty} \left(\mathcal{M}_{\varphi} \mathcal{L} \kappa^n r_n^{-d} \right) \rightarrow 0,$$

we obtain the final bounds on the shrinkage rate of the mollification bandwidth r_n :

$$\mathcal{O}(\gamma^n) \leq r_n < \mathcal{O}(1),$$

where $\kappa \leq \gamma < 1$ should be greater or equal to the spectral gap κ of the original transition kernel $K(\cdot, \cdot)$. \square

Conditions of this theorem assert that the mutation strategy should not only cover the whole path space, but the probability of returning to the same path has to be non-zero. The existing sets of perturbation-based path generators, including Kelemen's and Veach's mutation sets, comply.

Therefore, the target distribution (tempered by the shrinkage rate r_n) should change slower than the Markov chain mixing rate. The shrinkage rate of the mollification bandwidth for a single Markov chain

$$\mathcal{O}(\gamma^n) \leq r_n < \mathcal{O}(1), \tag{6.7}$$

ensures that the change in the target distribution for mutation n is small enough. We use an ad-hoc sequence $r_n = r_0 \gamma^n$, where the constant r_0 is selected in the same manner as for MC integration, preferably large enough to allow the Markov chain to easily reach any of the mollified paths from any state during the initial steps. The parameter γ is problem-dependent and has a different meaning compared to Monte Carlo integration, that is, $\gamma \in (0; 1)$ should be greater than the spectral gap of the transition kernel $K(\cdot, \cdot)$. In practice, it should be close enough to 1 to achieve convergence in finite time.

Normalization Constant. The initialization of MCMC methods is done with unbiased methods described in Section 3.3.3 to estimate the image normalization constant correctly. Veach's original normalization [Veach 1998] also works with

regularization, as it is based on a scaling ratio to the alternative sampling method. However, there are cases when regularized features dominate the resulting image, therefore, it is hard or even impossible to determine the normalization constant solely based on paths that can be sampled in an unbiased way. In order to avoid such cases, we initialize MLT with a selectively regularized integrand (using regularized BPT). We then refine the normalization constant sparsely, e.g., once per several hundred mutations, by sampling additional paths using regularized BPT. During this process, both mollification bandwidths for MLT and BPT are shrunk as described in Section 6.3.1 and Section 6.3.2, correspondingly. This way the normalization constant is estimated in a consistent way and provides a robust estimate for average image intensity (see Figure 6.6).

6.3.2.1 Discussion on the Convergence of the Regularized Markov Chain Monte Carlo Methods

Selecting the regularization parameters. One practical sequence of bandwidths would be $r_n = r_0 \gamma^n$. There are no theoretical restrictions on the selection of the initial bandwidth r_0 . However, from practical considerations, we recommend to select it such that the set of mutation strategies can easily sample the subset of the path space, which is covered by the smallest regularized feature in the image. According to our experiments, this parameter can be determined using the same heuristic as the one used for computing the initial global radius in progressive photon mapping (PPM) [Hachisuka et al. 2008].

Note that the parameter $\gamma \in (0; 1)$ has a different meaning comparing to the Monte Carlo integration. This parameter ideally should be selected close enough to the spectral gap of the transition kernel (the distance between its two largest eigenvalues), which denotes the base of the geometric convergence rate of the probability measure $K^n(x_0, \cdot)$ with any valid initial state x_0 to the target distribution in total variation [Bremaud 1999]. It has to be chosen close enough to one to guarantee practical convergence. Intuitively, this means that depending on the mutation strategies used, the parameter γ should be selected such that the regular part of the integral (the part of the original MLT method) converges faster than the

singularities from ill-posed paths start to appear in the integrand. In case if this value is selected wrongly (smaller than the spectral gap), the practical mixing of the chain is dominated by the regularized features and eventually lead to a wrong image where regularized features would be oversampled while the rest of the image would remain undersampled.

Requirements to the MLT mutation strategies. The proof of Theorem 3 is based on the assumption that the set of mutation strategies used in some particular implementation of Metropolis light transport meets the practical conditions for Harris recurrence of the transition kernel. In case of Metropolis-Hastings sampling used in MLT this means that there should be a non-zero probability (bounded with some positive constant from below) that the chain will return into any current state it can ever land into.

For the original MLT [Veach 1998] this is practically achieved by using the bidirectional mutation strategy, which can regenerate the path from scratch and thus guarantees that the visited path can be sampled again.

In case of Kelemen mutations in the hypercube of random numbers [Kelemen et al. 2002], this condition is satisfied by the so-called “large step” mutation, which also regenerates a complete path from scratch, providing the positive probability of returning into any state.

6.4 Implementation Details

We use the following rendering methods for comparisons: standard path tracing with next event estimation from Section 4.3.1 (denoted in the figures as PT); bidirectional path tracing (denoted as BPT) with multiple importance sampling (power heuristic, $\beta = 2$) from Section 4.3.2; Metropolis light transport (denoted as MLT) from Section 3.3.3 with the original set of mutations from Section 4.6.1 as well as with simple mutations in the unit hypercube and bidirectional connections from Section 4.6.3; stochastic progressive photon mapping (denoted as PPM) [Hachisuka and Jensen 2009, Hachisuka et al. 2008]; and its variant with Metropolis photon

Algorithm 2 Selective angular mollification of a single specular interaction for bidirectional path tracing; reflection direction \mathbf{o} and direction to vertex \mathbf{o}' .

```

if current vertex is "S" then
  if path contains no "DD" then
     $r_n \leftarrow r_0 \cdot n^{-\lambda}$                                  $\triangleright$  Shrink bandwidth with iterations
     $\epsilon_n \leftarrow \arctan \frac{r_n}{\text{connection\_distance}}$          $\triangleright$  Compute mollification angle
    return  $\varphi_{\epsilon_n}(\mathbf{o}, \mathbf{o}') / \langle \mathbf{o} \cdot \mathbf{n} \rangle$            $\triangleright$  Evaluate mollifier
  else return 0                                               $\triangleright$  Mollification is not required

```

tracing (denoted as MCPPM) [Hachisuka and Jensen 2011]; and vertex connection and merging (denoted as VCM) [Georgiev et al. 2012]. Our method is orthogonal to the sampling strategy, i.e., one can use the original set of mutations [Veach and Guibas 1997] as well as the manifold exploration mutation described in Section 4.6.2.

The renderer is implemented using NVIDIA OptiX ray tracing platform and all results are measured using a GeForce 580 GTX GPU. Unless stated otherwise, all images are rendered at 1024×768 resolution.

In Algorithm 2 we provide a simple listing for Monte Carlo integration, which can replace the connection routine for the specular material in an existing unbiased renderer. The same pseudocode can be reused for MCMC integration with the shrinkage rate of r_n from Eq. 6.7. Special attention should be paid to the machine precision during the integration. Since the mollification angle can become very narrow after many iterations, it can run out of floating point precision. We suggest to avoid dividing by the connection distance in Algorithm 2, instead work on a scaled sphere that contains the target connection vertex.

Parameters. Both r_0 and λ (for MC integration) / γ (for MCMC integration) are user-defined parameters; r_0 controls the mollification blurring in spatial units. In the case of MC integration λ controls the shrinkage rate and is equivalent to this in PPM from Section 5.3.2. For MCMC, however, γ should be greater than the second largest eigenvalue of the transition kernel, the mixing rate of the Markov chain. It depends on the scene and cannot be known in advance. It is practically hard to impossible to obtain the spectral gap from a given transition kernel. Therefore,

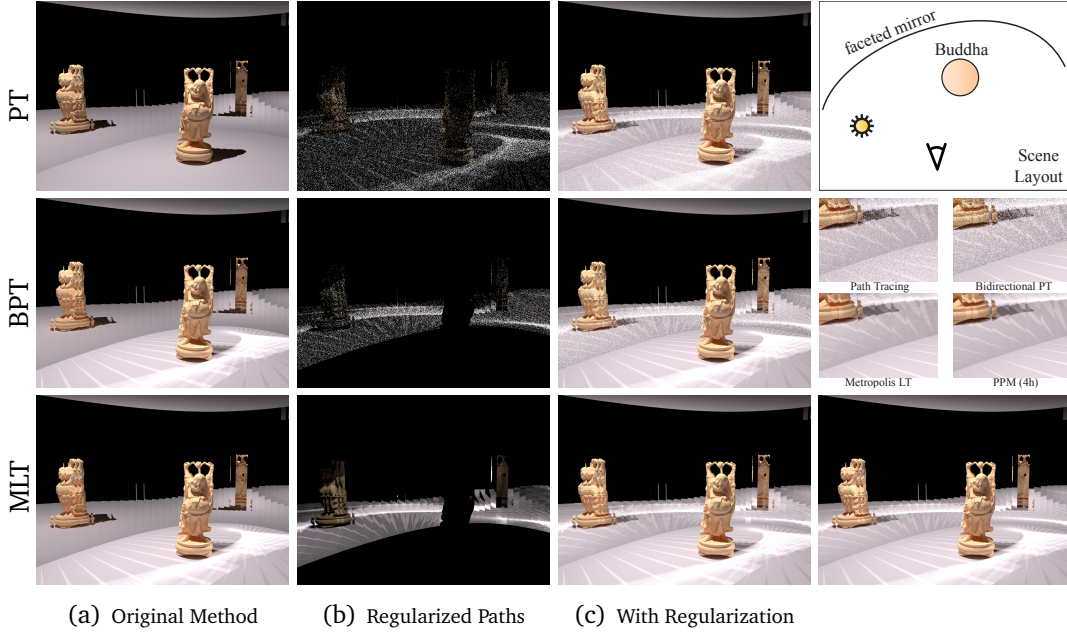


Figure 6.4: Equal time rendering (30 min.) of the BUDDHA scene at 1024×768. This scene illustrates the additional light transport paths enabled by the regularization. It is simple enough to be handled with progressive photon mapping to compute the “reference” (4 hours). PPM reference rendered in 4 hours is on the bottom right.

we select it conservatively close enough to 1 to achieve the practical convergence. We use the value of 1% of the scene bounding radius for the initial on-surface bandwidth r_0 , $\lambda = 1/6$ for MC integration with 2D regularization and $\gamma = 1 - 10^{-4}$ for MCMC integration in all experiments.

6.5 Results and Discussion

Figure 6.4 shows PT, BPT, and MLT with and without regularization to demonstrate the difference when accounting for all light transport paths (which in this example can also be handled by kernel estimation).

Figure 6.5 demonstrates the strength of our method in finding difficult paths. Due to the specular BRDF and the pinhole camera, unbiased methods typically have difficulties handling the reflection of the LEDs on the curved surface.

Figure 6.6 shows a comparison of MLT with regularization and Markov chain

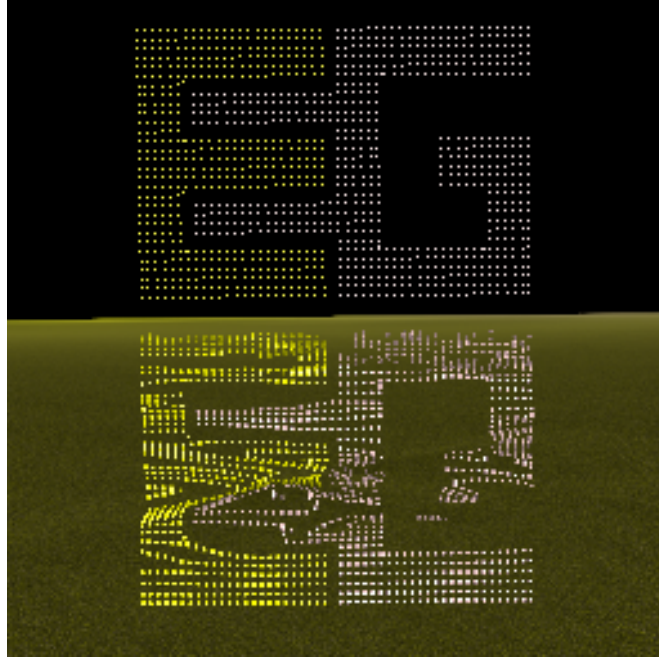


Figure 6.5: The EG logo is made of 4194 LEDs and placed over a curved specular water surface with a subtle diffuse component. The image was rendered using MLT with regularization in 10 minutes at a resolution of 256×256 pixels.

PPM (MCPMM) demonstrating the convergence of the former. MCPMM as well as the original MLT method does not handle all light transport paths (see also Figure 6.1). Bias introduced by the regularization is apparent as blurring of the floor caustics reflected in the wall; due to a spatially-controlled mollification it is similar to the bias of PPM. However, in contrast to kernel estimation methods, the majority of lighting in regularized MLT is rendered in an unbiased way, which can be seen by comparing the sharpness of the edge of the directly visible caustic on the floor. Bright splotches in both regularized and original MLT images are due to the set of mutations in primary sample space; the original mutation set with manifold exploration path generator (the last two images in the bottom row of Figure 6.6) reduces these artifacts.

Other Consistent Methods as a Special Case. Photon mapping methods can be considered as a special case of regularization. These methods do not connect eye-subpaths with randomly shot light-subpaths (as BPT), but one more ray is shot from the eye-subpath according to the local importance function and then a range query

gathers from proximate light-subpaths at the intersection point. This can be interpreted as on-surface regularization after sampling, leading to the same regularization procedure that we propose for specular BSDFs. However, this regularization is *applied to all paths* regardless whether they can be sampled or not. Our method can also be combined with more advanced sampling methods such as MLT, which has a superior convergence than PPM in non-trivial illumination scenarios (Figure 6.6) and does not suffer from the absence of the cache as BPT (Figure 6.4). Virtual spherical lights (VSL) [Hasan et al. 2009] are used to replace virtual point lights for faster rendering of glossy scenes. This is equivalent to an angular mollification of specular BSDFs (at the connection point) and convergence is also achieved by gradually shrinking the VSL radius as in Eq. 6.5.

Potential Improvements and Limitations. One interesting future work for MCMC methods is a more tight analysis of the spectral gap of the transition kernel. This would enable the analysis of the convergence rate, as well as a more accurate choice of the bandwidth shrinkage rate parameter γ .

One limitation of our method can be observed with Energy Redistribution Path Tracing (ERPT) [Cline et al. 2005]. It can also benefit from regularization, but due to the design of the algorithm, the bias can be high: ERPT computes a predetermined number of mutations for many short chains, therefore, the mollification bandwidth has to be reset to r_0 for every new chain.

Conclusions. Selective regularization in path space using mollification enables integration of the singularities in the light transport integrand, and can be applied to any integration method. The regularization is only required for a small subset of the path space, avoiding bias for the majority of light transport, while handling hard light paths in a consistent way. This provides a simple yet powerful framework, which enables consistent sampling of paths, hitherto non-samplable by advanced techniques, such as Metropolis light transport.

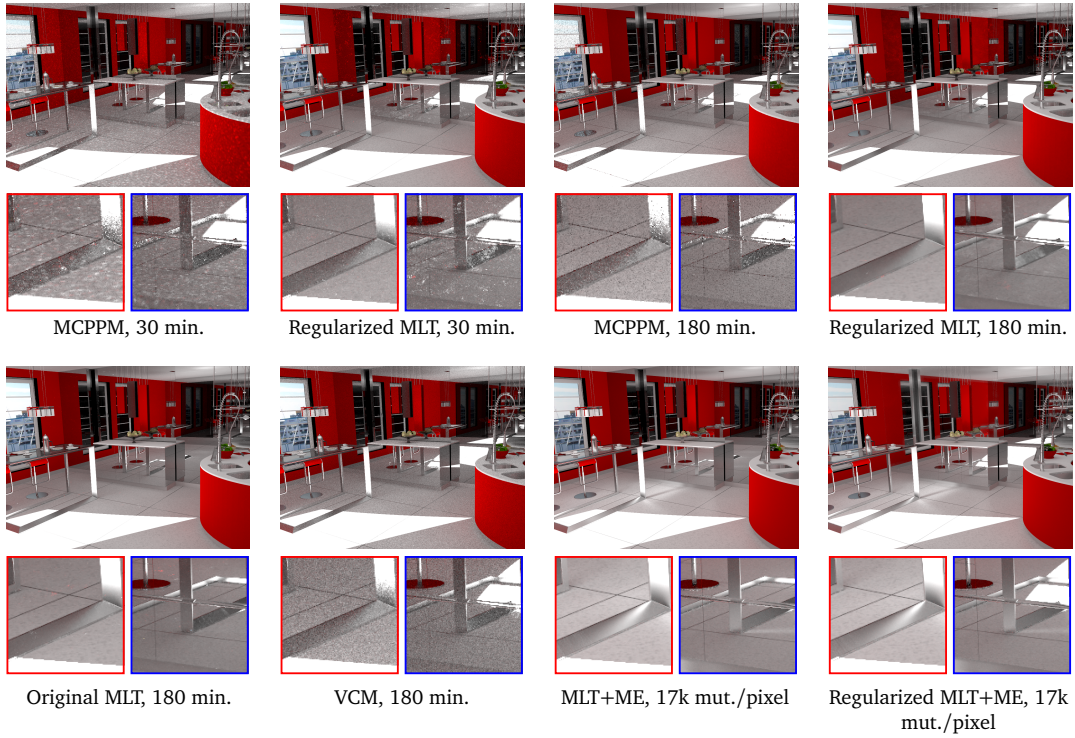


Figure 6.6: Equal time comparison: a point light illuminates the KITCHEN scene from outside through glass windows. Note how MCPPM introduces bias where MLT does not: here bias becomes noticeable as blurred edges of the floor caustic (red inset). The right inset shows regions where both MCPPM and regularized MLT introduce bias; original MLT cannot handle the caustic reflected off the mirror wall. For MCPPM we update eye subpaths once per 2000 mutations of the Markov chain. We use 16384 Markov chains mutating in parallel for both MLT and MCPPM. VCM, an advanced non-Metropolis technique, generates noisy results which is due to the inefficiency of random path sampling (compared to exploring the small subset of light paths through the window contributing to the lighting in the image). The two rightmost figures in the bottom row were rendered using original MLT with the recent manifold exploration using the Mitsuba renderer [Jakob 2010]. We use approximately the same number of mutations per pixel as for other images rendered with our renderer in 3 hours. The slight difference to the other images is due to the different material descriptions used. Note how manifold exploration is able to efficiently explore regularized paths.

II

Half Vector Space Light Transport

Light Transport in Half Vector Space

...if one were to refuse to have direct, geometric, intuitive insights, if one were reduced to pure logic, which does not permit a choice among every thing that is exact, one would hardly think of many questions, and certain notions... would escape us completely.

— HENRI LEBESGUE

Société Mathématique (Copenhagen, 8 May 1926)

The path integral formulation of light transport is the basis for (Markov chain) Monte Carlo global illumination methods. In this chapter we present the foundation for *half vector space light transport* (HSLT), a domain for constructing light transport paths on surfaces. The key is a partitioning of the path space into subspaces in which a path is represented by a sequence of half vectors between its start and endpoint.

We show how light transport on surfaces (in the absence of participating media) can be reformulated in a domain that is better suited for difficult phenomena. Consider for example a perfect specular reflection: in this case the trajectory of light is dictated by Fermat's principle, that is, the angle of incidence is equal to the angle of reflection. The configuration can likewise be expressed using the half vector between the incident and outgoing directions, which should coincide with the surface normal. This geometric constraint is always met for specular reflections and we thus call it a *natural constraint*.¹

¹In case of refraction, Snell's law serves as a special case of Fermat's principle, where the *generalized* half vector aligns with the surface normal.

These constraints have long been used in optical design, e.g., in ABCD matrix analysis or transfer/system matrices, and also were applied to image formation with pencil tracing [Shinya et al. 1987]. Recently, these geometric considerations of Fermat’s principle led to an improved sampling technique for specular interactions with manifold exploration [Jakob and Marschner 2012] in MLT.

In light transport simulation, a path is typically represented as a list of vertices (locations of interactions), or by its start vertex and a sequence of sampled directions. We propose a different representation, where the relation between incident and outgoing directions at interactions is specified by the direction of the half vector [Sommerfeld and Runge 1911] in the local tangent frame of a surface. This representation is a local parametrization and unique only within a subspace of the path space (paths through the surface patches with the same convexity, Figure 7.2). Usually, we need to sample the entire path space for light transport simulation. However, considering the light transport *within* each local subspace of this representation provides a number of beneficial properties:

- many practical bidirectional scattering distribution functions (BSDFs) can be expressed with half vectors naturally [Walter et al. 2007] or more compactly [Rusinkiewicz 1998],
- it enables importance sampling both geometry (e.g., curvature) and BSDFs at all interactions along the path,
- we can directly generate valid paths by perturbing the half vectors (see Figure 7.2), even with glossy and specular BSDFs on curved surfaces in presence of small light sources.

In addition, in Section 7.2.1 we provide both an analytic and a numerical analysis of the transformation between the half vector space and the standard measurement spaces from Section 2.2.7 commonly used for integration in the path space.

Throughout the chapter, we ignore visibility and assume the BSDFs can be well represented in half vector domain (up to smoothly varying parameters, such as the Fresnel term).

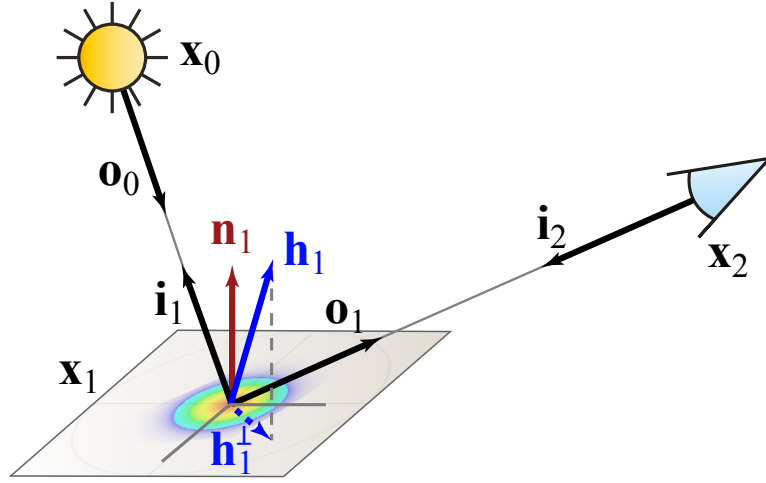


Figure 7.1: Illustration of the additional notation on a simple light path. See Table 2.2 for an overview of the notation.

Later in Chapter 8, we will introduce the new path construction method, which makes use of the new domain within Metropolis light transport framework.

7.1 Preliminaries

7.1.1 Path Integral in Path Space

In order to compute the path integral in Eq. 2.10, we have to construct a path from the path space Ω , and compute its measurement contribution in the on-surface area domain. However, the path construction is typically performed in the *domain of outgoing directions* (or projected solid angle), where most of the path construction methods operate: eye subpaths and/or light subpaths are constructed by sampling a direction at an interaction location. In order to compute the path measurement contribution we still need to do a change of variables at every vertex \mathbf{x}_i from the sampling domain of directions to the surface area measure at the next vertex $d\mathbf{o}_i^\perp \mapsto d\mathbf{x}_{i+1}$, which is done using the geometric term from Eq. 2.9 as

$$G(\mathbf{x}_i, \mathbf{x}_{i+1}) = \left| \frac{d\mathbf{o}_i^\perp}{d\mathbf{x}_{i+1}} \right| = \left| \frac{d\mathbf{o}_i \langle \mathbf{o}_i, \mathbf{n}_i \rangle}{d\mathbf{x}_{i+1}} \right| = \frac{|\langle \mathbf{i}_{i+1}, \mathbf{n}_{i+1} \rangle \langle \mathbf{o}_i, \mathbf{n}_i \rangle|}{\|\mathbf{x}_i - \mathbf{x}_{i+1}\|^2}.$$

These geometric terms only appear explicitly in next event estimation in path tracing or in the subpath connection in BDPT as they otherwise cancel out with the one-sided geometric term of the path pdf when computing the path throughput.

Unfortunately, these geometric terms can introduce weak singularities in measurements as they are not always well-behaved on geometric junctions, e.g., at corners. Nevertheless, the measurement space is the only space for a comparison of the measurement and the pdfs of two paths, therefore, is an important space for techniques that require such comparisons, e.g., multiple importance sampling and Metropolis light transport [Veach 1998].

On the other hand, the sampling domain of outgoing directions is well-suited for local sampling at scattering interactions. A path is represented in this space indirectly (e.g., as an ordered set of vertices), i.e., it has to be raytraced to resolve all path vertices. As a consequence, altering directions can lead to changes in the path configuration caused, e.g., by hitting a different surface, or a drastic change of an endpoint position. This is inconvenient when constructing paths for point-to-point light transport (connection from a light source to a sensor), because the trajectory between the endpoints cannot be easily controlled without changing the endpoints themselves.

This domain also entails one limitation on most of the local path construction methods: at some sampling step a path is completed, i.e., it forms a point-to-point connection, and the values of BSDFs have to be evaluated post factum, which can lead to unpredictable changes in the measurement carried by such a path. Ideally, we would like to importance-sample all BSDF interactions along the path, while keeping the endpoints fixed.

7.1.2 Generalized Halfway Vector

The (non-projected) half vector \mathbf{h} is defined as the *generalized half vector* [Sommerfeld and Runge 1911, Walter et al. 2007], as

$$\mathbf{h} = (\mathbf{i} + \eta \mathbf{o}) / \|\mathbf{i} + \eta \mathbf{o}\|.$$

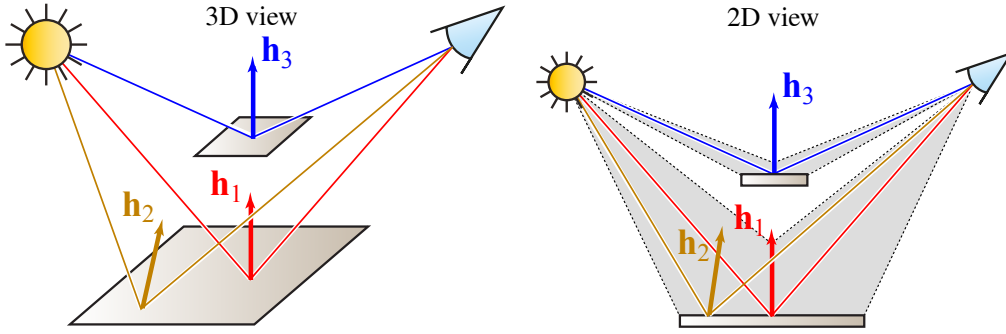


Figure 7.2: The new representation partitions the path space into subspaces, where a path can be uniquely maps to its natural constraints described in Section 7.2.1. On the right, we depict two subspaces (gray wedges) for paths of length 2, each bounded by one of the surfaces: \mathbf{h}_1 and \mathbf{h}_2 specify two paths in the same subspace. Red and blue paths have the same half vector $\mathbf{h}_1 = \mathbf{h}_3$, but belong to different subspaces. Our method enables to easily construct paths within the same subspace.

The term η_i is the ratio of the indices of refraction at vertex i , corresponding to the previous and next edge of the path. It is 1 for reflection and η_o/η_i for transmission events, where η_o corresponds to the edge along the outgoing direction \mathbf{o} .

We can transform the half vector density $d\mathbf{h}$ to the density of an outgoing direction $d\mathbf{o}$ by applying Jacobians from BSDF theory [Walter et al. 2007]:

$$\left| \frac{d\mathbf{h}}{d\mathbf{o}} \right| = \frac{\eta^2 |\langle \mathbf{h}, \mathbf{o} \rangle|}{(\langle \mathbf{h}, \mathbf{i} \rangle + \eta \langle \mathbf{h}, \mathbf{o} \rangle)^2}. \quad (7.1)$$

7.2 Light Transport in Half Vector Space

7.2.1 Half Vector Domain for Light Paths

For light transport simulation we need to (1) construct a path, and (2) compute the measurement contribution of the path that, for the domain of outgoing directions, potentially involves a change of variables introducing singularities. Instead, we use a different domain for path construction, which does not suffer from the aforementioned singularities and performs a global path construction, enabling us to importance-sample all interactions along the path in between two endpoints (e.g., a pixel on a sensor and a point on a light source), while keeping the endpoints fixed.

This *half vector space*, or a *space of natural constraints*, is based on the observation that a complete path—in a subspace of the path space—can be represented by an ordered set of vertices, $\mathbf{X} = (\mathbf{x}_0, \dots, \mathbf{x}_k)$, or, alternatively, it can also be uniquely represented using the two endpoints \mathbf{x}_0 and \mathbf{x}_k and a sequence of half vectors \mathbf{h}_i at the interactions (see Figure 7.1). We introduce the space of natural constraints, or the half vector domain, as

$$\mathbf{H}^\perp = (\mathbf{x}_0, \mathbf{h}_1^\perp, \mathbf{h}_2^\perp, \dots, \mathbf{h}_{k-1}^\perp, \mathbf{x}_k) \in \Omega(\mathbf{H}^\perp) \subset \Omega,$$

where a *submanifold* $\Omega(\mathbf{H}^\perp)$ is a subspace of a path space, where every path in $\Omega(\mathbf{H}^\perp)$ is uniquely represented in a form \mathbf{H}^\perp .

This domain has two key properties that we will introduce in the following sections:

- due to the path construction in the half vector space, the integrand becomes flatter and decomposed into highly decorrelated subintegrands,
- the domain of constraints enables an easy exploration of all paths of the same topology in the same submanifold $\Omega(\mathbf{H}^\perp)$.

7.2.2 Analysis of the New Domain

Next, we introduce the transformation from the half vector domain to the measurement domain and analyze the properties of this mapping. The transformation is obtained from the Jacobian of the half vector constraint matrix from projected half vector measure $d\mathbf{H}^\perp$ into surface area measure $d\mathbf{X}$:

$$\int_{\Omega(\mathbf{X}_0)} f(\mathbf{X}) d\mathbf{X} = \int_{\Omega(\mathbf{H}_0^\perp)} f(\mathbf{X}) \left| \frac{d\mathbf{X}}{d\mathbf{H}^\perp} \right| d\mathbf{H}^\perp, \quad (7.2)$$

where $\Omega(\mathbf{X}_0)$ and $\Omega(\mathbf{H}_0^\perp)$ denote the submanifold around the path \mathbf{X}_0 in the path space. We will detail in Section 8.1 how this requirement is practically met.

The Jacobian for the change of variables from vertex positions density $d\mathbf{x}_i$ in local tangent space to the density of projected half vectors $d\sigma^\perp(\mathbf{h}_i) \equiv d\mathbf{h}_i^\perp = d\mathbf{h}_i |\langle \mathbf{h}_i, \mathbf{n}_i \rangle|$

is written as

$$J = \frac{d\mathbf{H}^\perp}{d\mathbf{X}} = \frac{d(\mathbf{h}_1^\perp \dots \mathbf{h}_{k-1}^\perp)}{d(\mathbf{x}_1 \dots \mathbf{x}_{k-1})} = \begin{pmatrix} B_1 & C_1 & & \\ A_2 & B_2 & C_2 & \\ & \ddots & \ddots & \ddots \\ & & A_{k-2} & B_{k-2} & C_{k-2} \\ & & & A_{k-1} & B_{k-1} \end{pmatrix}, \quad (7.3)$$

which has a structure of a block tridiagonal matrix with 2×2 blocks A_i, B_i, C_i . More details on this matrix can be found in Wenzel Jakob's thesis [Jakob 2013, Section 4.3]. It can also be used as a first order approximation that maps finite changes of path vertices $\Delta\mathbf{X} = (\Delta\mathbf{x}_0 \dots \Delta\mathbf{x}_k)$ to offsets $\Delta\mathbf{H}^\perp = (\Delta\mathbf{h}_1^\perp \dots \Delta\mathbf{h}_{k-1}^\perp)$ in projected half vector space as $\Delta\mathbf{H}^\perp = J \cdot \Delta\mathbf{X}$.

The complete measurement contribution then includes two Jacobians, the Jacobian of projected half vectors $d\mathbf{x}_i \mapsto d\mathbf{h}_i^\perp$, and the product of geometric terms $d\mathbf{o}_{i-1}^\perp \mapsto d\mathbf{x}_i$:

$$f(\mathbf{X}) \left| \frac{d\mathbf{X}}{d\mathbf{H}^\perp} \right| = \rho(\mathbf{X}) \left(\prod_{i=0}^{k-1} G(\mathbf{x}_i, \mathbf{x}_{i+1}) \right) \left| \frac{d(\mathbf{x}_1 \dots \mathbf{x}_{k-1})}{d(\mathbf{h}_1^\perp \dots \mathbf{h}_{k-1}^\perp)} \right|. \quad (7.4)$$

Here $\rho(\mathbf{X})$ denotes the product of BSDFs f_s at each surface vertex, the sensor responsivity W_e at the endpoint \mathbf{x}_k , and emitter radiance L_e at the start point \mathbf{x}_0 . These components of $\rho(\mathbf{X})$ are straightforward to evaluate and common to all global illumination methods. We will thus focus on the product of the two Jacobians.

7.2.2.1 Properties of the Product of Jacobians

Consider the most common case of point-to-point transport, where constraints are two endpoints and the projected half vectors (on-surface microfacets) for interactions in between. For a complete path $X = (\mathbf{x}_0, \dots, \mathbf{x}_k)$ the product of Jacobians in

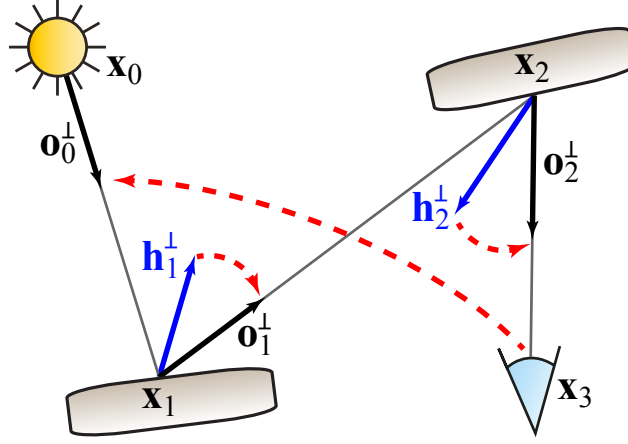


Figure 7.3: The simplified product of Jacobians first transforms the displacement at the last vertex \mathbf{x}_3 to the outgoing direction at the first one (\mathbf{x}_0), allowing to track the changes in flux density throughout the path. Then a transformation from projected half vectors to projected outgoing direction at every scattering vertex is applied.

Eq. 7.4 can be written as

$$\begin{aligned}
 (\text{Eq. 7.4})/\rho(\mathbf{X}) &= \left| \frac{d(\mathbf{o}_0^\perp \dots \mathbf{o}_{k-1}^\perp)}{d(\mathbf{x}_1 \dots \mathbf{x}_k)} \right| \left| \frac{d(\mathbf{x}_1 \dots \mathbf{x}_{k-1})}{d(\mathbf{h}_1^\perp \dots \mathbf{h}_{k-1}^\perp)} \right| \\
 &= \left| \frac{d(\mathbf{o}_0^\perp, \mathbf{o}_1^\perp \dots \mathbf{o}_{k-1}^\perp)}{d(\mathbf{x}_k, \mathbf{x}_1 \dots \mathbf{x}_{k-1})} \right| \left| \frac{d(\mathbf{x}_1 \dots \mathbf{x}_{k-1})}{d(\mathbf{h}_1^\perp \dots \mathbf{h}_{k-1}^\perp)} \right| \\
 &= \left| \frac{d\mathbf{o}_0^\perp}{d\mathbf{x}_k} \right| \prod_{i=1}^{k-1} \left| \frac{d\mathbf{o}_i}{d\mathbf{h}_i} \right| \left| \frac{\langle \mathbf{o}_i, \mathbf{n}_i \rangle}{\langle \mathbf{h}_i, \mathbf{n}_i \rangle} \right|, \tag{7.5}
 \end{aligned}$$

where we first swapped matrix columns (property of the Jacobian determinant) and then applied the chain rule in this rearrangement. The first term of Eq. 7.5 is a *generalized geometric term* [Jakob and Marschner 2012] and for perfect specular chains

$$\left| \frac{d\mathbf{o}_0^\perp}{d\mathbf{x}_k} \right| = \left| \frac{d\mathbf{o}_0^\perp}{d\mathbf{x}_1} \right| \left| \frac{d\mathbf{x}_1}{d\mathbf{x}_k} \right| = G(\mathbf{x}_0, \mathbf{x}_1) |T_1|, \tag{7.6}$$

where T_1 is a *transfer matrix* and can be computed using pencil tracing [Shinya et al. 1987]. The transfer matrix T_1 contains the differential transformation of an on-surface tangent patch at \mathbf{x}_1 to \mathbf{x}_k based on the differential geometry. In other words, it tracks the changes of area density along the light trajectory from the last

vertex \mathbf{x}_k on the camera to the first on-surface vertex after the camera \mathbf{x}_{k-1} .

Therefore, Eq. 7.5 can be seen as the two steps: first, transform the area density as if considering a specular chain using transfer matrices (Figure 7.3); then, apply the individual transformations from half vectors to outgoing directions to account for glossy transport.

As an example, assume a path has only surface reflections (as in Figure 7.3). If we insert the reflection Jacobians $|\mathrm{d}\mathbf{h}_i/\mathrm{d}\mathbf{o}_i| = 1/|4 \cdot \langle \mathbf{h}_i, \mathbf{o}_i \rangle|$ for transforming half vector density to outgoing direction (see [Walter et al. 2007] and Eq. 7.1) into Eq. 7.5, we obtain:

$$\left| \frac{\mathrm{d}\mathbf{o}_0^\perp}{\mathrm{d}\mathbf{x}_k} \right| \prod_{i=1}^{k-1} 4 \cdot \left| \frac{\langle \mathbf{o}_i, \mathbf{n}_i \rangle \langle \mathbf{o}_i, \mathbf{h}_i \rangle}{\langle \mathbf{h}_i, \mathbf{n}_i \rangle} \right|. \quad (7.7)$$

Note that all singularities caused by geometric terms except for $G(\mathbf{x}_0, \mathbf{x}_1)$ cancel out, which is a desirable property for integration.

7.2.2.2 Qualitative and Quantitative Analysis

The simplified product of Jacobians in Eq. 7.5 is numerically more stable than evaluating geometric terms for difficult geometric configurations, such as corners, because transfer matrices in these configurations degenerate to identities, while geometric terms become singular (see Figure 7.4).

Another advantage of the new domain is that the integrand dimensions are highly decorrelated from each other, i.e., changing the value of one constraint (e.g., a half vector) causes the change in the complete path measurement contribution that comes mostly from the local change of the corresponding BSDF at the changed half vector. In other words, modifications to a half vector \mathbf{h}_i^\perp mostly influences the BSDF at a single vertex \mathbf{x}_i . For importance sampling, we do not consider visibility and terms that are not expressible in half vector domain, such as the Fresnel coefficient, directional emission distribution of the light source and directional sensor responsivity. In practice, we did not notice significant loss of path construction efficiency, and rely on Metropolis sampling to account for these terms (see Chapter 8).

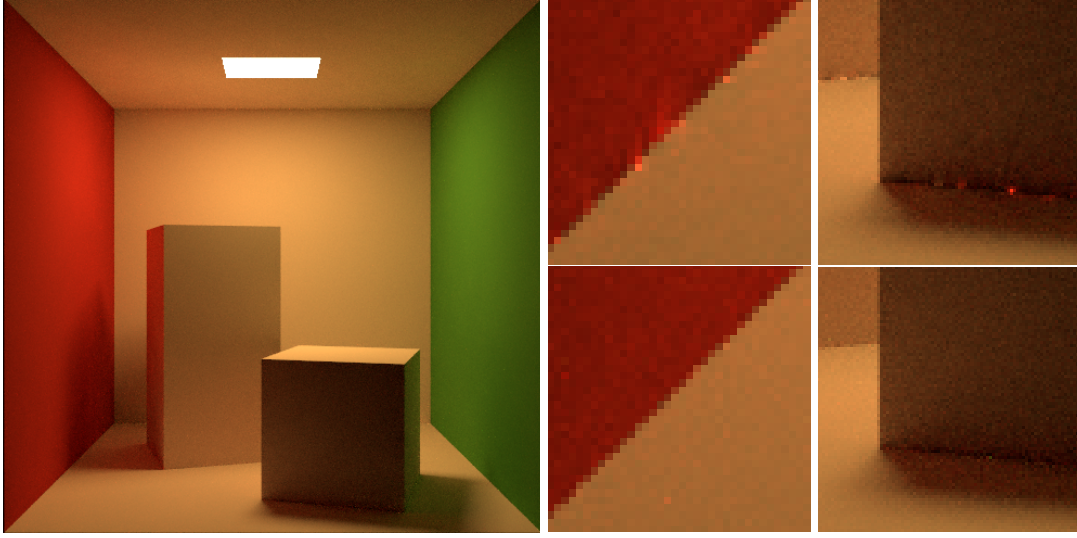


Figure 7.4: The CORNELL BOX (256 samples/pixel) with diffuse surfaces rendered using MLT with the original set of mutations (equivalent to manifold exploration for diffuse surfaces) in on-surface area measure (closeups top row), and with our method (bottom row). Note how the singularities in the corners cancel out thanks to the well-behaved product of Jacobians with the half vector domain.

The remainder of the integrand (Eq. 7.5) depends on the differential geometry, which is well-behaved within a submanifold. Analysis of this remainder (Eq. 7.7 shows an example) reveals that all components of this product change smoothly: the first term contains one initial geometric term and the determinant of one transfer matrix. The cosine terms are known to introduce smooth low-frequency changes in the half vector domain.

To support these claims, we provide a numerical evaluation. Table 7.1 shows how the product of Jacobians influences the shape of the integrand in half vector domain. The two distributions are very similar, which confirms that the product of Jacobians is relatively flat. This also confirms the observations about the decorrelated dimensions in the new domain. In addition, the effect of the Jacobians in the case of glossy transport is negligible. On the other hand, even for highly diffuse transport the product remains well-behaved.

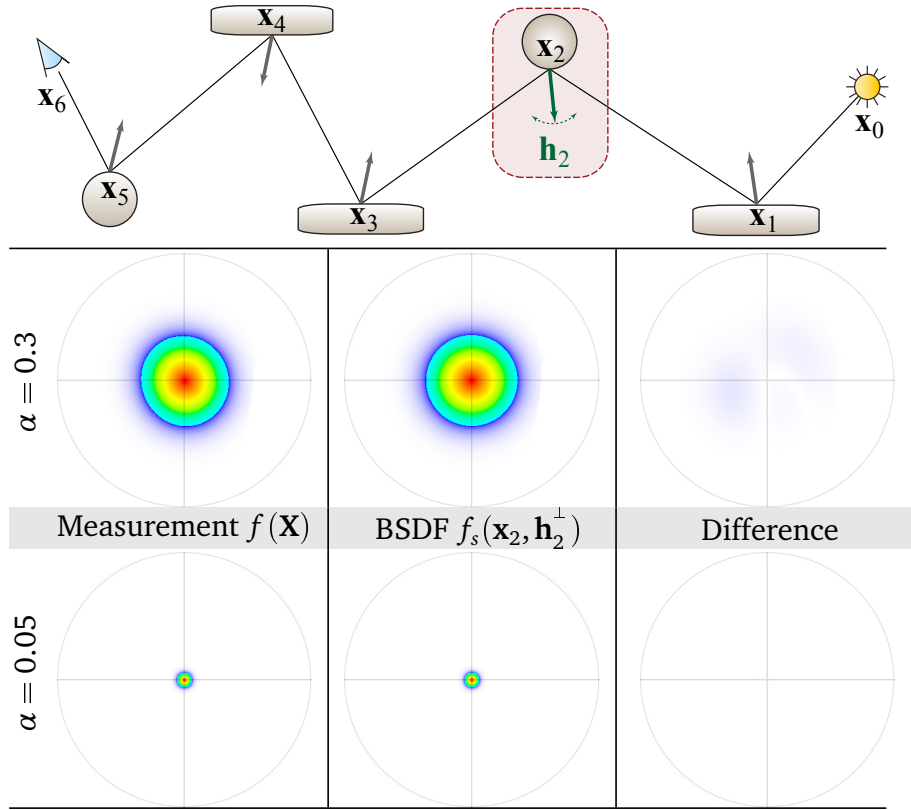


Table 7.1: Changes of the measurement contribution $f(\mathbf{X})$ (left) and the Blinn-Phong BSDF $f_s(\mathbf{x}_2)$ (center) caused by perturbing only the half vector at the highlighted vertex. We visualize normalized values in false-color inside the domain of the projected half vector (the unit circle). The differences (right) between the normalized distributions show that the changes of the full integrand are mostly due to the changes in the local BSDF at the perturbed half vector (Eq. 7.4).

7.2.3 Specular Interactions in the New Domain

Specular interactions are usually considered as a special case in light transport. A specular interaction reduces the dimensionality of the integration and entails that the respective constraint (half vector) should not be perturbed when sampling paths. In fact, we will see that one of the terms in the product in Eq. 7.5 cancels out with a specular BSDF.

This cancellation is observed when the specular BSDF $f_s(\mathbf{i} \rightarrow \mathbf{o})$ with its delta distribution is expressed in the half vector domain. The respective transformation from the domain of outgoing directions to the domain of half vectors adds the

Jacobian $d\mathbf{o} \mapsto d\mathbf{h}$ as

$$f_s = \kappa \frac{\delta_{\mathbf{o}}(\mathbf{s})}{|\langle \mathbf{o}, \mathbf{n} \rangle|} = \kappa \frac{\delta_{\mathbf{h}^\perp}(\mathbf{0})}{|\langle \mathbf{o}, \mathbf{n} \rangle|} \left| \frac{d\mathbf{h}}{d\mathbf{o}} \right|, \quad (7.8)$$

where \mathbf{s} is the outgoing direction of a perfect specular interaction and κ folds all other factors, such as the reflection/transmission coefficient and the Fresnel term. For pure specular interaction the dot product caused by the projected half vector $|\langle \mathbf{h}, \mathbf{n} \rangle| \equiv 1$, therefore, we omitted it in the numerator.

Eq. 7.8 contains the inverse Jacobian, $|d\mathbf{h}/d\mathbf{o}|$, to the respective Jacobian in Eq. 7.5. For the special case of a purely specular chain in a path, due to such cancellations in the resulting sub-Jacobians, the corresponding terms in Eq. 7.5 become equal to Eq. 7.6, resulting in a generalized geometric term used in manifold exploration [Jakob and Marschner 2012]. In practice, this allows us to directly evaluate materials with delta distributions in this domain by combining Eq. 7.8 with Eq. 7.1 as they are reduced to the canonical form. That is, they do not require treatment in a special domain, such as a “discrete measure” used in the Mitsuba renderer [2010].

7.2.4 Discussion

The aforementioned properties enable a path construction strategy that importance-samples all interactions of a path in between endpoints all together. For example, it is possible to predict the change in path measurement caused by altering the value of a single half vector constraint. From a practical standpoint, many BSDFs can be expressed [Walter et al. 2012], represented [Rusinkiewicz 1998], or approximated (e.g., by a Beckmann roughness equivalent [Jakob 2010]) in the half vector domain.

One advantage of computing light transport in this domain is that we can modify individual constraints, while preserving others, e.g., the endpoints of a path. This is in contrast to other domains: in the domain of outgoing directions the endpoint position depends on chosen outgoing directions; in the on-surface domain, moving one vertex affects the directions towards two adjacent vertices, leading to changes in the corresponding BSDFs.

In this section we detailed only a common case of point-to-point transport, where two endpoints are fixed. Other types of boundary constraints follow the same derivation, e.g., directional constraint imposed by a directional light source or an orthographic camera model, which replace the positional constraints. In this case, the positional constraint at the corresponding endpoint is replaced by a directional constraint (please see [Jakob 2013, p.70] for more details).

With Eq. 7.5, we compute the path measurement contribution $f(\mathbf{X})$ using transfer matrices (or system matrices), which were prior used only for specular transport paths [Shinya et al. 1987]. To our knowledge, we are the first to employ transfer matrices for the general light transport framework with arbitrary scattering (such as glossy and diffuse) along the path. In other words, the path is always treated as a pure specular transport going through the specular microfacets defined by a set of preselected half vectors (see Figure 7.3).

7.3 Conclusion

In this chapter, we presented and analyzed a novel domain for constructing light transport paths. We introduced the half vector domain, an attractive alternative domain for computing and analyzing light transport. We derived the transformation from this domain to the measurement domain and examined the resulting transformation Jacobian determinant.

Half Vector Space Path Construction for Efficient Light Transport

*The mathematicians know a great deal about very little and the
physicists very little about a great deal.*

— STANISŁAW MARCIN ULAM

Analogies between Analogies, LA-2055 report (10 May 1990)

The path integral formulation of light transport [Veach 1998] and numerical integration and sampling techniques, such as (Markov chain) Monte Carlo methods, form the basis of nowadays global illumination methods. In all their diversity, these methods share the concept of stochastically creating paths, connecting the sensor to the lights. With limited computational budgets, i.e., finite sample count, the sampling strategy is of utmost importance. Metropolis light transport (MLT) [Veach and Guibas 1997] was the first application of Markov chain Monte Carlo (MCMC) sampling to light transport problem. This class of methods constructs a new path by perturbing the existing path and concentrates on paths with high contributions to the image. MLT is able to locally explore regions of the path space that are otherwise difficult to reach, such as highly occluded regions of the scene.

Based on the previous chapter, we propose a new perturbation-based path construction to be used with Markov chain Monte Carlo methods such as Metropolis light transport, which is well-suited for all types of surface transport paths (with diffuse/glossy/specular interactions). A new path is constructed by perturbing half vectors of the existing path individually, while keeping the first point on the light

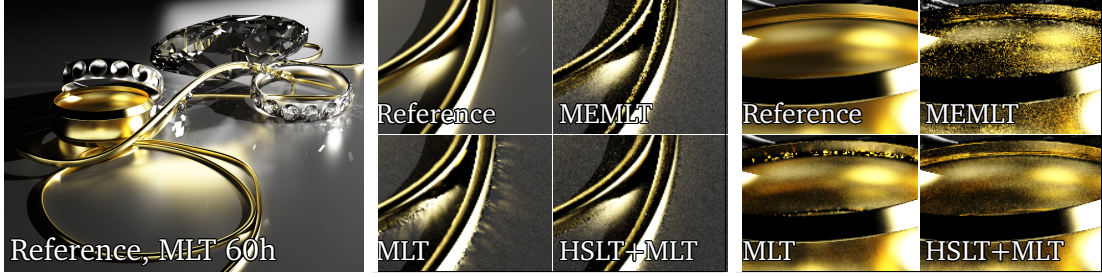


Figure 8.1: Equal-time comparison of the JEWELRY scene with difficult glossy paths. Left: Reference computed with the original Metropolis Light Transport (MLT) in 60 hours. The two closeups show 10 minute renders using original MLT, Manifold Exploration (MEMLT) and MLT with the proposed path construction method (HSLT+MLT). All MLT variants used 1 million initial samples, MEMLT used $\lambda = 50$. There was no special handling of direct illumination. The average number of samples per pixel: MEMLT 621, MLT 1424, HSLT+MLT 612. The new path construction method estimates the optimal sampling density of difficult specular and glossy transport and excels at capturing hard features like thin highlights along highly curved rims.

source and the last point on the sensor. That is, the method explores the subspace containing the current path (see Figure 7.2) by perturbing microfacets. One consequent characteristic of path construction in this new domain is that the Fourier-domain properties of the path integral can be practically estimated. These properties are used to achieve optimal correlation of the samples due to well-chosen perturbation step sizes, leading to more efficient exploration of light transport features. We also propose a novel approach to control the image-space stratification in MLT in the new path construction domain.

In Section 8.2.2 we detail how this path construction can be controlled by balancing the three aspects of importance sampling: exploring the BSDFs, achieving stratification on the image plane, and sampling according to the frequency spectrum. These goals can be achieved simultaneously in the half vector domain.

Main advantages of the new path construction strategy are:

- a novel integration method using the new representation to robustly handle complex *inter-surface* light transport,
- a mutation that operates on entire paths without explicit connections (Section 8.1),
- a novel stratification scheme for MCMC rendering (Section 8.2.2),

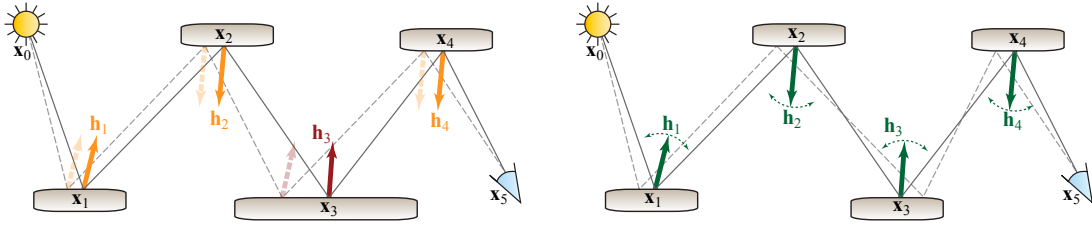


Figure 8.2: Manifold exploration (left) fixes the half vectors (orange) for two selected sub-paths by stochastically classifying vertices as specular. It then perturbs the direction at one of the end points and constructs a new path. The scattering at the vertex between these two subchains (here: x_3 , red half vector) is not importance-sampled. Our new strategy takes all BSDF interactions along the path into account. Moreover, while manifold exploration keeps all half vectors but one (orange) fixed, uncontrollably changing this one half vector (red), we perturb all half vectors simultaneously. This reduces the sample correlation and leads to faster exploration of illumination features.

- we employ recent results on the Fourier analysis of Monte Carlo simulation [Subr and Kautz 2013] to optimally redistribute sampling density (Section 8.2.3).

8.1 Path Construction in Half Vector Space

The new representation considers isolated subspaces of the path space, and only within a subspace the transformation Jacobian from the previous chapter is valid. To sample a path from the subspace, we employ a Markov chain method and explore the subspace by using small perturbations to the half vectors. In order to sample from all subspaces, similarly to original MLT or ERPT [Cline et al. 2005], we augment our path construction with another method, such as the bidirectional mutation or BDPT, and then transform the new path into the half vector representation. We then perturb the constraints of the path (the endpoints as well as the half vectors) and by this efficiently explore the new subspace.

8.1.1 Half Vector Space Mutation

The key to an efficient intra-subspace exploration is a new perturbation-based path construction method, which adds to the framework of MLT mutations. Starting with

the current path of a Markov chain, we compute geometric derivatives of the half vector constraints in the tangent spaces of all vertices on the path. This step is analogous to manifold exploration and results in a block-tridiagonal matrix J , which expresses the first-order change of half vectors with respect to vertex positions in their respective tangent spaces (Eq. 7.3, described in Section 7.1.2 and [Jakob and Marschner 2012]). During this process we also compute and store the half vectors of the current path.

We proceed by perturbing all half vector constraints simultaneously, i.e., by sampling a small region around each individual half vector of the current path. Note that all path vertices are affected even if only one half vector is perturbed. We first focus on how a new path is constructed with the new half vectors, and then describe the sampling strategy for half vector constraints, which takes into account both surface roughness as well as stratification over the image plane.

Obtaining Path Vertices. Given perturbed half vectors, we need to determine the locations of the new path vertices on the scene surfaces. These locations are required, for example, to test for visibility and to perform surface shading. To this end, we use a modified predictor-corrector Newtonian method from manifold exploration [Jakob and Marschner 2012]. This works by first *predicting* new vertex positions using J as the Jacobian in a Newtonian method. And second, *correcting* these positions by using ray tracing to project the predicted points back onto the scene surfaces.

If we applied the original correction scheme of manifold exploration to project vertex positions, it would retrace the path starting at one of the endpoints, e.g., from \mathbf{x}_0 towards \mathbf{x}_1 , and enforce the newly sampled half vectors at every interaction. This results in a displaced end vertex \mathbf{x}'_k and the correction $T_1 \cdot (\mathbf{x}'_k - \mathbf{x}_k)$ can be used to iteratively refine the path vertices \mathbf{x}'_i . This original algorithm converges when the error $(\mathbf{x}'_k - \mathbf{x}_k)$ is not too large; this approach typically works very well for purely specular surfaces.

Instead, we use another approach, which first explicitly predicts the change of all vertex positions at once using $\Delta \mathbf{X} = J^{-1} \Delta \mathbf{H}^\perp$ (see Eq. 7.3 in the previous chapter)

	From light	From eye	Proposed
Successful generation	47.8%	51.2%	52.8%
Acceptance rate	45.3%	46.8%	47.4%
Average # of iterations	2.20	2.19	2.18

Table 8.1: Statistics for different tracing direction for the corrector in the KITCHEN scene. The statistics was gathered on the same number of paths (10.84M) rendered with the same initial seed paths. The left column refers to the ray-traced projection that always starts from the direction of the light source. The middle column is for projecting from the sensor (eye); and the rightmost column shows the advantage of our direction selection heuristic.

and then project the new vertex positions onto the surfaces. The above prediction-correction steps have to be iterated until the half vectors of the resulting path match the new constraints. This strategy, although a little more costly to evaluate, typically has better convergence to desired half vectors, especially, at grazing interaction angles. Note that, a full inversion of the Jacobian matrix can be avoided (Section 8.3) due to the block-tridiagonal structure of the matrix.

We select the projection direction (either from the sensor or from the light source) by computing $|\mathbf{dx}_1/\mathbf{dx}_{k-1}| = |T_1| \cdot |T_{k-1}|^{-1}$, which is the first-order estimation of the movement of \mathbf{x}_1 when moving \mathbf{x}_{k-1} . If this ratio is less than one, we start by projecting vertices with \mathbf{x}_{k-1} , \mathbf{x}_{k-2} etc. (from the sensor), as this minimizes the projection error. Otherwise, we project starting from the light (\mathbf{x}_1 , \mathbf{x}_2 , ...). This allows us to improve the convergence of the predictor-corrector method (see Table 8.1).

Acceptance Probability. To compute the acceptance probability for a Metropolis move from the current path \mathbf{X}_t to a proposed path \mathbf{X}_{t+1} according to the Eq. 3.8, both the transition probability T and the path contribution f are computed in the product on-surface area measure $d\mathbf{X}$ for both transition directions. When mutating in the half vector space $\Omega(\mathbf{H}^\perp)$ the transition probability $T(\mathbf{H}_t^\perp \rightarrow \mathbf{H}_{t+1}^\perp)$ is easy to compute, as the joint probability is just the product of the probabilities of the individual perturbations $p(\mathbf{h}_{t+1}^\perp|\mathbf{h}_t^\perp)$. In order to use it in the acceptance probability a (which is in product on-surface area measure), we transform $T(\mathbf{H}_t^\perp \rightarrow \mathbf{H}_{t+1}^\perp)$ back

to $\Omega(\mathbf{X})$ and evaluate

$$\begin{aligned} R_{t+1} &= f(\mathbf{X}_{t+1}) / \left(T(\mathbf{H}_t^\perp \rightarrow \mathbf{H}_{t+1}^\perp) \left| \frac{d\mathbf{H}^\perp}{d\mathbf{X}} \right| \right) \\ &= f(\mathbf{X}_{t+1}) \left| \frac{d\mathbf{X}}{d\mathbf{H}^\perp} \right| / T(\mathbf{H}_t^\perp \rightarrow \mathbf{H}_{t+1}^\perp), \end{aligned}$$

where the second line shows the interpretation of integration in half vector space, as described in Section 7.2.2 of the previous chapter. This formula uses the simple transition probability $T(\mathbf{H}_t^\perp \rightarrow \mathbf{H}_{t+1}^\perp)$. In addition, the numerator takes advantage of the simplified path measurement from Eq. 7.5. This allows to decrease the amount of computations as well as improves the numerical robustness of the method.

In order to guarantee convergence with MLT, the set of mutations has to cover the whole path space. The proposed mutation does not meet this requirement, e.g., it does not mutate the path length. Therefore, we always use the bidirectional mutation [Veach 1998] as an augmenting mutation. Similar to manifold exploration, our method does not introduce visible bias in practice, even though new paths are constructed up to a limited numerical precision.

Comparison to Manifold Exploration. Since the formulation used to derive manifold exploration (ME) is based around removing Dirac deltas from the integration, it is conceptually limited to specular surfaces (i.e., the special case described in Section 7.2.3). The extension of ME to glossy materials stochastically classifies all but one glossy interaction as specular, and keeps the half vectors for these “quasi-specular” interactions fixed. In every new mutation, a different half vector is non-fixed. The BSDF at this interaction is evaluated post factum, after the path construction is complete. Note that this can lead to uncontrollable change in the measurement contribution $f(\mathbf{X})$ due to the lack of importance sampling at this BSDF. Figure 8.2 (left) shows an example, where vertex \mathbf{x}_3 is classified as non-specular, which means that ME will not importance-sample the BSDF at \mathbf{x}_3 , nor try to keep the half vector constant. Moreover, because the half vectors

at the other vertices are kept constant, they might be insufficiently resampled by the method, which can lead to visible correlation artifacts. For example, the scratch-like artifacts in the gold ring closeup in Figure 8.1 and in the reflection on the wall in Figure 8.8 appear due to these correlations. Also, treating glossy vertices as specular still requires to account for the Jacobian determinant, which in turn requires a costly full matrix inversion when specular and glossy vertices are mixed [Jakob and Marschner 2012].

Path construction in the half vector domain does not depend on classifying vertices as specular or non-specular, and can directly explore a multi-glossy caustic by importance-sampling all BSDF interactions at once. Specular interactions do not require special treatment, because their half vectors always remain at a fixed position (at zero), effectively reducing the dimensionality of the integration. In addition, the simplified measurement contribution in half vector space in Eq. 7.5 treats all types of interactions and BSDFs (diffuse, glossy, and specular) equally, without a need to handle any special case.

8.1.2 Transitioning between Subspaces

As noted in Section 7.2.2, the integration in the domain of half vectors is valid only within a small subspace $\Omega(\mathbf{H}_t^\perp)$ (Figure 7.2). First, within this subspace of the path space the implicit function theorem should hold [Spivak 1965]. That is, a subspace is a region of path space where a predictor-corrector can uniquely walk from a path \mathbf{H}^\perp to a path $\mathbf{H}_{\text{new}}^\perp$. Otherwise, paths \mathbf{H}^\perp and $\mathbf{H}_{\text{new}}^\perp$ belong in different subspaces.

The new path construction works directly in the domain of constraints and relies on the derivatives of the constraints at every vertex along the path. Therefore, the discontinuities in these derivatives, e.g., in normal derivatives, become more apparent (see Figure 8.4, left). The remainder of this section is devoted to this problem and introduces an extension to our random walk that enables it to switch between different subspaces, leading to easier exploration of the path space.

Analogous to manifold exploration, we test whether the walk with our projection method from a current path \mathbf{H}_t^\perp to a proposal path \mathbf{H}_{t+1}^\perp is reversible, i.e., if we

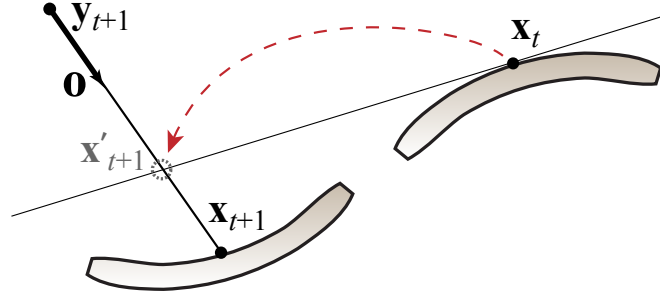


Figure 8.3: When jumping from one submanifold to another, first the point \mathbf{x}'_{t+1} is predicted by the constraint Jacobian matrix J at point \mathbf{x}_t . Then the projection step traces a ray from the point of the previous vertex of the proposal \mathbf{y}_{t+1} to find the projected point \mathbf{x}_{t+1} on the new subspace.

can also walk back from \mathbf{H}_{t+1}^\perp to \mathbf{H}_t^\perp . Note that the predictor-corrector scheme can potentially find a new path that satisfies the proposed constraints \mathbf{H}_{t+1}^\perp but belongs in a different subspace. In this case, if we then perform a reverse walk, we might end up with a path that satisfies the original constraints \mathbf{H}_t^\perp but still lies in this different subspace. We determine such cases by testing whether the vertex positions match the positions of the original path and reject the proposal based on that. This ensures the detailed balance requirement of the Metropolis-Hastings algorithm.

In the case of a non-reversible walk, manifold exploration rejects the proposal path. In contrast, in our method, if the walk fails (i.e., when our predictor-corrector scheme does not converge to the given new constraints \mathbf{H}_{t+1}^\perp), we reject the proposal but attempt to use the predicted path differently instead of rejecting the walk. For that, we test whether the first-order prediction of the new path given by the Jacobian of the constraints results in a valid path, i.e., with non-zero throughput after projection; this path is also a result of the prediction of the first predictor-corrector iteration. If this path is valid, we make sure it is non-reversible and take it as a *suggested* perturbation of the path. We interpret this case as an indication of a jump from one subspace to another and apply different transition rules (described below).

For performing a random walk, we compute the new transition probability density $T(\mathbf{X}_t \rightarrow \mathbf{X}_{t+1})$ of sampling this suggested path in the on-surface measure. This probability density consists of the probability densities of (1) predicting each vertex

\mathbf{x}'_{t+1} and (2) projecting \mathbf{x}'_{t+1} to obtain \mathbf{x}_{t+1} . The former is the probability density of sampling the point \mathbf{x}'_{t+1} on the tangent plane of the current path's vertex \mathbf{x}_t . This simple change of the domain is computed using the constraint Jacobian as

$$p(\mathbf{x}'_{t+1}|\mathbf{x}_t) = p(\mathbf{H}_{t+1}^\perp|\mathbf{H}_t^\perp) \left| \frac{d\mathbf{X}}{d\mathbf{H}^\perp} \right| = p(\mathbf{H}_{t+1}^\perp|\mathbf{H}_t^\perp) |J_{\mathbf{H}_t^\perp}|^{-1}.$$

The correction step obtains the projected vertex \mathbf{x}_{t+1} by casting a ray from the previous vertex \mathbf{y}_{t+1} towards the outgoing direction $\mathbf{o} = \overline{\mathbf{y}_{t+1}\mathbf{x}'_{t+1}}$ (Figure 8.3). Note that \mathbf{x}'_{t+1} might be different from \mathbf{x}_{t+1} , i.e., it might not lie on a surface. The second probability density accounts for this and converts the on-surface probability density of the predicted vertex from the virtual surface (defined by the tangent plane at \mathbf{x}_t) to the density at the new projected vertex \mathbf{x}_{t+1} on the actual surface. As the projection is done by tracing a ray, we transform the density using the shared domain of fixed outgoing direction \mathbf{o} (Figure 8.3), which leads to a simple ratio of geometric terms for the points \mathbf{x}'_{t+1} and \mathbf{x}_{t+1} :

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t) = p(\mathbf{x}'_{t+1}|\mathbf{x}_t) \left| \frac{d\mathbf{o} d\mathbf{x}'_{t+1}}{d\mathbf{x}_{t+1} d\mathbf{o}} \right| = p(\mathbf{x}'_{t+1}|\mathbf{x}_t) \frac{G(\mathbf{y}, \mathbf{x}_{t+1})}{G(\mathbf{y}, \mathbf{x}'_{t+1})}.$$

The transition probability of the full proposal path $T(\mathbf{X}_t \rightarrow \mathbf{X}_{t+1})$ is then computed sequentially as a joint product of probabilities at every vertex along this path, starting from the same direction as the projection step. In order to satisfy the detailed balance, we divide by the current transition probabilities for both ways when computing the acceptance probability a as in Eq. 3.8, i.e., we compute both $T(\mathbf{X}_t \rightarrow \mathbf{X}_{t+1})$ and the reverse probability $T(\mathbf{X}_{t+1} \rightarrow \mathbf{X}_t)$. Moreover, to make this random walk symmetric, we test that the predictor-corrector method cannot converge *both ways* guaranteeing that the walk cannot be done from the suggested path back to the current path using the regular manifold walk (we denote such walks as *bidirectionally invalid*). This extension allows for better exploration of fragmented and highly displaced geometry (see salad leaves in Figure 8.4).

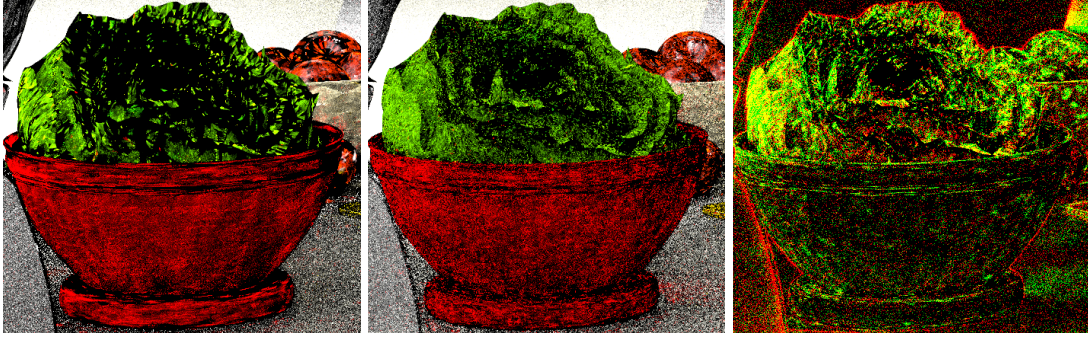


Figure 8.4: SALAD BOWL: the exploration across geometric edges of the salad leaves is difficult as a walk cannot easily jump to another subspace (left); with our improvement transition, the walk can more easily jump between submanifolds leading to more uniform exploration of the illumination (center); right: false colors showing where the walk attempted to jump (red) and where the jump was successful (green).

Discussion. If a walk is not possible, we try to take a path suggested by the constraints derivatives and treat it as a path generated by perturbing directions, as in other path construction methods. This facilitates jumps between different subspaces.

This procedure is used only if a walk is bidirectionally invalid, which indicates the boundary of a subspace. The fraction of non-reversible walks depends on the structure and smoothness of the scene surfaces and is slightly higher for our method compared to manifold exploration (see Table 8.2) due to higher reliance on the differential geometry around the path.

8.2 Sampling Improvements

In order to make the new perturbation efficient for computing images, we describe how to adapt the mutation step sizes to the individual BSDFs, stratify samples over the image plane, and consider the spectral bandwidth of the signal to distribute the perturbation across multiple interactions.

	Ours	ME
Average # of iterations	2.18/6.63	2.61/2.98
Path construction rate	58.8%	46.2/59.2%
Non-reversible walks	3.3%	2.1%
- Submanifold jump	88.3% (of 3.3%)	—
Acceptance rate	47.1%	35.7%
Mutations	453.6 M	488.6 M

Table 8.2: Statistics for our path construction and the corresponding numbers for manifold exploration for the equal-time renders in Figure 8.1. The number of iterations is counted both per successful proposal and total. Note that our method can create almost the same total number of mutations as manifold exploration.

8.2.1 Optimal Expected Step Size for Interactions

Integration in the domain of half vectors has the property of decorrelated dimensions, and thus, we are able to optimize the integration for individual interactions. That is, a change in one half vector \mathbf{h}_i^\perp influences mostly the local value of the BSDF at the vertex \mathbf{x}_i only (see Section 7.2.2.2). We apply this *local* assumption to every constraint independently to improve the sampling of paths as a whole.

In addition, we assume that all BSDFs can express their roughness as Beckmann equivalent α , i.e., the microfacet (or half vector) distribution of a BSDF can be approximated by a Beckmann distribution. As Walter et al. [2012] noticed, most microfacet-based BSDFs are easily expressible in this way. This assumption is also practical for a renderer and is, e.g., used in Mitsuba renderer [Jakob 2010].

Then, for a single interaction, e.g., a glossy reflection, we can estimate the optimal expected value of the mutation step size. Projected onto the parallel plane at distance one to the tangent plane, a Beckmann lobe is a 2D Gaussian around the center with variance $\sigma^2 = \alpha^2/2$. We use the projected half vector \mathbf{h}^\perp which is related to this plane-plane domain as $\mathbf{h}_{p-p} = \mathbf{h}/\sqrt{1 - |\mathbf{h}^\perp|^2}$.

The optimal MCMC acceptance rate for such a 2D Gaussian is known to be about 35% [Bremaud 1999] and the optimal expected step size for a Gaussian proposal is $s = (2/\pi)|\Sigma|$, where Σ is the covariance matrix of the 2D Gaussian. We can derive the expected step size for a single half vector dimension in the parallel plane

domain as

$$s_{\text{p-p}, \text{max}} = \alpha / \sqrt{\pi}.$$

When mutating a half vector constraint, the expected value of the change to the half vector (i.e., the distance between the current and perturbed half vector in the parallel plane) should have an expected value of $s_{\text{p-p}, \text{max}}$ to achieve the fastest mixing of the Markov chain (we describe the actual perturbation after analyzing stratification in Section 8.2.2). This optimal step size is designed to achieve the maximum exploration of a 2D Gaussian.

Because visibility is ignored in our analysis, it can narrow the actual support of a BSDF. We account for this by reducing the step size as

$$s_{\text{p-p}} = \gamma \cdot s_{\text{p-p}, \text{max}}, \quad \text{with } \gamma \in (0, 1].$$

The scaling parameter γ has a subtle effect on the sampling efficiency. We found that $\gamma = 0.1$ (10% of the optimal step size) gives the best results for difficult visibility cases in practice and therefore was used for all rendered images in this chapter.

Note that $s_{\text{p-p}}$ is defined in the plane-plane domain. It can be converted to half vector domain using the projected half vector and the relation provided above. However, in practice we use a numerically more stable Möbius transformation described in Section 8.3. We demonstrate the importance of the optimal step size selected for each individual interaction in Figure 8.5.

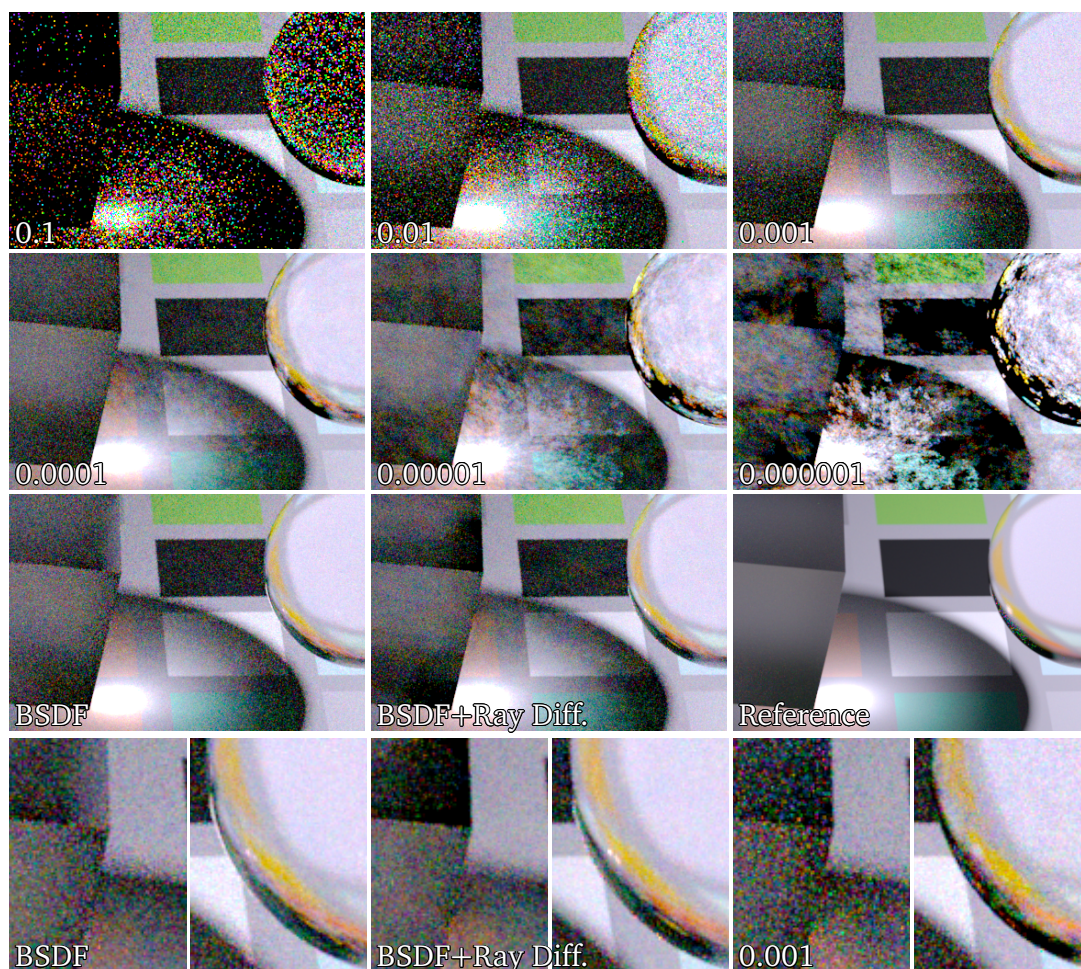


Figure 8.5: A simple scene with a diffuse ground plane, and a sphere and a cube with a rough dielectric with different roughness values (0.01 and 0.1), rendered with 128 samples per pixel and various fixed step sizes as indicated (two top rows). The third row shows our optimal step sizes for BxDF bandwidth and the same combined with ray differentials, as well as the reference image. The insets in the bottom row show that the automatic BxDF estimation is comparable to the best manually chosen parameter. Large step sizes for the BxDF maximize the exploration of the submanifold, but are limited by the ray differentials in favor of stratification on the image plane.

8.2.2 Stratification over the Image Plane

Another important requirement for a good path construction method is to ensure stratification of the generated paths on the image plane. Therefore, we compute a required perturbation to the half vectors from the desired stratification on the image plane, and combine these perturbations with the step size dictated by BSDFs.

Given a current path, as a first step to project a step size in image space to half vector domain, we compute the ray differentials [Igehy 1999] from the current position on the sensor \mathbf{x}_k to the first vertex in the scene. Next, we project the two ray differential vectors along the $\mathbf{u} \times \mathbf{v}$ basis of the image plane (\mathbf{u} and \mathbf{v} are horizontal and vertical offsets on the image plane with a length of 1 pixel) onto a tangent frame of the first vertex in the scene \mathbf{x}_{k-1} . This gives us a pair of offset vectors $\Delta\mathbf{x}_{\mathbf{u},k-1}$ and $\Delta\mathbf{x}_{\mathbf{v},k-1}$ on the surface. Veach [1998] suggested to use a stratification step size of 5% of the image size (by scaling these vectors) and use this as the desired change of \mathbf{x}_{k-1} . However, his experiments were mostly on diffuse interactions. We found that a scaling of 2% is more practical in scenes with highly glossy surfaces, due to potentially more localized highlights.

Next, we derive how \mathbf{x}_{k-1} changes when we perturb individual half vectors. Therefore, we compute the half vector perturbation $\Delta\mathbf{h}_i^\perp$ from $\Delta\mathbf{x}_{k-1}$ using the Jacobian for the change of variables from vertex position density in local tangent

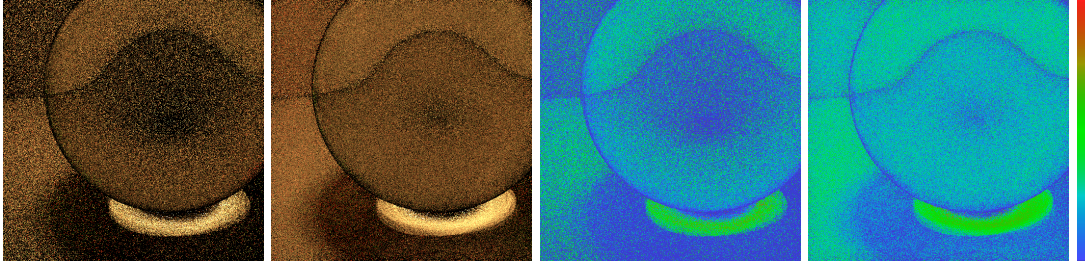


Figure 8.6: False color visualization of sample counts in the CORNELL BOX with a glass sphere rendered with 8 spp without (left) and with (right) the proposed stratification.

space to the density of half vectors (see Eq. 7.3):

$$\begin{array}{c} \vdots \\ k-1 \\ k \end{array} \underbrace{\begin{array}{|c|} \hline \\ \hline D_i \\ \hline \end{array}}_{=J^{-1}} \cdot \begin{array}{|c|} \hline 0 \\ \hline \Delta \mathbf{h}_i^\perp \\ \hline 0 \\ \hline \vdots \\ \hline \end{array} = \begin{array}{|c|} \hline \Delta \mathbf{x}_0 \\ \hline \vdots \\ \hline k-1 \\ \hline k \end{array}, \quad \Delta \mathbf{h}_i^\perp = D_i^{-1} \cdot \Delta \mathbf{x}_{k-1}. \quad (8.1)$$

where D_i is the i -th 2×2 sub-block in the $(k-1)$ -th row of the inverted constraint Jacobian J^{-1} , indicating how the last vertex \mathbf{x}_{k-1} changes when changing a constraint \mathbf{h}_i^\perp . We use this first-order approximation to independently approximate every $\Delta \mathbf{h}_i^\perp$. Due to the sparsity of J , it is not necessary to perform a full matrix inversion (see Section 8.3). Using D_i^{-1} , we convert the ray differential offsets $\Delta \mathbf{x}_{\mathbf{u},k-1} \times \Delta \mathbf{x}_{\mathbf{v},k-1}$ to the half vector offsets at each vertex obtaining $\Delta \mathbf{h}_{\mathbf{u},i}^\perp \times \Delta \mathbf{h}_{\mathbf{v},i}^\perp$.

At this point, we have two ways of computing a step size for each vertex: considering the BSDF *or* image plane stratification. Since we want to make sure to stay on the BSDF lobe *and* inside the image boundaries, we use the smallest of the two step sizes. This, e.g., ensures that purely specular surfaces keep a constant half vector. As ray differentials are often anisotropic, we first rotate the sampling scheme to align with the basis vectors $\Delta \mathbf{h}_{\mathbf{u},i}^\perp \times \Delta \mathbf{h}_{\mathbf{v},i}^\perp$, and take the minimum of the step sizes proposed by the BSDF and ray differentials for these axes individually. Section 8.3 explains this sampling in detail.

Figure 8.6 shows the ray differential-based stratification in the half vector domain. It ensures that samples are more evenly distributed over the image plane, leading to a uniform convergence and lower rejection rates caused by large jumps.

The proposed stratification step size at a single half vector leads to the complete desired change at \mathbf{x}_{k-1} . Therefore, we have to distribute the stratification step size among path constraints $\mathbf{h}_1^\perp \dots \mathbf{h}_{k-1}^\perp$, which we detail in the next section.

8.2.3 Sampling Motivated by the Integrand Spectrum

In order to distribute the estimated step sizes, we assign a per-constraint weight w_i to each step size with $\sum_{i=1}^{k-1} w_i = 1$. To find these w_i , we recall that to achieve a low variance of the resulting estimator, the variance of the sampling spectrum needs to be low where the integrand's power spectral density is high [Subr and Kautz 2013]. That is, we need to focus the sampling density to places with narrower spectral bandwidth.

We reformulate the path integral for point-to-point transport as a nested convolution of the emitted light L_e with BSDFs along the path as

$$\text{Eq. 7.2} = \langle W_e * \langle f_{s,k-1} \cdots * \langle f_{s,2} * \langle f_{s,1} * L_e \rangle \rangle \rangle \rangle.$$

We are interested in the resulting value at the sensor. We consider each of these nested convolutions in half vector domain as an isolated integration process due to the property of highly decorrelated dimensions. Because all BSDFs are either expressed or approximated with a 2D Gaussian, it is easy to estimate the bandwidth of this signal at a particular interaction \mathbf{h}_i^\perp , which is proportional to the standard deviation σ_i of the corresponding Gaussian. Our heuristic is therefore to use the weight $w_i = \sigma_i / \sum_{j=1}^{k-1} \sigma_j$ for re-scaling the step size from Section 8.2.2 for every half vector. This way we redistribute the desired stratification offset according to the local constraints at every half vector constraint.

8.3 Implementation Details

We implemented the method as a new mutation strategy in the Mitsuba renderer [Jakob 2010], to be used in the path space MLT framework. The proposed implementation tries to avoid unnecessary computational overhead. We provide a high level overview of our implementation.

```

sample_mutation( $\mathbf{X}_t, \mathbf{X}_{t+1}$ )
    // sample a new path
     $\mathbf{H}_{t+1}^\perp = \text{mutate\_half\_vectors}(\mathbf{X}_t)$  (Section 8.2.2, Section 8.3)
     $\mathbf{X}_{t+1} = \text{h\_to\_positions}(\mathbf{X}_t, \mathbf{H}_{t+1}^\perp)$  (Section 8.1.2)
    // reverse check
     $\text{h\_to\_positions}(\mathbf{X}_{t+1}, \mathbf{H}_t^\perp) == \mathbf{X}_t$ 
    // compute transition probabilities
     $T_1, D_i^{-1} = \text{ray\_differentials}(\mathbf{X}_{t+1})$  (Section 8.3)
     $r = T(\mathbf{H}_{t+1}^\perp \rightarrow \mathbf{H}_t^\perp) / T(\mathbf{H}_t^\perp \rightarrow \mathbf{H}_{t+1}^\perp)$ 
    // compute half vector measurement
     $f_H^\perp(\mathbf{X}_{t+1}) \leftarrow T_1, \mathbf{X}_{t+1}$  (Eq. 7.4, Eq. 7.5, Eq. 7.6)
     $a = f_H^\perp(\mathbf{X}_{t+1}) / f_H^\perp(\mathbf{X}_t) \cdot r$ 

```

Figure 8.7: High-level overview of the new path construction strategy.

Core Algorithm. Figure 8.7 shows the pseudo-code for our path construction. This procedure is executed after the initial seeding phase and receives a valid path \mathbf{X}_i as a current sample. It terminates the construction and returns an acceptance probability of zero in case if any operation is not feasible, e.g., a singular Jacobian matrix is attempted to be inverted or a non-reversible walk is encountered. This code also assumes that J , D_i , T_1 , and f_H^\perp are cached for a path, and recomputed for the current path \mathbf{X}_i only in case it originates from another mutation strategy. The function `h_to_positions` contains the predictor/corrector Newtonian method (Section 8.1) and recomputes the tangent space derivative J often (see Eq. 7.3 and [Jakob and Marschner 2012]) to derive the updated vertex positions from the perturbed half vector offsets. It does, however, not require updating transfer matrices or ray differentials. The same function is also used to check for reversibility. Finally the acceptance probability is computed by evaluating both the measurement contribution and the transition probabilities in half vector domain.

Efficient Transformation between Domain. When converting ray differentials to half vector domain, it is crucial to avoid a slow full matrix inversion. We take advantage of the sparse nature of the matrix J and use a stripped-down version of LU decomposition to solve only for the blocks D_i from Eq. 8.1.

In addition, in Section 8.1 we demonstrate how to compute vertex offsets as

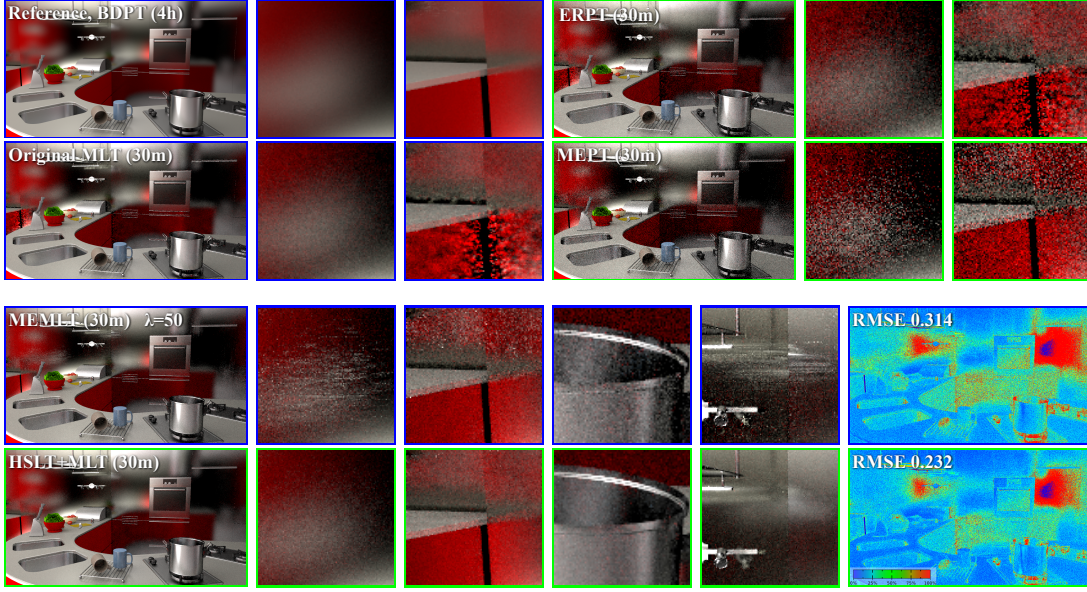


Figure 8.8: Equal-time rendering of the KITCHEN scene with difficult glossy paths. The top row shows BDPT, MLT, ERPT and MEPT for comparison. The two bottom rows compare manifold exploration (MEMLT) (237 spp, acceptance rate 42.7%) to our method (HSLT) (229 spp, acceptance rate 67.3%). Note how efficient the new mutation can estimate the optimal sampling density of difficult regions and handle hard features such as glossy caustics (please see the supplementary for more results).

$\Delta \mathbf{X} = \mathbf{J}^{-1} \Delta \mathbf{H}^\perp$ in the prediction step of the Newtonian method. Again, there is no full matrix inversion required, and we employ the LU decomposition to compute $\Delta \mathbf{X}$.

Anisotropic Half Vector Sampling. We perform a random walk in \mathbf{H}^\perp rather than resampling each half vector from scratch. We sample an anisotropic offset around the current half vector \mathbf{h}^\perp of the sample \mathbf{X}_i . We use a zero-centered anisotropic Phong lobe [Ashikhmin and Shirley 2000] and transform it into the space of orthonormalized ray differentials. We use the step sizes derived in Section 8.2.2 (i.e., the minimum of suggested BSDF step and the ray differential lengths $\|\Delta \mathbf{h}_u\|$ and $\|\Delta \mathbf{h}_v\|$ along the respective axes). Then we apply a Möbius transformation [Hanika 2011, p.40] to move the center of this distribution to the current half vector \mathbf{h}^\perp . This has the full disk as support and never samples outside it.

8.4 Results and Discussion

We implemented our techniques in Mitsuba renderer and provide all equal-time renders using this framework on a workstation with Intel i7-3930K hexacore CPU at 3.20GHz using 12 threads. To facilitate future work and comparisons, our full implementation is available as part of Mitsuba renderer.

We compare our method, i.e., half vector mutations with seeding by bidirectional mutation (see Section 8.1.1), to MLT with manifold exploration (ME) and Kelemen’s variant of MLT perturbing in the primary sample space (PSSMLT) [Kelemen et al. 2002] and we additionally show results of bidirectional path tracing (BDPT).

User Parameters. We use the default parameters in the Mitsuba renderer which are the recommended optimal parameters from the corresponding publications. According to our experiments, they also yield the best results on our test scenes. We use 1.0 for the average number of chains and 100 mutations per chain for ERPT. We use $\lambda = 50$ for manifold exploration.

Test Scenes. Figure 8.1 shows a scene with a difficult mixture of glossy and purely specular surfaces, lit by a point spot light and area lights. Figure 8.8 shows a setup with glossy transport and complex distant illumination coming from the outside. Most of the illumination in the image is due to a caustic reflected off the glossy floor. When using a unidirectional path tracer as mapping of primary space samples to path space in PSSMLT, BSDFs (and thus half vectors) are importance-sampled in the domain of outgoing directions. As a consequence, it always misses the point light source. If a bidirectional path sampler is used, a deterministic connection is performed, evaluating a BSDF instead of importance-sampling it. Manifold exploration explores such scenes by keeping the light vertex \mathbf{x}_0 fixed, but cannot importance-sample all BSDFs simultaneously, leading to visible correlation in the rendered highlights.

Manifold exploration and our method are designed for the exploration of reflected caustics (specular-diffuse-specular paths); finding them is an orthogonal problem

and can be solved using path space regularization from Chapter 6.

Scenes with complex occlusion are challenging for our mutation strategy, as we do not specifically design it to work with difficult visibility. Unfortunately, representing the path integral in half vector domain incurs that all visibility constraints show up in all decorrelated 2D subintegrals of the integration domain (i.e., it appears in each BSDF interaction as the one depicted in Figure 7.1). The AJAR DOOR scene (Figure 8.9) belongs to this category of difficult scenes. Nonetheless, our mutation strategy handles this case well and performs on par with ME in some regions of the image, e.g., on the back wall which is explored uniformly.

Although our implementation leaves room for optimization, our mutation strategy involves more costly computation than other strategies in MLT. Therefore, we show equal-time comparisons, not images with an equal number of samples. That is, the more costly yet higher-quality samples with our method ultimately pay off.

Recent Advances and Future Work. Recently, several advances have been presented that improve and extend the original Half Vector Space Light Transport. Holzschuch et al. [2016] proposed a practical method for computing a more accurate optimal step size for non-Beckmann BSDFs, such as the Trowbridge-Reitz and measured BSDFs. Lastly, manifold next event estimation [Hanika et al. 2015a] employed the half vector domain in ordinary Monte Carlo methods to connect a path to a light source through multiple interfaces with specular or rough refraction.

We demonstrate that the new path construction method performs well on constructing complete paths. It is also practical to use it as a drop-in replacement for the code block of the deterministic manifold walk used in manifold exploration for connecting two points.

In our implementation used to generate the results, we always retrace the entire path to reduce the correlation of the generated samples. Diffuse interactions, however, can be easier explored by only retracing a subpath of a path and then performing a connection. This contrasts the problem that keeping a part of a path fixed increases the correlation between samples (see Figure 8.8), which is a well-known disadvantage of MCMC methods. Improved Half Vector Space Light Trans-

port [Hanika et al. 2015b] proposes to break up the path at diffuse constraints, improve the domain by representing the half vectors in parallel plane domain, which is a more natural slope domain for microfacet BSDFs. This also allows to compute the anisotropic stratification step size more robustly.

In presence of participating media, the dimensionality of the problem increases, since there are no explicitly defined differential manifolds as in the case of surfaces. We can imagine that this framework could be extended by additional constraints for participating media.

8.5 Conclusion

The new global path construction method has several benefits and enabled us to design a mutation strategy for Metropolis light transport which importance-samples all BSDF interactions along a path, while controlling image space stratification and incorporating recent advances in frequency analysis of light transport. It is tailored for glossy and specular interactions, yet also handles diffuse transport robustly.



Figure 8.9: The AJAR DOOR rendered in 30 minutes. Top: reference (rendered with PSSMLT in 12 hours). Middle: using MEMLT with the original set of mutations, RMSE 0.244. Bottom: with HSLT (only our mutation and bidirectional mutation), RMSE 0.176. Note that although this scene is challenging for our method (all half vectors have limited variation due to difficult visibility), it is still handled well.

III

Additional Applications

Analyzing and Controlling Light Transport

Painting is concerned with all the ten attributes of sight; which are: Darkness, Light, Solidity and Color, Form and Position, Distance and Propinquity, Motion and Rest.

— LEONARDO DA VINCI

The Notebooks of Leonardo da Vinci, c. 1478 – 1518

In the previous chapters we were focusing on improving the robustness of the complex light transport problems. While a physically accurate global illumination is a correct and exact solution, it can be hard to comprehend, analyze, and control for a human due to a complex interplay of light and materials.

A final rendered image contains a lot of information: we can potentially infer the scene shape, and its materials and light sources. Many illumination phenomena are recognizable for a trained viewer, however, even lighting professionals often lack “the big picture”. For example, how the illumination is being formed? An accessible visual analysis of the light transport solution can be a useful component of a workflow in many industries.

Industry-leading studios intensively adopt physically-based global illumination across their art generation pipelines. As a consequence, many existing artistic light editing tools become unsuitable as they address only local effects without considering the underlying physically based concepts and consequences. An artistic control over physically based global illumination can also be a desired tool when creat-

ing artwork, such as visual effects and computer-animated films, with generated imagery.

In this chapter, we present interactive light transport visualization tools, as well as a light transport manipulation technique that operates directly in the path space.

9.1 Interactive Visualization of Light Transport

Light transport visualization is a difficult task: even in a static scene, the data at hand is a five-dimensional light field caused by light propagating through space in every possible direction at the same time and reflecting at scene surfaces. On the other hand, conveying this information in a visually meaningful way can be of great help for the end users of the light transport simulation, e.g., digital artists, architects, engineers, and lighting designers.

9.1.1 Selection and Classification of Light Transport

A key observation is that, in order to understand the light transport, it is necessary to analyze it locally. To this end, all tools support what we refer to as *selective* visualization: they can visualize and manipulate only a particular, user-selected subset of light paths. This selection helps the user focusing on lighting phenomena of particular interest.

9.1.1.1 Extended Path Classification and Enriched Path Vertices

In order to provide an efficient selection and filtering of light transport, we extend Heckbert's notation [1990] for light paths, by enriching them with additional information, such as interaction types and object IDs. This provides a detailed differentiation of individual path interactions. We distinguish between diffuse (D), glossy (G), and specular (S) interactions. We additionally classify interactions as reflections (superscript \square^R) or transmissions (\square^T). We also define a token X_p corresponding to *any* surface interaction (D , G , or S) that (optionally) belongs to a

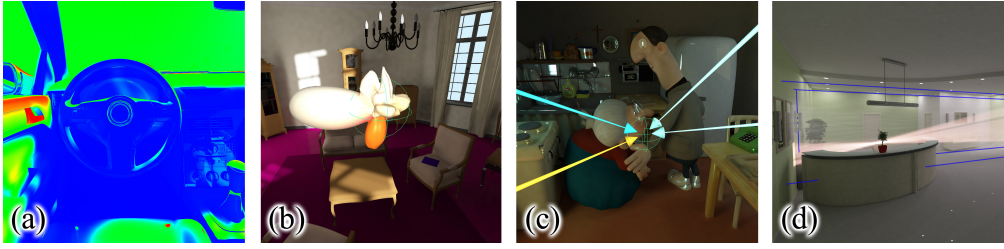


Figure 9.1: Light transport visualization tools provide key information about illumination structure within virtual scenes. From left to right: (a) the false-color intensity rendering, (b) spherical intensity plot, (c) light path inspection, and (d) volumetric inspection tools. Adapted from [Reiner et al. 2012].

user-selected volumetric region P within the scene.

We assign all objects and light sources in the scene a unique ID and store the ID of the object at each interaction. The scene object IDs carry semantic information for manually modeled scenes, and thus typically support intuitive visualization.

We store *enriched path vertices* with the following additional attributes:

- all material interaction types along the path (e.g., diffuse, glossy, specular reflections, refractions, directly visible light, etc.).
- the IDs of the hit objects (or light sources) for each interaction
- additional attributes for each interaction along the path (including hit position, and path throughput)

9.1.1.2 Clustering of Light Paths

When dealing with visualization of a collection of light paths, it is important to avoid clutter. Therefore, it is useful to cluster the gathered paths and group them into bundles for more intuitive visualization.

Clustering Based on Light Path Expressions. The first heuristic clusters by the last interactions of a light or an eye subpath. For example, if the last interactions are $LDDD$, $LDSS$, and $LDSD$, we obtain two clusters: (1) diffuse interactions $LDDD$ and $LDSD$, and (2) specular interactions $LDSS$.



Figure 9.2: The light path inspection tool visualizes the source of the caustic formed by a reflection of the sunlight off a mirror at the back wall. Adapted from [Reiner et al. 2012].

To cluster further interactions along the path, we apply the clustering algorithm recursively. In our example, we would partition cluster (1) according to the second last interactions, which yields two subclusters *LDDD* and *LDSD*.

Clustering Based on Object IDs. The second clustering heuristic is based on the object IDs obtained from the scene authoring process. The recursive clustering works the same way as in the previous method, except that the object ID is used instead of the interaction type.

9.1.2 Light Transport Visualization Tools

In order to efficiently examine the whys and wherefores of illumination phenomena, we introduce a set of four *light visualization tools*, presented in Figure 9.1.

False-Color Rendering. With *false-color rendering* of the on-surface intensity, we map the pixels' RGB values, i.e., outgoing radiance, to a scalar value (either equally weighted or according to perceived luminance). It is then used to index into color maps [Silva et al. 2011] (Figure 9.1a). Our motivation to include it into the set of tools is its simplicity, and the fact that engineering software often provides simi-



Figure 9.3: Volumetric inspection set to visualize only specular caustics (left), and all lighting components (right). Note how selective inspection emphasizes lighting phenomena. Adapted from [Reiner et al. 2012].

lar depictions of the surface irradiance or radiant exitance (e.g., Autodesk Ecotect Analysis).

Spherical Plot. We introduce the *spherical plot* tool to depict directional distribution of the radiance passing through a spherical region (sphere-shaped probe) or a disc-shaped probe if the region is on surface. The plot radius for a given direction is determined by the magnitude of the radiance and controlled by a user-defined parameter. Figure 9.1b shows spherical plots with different phenomena selected.

Light Path Inspection. The *light path inspection* tool visualizes the key light paths that contribute to the lighting of a user-specified region. It displays the gathered paths using several bundle-like arrows. In order to reduce clutter, we cluster the collected paths based on their type and create arrows for each clustered bundle (Figure 9.2). We use two path clustering strategies described in Section 9.1.1.

Volumetric Inspection Tool. The *volumetric inspector* is a virtual box filled with a single scattering homogeneous participating medium. The user can freely position it to analyze strong directional effects, such as caustics and shadows in free space. To capture these phenomena, we rasterize the paths going through the volume and accumulate their throughputs to voxels of a uniform 3D grid. Figure 9.3 shows an example of volume caustics detected in free space.



Figure 9.4: Example edits in the GARAGE scene. Before/after close-ups (right): removing reflections caused by the car, moving sunlight refracted through the windows, transforming a glossy caustic, and altering the mirror reflection. Adapted from [Schmidt et al. 2013].

Evaluation of the Tools. We performed a user study (see [Reiner et al. 2012] for details) that indicates that different inspection tools are required and best suited for different application scenarios and tasks. False-color rendering was helpful where the strength of illumination was examined, giving insufficient insights for more complex lighting. The spherical plot tool performed well when a local directional aspect was examined, but performed poor for all tasks requiring “the big picture”. Light path inspection worked well in many cases and is definitely worth further investigation, with the most difficult task for it being the soft diffuse illumination. Volumetric inspection is especially well suited to comprehend complex caustics, while it also becomes uninformative for soft diffuse illumination.

9.2 Artistic Manipulation of Light Transport

The availability of efficient physically based rendering (PBR) systems with interactive preview has promoted its rapid adoption in the feature-film and gaming industries [Krivánek et al. 2010, McAuley et al. 2012]. Artists’ productivity depends on the flexibility of the light editing tools. Many existing tools either target non-PBR systems or consider only specific PBR effects. Our motivation for light editing is to keep the manipulated illumination as physically plausible as possible, without restricting artistic freedom.

We present *path retargeting*, an artistic light editing tool built on top of physically based light transport, that enables intuitive manipulation of illumination, including effects that result from complex light paths (see Figure 9.4). Path retargeting

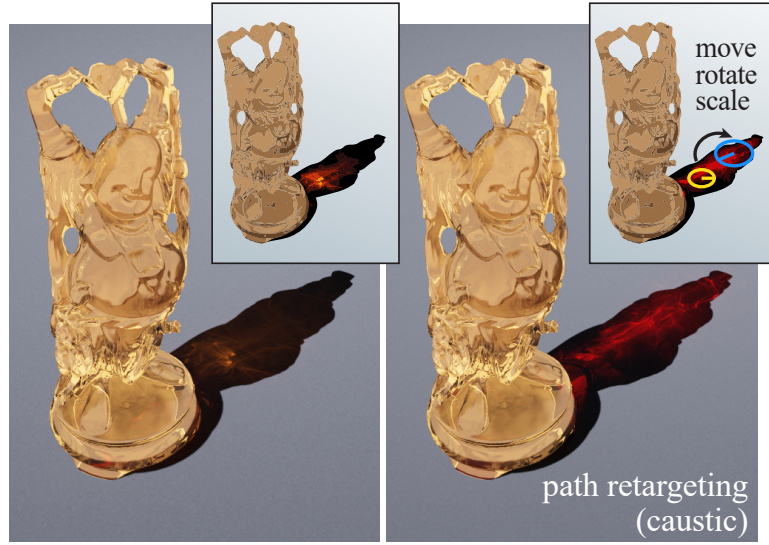


Figure 9.5: Left: original render. Right: using path retargeting to displace light paths forming the caustic. Adapted from [Schmidt et al. 2013].

operates *directly* on light paths and allows the user to select an illumination feature using visualization and selection techniques from Section 9.1.1 and then manipulate the selected paths related to this feature (Figure 9.5). We also discuss how to render manipulated paths using bidirectional light transport algorithms.

9.2.1 Manipulation with Path Retargeting

Path retargeting provides control over lighting features by manipulating on an optional subset of the path space as well as an optional user-defined region(s) of the scene.

First, the transport phenomenon is selected, by filtering the subpaths that will be manipulated using the extended path notation (Section 9.1.1). For example, the user can select a light (e.g., a diffuse indirect illumination LD^RX_p) or an eye subpath (e.g., specular reflection X_pS^RE). If the path selection is empty, all paths are manipulated.

Then, the user can also specify a volumetric region of interest, a *source region*, serving as the origin of the retargeting transformation. The transformation is then an affine mapping defined by placing a *target region* (Figure 9.5) in the scene. The

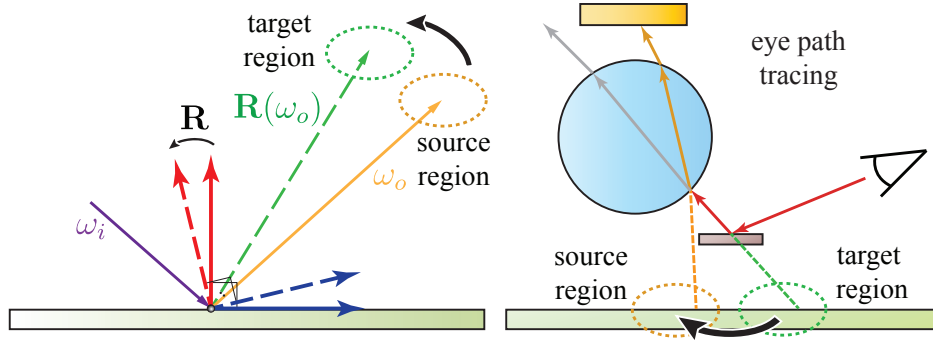


Figure 9.6: Left: path retargeting can be viewed as a transformation of either the incident or exitant shading tangent frame, depending on the manipulation direction and the direction of the path construction method. Right: when constructing a path in a direction opposite to the manipulation direction, at each path edge, we apply the inverse transformation to check for potential manipulations from the opposite direction. Adapted from [Schmidt et al. 2013].

tool redirects path segments, therefore, implicitly propagating the manipulation to the secondary effects, such as inter-reflections and indirect shadows. Both source and target regions are not linked to any scene object, however, their transformations can be keyframed for animation, and optionally linked to the transformation of an object. In addition, we introduce basic appearance modifiers for path throughput, such as intensity scaling and hue editing, in the spirit of [Obert et al. 2008].

9.2.1.1 Robust Bidirectional Manipulation

Retargeting works by redirecting either a light or an importance flow in the light transport simulation based on the selected manipulation. Therefore, it is easy to apply the manipulation during rendering in case if the path construction direction coincides with the direction of the manipulation. However, a path can usually be constructed using multiple techniques, by connecting subpaths of different lengths, therefore, it is not always the case that the direction of path construction would coincide with the manipulation direction. In this case, we have to ensure that all bidirectional path construction techniques can consistently apply the manipulations.

Therefore, we formulate path retargeting as a modifier that acts on the scene surfaces. This allows to seamlessly use path retargeting in various light transport methods such as path tracing, bidirectional path tracing, and Metropolis Light Transport [1997]. Path retargeting effectively transforms *a surface's shading tan-*

gent frame such that an outgoing segment points towards the user-specified target (Figure 9.6, left). This introduces a non-symmetric BRDF [Veach 1998] for bidirectional transport, similarly to modified or bump-mapped shading normals. Given a surface point with normal \mathbf{n} , BSDF f , and a path with incident and outgoing directions \mathbf{i} and \mathbf{o} , path retargeting defines a transformation operator \mathbf{R} of the tangent frame that depends on the source and the target regions, and modifies the BSDF as

$$f'(\mathbf{i} \rightarrow \mathbf{o}) = f(\mathbf{i} \rightarrow \mathbf{R}(\mathbf{o})),$$

and its adjoint is derived similarly to [Veach 1998, Section 5.3.2] as

$$f'^*(\mathbf{i} \rightarrow \mathbf{o}) = f(\mathbf{R}^T(\mathbf{o}) \rightarrow \mathbf{i}) \frac{|\mathbf{R}^T(\mathbf{o}) \cdot \mathbf{n}|}{|\mathbf{o} \cdot \mathbf{n}|}.$$

The adjoint BSDF f'^* is used when the path construction direction does not coincide with the manipulation direction. If the user specifies a source region, the manipulation must be handled using rejection sampling for paths constructed from the direction opposite the manipulation direction. For example, if an eye subpath is constructed for a path that was retargeted from the light direction, we apply the inverse source-target transformation to ensure that the unmodified light subpath would hit the source region (Figure 9.6, right); otherwise, the retargeting modification is rejected.

9.2.2 Example Edits

Figure 9.4 demonstrates multiple sequential edits, here modifications are applied to $LS^T G^R X_p$ and $LS^T X_p$ subsets of the path space (glossy reflection off the car and sunlight through the window), followed by a rotation and scaling of a glossy reflection on the floor, as well as the modification of the reflection in the mirror ($X_p S^R E$ paths).

Figure 9.7 shows light editing using path retargeting in an architectural scene. We applied two local transformations to indirect diffuse illumination ($LD^R X_p$ paths), first “stretching” the color bleeding on the wall (top green helper gizmo) and scaling

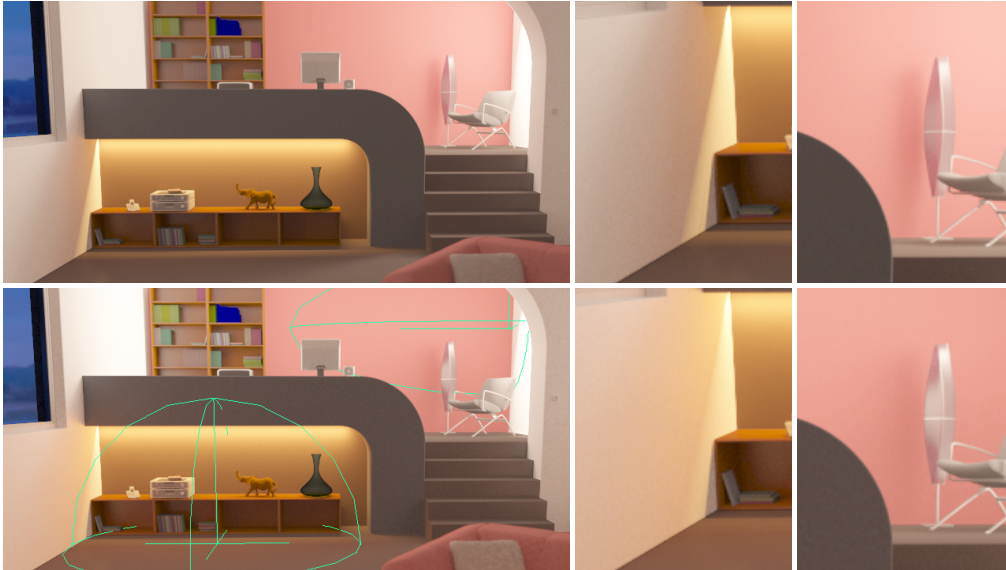


Figure 9.7: Manipulating diffuse GI in the LIVING ROOM scene. Top to bottom: unmodified light transport; indirect diffuse illumination above the shelf and on the back wall are manipulated with path retargeting. Adapted from [Schmidt et al. 2013].

the indirect lighting down the stairs (bottom green helper gizmo).

9.3 Conclusion

We presented an interactive set of tools for intuitive visualization, selection, and manipulation of light transport. Light visualization tools use interactive visualization techniques. They support users in understanding the light transport in virtual scenes. Individual tools are useful and appropriate for different tasks. Operating directly on path-space solutions of the rendering equation enables the manipulation of complex transport phenomena, including secondary shading effects.

Conclusion

The measure of greatness in a scientific idea is the extent to which it stimulates thought and opens up new lines of research.

— PAUL ADRIEN MAURICE DIRAC (1968)

In this thesis, we provide solutions to a spectrum of light transport problems of handling difficult illumination for high-quality global illumination in complex scenarios.

We analyzed kernel estimation methods, such as photon mapping, and an optimal convergence rate for them, which turned out to be asymptotically slower than the convergence rate of unbiased Monte Carlo methods. Moreover, we provided a non-asymptotic analysis and proposed an on-line bandwidth selection method for the kernel bandwidth, which locally adapts to the shape of the light field around the kernel estimation region. Due to the generalization of the kernel estimation to the path integral, the bandwidth selection method supports effects, such as glossy reflections and depth of field, which would otherwise be challenging to incorporate. This allows achieving faster practical renderings, which we demonstrated with progressive photon mapping.

We have also extended kernel estimation to the path space regularization framework that is compatible with advanced integration methods, such as Metropolis light transport. We also provide the criteria for consistent convergence for both Monte Carlo and Markov chain Monte Carlo methods. The proposed selective regularization on arbitrary domains introduces bias only where it is necessary, e.g., for paths that are impossible to sample with unbiased path construction methods. The

regularization of these paths allows rendering scenes with specular materials and point light sources. The required modifications to existing renderers are minimal, making it easy to incorporate in practice.

In addition, we introduced half vector space light transport, a new representation for light paths as well as a new method for computing light transport. This representation allows for an easier theoretical analysis of the path integral, as well as enables practically more numerically robust computation of light transport. When represented in the half vector domain, the path integral can be decomposed into weakly correlated two-dimensional integrals, and thus allows for an easy joint importance sampling as well as for the estimation of spectral properties of the integral. Furthermore, the proposed mutation strategy for Metropolis light transport practically handles long and complex chains of inter-reflections between highly glossy materials by mutating the path directly in the half vector domain. It is also possible to better control image-space stratification of the random walk in Metropolis light transport in this domain.

Finally, we considered the recent transition of multiple industries to physically based rendering, and the related problem of understanding, creating, and artistically adjusting the illumination in this setting. We proposed a practical set of selective visualization tools for light transport that allows the digital content creators to better understand and assess the light distribution in the scene. This is achieved by selecting and visualizing only the phenomena of interest, which allows to provide a clutter-free visualization of the structure and the parameters even for complex illumination features. Our manipulation tool is designed to easily achieve artistic goals and appearance in the context of physically based light transport. We relax the physical constraints by allowing the artists to alter the illumination, however, in a systematic and robust manner by changing selective subspaces of the path space. This allows for consistent and plausible editing, while also enabling the integration with advanced light transport methods, such as bidirectional path construction methods.

Bibliography

- Ament, M., Bergmann, C., and Weiskopf, D. *Refractive radiative transfer equation*. ACM Trans. on Graphics, 33(2):17:1–17:22, 2014. [24](#)
- Arvo, J. *Backward ray tracing*. In SIGGRAPH Course Notes, pages 259–263. 1986. [77](#)
- Ashikhmin, M. and Shirley, P. *An anisotropic Phong BRDF model*. Journal of Graphics Tools, 5(2):25–32, 2000. [188](#)
- Becker, K., Becker, M., and Schwarz, J. H. *String theory and M-theory: A modern introduction*. Cambridge University Press, 2007. [13](#)
- Beckmann, P. and Spizzichino, A. *The scattering of electromagnetic waves from rough surfaces*. International series of monographs on electromagnetic waves. Pergamon Press, 1963. [33](#), [35](#)
- Belcour, L. and Soler, C. *Frequency-based kernel estimation for progressive photon mapping*. In ACM SIGGRAPH Asia Posters. 2011. [85](#)
- Berestetskiy, V., Lifshitz, E., and Pitaevskiy, L. *Quantum Electrodynamics*. Course of theoretical physics. Butterworth-Heinemann, 1982. ISBN 9780750633710. [13](#)
- Bremaud, P. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, 1999. ISBN 0-387-98509-3. [62](#), [141](#), [142](#), [144](#), [181](#)
- Cappé, O., Guillin, A., Marin, J.-M., and Robert, C. P. *Population Monte Carlo*. Journal of Computational and Graphical Statistics, 13(4), 2004. [92](#)
- Case, K. and Zweifel, P. *Linear transport theory*. Addison-Wesley series in nuclear engineering. Addison-Wesley Pub. Co., 1967. [18](#)
- Chandrasekhar, S. *Radiative Transfer*. Dover Publications Inc., 1960. ISBN 0-486-60590-6. [28](#), [48](#)
- Chen, J., Wang, B., and Yong, J.-H. *Improved stochastic progressive photon mapping with Metropolis sampling*. Computer Graphics Forum, 30(4):1205–1213, 2011. [92](#)
- CIE. Commission Internationale de l’Eclairage Proceedings. Rapport technique, Cambridge University Press, Cambridge, 1931. [44](#)

- Cline, D., Talbot, J., and Egbert, P. K. *Energy redistribution path tracing*. ACM Trans. on Graphics (Proc. SIGGRAPH), 24(3):1186–1195, 2005. 92, 149, 173
- Cook, R. L. and Torrance, K. E. *A reflectance model for computer graphics*. Computer Graphics (Proc. SIGGRAPH), 1(1):7–24, 1982. 34
- Cornuet, J.-M., Marin, J.-M., Mira, A., and Robert, C. P. *Adaptive multiple importance sampling*. Scandinavian Journal of Statistics, 39(4):798–812, 2012. 56
- Cortes, C. and Vapnik, V. *Support-vector networks*. Machine Learning, 20(3):273–297, 1995. 69
- Cuypers, T., Haber, T., Bekaert, P., Oh, S. B., and Raskar, R. *Reflectance model for diffraction*. ACM Trans. on Graphics (Proc. SIGGRAPH), 31(5):122:1–122:11, 2012. 31
- Dachsbacher, C., Krivánek, J., Hasan, M., Arbree, A., Walter, B., and Novák, J. *Scalable realistic rendering with many-light methods*. Computer Graphics Forum, DOI: 10.1111/cgf.12256, 2013. 86
- Dobrushin, R. L. *Central limit theorems for non-stationary Markov chains II (Russian)*. Theory of Probability and its Applications, 1(4):356–425, 1956. 61
- Donoho, D., Johnstone, I., and Johnstone, I. M. *Ideal spatial adaptation by wavelet shrinkage*. Biometrika, 81:425–455, 1993. 69, 112
- Fan, S., Chenney, S., and chi Lai, Y. *Metropolis photon sampling with optional user guidance*. In Proc. Eurographics Symposium on Rendering, pages 127–138. 2005. 92
- Feynman, R. and Hibbs, A. *Quantum mechanics and path integrals*. International series in pure and applied physics. McGraw-Hill, 1965. 37
- Georgiev, I., Krivanek, J., Davidovic, T., and Slusallek, P. *Light transport simulation with vertex connection and merging*. ACM Trans. on Graphics (Proc. SIGGRAPH Asia), 31(6):192:1–192:10, 2012. 83, 84, 85, 135, 146
- Gerrard, A. and Burch, J. *Introduction to Matrix Methods in Optics*. A Wiley-Interscience publication. Wiley, 1975. ISBN 9780471296850. 48
- Goral, C. M., Torrance, K. E., Greenberg, D. P., and Battaile, B. *Modeling the interaction of light between diffuse surfaces*. Computer Graphics (Proc. SIGGRAPH), 18(3):213–222, 1984. 47

- Hachisuka, T., Jarosz, W., and Jensen, H. W. *A progressive error estimation framework for photon density estimation*. ACM Trans. on Graphics (Proc. SIGGRAPH Asia), 29(6):144:1–144:12, 2010. 85, 97, 102, 110, 111, 115, 121
- Hachisuka, T. and Jensen, H. W. *Stochastic progressive photon mapping*. ACM Trans. on Graphics (Proc. SIGGRAPH Asia), 28(5):141:1–141:8, 2009. 85, 98, 118, 128, 145
- Hachisuka, T. and Jensen, H. W. *Robust adaptive photon tracing using photon path visibility*. ACM Trans. on Graphics, 30(5):114:1–114:11, 2011. 85, 92, 119, 120, 146
- Hachisuka, T., Ogaki, S., and Jensen, H. W. *Progressive photon mapping*. ACM Trans. on Graphics (Proc. SIGGRAPH Asia), 27(5):130:1–130:8, 2008. 84, 85, 97, 98, 99, 100, 101, 103, 125, 128, 144, 145
- Hachisuka, T., Pantaleoni, J., and Jensen, H. W. *A path space extension for robust light transport simulation*. ACM Trans. on Graphics (Proc. SIGGRAPH Asia), 31(6):191:1–191:10, 2012. 81, 84, 85
- Hall, P. and Patil, P. *On the efficiency of on-line density estimators*. IEEE Transactions on Information Theory, 40(5):1504–1512, 1994. 97, 103, 110, 112, 115
- Hanika, J. *Spectral light transport simulation using a precision-based ray tracing architecture*. Ph.D. thesis, Ulm University, 2011. VTS-ID/7539. 188
- Hanika, J. and Dachsbacher, C. *Efficient monte carlo rendering with realistic lenses*. Computer Graphics Forum (Proc. Eurographics Symposium on Rendering), 33(2):323–332, 2014. 43
- Hanika, J., Droske, M., and Fascione, L. *Manifold next event estimation*. Computer Graphics Forum (Proc. Eurographics Symposium on Rendering), 34(4):87–97, 2015a. 190
- Hanika, J., Kaplanyan, A., and Dachsbacher, C. *Improved half vector space light transport*. Computer Graphics Forum (Proc. Eurographics Symposium on Rendering), 34(4):65–74, 2015b. 191
- Hasan, M., Krivanek, J., Walter, B., and Bala, K. *Virtual spherical lights for many-light rendering of glossy scenes*. ACM Trans. on Graphics (Proc. SIGGRAPH Asia), 28(5):143:1–143:6, 2009. 149

- Hastings, W. K. *Monte Carlo sampling methods using Markov chains and their applications*. Biometrika, 57(1):97–109, 1970. 57
- Heckbert, P. S. *Adaptive radiosity textures for bidirectional ray tracing*. Computer Graphics (Proc. SIGGRAPH), 24(4):145–154, 1990. 127, 198
- Heckbert, P. S. and Winget, J. M. *Finite element methods for global illumination*. Technical Report, Berkeley, CA, USA, 1991. 47
- Hege, H.-C., Höllerer, T., and Stalling, D. *Volume rendering: Mathematical models and algorithmic aspects*. 1996. 28
- Hegland, M. and Anderssen, R. *A mollification framework for improperly posed problems*. Numerical Mathematics, 78:549–575, 1996. 129, 135
- Heneyey, L. G. and Greenstein, J. L. *Diffuse radiation in the galaxy*. Astrophysical Journal, 93:70–83, 1941. 30
- Holzschuch, N., Kaplanyan, A. S., Hanika, J., and Dachsbacher, C. *Estimating local Beckmann roughness for complex BSDFs*. In ACM SIGGRAPH Talks, pages 66:1–66:2. 2016. 190
- Hrennikoff, A. *Solution of problems of elasticity by the framework method*. Journal of Applied Mechanics, 8(4):A619–A715, 1941. 47
- Igehy, H. *Tracing ray differentials*. Proc. SIGGRAPH, pages 179–186, 1999. 184
- Immel, D. S., Cohen, M. F., and Greenberg, D. P. *A radiosity method for non-diffuse environments*. Computer Graphics (Proc. of SIGGRAPH), 20(4):133–142, 1986. 29
- Jakob, W. *Mitsuba renderer*. 2010. URL <http://www.mitsuba-renderer.org>. 150, 166, 181, 186
- Jakob, W. *Light transport on path-space manifolds*. Ph.D. thesis, Cornell University, 2013. 161, 167
- Jakob, W., Arbre, A., Moon, J. T., Bala, K., and Marschner, S. *A radiative transfer framework for rendering materials with anisotropic structure*. ACM Trans. on Graphics (Proc. SIGGRAPH), 29(4):53:1–53:13, 2010. 28, 30

- Jakob, W. and Marschner, S. *Manifold exploration: a Markov chain Monte Carlo technique for rendering scenes with difficult specular transport*. ACM Trans. on Graphics (Proc. SIGGRAPH), 31(4):58:1–58:13, 2012. 89, 128, 133, 156, 162, 166, 174, 177, 187
- Jarosz, W., Nowrouzezahrai, D., Thomas, R., Sloan, P.-P., and Zwicker, M. *Progressive photon beams*. ACM Trans. on Graphics (Proc. SIGGRAPH Asia), 30(6), 2011. 85, 106, 120
- Jensen, H. W. *Global illumination using photon maps*. In Proc. Eurographics Workshop on Rendering, pages 21–30. 1996. 81, 84
- Jensen, H. W. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., 2001. ISBN 1-56881-147-0. 5
- Jensen, H. W. and Christensen, P. H. *Efficient simulation of light transport in scences with participating media using photon maps*. Proc. SIGGRAPH, pages 311–320, 1998. 132
- Jones, M. C., Marron, J. S., and Sheather, S. J. *A brief survey of bandwidth selection for density estimation*. Journal of the American Statistical Association, 91(433):401–407, 1996. 112
- Jones, M. C. and Sheather, S. J. *Using non-stochastic terms to advantage in kernel-based estimation of integrated squared density derivatives*. Statistics and Probability Letters, 11(6):511–514, 1991. 97, 108, 112
- Kajiya, J. T. *The rendering equation*. Computer Graphics (Proc. of SIGGRAPH), pages 143–150, 1986. 3, 29, 73, 76
- Kaplanyan, A. S. and Dachsbacher, C. *Adaptive progressive photon mapping*. ACM Trans. on Graphics, 32(2):16:1–16:13, 2013a. xix, xxiii
- Kaplanyan, A. S. and Dachsbacher, C. *Path space regularization for holistic and robust light transport*. Computer Graphics Forum (Proc. of Eurographics), 32(2):63–72, 2013b. xviii, xxiii
- Kaplanyan, A. S., Hanika, J., and Dachsbacher, C. *The natural-constraint representation of the path space for efficient light transport simulation*. ACM Trans. on Graphics (Proc. SIGGRAPH), 33(4), 2014. xix, xxiv
- Katkovnik, V. and Shmulevich, I. *Kernel density estimation with adaptive varying window size*. Pattern Recognition Letters, 23(14):1641–1648, 2002. 69, 112

- Kelemen, C., Szirmay-Kalos, L., Antal, G., and Csonka, F. *A simple and robust mutation strategy for the Metropolis light transport algorithm*. Computer Graphics Forum, 21(3):531–540, 2002. 90, 91, 145, 189
- Keller, A. *Instant radiosity*. Proc. SIGGRAPH, pages 49–56, 1997. 52, 85
- Keller, A., Fascione, L., Fajardo, M., Georgiev, I., Christensen, P., Hanika, J., Eisenacher, C., and Nichols, G. *The path tracing revolution in the movie industry*. In ACM SIGGRAPH Courses, SIGGRAPH, pages 24:1–24:7. 2015. 4
- Kitaoka, S., Kitamura, Y., and Kishino, F. *Replica exchange light transport*. Computer Graphics Forum, 28(8):2330–2342, 2009. 92
- Knaus, C. and Zwicker, M. *Progressive photon mapping: A probabilistic approach*. ACM Trans. on Graphics, 30(3):25:1–25:13, 2011. 84, 85, 97, 98, 99, 101, 102, 103, 104, 105, 108, 109, 110, 117, 120
- Kolb, C., Mitchell, D., and Hanrahan, P. *A realistic camera model for computer graphics*. pages 317–324. 1995. 43
- Krivánek, J., Fajardo, M., Christensen, P. H., Tabellion, E., Bunnell, M., Larsson, D., and Kaplanyan, A. S. *Global illumination across industries*. In ACM SIGGRAPH Courses. 2010. 202
- Krivánek, J., Georgiev, I., Kaplanyan, A. S., and Cañada, J. *Recent advances in light transport simulation: Theory and practice*. In ACM SIGGRAPH Courses. 2013. 86
- Lafortune, E. P. and Willems, Y. D. *Bi-directional path tracing*. In Proc. of Conference on Computational Graphics and Visualization Techniques, pages 145–153. 1993. 3, 78, 80
- Lafortune, E. P. F., Foo, S.-C., Torrance, K. E., and Greenberg, D. P. *Non-linear approximation of reflectance functions*. In Proc. SIGGRAPH, pages 117–126. 1997. 31
- Lehtinen, J., Karras, T., Laine, S., Aittala, M., Durand, F., and Aila, T. *Gradient-domain Metropolis light transport*. ACM Trans. on Graphics (Proc. SIGGRAPH), 32(4):95:1–95:12, 2013. 45, 92
- Lessig, C., Fiume, E., and Desbrun, M. *On the mathematical formulation of radiance*. Phys. Lett., 2012. 29, 31

- Maxwell, J. C. *A Dynamical Theory of the Electromagnetic Field*. Philosophical Transactions of the Royal Society of London, 155:459–512, 1865. 14
- McAuley, S., Hill, S., Hoffman, N., Gotanda, Y., Smits, B., Burley, B., and Martinez, A. *Practical physically-based shading in film and game production*. In ACM SIGGRAPH Courses. 2012. 202
- Meng, J., Simon, F., Hanika, J., and Dachsbacher, C. *Physically meaningful rendering using tristimulus colours*. Computer Graphics Forum (Proc. Eurographics Symposium on Rendering), 34(4):31–40, 2015. 44
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. *Equation of state calculations by fast computing machines*. The Journal of Chemical Physics, 21(6):1087–1092, 1953. 57, 141
- Metropolis, N. and Ulam, S. *The Monte Carlo method*. Journal of the American statistical Association, 44(247):335–341, 1949. 50
- Mitchell, D. and Hanrahan, P. *Illumination from curved reflectors*. Computer Graphics (Proc. SIGGRAPH), pages 283–291, 1992. 107
- Mitra, D., Romeo, F., and Sangiovanni-Vincentelli, A. *Convergence and finite-time behavior of simulated annealing*. Advances in Applied Probability, 18(3):747–771, 1986. 142
- Neyret, F. *Modeling animating and rendering complex scenes using volumetric textures*. IEEE Transactions on Visualization and Computer Graphics, 4(1):55 – 70, 1998. 35
- Ngerng, M. H. *Recursive nonparametric estimation of local first derivative under dependence conditions*. Communications in Statistics – Theory and Methods, 40(7):1159–1168, 2011. 114
- Obert, J., Krivánek, J., Pellacini, F., Sýkora, D., and Pattanaik, S. N. *iCheat: A representation for artistic control of indirect cinematic lighting*. Computer Graphics Forum (Proc. Eurographics Symposium on Rendering), 27(4):1217–1223, 2008. 204
- Park, B. U. and Marron, J. S. *Comparison of data-driven bandwidth selectors*. Journal of the American Statistical Association, 85(409):66–72, 1990. 69

- Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., and Stich, M. *OptiX: a general purpose ray tracing engine*. ACM Trans. on Graphics (Proc. SIGGRAPH), 29(4):66:1–66:13, 2010. 118
- Parzen, E. *On estimation of a probability density function and mode*. The Annals of Mathematical Statistics, 33(3):1065–1076, 1962. 66, 69
- Perlin, K. *Improving noise*. ACM Trans. on Graphics, 21(3):681–682, 2002. 102
- Peskun, P. H. *Optimum Monte-Carlo sampling using Markov chains*. Biometrika, 60(3):607–612, 1973. 59
- Pharr, M. and Humphreys, G. *Physically Based Rendering, Second Edition: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., 2nd edition, 2010. 4, 46, 72, 82
- Phong, B. T. *Illumination for computer generated pictures*. Computer Graphics (Proc. SIGGRAPH), 18(6):311–317, 1975. 31
- Reif, J. H., Tygar, J. D., and Yoshida, A. *Computability and complexity of ray tracing*. Discrete Computational Geometry, 11(1):265–288, 1994. 73, 108
- Reiner, T., Kaplanyan, A. S., Reinhard, M., and Dachsbacher, C. *Selective inspection and interactive visualization of light transport in virtual scenes*. Computer Graphics Forum (Proc. of Eurographics), 31(2):711–718, 2012. xx, xxv, 199, 200, 201, 202
- Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J. *Photographic tone reproduction for digital images*. ACM Trans. on Graphics (Proc. SIGGRAPH), 21(3):267–276, 2002. 45
- Ritschel, T., Smith, K., Ihrke, M., Grosch, T., Myszkowski, K., and Seidel, H.-P. *3D Unsharp Masking for Scene Coherent Enhancement*. ACM Trans. on Graphics (Proc. SIGGRAPH), 27(3):90:1–90:8, 2008. 45
- Robert, C. P and Casella, G. *Monte Carlo Statistical Methods*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396. 51
- Rudemo, M. *Empirical choice of histograms and kernel density estimators*. Scandinavian Journal of Statistics, 9(2):65–78, 1982. 69
- Rusinkiewicz, S. *A new change of variables for efficient BRDF representation*. In Proc. Eurographics Workshop on Rendering, pages 11–22. 1998. 31, 156, 166

- Schmidt, T.-W., Novak, J., Meng, J., Kaplanyan, A. S., Reiner, T., Nowrouzezahrai, D., and Dachsbacher, C. *Path-space manipulation of physically-based light transport*. ACM Trans. on Graphics (Proc. SIGGRAPH), 32(4):129:1–129:11, 2013. [xx](#), [xxv](#), [202](#), [203](#), [204](#), [206](#)
- Sheather, S. J. and Jones, M. C. *A reliable data-based bandwidth selection method for kernel density estimation*. Journal of the Royal Statistical Society. Series B, 53(3):683–690, 1991. [69](#), [112](#)
- Shinya, M., Takahashi, T., and Naito, S. *Principles and applications of pencil tracing*. Computer Graphics (Proc. SIGGRAPH), pages 45–54, 1987. [49](#), [156](#), [162](#), [167](#)
- Shirley, P., Wade, B., Hubbard, P. M., Zareski, D., Walter, B., and Greenberg, D. P. *Global illumination via density-estimation*. In Proc. Eurographics Workshop on Rendering, pages 219–230. 1995. [5](#), [84](#)
- Silva, S., Sousa Santos, B., and Madeira, J. *Using color in visualization: A survey*. Computers and Graphics, 35(2):320–333, 2011. [200](#)
- Silverman, B. *Density estimation for statistics and data analysis*. Monographs on statistics and applied probability. Chapman and Hall, 1986. ISBN 9780412246203. [66](#), [68](#), [69](#), [81](#), [102](#), [107](#), [108](#), [111](#), [115](#)
- Smith, B. *Geometrical shadowing of a random rough surface*. IEEE Transactions on Antennas and Propagation, 15(5):668–671, 1967. [33](#), [35](#)
- Sommerfeld, A. and Runge, J. *Anwendung der Vektorrechnung auf die Grundlagen der geometrischen Optik*. Annalen der Physik, 340:277–298, 1911. [32](#), [156](#), [158](#)
- Spivak, M. *Calculus on manifolds*. Addison-Wesley, 1965. ISBN 978-0-81-334612-0. [177](#)
- Subr, K. and Kautz, J. *Fourier analysis of stochastic sampling strategies for assessing bias and variance in integration*. ACM Trans. on Graphics (Proc. SIGGRAPH), 32(4):128:1–128:12, 2013. [173](#), [186](#)
- Tornberg, A.-K. *Multi-dimensional quadrature of singular and discontinuous functions*. BIT Numerical Mathematics, 42(3):644–669, 2002. [136](#)
- Torrance, K. E. and Sparrow, E. M. *Theory for off-specular reflection from roughened surfaces*. Journal of the Optical Society of America, 57(9):1105–1112, 1967. [34](#)

- Trowbridge, T. S. and Reitz, K. P. *Average irregularity representation of a rough surface for ray reflection*. Journal of the Optical Society of America, 65(5):531–536, 1975. 35
- van Eeden, C. *Mean integrated squared error of kernel estimators when the density and its derivative are not necessarily continuous*. Annals of the Institute of Statistical Mathematics, 37(1):461–472, 1985. 114
- Veach, E. *Robust Monte Carlo methods for light transport simulation*. Ph.D. thesis, Stanford University, 1998. AAI9837162. xvii, xviii, xxi, xxii, 3, 24, 32, 36, 37, 40, 65, 76, 80, 127, 128, 143, 145, 158, 171, 176, 184, 205
- Veach, E. and Guibas, L. *Bidirectional estimators for light transport*. In Proc. Eurographics Workshop on Rendering, pages 147–162. 1994. 3, 78, 80, 134
- Veach, E. and Guibas, L. J. *Optimally combining sampling techniques for Monte Carlo rendering*. Proc. SIGGRAPH, pages 419–428, 1995. 55, 56
- Veach, E. and Guibas, L. J. *Metropolis light transport*. Proc. SIGGRAPH, pages 65–76, 1997. 62, 88, 92, 146, 171, 204
- Wald, I. *Photorealistic Rendering using the Photon Map*. Ph.D. thesis, Master’s thesis, Numerical Algorithms Group, University of Kaiserslautern, Germany, 1999. 84
- Walter, B., Fernandez, S., Arbree, A., Bala, K., Donikian, M., and Greenberg, D. P. *Lightcuts: a scalable approach to illumination*. ACM Trans. on Graphics (Proc. SIGGRAPH), 24(3):1098–1107, 2005. 86
- Walter, B., Khungurn, P., and Bala, K. *Bidirectional lightcuts*. ACM Trans. on Graphics (Proc. SIGGRAPH), 31(4):59:1–59:11, 2012. 86, 166, 181
- Walter, B., Marschner, S., Li, H., and Torrance, K. *Microfacet models for refraction through rough surfaces*. In Proc. Eurographics Symposium on Rendering, pages 195–206. 2007. 32, 33, 34, 35, 36, 156, 158, 159, 163
- Wand, M. P. and Jones, M. C. *Multivariate plug-in bandwidth selection*. Computational Statistics, 9:97–116, 1994. 97, 105, 107, 112, 113, 115, 120
- Wandell, B. *Foundations of Vision*. Sinauer Associates, 1995. ISBN 9780878938537. 43

Williamson, S. and Cummins, H. *Light and Color in Nature and Art*. Wiley, 1983. ISBN 9780471083740. 44

Wolverton, C. and Wagner, T. *Recursive estimates of probability densities*. IEEE Transactions on Systems Science and Cybernetics, 5(3):246–247, 1969. 103

Yamato, H. *Sequential estimation of a continuous probability density function and mode*. Bulletin of Mathematical Statistics, 14:1–12, 1971. 103

Yellott, J. I., Wandell, B. A., and Cornsweet, T. N. *The Beginnings of Visual Perception: The Retinal Image and its Initial Encoding. Appendix: Fourier Transforms and Shift-Invariant Linear Operators*, pages 257–316. John Wiley and Sons, Inc., 1980. 45

Index

ABCD matrix, 48, 156, 162

bidirectional path tracing, 78

- camera subpath, 78

- eye subpath, 78

- light subpath, 78

- techniques, 79

bidirectional scattering distribution function, 29, 30

- Lambert cosine law, 16

- Lambertian reflectance model, 31

- measured data, 31

- parametric models, 31

- perfect specular reflection, 32

cumulative distribution function, 50, 54

curse of dimensionality, 47, 50, 51, 57, 69, 120

energy redistribution path tracing, 92, 149

Fermat's principle, 16, 155

fluorescence, 18

Fresnel effect, 19, 32

geometric term, 38, 157

- generalized geometric term, 162, 166

- Nusselt analog, 39

- one-sided geometric term, 74, 158

global illumination, 3

half vector, 158

- generalized half vector, 158

half vector space light transport, 155

half vector space, 160

- space of natural constraints, 160

- submanifold, 160

integration

- discrete ordinates methods, 48

- finite element, 47

kernel estimation, 66

- adaptive kernel estimators, 69

- asymptotic analysis, 104

- asymptotic mean squared error, 105

- bandwidth selection, 108

- bias, 67, 84

- bias-variance trade-off, 67

- consistent, 103

- cross-validation method, 69

- intersection of confidence intervals, 69

- kernel bandwidth, 66, 102, 129

- mean squared error, 67

- plug-in bandwidth selection, 69, 112

- recursive estimation, 103

- smoothing kernel, 66

- wavelet shrinkage, 69

Kramers-Kronig relations, 33

law of reflection, 32

light field, 19

light tracing, 77

light transport

- one-velocity transport, 18

- spectral rendering, 18

- temporal dimension, 18

light visualization tools, 200

- false-color rendering, 200

- light path inspection, 201

- spherical plot, 201
- volumetric inspector, 201
- many-light methods, 85
 - instant radiosity, 85
 - virtual point lights, 85
 - virtual spherical lights, 149
- Markov chain
 - acceptance probability, 58
 - detailed balance, 58
 - Dobrushin's ergodic coefficient, 61
 - equilibrium, 57
 - Harris-recurrent, 60
 - Markov kernel, 60
 - n-step transition kernel, 60
 - non-homogeneous, 61
 - reversible, 58
 - state space, 57
 - stationary distribution, 57
 - weak ergodicity, 61
- Markov chain Monte Carlo, 57
- measure theory
 - area measure, 22
 - measurable space, 21
 - measure, 21
 - probability measure, 21
 - product measure, 21, 139
 - projected area measure, 22
 - projected solid angle, 23, 24, 157
 - Radon-Nikodym derivative, 22
 - solid angle measure, 23
- Metropolis light transport, 62, 158
 - bidirectional mutation, 88
 - caustic perturbation, 87
 - correlated paths, 86
 - gradient-domain Metropolis light transport, 45, 92
 - large step, 91
 - lens mutation, 92
 - lens perturbation, 87
 - manifold exploration, 89, 133, 156
 - manifold walk, 89
 - multichain perturbation, 87
 - mutation strategy, 62
 - non-reversible walk, 178
 - path space perturbations, 86
 - primary sample space, 90
 - seeding path, 86
 - small step, 91
- Metropolis-Hastings algorithm, 57
 - acceptance probability, 58
 - normalization constant, 59, 64
 - posterior distribution, 59
 - proposal distribution, 58
 - start-up bias, 59, 64
- microfacet theory, 33
 - Beckmann distribution, 35
 - Cook-Torrance BSDF, 34
 - microfacet models, 31
 - normal distribution function, 33
 - shadowing-masking term, 34
 - slope domain, 34
 - Trowbridge-Reitz distribution, 35
- mollifier, 129
 - mollification bandwidth, 129
- Monte Carlo, 50
 - estimator, 51
 - path throughput, 51
- optics

- diffraction, 15
 - geometrical optics, 13, 15
 - interference, 15
 - paraxial optics, 49
 - physical optics, 15
 - polarization, 15
 - quantum electrodynamics, 12
 - ray optics, 15
 - visible light spectrum, 11
 - wave optics, 13, 15
- parallel-plane domain, 34
- path
- enriched path vertices, 199
 - non-samplable, 133
 - undecidable paths, 107
- path integral, 40
- global path construction, 72
 - light path, 71
 - local path construction, 72
 - measurement contribution function, 40
 - natural constraint, 155
 - path probability density, 72
 - path space, 40, 129
- path retargeting, 202, 203
- source region, 203
 - target region, 203
- path tracing, 73
- explicit light sampling, 76
 - implicit light sampling, 74
 - next event estimation, 76
- pencil tracing, 156, 162
- perception
- color temperature, 12
 - contrast sensitivity function, 45
 - Weber-Fechner law, 44
- phase function, 27, 29
- Henyey-Greenstein phase function, 30
 - phase angle, 27
- phosphorescence, 18
- photon mapping, 81, 84
- Laplacian of the pixel measurement, 113
 - photon map, 81
 - progressive photon mapping, 85, 97
 - vertex-Laplacian, 113
- physically based rendering, 4, 202
- pinhole camera, 43
- plenoptic illumination function, 19
- population Monte Carlo, 92
- probability density function, 21
- probability mass function, 89
- radiative transfer, 26
- Beer-Lambert law, 28
 - elastic scattering, 18
 - extinction coefficient, 27, 32
 - heat transfer, 57
 - inelastic scattering, 18, 28
 - mean free path, 27
 - radiative transfer equation, 28
 - transmittance, 28
- radiometry
- basic radiance, 24, 36
 - energy, 24
 - flux, 25
 - irradiance, 25
 - radiance, 24

- radiant emittance, 25
- radiant energy, 25
- radiant energy density, 25
- radiant intensity, 25
- radiosity, 47
- ray tracing, 72
 - acceleration structure, 72
- rendering equation, 3, 28
- replica exchange, 92
- responsivity, 38
- sampling, 55, 135
 - adaptive sampling, 46
 - balance heuristic, 55
 - combined sample density, 55
 - deterministic mixture model, 55
 - importance sampling, 53
 - joint importance sampling, 54
 - multiple importance sampling, 54, 134, 158
 - perfect importance sampling, 53
 - power heuristic, 56
 - proposal probability density, 53
 - zero-variance importance sampling, 53
- shading normal, 37
- shutter time, 43
- simulated annealing, 140
- spherical coordinates, 23
- thermodynamic equilibrium, 18
- tone mapping, 45
- transfer matrix, 167
- visibility term, 38

