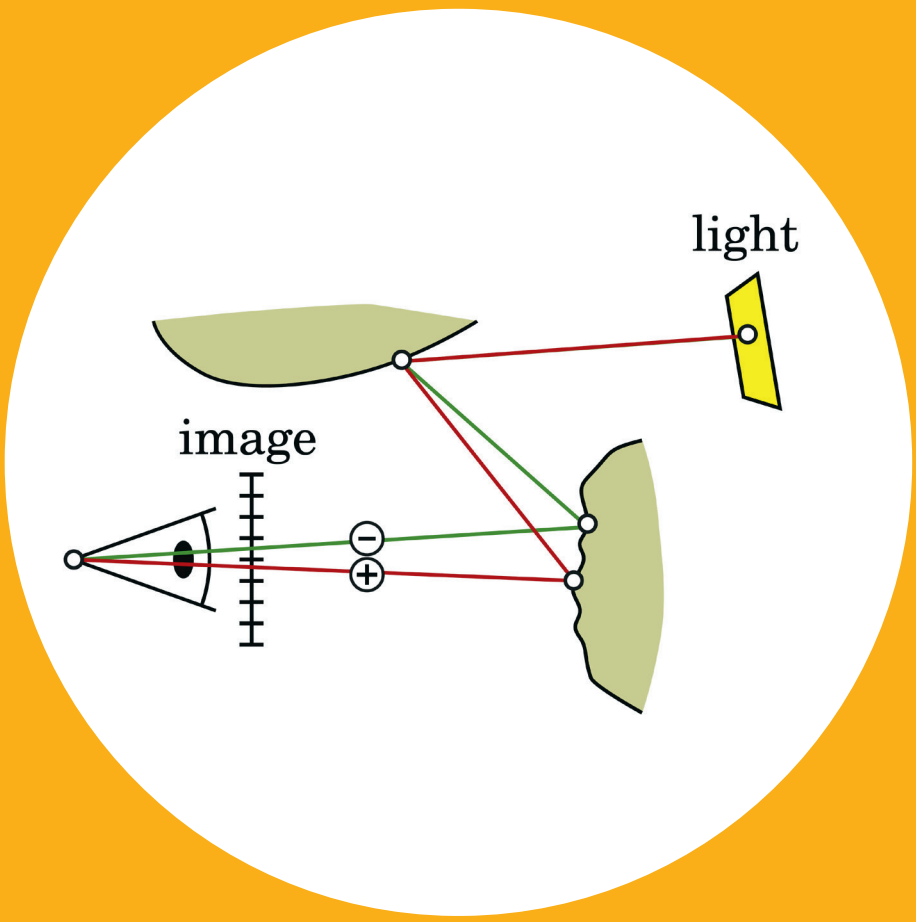
Department of Computer Science

Gradient-Domain Methods for Realistic Image Synthesis

Markus Kettunen



Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 5/2020

Gradient-Domain Methods for Realistic Image Synthesis

Markus Kettunen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall TU2 of the school on 12 February 2020 at 12 noon.

Aalto University
School of Science
Department of Computer Science

Supervising professor
Prof. Jaakko Lehtinen

Thesis advisor
Prof. Jaakko Lehtinen

Preliminary examiners
Prof. Carsten Dachsbacher, Karlsruhe Institute of Technology, Germany
Prof. Toshiya Hachisuka, The University of Tokyo, Japan

Opponent
Dr. George Drettakis, INRIA, France

Aalto University publication series
DOCTORAL DISSERTATIONS 5/2020

ISBN 978-952-60-8907-2 (printed)
ISBN 978-952-60-8908-9 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-8908-9

Unigrafia Oy
Helsinki 2020

Finland

NORDIC SWAN ECOLABEL
Printed matter
4041-0619

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Markus Kettunen

Name of the doctoral dissertation
Gradient-Domain Methods for Realistic Image Synthesis

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 5/2020

Field of research Computer Graphics

Manuscript submitted 2 December 2019 **Date of the defence** 12 February 2020

Permission for public defence granted (date) 13 December 2019 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

Realistic image synthesis, i.e., the creation of photographs of virtual environments by numerical simulation of light, is present in the lives of most people in the form of movies and advertising. Its results are often almost indistinguishable from reality, but the methods are computationally intensive and often require supercomputers. This thesis presents new methods for realistic image synthesis with the potential to reduce rendering times, costs, and environmental footprint.

The rendering equation describes the transportation of light as it repeatedly scatters around a virtual environment. The light arriving at a virtual sensor produces a virtual photograph. Typical solutions evaluate the pixel colors by randomly sampling numerous paths by which light can reach the sensor and evaluate their average contribution.

The methods presented in this thesis work in the gradient-domain: In addition to sampling the colors, they directly evaluate the finite differences between adjacent pixels and reconstruct the image as an integration problem. This allows the solution to utilize the similarity of the light transport behind close-by pixels. Gradient-domain rendering was recently proposed in the context of Markov Chain Monte Carlo, but the feasibility of gradient-domain rendering in the more common traditional Monte Carlo context was left unanswered. This thesis presents four new gradient-domain Monte Carlo rendering methods.

The first two methods evaluate the image gradients by constructing highly correlated pairs of paths for adjacent pixels. Subtracting their contributions produces typically lower-noise gradients since correlation decreases variance in subtraction. A screened Poisson equation combines the high frequencies captured by the gradient samples with the low frequencies of the color samples. This results in images that typically have less high-frequency noise.

The third method is an extension to animation. It evaluates gradients also in the time-dimension by rendering each image in two parts, with random seeds shared between the previous and the next frame. Subtracting the images rendered with the same random seed produces the time component of the gradients. A spatiotemporal reconstruction results in decreased flickering in animation.

The last method is a deep convolutional neural network that replaces the screened Poisson reconstruction. The network is trained to map the sampled colors and gradients to noise-free reconstructions by minimizing a neural perceptual image distance function. This improves the sharpness of the reconstructions. The gradient inputs improve especially the quality of shadows.

Keywords realistic image synthesis, gradient-domain rendering, ray tracing

ISBN (printed) 978-952-60-8907-2 **ISBN (pdf)** 978-952-60-8908-9

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2020

Pages 118 **urn** http://urn.fi/URN:ISBN:978-952-60-8908-9

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Markus Kettunen

Väitöskirjan nimi
Gradienttiavaruuden menetelmät realistiseen kuvasynteesiin

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 5/2020

Tutkimusala Tietokonegrafiikka

Käsikirjoituksen pvm 02.12.2019 **Väitöspäivä** 12.02.2020

Väittelyluvan myöntämispäivä 13.12.2019 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Realistinen kuvasynteesi eli realististen virtuaalisten valokuvien luominen laskennallisesti on läsnä monien ihmisten arjessa esimerkiksi elokuvien ja mainosten muodossa. Sen tuloksia on usein lähes mahdoton erottaa aidoista, mutta menetelmät ovat raskaita ja vaativat usein supertietokoneen käyttöä. Tämä väitöskirja esittelee uusia laskennallisia menetelmiä, jotka voisivat joissakin tapauksissa pienentää realistisen kuvansynteesin laskenta-aikaa, kuluja ja ympäristöjalanjälkeä.

Valaistusyhtälö (rendering equation) kuvaa valon kulkua sen sirotessa pintojen välillä virtuaalisessa ympäristössä. Virtuaalinen valokuva muodostuu valon saavuttaessa virtuaalisen kameran kennon. Tyypilliset ratkaisut muodostavat satunnaisia polkuja, joita pitkin valo voisi saapua kennolle, ja laskevat kuvapisteiden värin polkuja pitkin saapuvan valomäärän keskiarvona.

Tämän väitöskirjan menetelmät toimivat gradienttiavaruudessa: kuvapisteiden värin lisäksi ne laskevat kuvapisteiden välisiä erotuksia ja muodostavat kuvan integrointiongelman ratkaisuna. Ratkaisun teho perustuu vierekkäisiin pikseleihin vaikuttavien valopolkujen samankaltaisuuteen. Gradienttiavaruuden kuvasynteesi esitettiin aluksi Markovin ketju Monte Carlo -yhteydessä, mutta gradienttiavaruuden hyödyllisyys tavallisessa Monte Carlo -kontekstissa jäi avoimeksi. Tämä väitöskirja esittelee neljä tällaista realistisen kuvasynteesin menetelmää.

Kaksi ensimmäistä menetelmää arvioivat kuvan gradientteja muodostamalla vierekkäisten kuvapisteiden välille samankaltaisten polkujen pareja. Näin lasketut gradientit kohisevat usein vähemmän, sillä korrelaatio pienentää vähennyslaskun varianssia. Vaimennetun (screened) Poissonin yhtälön ratkaisu yhdistää gradienttien korkean taajuuden informaation värinäytteiden mataliin taajuuksiin. Seurauksena lopullisessa kuvassa on vähemmän korkean taajuuden kohinaa.

Kolmas, animaatioihin keskittyvä menetelmä laajentaa gradienttien laskemisen aikaulottuvuuteen muodostamalla kuvat kahdessa osassa. Ensimmäinen osa jakaa satunnaisluvut animaation aikaisemman kuvan kanssa ja jälkimmäinen seuraavan kuvan kanssa. Samoilla satunnaisluvuilla muodostettujen kuvien erotus tuottaa gradientin aikakomponentin. Aikaulottuvuuteen laajennettu rekonstruktio tuottaa usein käyttökelpoisen lopputuloksen lyhyemmässä ajassa.

Neljäs menetelmä on syviin neuroverkkoihin perustuva rekonstruktio. Neuroverkko opetetaan minimoimaan mallikuvan ja verkon ulostulon välinen neuroverkkoihin perustuva etäisyysmitta. Tämän etäisyysfunktion käyttäminen parantaa rekonstruktioiden terävyyttä. Gradienttien käyttäminen neuroverkon syötteessä parantaa erityisesti lopullisen kuvan varjojen laatua.

Avainsanat realistinen kuvasynteesi, gradienttiavaruuden renderöinti, säteenseuranta

ISBN (painettu) 978-952-60-8907-2 **ISBN (pdf)** 978-952-60-8908-9

ISSN (painettu) 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2020

Sivumäärä 118 **urn** http://urn.fi/URN:ISBN:978-952-60-8908-9

Preface

It all started from a careless slip to a friend, “I would like to do computer graphics research for a living.”

The little bird did its job and I was soon contacted by Dr. Samuli Laine from NVIDIA Research. Having read my master’s thesis on the rendering equation, he introduced me to his colleague Dr. Jaakko Lehtinen, who was about to start as a professor at Aalto University. Six months later, having finally given in to the temptation, I found myself reading research articles in professor Lehtinen’s newly founded research group.

Time passed and I grew as a researcher. I put in the hours, and ideas started to take shape. The most important results are now compiled into the book that you are holding, and that book is precious to me: it contains the joys and sorrows of my time at Aalto, written on its pages in invisible ink, only to be experienced with the heart.

I thank professor Jaakko Lehtinen for the countless hours he has invested in supervising my research and helping me grow as a researcher and as a person. I also thank professors Matthias Zwicker and Frédo Durand and my colleagues Miika Aittala, Marco Manzi and Erik Härkönen for co-authoring the publications in this dissertation.

I would also like to show my gratitude to Dr. George Drettakis for accepting to be my opponent, and professors Carsten Dachsbacher and Toshiya Hachisuka for pre-examining this dissertation and providing invaluable feedback. I also thank the Academy of Finland and the HICT doctoral education network for funding my research, and the Aalto Science IT project for providing the computing cluster that made this dissertation possible.

Above all, I thank my wife Henna for her limitless support in making this dream, which started from a careless slip to a friend, come true.

Espoo, December 19, 2019,

Markus Kettunen

Contents

Preface **7**

Contents **9**

List of Publications **11**

Author's Contribution **13**

List of Figures **15**

Abbreviations **17**

1. Introduction **19**
- 1.1 Light Simulation 20
- 1.2 Gradient-Domain Rendering 22
- 1.3 Research Gap 23
- 1.4 Objectives and Scope 23
- 1.5 Research Process 25
- 1.6 Summary 27

2. Theoretical Foundation **29**
- 2.1 Geometrical Optics 29
- 2.2 Rendering Equation 30
- 2.3 Path Tracing 31
- 2.4 Bidirectional Path Tracing 33
- 2.5 Metropolis Light Transport 33
- 2.6 Gradient-Domain Metropolis Light Transport 34
- 2.7 Neural Networks 37
- 2.8 U-Net 38
- 2.9 Reconstruction and Denoising 39
- 2.10 Summary 40

3. Publication Summaries **43**

3.1 Gradient-Domain Path Tracing 43
3.2 Gradient-Domain Bidirectional Path Tracing 45
3.3 Temporal Gradient-Domain Path Tracing 46
3.4 Deep Convolutional Reconstruction for Gradient-Domain Rendering 46
3.5 Summary 47

4. Discussion **49**
4.1 Gradient-Domain Rendering as Path Space Denoising . . 49
4.2 Recent Research by Others 50
4.3 Recommendations for Future Research 50
4.4 Practical Implications 54
4.5 Reliability and Validity 54

References **57**

Errata **63**

Publications **65**

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, Matthias Zwicker. Gradient-Domain Path Tracing. *ACM Transactions on Graphics (TOG)*, Volume 34, Issue 4, Article 123, August 2015.

II Marco Manzi, Markus Kettunen, Miika Aittala, Jaakko Lehtinen, Frédo Durand, Matthias Zwicker. Gradient-Domain Bidirectional Path Tracing. In *Eurographics Symposium on Rendering – Experimental Ideas & Implementations*, June 2015.

III Marco Manzi, Markus Kettunen, Frédo Durand, Matthias Zwicker, Jaakko Lehtinen. Temporal Gradient-Domain Path Tracing. *ACM Transactions on Graphics (TOG)*, Volume 35, Issue 6, Article 246, November 2016.

IV Markus Kettunen, Erik Härkönen, Jaakko Lehtinen. Deep Convolutional Reconstruction For Gradient-Domain Rendering. *ACM Transactions on Graphics (TOG)*, Volume 38, Issue 4, Article 126, July 2019.

Author's Contribution

Publication I: "Gradient-Domain Path Tracing"

The author designed and implemented the Gradient-Domain Path Tracing algorithm. The author also ran most of the experiments and helped with the writing.

Publication II: "Gradient-Domain Bidirectional Path Tracing"

The author participated in deriving and implementing the method and running the experiments.

Publication III: "Temporal Gradient-Domain Path Tracing"

Shared first authorship with Marco Manzi. The author implemented the base version of the method and helped with the motion vector and adaptive sampling extensions. The author ran and implemented many of the experiments and participated in the writing. The author invented the main principle behind the method, sharing random numbers between the next and previous video frames to sample for change in time.

Publication IV: "Deep Convolutional Reconstruction For Gradient-Domain Rendering"

The author designed and implemented the method described in the paper and ran most of the experiments. The author wrote the manuscript draft and made all the figures. The author also polished the final text of the paper together with the co-authors.

List of Figures

1.1 **A virtual photograph simulated with Path Tracing.** From short film Spring. 19

1.2 **Decomposition of lighting.** Full lighting is attained by summing light from directly visible emitters, direct light, and all orders of indirect light. 20

1.3 **Photograph of a glass with caustics.** The glass refracts light into complex caustic patterns on the table. . . 21

1.4 **Research Gap.** The intersection of gradient-domain rendering (bottom row) and Markov Chain Monte Carlo (left column) results in large benefits in the Gradient-Domain Metropolis Light Transport rendering algorithm (bottom left). 23

1.5 **Relations of the publications.** Publication PI extends gradient-domain rendering from Markov Chain Monte Carlo to standard Monte Carlo. 26

2.1 **Shift mapping.** Gradient-Domain Metropolis Light Transport evaluates gradients by shifting base paths (green) into similar offset paths (red) in the adjacent pixels. . . . 36

2.2 **Auxiliary inputs.** Example of a Monte Carlo sampled image (“Color”) and the albedo, depth and normal auxiliary inputs which are often used to guide in the reconstruction. 40

3.1 **Frequency-domain analysis of gradient-domain rendering.** Gradients (top left) capture the high frequencies of the image with little error but do not contain much useful information about the low frequencies (notice the noise singularity near the zero frequency). 44

3.2 **Subpixel-scale geometric detail.** *Left*: A small amount of subpixel-scale geometric detail does not prevent constructing useful offset paths. 45

Abbreviations

BDPT Bidirectional Path Tracing

BSDF Bidirectional Scattering Distribution Function

G-BDPT Gradient-Domain Bidirectional Path Tracing

G-PT Gradient-Domain Path Tracing

G-MLT Gradient-Domain Metropolis Light Transport

MC Monte Carlo

MCMC Markov Chain Monte Carlo

MIS Multiple Importance Sampling

MLT Metropolis Light Transport

MSE Mean Squared Error

PSSMLT Primary Sample Space Metropolis Light Transport

PT Path Tracing

RelMSE Relative Mean Squared Error

ReLU Rectified Linear Unit

SPP Samples Per Pixel

SVD Singular Value Decomposition

TGPT Temporal Gradient-Domain Path Tracing

1. Introduction

Realistic image synthesis – computationally producing realistic-looking virtual photographs of a virtual environment – is a core subject in computer graphics. Its products are consumed daily by most people in the form of movies and advertising (Figure 1.1). This thesis deals with the creation of virtual photographs by computational simulation of light. More specifically, this thesis describes novel synthesis methods that require less computation, which can potentially translate into cheaper production costs, improved image quality, and smaller environmental footprint.

Realistic image synthesis can be roughly divided into two categories: real-time rendering and offline rendering. Real-time methods are used when images need to be shown to the user quickly – typically in under 17 milliseconds – as a response to user input in applications such as games and virtual reality. To achieve such a short rendering time, real-time methods often resort to case-specific approximations and precomputation of content with offline methods. As such, the methods presented in this thesis do not target real-time due to the strict time constraints, but this may eventually change with the development of more computationally capable hardware.

Figure 1.1. A virtual photograph simulated with Path Tracing [38]. From short film Spring. © Blender Foundation | cloud.blender.org/spring. Licensed under CC BY 4.0 (`https://creativecommons.org/licenses/by/4.0/`).

When there is no requirement to react to user input in real-time, then more accurate and general, but also slower, rendering methods can be used. In movie and advertisement companies the rendering times of a single frame may range from hours to hundreds or even thousands of processor hours per frame, depending on the company. Feature films may consist of hundreds of thousands of frames, and typically require the capacity of expensive supercomputers to render. This uses a lot of electricity, which directly translates into high production and environmental cost. A more efficient rendering method – one that achieves equivalent image quality in a shorter time – directly translates into savings in production and environmental costs. Research on more efficient methods for realistic image synthesis, which this thesis is about, is thus not only a pursuit of scientific curiosity but also of strong practical interest.

1.1 Light Simulation

A higher computation budget enables a more realistic simulation of light by algorithms based on tracking paths of photons (or light rays) by a process called ray tracing. In a very simplified view, virtual light sources emit virtual light particles (photons) that interact with the virtual environment called the *scene*. Some of these photons end up into the sensor of a virtual camera, and counting the number of photons in each cell (pixel) gives rise to a virtual photograph.

The light simulation is described by the *rendering equation* [38] whose solution [70] simply states that an image is formed by summing up all light that directly reaches the sensor from the directly visible light sources, light that reaches the sensor after one surface interaction (*direct light*), and light that reaches the sensor after two, three, and so on, surface interactions called the first, second, and so on, order *indirect light*. See Figure 1.2.

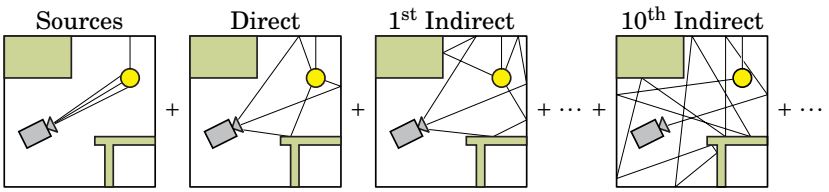


Figure 1.2. Decomposition of lighting. Full lighting is attained by summing light from directly visible light sources, direct light, and all orders of indirect light.

The amount of light arriving each pixel is thus defined by the sum, or integral, of the amount of light arriving through all possible paths that light can take. Each pixel has a separate – although highly similar – integral, whose value is typically solved by Monte Carlo simulation, i.e., random sampling potential light paths and evaluating the amount of light

arriving through each one of them. The Monte Carlo solution provides each pixel with a noisy estimate of the final color, and the amount of noise diminishes as more samples are taken.

The difference between most path-sampling-based Monte Carlo solutions of the rendering equation, such as classic Path Tracing [38] (PT), Bidirectional Path Tracing [44, 69, 70] (BDPT) and Metropolis Light Transport [71, 40] (MLT) can be reduced to the differences in the path samplers. A realized path distribution that better matches the real distribution of light generally results in less noise, and thus provides equivalent quality results with less computing time and power.

Each decrease of the remaining noise level to ten percent by taking more samples requires one hundred times more rendering time. Even using the best available sampler and the maximum available time budget may not always be enough. The noise then needs to be removed by other means. Such *denoising* methods typically remove the noise in the image by context-aware smoothing, although also other methods such as dimensionality reduction by truncated SVD [76] exist. Denoising methods are essentially more advanced methods of *reconstruction*, constructing the final image from the sampled data. These denoising methods have recently proven extremely powerful, reducing rendering time by up to orders of magnitude.

Traditional path samplers are not the only ways to solve the rendering equation. Other common solutions include for example photon density estimation methods [35, 36, 23, 21, 43] and the finite element method Radiosity [18] and its variants, e.g. [32, 4, 46]. The radiosity methods are not as common anymore due to challenges in scaling to modern-day geometric and material complexity. Photon density estimation methods may need longer rendering times time than traditional path sampling methods to avoid loss of detail, but they often solve for example caustics (see Figure 1.3) much faster. They are still commonly used today, especially due to the unification of the photon density estimation and traditional path sampling frameworks [14, 24].

Figure 1.3. Photograph of a glass with caustics. The glass refracts light into complex caustic patterns on the table. This concentration of light due to refraction is typically very hard to find for path-sampling-based methods.

1.2 Gradient-Domain Rendering

Lehtinen et al. [45] recently published a novel method for solving the rendering equation. They describe a *gradient-domain* modification of the Metropolis Light Transport [71] algorithm: Instead of *only* sampling the colors of the image pixels by traditional path sampling, they *also* sample for the *image gradients*, the finite differences between neighboring pixels in all directions. They then recover the final pixel colors by integrating the sampled image gradients by solving a screened Poisson equation.

This method exploits the typical similarity of the light paths which affect close-by pixels: Often a path that carries light to one pixel only needs to have its vertex positions moved a little to cast light into an adjacent pixel. While in traditional Monte Carlo rendering sampling such highly correlated pairs of paths would often be undesirable, gradient-domain rendering turns this into an advantage: The color differences of neighboring pixels are evaluated by subtraction, and subtracting correlated contributions of similar paths results in noise cancellation for the difference estimates. This often improves the quality of the sampled image gradients considerably and often leads to improved result images in equal time.

The final image is acquired from the samples by solving the screened Poisson equation which glues the color and gradient samples together. More precisely, the equation states that the final image's pixel colors should equal the traditional color estimates provided by the standard color samples, and the finite differences of the final image should equal the finite differences provided by the gradient samples. Solving this system of equations in the least-squares sense (L_2) provides an unbiased estimate for the final image, while a least absolute deviations (L_1) solution often produces visually more pleasing results while still converging to the exact solution. The novel *gradient-domain* method typically results in a significant improvement of image quality in equal time, or shorter time to equivalent quality, compared to the original *primal-domain* Metropolis Light Transport algorithm.

While providing good results, one may wonder why the authors decided to implement the novel gradient-domain renderer on top of the complex Metropolis Light Transport [71] algorithm instead of the much simpler Path Tracing [38] algorithm. The motivation, however, was solid: The Metropolis algorithm naturally concentrates samples to areas of high contribution, and as the gradients of an image contain the same information as the standard colors but are much more sparse, the gradient-domain sampler should be more efficient in gathering the information required for reconstructing the image. This argument does not, however, indicate nor rule out potential benefits of gradient-domain variants of classical Monte Carlo algorithms such as Path Tracing.

1.3 Research Gap

The intersection of gradient-domain rendering and classical Monte Carlo methods had not been covered by existing scientific literature before the publications in this thesis (see Figure 1.4 for an illustration). Earlier research in gradient-domain rendering [45, 48] did not predict nor oppose potential benefits from this intersection, which makes the research topic high-risk. The success of gradient-domain rendering in the context of Markov Chain Monte Carlo rendering, however, also suggests the possibility for a potentially high reward.

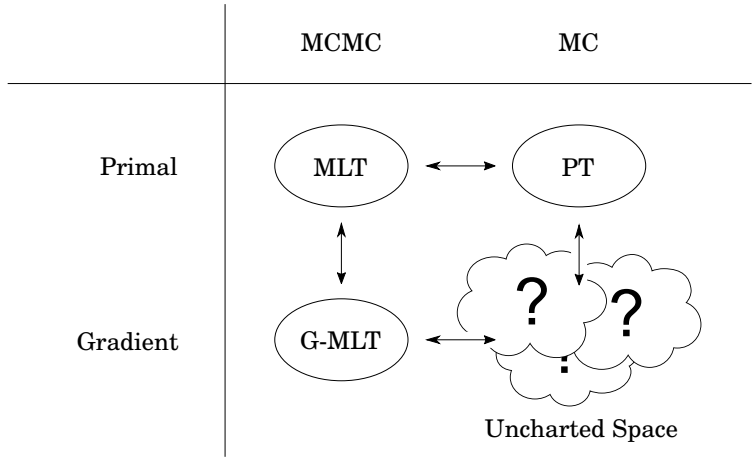


Figure 1.4. Research Gap. The intersection of gradient-domain rendering (bottom row) and Markov Chain Monte Carlo (left column) results in large benefits in the Gradient-Domain Metropolis Light Transport rendering algorithm (bottom left). What kind of benefits await in the uncharted intersection of gradient-domain rendering and standard Monte Carlo sampling (bottom right)?

1.4 Objectives and Scope

Rendering high-quality images is a very computationally intensive task, and movies may consist of hundreds of thousands of such images. Faster rendering methods are necessary to bring down the production and environmental costs of rendering animations and still images.

Many rendering companies – especially those specializing in animation – tend to use simpler Monte Carlo methods like Path Tracing and Bidirectional Path Tracing over complex methods like Metropolis Light Transport (e.g. [12, 11, 13, 9]). The full Metropolis Light Transport algorithm is notoriously hard to implement[1] and has a problem with predictability:

[1] Kelemen et al. [40] present a significantly simpler variant of Metropolis Light Transport, but it is often less effective.

different seemingly converged renderings of the same scene may feature the same areas of the image with very different brightnesses [45].

While unpredictability is undesirable in general, in animations this also results in flickering, which makes Metropolis Light Transport relatively unsuited for animation [67, 6]. Gradient-Domain Metropolis Light Transport often produces improved results in equal time but inherits these issues. Extending the scope of gradient-domain rendering to standard Monte Carlo in the form of Path Tracing and Bidirectional Path Tracing could result in methods that are more viable for industrial use, but special care should be given to animations to avoid potential flickering issues like that of Metropolis Light Transport. This establishes the three first research objectives:

Objective 1: Path Tracing is the de facto standard realistic image synthesis method in the industry [73]. Transform gradient-domain rendering from Markov Chain Monte Carlo to standard Monte Carlo by formulating and implementing a gradient-domain adaptation of Path Tracing. If path tracing in the gradient-domain turns out to be beneficial, the reason should be well studied.

Solution: Publication PI derives the Gradient-Domain Path Tracing algorithm and provides a frequency-domain analysis that explains the origins of its improved efficiency compared to standard Path Tracing.

Objective 2: Bidirectional Path Tracing is also sometimes used in production rendering (e.g. [12, 11]). Design and implement a gradient-domain adaptation of Bidirectional Path Tracing. This would also work as an example of how to extend more complex rendering methods to gradient-domain.

Solution: Publication PII presents the Gradient-Domain Bidirectional Path Tracing rendering method. It is often more efficient than both Bidirectional Path Tracing and Gradient-Domain Path Tracing.

Objective 3: Address potential animation-related issues, e.g. flickering, in gradient-domain rendering.

Solution: Publication PIII introduces the Temporal Gradient-Domain Path Tracing algorithm which extends gradient sampling and reconstruction to the time dimension. Reconstructing multiple frames at once this way removes most of the high-frequency temporal noise (flickering).

The industrial renderings with Path Tracing and Bidirectional Path Tracing are, in increasing numbers, starting to feature denoising – improved reconstructions from the same samples – often with extremely good results.

Even though the results of Gradient-Domain Path Tracing are often significantly better than standard Path Tracing, denoised Path Tracing results are often even better. If a modern reconstruction method can improve primal-domain rendering so much, how much could a modern reconstruction method improve gradient-domain rendering? Is gradient sampling still beneficial in the context of modern reconstruction? This sets the fourth and last research objective for this thesis:

Objective 4: Develop a new, more capable reconstruction method for gradient-domain rendering that features machinery similar to modern-day denoisers. Is gradient sampling still beneficial?

Solution: Publication PIV describes a novel machine-learning-based reconstruction method for gradient-domain rendering. With the new reconstruction method Gradient-Domain Path Tracing often reaches faster time to a given quality than when Path Tracing is joined with similar modern-day denoisers. Much of the improvement, e.g. the improved quality of shadows, follows directly from the gradients. Gradient sampling is still often beneficial.

In summary, the publications in this thesis extend the field of realistic image synthesis by studying the previously uncharted intersection of gradient-domain rendering and Monte Carlo image synthesis. Publication PI extends standard Monte Carlo Path Tracing to gradient-domain and explains why it is beneficial also in this context. Publication PII extends another Monte Carlo rendering method, Bidirectional Path Tracing, to gradient-domain. Publication PIII extends gradient sampling to the time-domain to better handle animations. Publication PIV replaces the old screened Poisson reconstruction with a much more powerful method based on machine learning and shows that gradients are still beneficial in the context of modern denoising. The new reconstruction method often produces higher quality results than primal-domain rendering followed with state of the art denoising. Figure 1.5 illustrates these relations.

1.5 Research Process

The research process used in the publications in this thesis follows the standard practices of the field.

In the scientific process, ideas need to be backed up with evaluation. In the context of realistic image synthesis, this means that the novel ideas need to be implemented as rendering algorithms, and their performance and properties need to be analyzed and evaluated.

Once the ideas have been implemented, the implementations need to be validated and potential bugs need to be fixed. In practice, this starts by

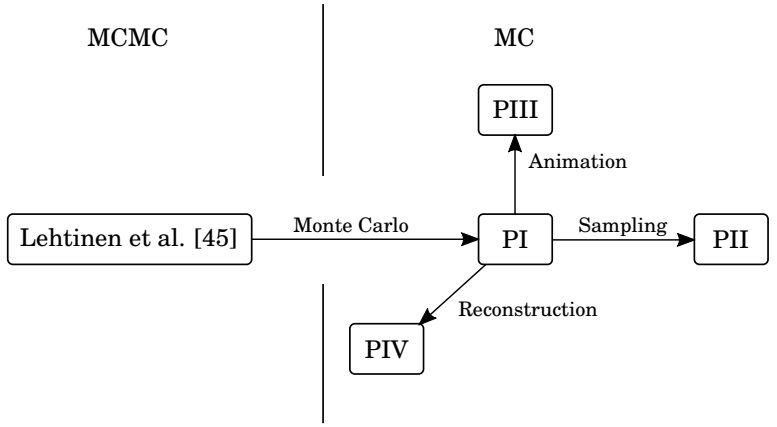


Figure 1.5. Relations of the publications. Publication PI extends gradient-domain rendering from Markov Chain Monte Carlo to standard Monte Carlo. Publications PII-PIV extend it in three orthogonal directions: PII introduces a bidirectional sampler, PIII improves the handling of animations, and PIV improves the reconstruction.

rendering high-quality ground truth images for a set of test scenes over a long period of time, often with a supercomputer. The images produced by the new implementation need to be compared to the ground truth images to catch potential systematic errors that are not explained by sampling noise. If the results do not show the expected improvement or show other surprising findings, the root causes need to be identified.

Once the method is considered ready, its performance needs to be analyzed and evaluated. A representative set of publicly available scenes are rendered with a wide range of rendering times to map the behavior of the new method. This examination is repeated for the relevant comparison methods in the scientific literature, with the same sequence of rendering times to facilitate a fair equal-time comparison.

The resulting images are analyzed and compared by objective numeric metrics to better understand the performance differences between the methods in different scenarios. Typical error metrics such as MSE and RelMSE [60] do not fully capture the human perception, and the analysis is followed by an additional visual examination to gain qualitative understanding of the properties of the novel method – its strengths and shortcomings. This approach both maps the area of applicability of the new method and paves the way to new research that might address potential shortcomings. Finally, the results of the research are documented and published in the form of a scientific article after a peer review.

1.6 Summary

The objective of this thesis is to develop new Monte Carlo rendering methods that work in the gradient-domain. The methods are based on path sampling which means that they sample random paths that connect lights to the sensor and evaluate their contributions. The methods work in the gradient-domain which means that they directly evaluate image gradients, i.e., color differences between adjacent pixels, and combine the sampled colors and gradients in a reconstruction step.

The next chapter provides a brief introduction to the most important topics for understanding the included publications. Chapter 3 summarizes their most important results, and Chapter 4 discusses their significance. Many researchers have since continued the work on Monte Carlo gradient-domain rendering, and these advances are summarized in Section 4.2.

2. Theoretical Foundation

This chapter contains a brief introduction to the most important topics for understanding the theoretical context of this thesis. The treatment in this chapter is brief and should be understood more as a refresher.

The reader is referred to the dissertation of Eric Veach [68] and the book by Pharr et al. [56] for excellent introductions to realistic image synthesis and light transport, the book by Goodfellow et al. [17] for an introduction to machine learning and neural networks, and O'Shea and Nash [53] for an excellent introduction to convolutional neural networks.

2.1 Geometrical Optics

Realistic image synthesis methods produce virtual photographs by simulating light. Light is electromagnetic radiation, carried by photons, and the human visual system is typically sensitive to wavelengths from approximately 380 nm (violet) to 740 nm (red) [63]. The behavior of photons is very accurately described by quantum mechanics, but this level of precision is rarely needed in practical applications of realistic image synthesis. A model based on geometrical optics, described in what follows, is often used instead.

The geometrical optics model treats light essentially as particles, or beams of light, essentially forgetting its wave-like properties. Light is modeled to travel in straight lines until it encounters a piece of matter. Upon contact, some of the light gets absorbed, and some of it scatters to different directions as defined by the properties of the material and the wavelength of the light.

This particle-like treatment of light leads to a solution that does not take into account light's wave-like phenomena such as diffraction and interference. Some wave effects, such as polarization [75, 34], can be added back as an extension. This simplification is required for practical and efficient realistic image synthesis and is usually not a significant limitation in practice.

The physical quantity describing the energy carried by a beam of light is *radiance* with unit $\mathrm{W\,sr^{-1}\,m^{-2}}$ and represents radiant power per solid angle and area of the beam. This models the energy density emitted or scattered as a beam – a narrow cone in space – from a small area near a point of study x to a small solid angle around a studied direction ω. It is this quantity, much like the energy of a photon, which stays constant along a beam as it travels in space. More technically, the definition of radiance $L(x,\omega)$ at position x to direction ω is

$$L(x,\omega) = \frac{\partial^2\Phi(x,\omega)}{\partial A^{\perp}_{\omega}(x)\;\partial\Omega(\omega)}, \tag{2.1}$$

where $\partial^2\Phi(x,\omega)$ is the radiant power flowing through a small area $\partial A^{\perp}_{\omega}(x)$ orthogonal to ω, around point x, to a small solid angle $\partial\Omega(\omega)$ around direction ω.

2.2 Rendering Equation

As light beams emitted by light sources interact with objects in the scene, part of their energy gets absorbed and the rest gets distributed to new directions. Assuming no *subsurface scattering*, i.e., no volumetric scattering inside surfaces such as human skin, this distribution of new directions is given by a *Bidirectional Scattering Distribution Function*, BSDF, which defines the visual appearance of a material.

The newly scattered light will again travel in space towards the new directions. Assuming no participating media such as smoke or mist, the light beams travel straight and do not lose energy on their way. As the light beams hit objects, they scatter again according to the BSDFs. This combination of traveling and scattering continues ad infinitum, until the residual energy of the light beams finally approaches zero in the limit. Assuming no above-mentioned volumetric effects[1], this process is described by the *rendering equation* [38].

The solution of the rendering equation at the camera sensor defines the pixel colors of the virtual photographs. The nowadays predominant form of the equation [32, 70] states that the amount of radiance that leaves a surface point x to a given direction ω is the sum of emitted radiance and the radiance that first arrives at x from any direction and then scatters to direction ω:

$$L_o(x,\,\omega) \;=\; L_e(x,\,\omega) \;+\; \int_{|\omega_i|=1} L_i(x,\,\omega_i)\, f(x,\,\omega_i \to \omega_o)\, |\omega_i \cdot N|\, \mathrm{d}\omega_i. \tag{2.2}$$

Here, L_o is the outgoing radiance, L_e is the emitted radiance, $L_i(x,\,\omega_i)$ is the radiance arriving at x from direction ω_i, N is the surface normal, $\omega_i \cdot N$

[1] The rendering equation may be extended to volumetric phenomena, but that is not required for this thesis. See e.g. Pharr et al. [56] for more information.

converts the incident radiance from area orthogonal to ω to the actual surface area around x, and the BSDF f tells the ratio of this differential irradiance that scatters towards direction ω_o. The integral sums this over all incident directions ω_i.

The incoming and outgoing radiances L_i and L_o are related by a simple condition on visibility: Since light travels straight in space, the radiance arriving point x from direction ω is the same radiance that was sent towards the receiver from the closest scene point in that direction:

$$L_i(x,\omega) = L_o(\gamma(x,\omega), -\omega). \tag{2.3}$$

Here x is the receiver location, ω is the direction of study, $\gamma(x,\omega)$ is the *ray-casting function* which returns the first intersection with scene geometry from x towards direction ω, and $-\omega$ is the direction from that point back towards x.

The algebraic solution of this equation system by e.g. the Neumann series is simple (see e.g. Veach [68]) and agrees with common sense: lighting is defined by emitted light and its all orders of scattering.

The rendering equation is usually applied per-wavelength, i.e., all radiances and BSDFs above are understood to have an implicit dependency on a wavelength parameter. This independent treatment of different wavelengths suffices for modeling most everyday physical effects, but e.g. fluorescence and phosphorescence require an extension [16].

2.3 Path Tracing

Path Tracing [38] is a direct solution to the rendering equation by Monte Carlo integration.

In Monte Carlo integration the integral is first written as the expectation of a random variable, and the expectation is estimated as an average over a number of random samples. The estimate contains noise but is unbiased, i.e., its expected value is precisely the value of the integral. The estimate is also consistent, meaning that the realized error converges towards zero when more samples are taken.

Uniformly distributed random samples would often result in very noisy estimates, and hence *importance sampling* is used: The samples $\{x_i\}$ are taken from a probability distribution constructed to resemble the integrand as well as possible, and the Monte Carlo estimate of an integral becomes

$$\int_\Omega f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}, \tag{2.4}$$

where $p(x_i)$ is the probability density of the random samples $\{x_i\}$. The variance of the estimate is at its minimum if $p(x)$ is proportional to $f(x)$ [37].

While an exact match is usually impossible, closer sampling distributions tend to lead to less noise.

In realistic image synthesis, the pixel colors of the virtual photograph are defined by the incoming radiance $L_i(x,\omega)$ given by the rendering equation. For a pinhole camera, the parameter x is simply the camera position and ω is the direction of measurement which depends on the (sub-)pixel location of the sample: the color of a pixel is acquired by integrating L_i over a set of directions corresponding to the pixel, weighted by a *pixel filter*. More realistic camera models also integrate over an aperture but still reduce to evaluating the incident radiance $L_i(x,\omega)$.

Path Tracing evaluates L_i stochastically, essentially by replacing the integral over incoming directions in Equation 2.2 by a randomly sampled direction ω_i:

$$L_o(x,\,\omega) \approx L_e(x,\,\omega) + \frac{1}{p(\omega_i)} L_i(x,\,\omega_i)\, f(x,\,\omega_i \to \omega_o)\, |\omega_i \cdot N|\,. \qquad (2.5)$$

Together with Equation 2.3, this leads to a straightforward algorithm: To evaluate $L_i(x,\,\omega)$, trace a ray from x towards ω to find the closest intersection with the scene geometry in that direction. Evaluating the outgoing radiance towards the arrival direction by Equation 2.5 gives an unbiased estimate for $L_i(x,\omega)$. This algorithm leads to an infinite recursion for the evaluation of the L_i terms, with typically ever-decreasing contributions. The recursion may be terminated probabilistically by *Russian roulette*, terminating at each iteration with some probability, and in case of a non-termination, increasing the weight of the rest of the contributions correspondingly to retain the correct expected value.

Two main strategies are often used for sampling the directions ω_i: *BSDF sampling*, i.e., sampling the direction from a distribution mimicking the shape of the BSDF, and *light sampling*, i.e., sampling a direction towards a known emitter in the scene. The two strategies are commonly used together: one or more light samples are taken towards known emitters, but the recursion is continued to a direction sampled according to the BSDF. These samples can be combined with *multiple importance sampling* [70, 68] which, if the *balance heuristic* is used, essentially reduces to using the marginal probability of having sampled the path by any of the used sampling techniques.

All in all, Path Tracing is a method that samples *light paths* between the sensor and the lights by a given unidirectional method. The sampler provides random paths x with known probability densities $p(x)$, whose radiances $f(x)$ are evaluated and the contributions $f(x)\,/\,p(x)$ are added to the pixels. This *path sampling framework* directly generalizes to many more advanced methods such as Bidirectional Path Tracing [44, 69, 70] and Metropolis Light Transport [71], whose main difference to Path Tracing is that Bidirectional Path Tracing uses a bidirectional path sampler, and Metropolis Light Transport uses a Metropolis-Hastings [50, 25] based

Markov chain path sampler. See Pharr et al. [56] for an excellent introduction to the topic.

2.4 Bidirectional Path Tracing

Bidirectional Path Tracing [44, 69, 70] is a related method for solving the rendering equation and is a direct application of a bidirectional path sampler to the previous path sampling framework. It works by simultaneously sampling sub-paths from the sensor and the lights, and connecting these paths in all possible ways. The different sampling strategies are combined with multiple importance sampling.

A unidirectional path sampler often produces very noisy images for instance in scenes in which important light sources are occluded by lampshades. Such occluders often make the light sampling connections of unidirectional path tracing to fail, producing a large number of paths with zero contribution. As the bidirectional sampler constructs its sub-paths starting from the light sources, many of the sub-paths avoid the occluder and enter more open areas. The vertices of the sensor-side sub-path can often be more easily connected to the vertices in the open areas than the occluded light sources.

The bidirectional sampling strategy thus often increases the probability of finding paths that carry light from the light sources to the sensor. The resulting decrease in variance can result in quite dramatically decreased rendering times, since decreasing the noise level by taking more samples is very inefficient – each halving of the noise level requires four times more rendering time. See Veach [68] for an excellent introduction to Bidirectional Path Tracing.

2.5 Metropolis Light Transport

Metropolis Light Transport (MLT) [71] is a rendering algorithm based on the same path sampling framework as Path Tracing and Bidirectional Path Tracing, but it generates paths by the Metropolis-Hastings algorithm [25]. Metropolis-Hastings iteratively mutates a Markov chain random variable, producing a sequence of correlated states (e.g. light paths) whose probability density asymptotically converges to the one defined by a given target function f (e.g. the amount of light carried by a path):

$$\lim_{k \to \infty} p_k(x) = cf(x), \tag{2.6}$$

where p_k is the probability density of the kth state and c is a normalization constant. After running for a long time, the states (e.g. paths) will thus be distributed approximately according to f.

Disregarding minor details on the handling of color instead of grayscale images, the histogram of the produced paths converges to a version of the virtual photograph. The final image is produced by fixing the image brightness which is often evaluated with Bidirectional Path Tracing.

The algorithm produces paths by randomly mutating the current state X into a candidate state Y which is accepted as the new state with probability

$$a(X \to Y) = \min\left(1, \frac{f(Y)K(Y \to X)}{f(X)K(X \to Y)}\right), \tag{2.7}$$

where f is the target function, e.g. the path contribution, and $K(X \to Y)$ is the probability density of suggesting the mutation from X to Y. If the candidate is not accepted, the previous state is reused. This acceptance probability has been carefully chosen to ensure convergence to the distribution defined by f, with some relatively mild assumptions (see [71]).

The Markov chain will, asymptotically, spend at each path a time proportional to its light contribution. If the chain by chance happens to enter a hard-to-find area with a high contribution, it will compensate by exploring that area for a random extended time. While this automatic compensation helps in exploring hard-to-find modes of light transport, it also leads to unpredictability since relative brightnesses of different areas in the image may be wrong even if the image looks otherwise ready.

The original path space Metropolis Light Transport is a rather complex method due to the numerous details in many of its path mutation strategies. Primary Sample Space Metropolis Light Transport (PSSMLT) [40] is a simplified variation which does not directly mutate paths but instead mutates the random numbers used by e.g. Bidirectional Path Tracing. Unfortunately, PSSMLT often produces worse results in equal time.

The recent Multiplexed Metropolis Light Transport algorithm [22] and its follow-up work [6, 54] improve PSSMLT's exploration of the path space and may boost its result quality close to path space MLT while retaining much of the relative simplicity. The primary sample space and path space methods have also been joined into a hybrid algorithm [54].

2.6 Gradient-Domain Metropolis Light Transport

Gradient-Domain Metropolis Light Transport (G-MLT) [45] is a variant of Metropolis Light Transport [71] and introduced the concept of gradient-domain rendering. In addition to sampling the color image like previous methods, it also evaluates discrete image gradients, that is, finite differences between adjacent pixels, by constructing pairs of similar paths for adjacent pixels and subtracting their light contributions. The similarity of the paths cancels noise in the subtraction and produces improved quality gradient estimates. The final image is produced by integrating the sampled

colors and gradients by solving a screened Poisson equation.

The gradients that Gradient-Domain MLT estimates are simply the differences between adjacent pixels: the gradient at pixel (i, j) is

$$g_{i,j} = \begin{bmatrix} I_{i+1,j} - I_{i,j} \\ I_{i,j+1} - I_{i,j} \end{bmatrix} \tag{2.8}$$

where $I_{i,j}$ is the color of pixel (i, j). Estimating these gradients efficiently requires constructing similar paths for nearby pixels and is described in the following.

The pixel color $I_{i,j}$ can be written, following the area formulation of the solution of the rendering equation [69, 45], as

$$I_{i,j} = \int_{\Omega} h_{i,j}(x) f(x) \, \mathrm{d}\mu \tag{2.9}$$

where the integral is over paths x in the space Ω of all possible paths that join the lights and the sensor, $h_{i,j}$ is the *pixel filter* which returns the path's weight to pixel (i, j) based on its screen space location, f is the *light contribution function* and μ is the *product area measure*.

The finite difference between pixels $i+1$ and i (we omit the other dimension for brevity) is first written in terms of Equation 2.9, and simple manipulation leads to a form suitable for direct evaluation (see the explanation below):

$$I_{i+1} - I_i = \int_{\Omega} h_{i+1}(y) f(y) \, \mathrm{d}\mu - \int_{\Omega} h_i(x) f(x) \, \mathrm{d}\mu \tag{2.10}$$

$$= \int_{T^{-1}(\Omega)} h_{i+1}(T(x)) \, f(T(x)) \left|T'(x)\right| \mathrm{d}\mu - \int_{\Omega} h_i(x) f(x) \, \mathrm{d}\mu \tag{2.11}$$

$$= \int_{\Omega} h_i(x) \left[f(T(x)) \left|T'(x)\right| - f(x) \right] \mathrm{d}\mu. \tag{2.12}$$

The *offset paths* from pixel $i+1$ are expressed in terms of *base paths* of pixel i by the *shift mapping* $y = T(x)$. This change of integration variable requires the Jacobian determinant $\left|T'(x)\right|$ to compensate for the change in path density. The shift mapping is assumed to retain the sub-pixel coordinates of the paths, so $h_{i+1}(T(x)) = h_i(x)$. Also, for brevity of expression, the shift mapping is assumed bijective so $T^{-1}(\Omega) = \Omega$ and Equation 2.12 follows.

This expression allows directly estimating finite differences between pixels by sampling random paths x and evaluating their gradient contribution

$$g(x) = f(T(x)) \left|T'(x)\right| - f(x). \tag{2.13}$$

In practice the shift mapping T can usually be made bijective only in a subset of the path space. The offset paths of non-invertible shifts are

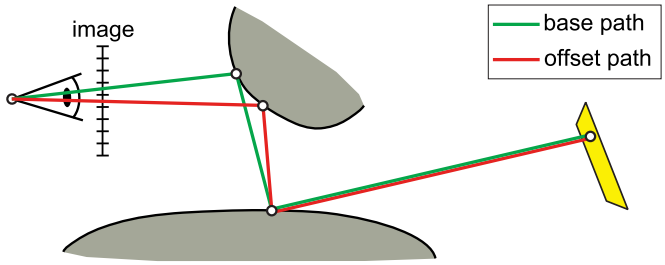


Figure 2.1. Shift mapping. Gradient-Domain Metropolis Light Transport evaluates gradients by shifting base paths (green) into similar offset paths (red) in the adjacent pixels. When the materials are rough, the offset path is constructed by simply reconnecting the offset path to the base path at the first vertex which is not determined by the pixel location. Figure from [45]. Republished with permission of Association for Computing Machinery; Permission conveyed through Copyright Clearance Center, Inc.

treated as non-existent, i.e., $f(T(x)) = 0$. The gradients are sampled in both ways, from pixel i to pixel $i+1$ and from pixel $i+1$ to pixel i. The samples in both directions are weighted appropriately to count each path pair exactly once. See Publication PI, Hua and Gruson et al. [30] or Lehtinen et al. [45] for details.

The primal-domain Metropolis Light Transport method runs a Metropolis-Hastings Markov chain in the space of light paths to directly evaluate the pixel integrals (Equation 2.9). Gradient-Domain Metropolis Light Transport uses similar machinery to evaluate the gradient integrals (Equation 2.12). While the primal-domain method drives the Markov chain with the light contribution function f, gradients are sparse in the path space and driving the chain with the gradient contribution function g (Equation 2.13) would make the chain to often get stuck. To facilitate easier exploring, G-MLT drives the chain with a mix of gradient and color magnitudes. As a result of this, the color and gradient images are no longer mere histograms of the Markov chain, but the samples need to be weighted by the ratio of the color or gradient contributions and the target function driving the Markov chain. These weighted contributions converge to the color and gradient images.

The gradient contribution function $g(x)$ (Equation 2.13) requires shifting paths from the *base* pixel to similar paths in the adjacent *offset* pixels by the shift mapping T. At simplest this can be easy: the offset path, whose screen space location has been changed by one pixel, can *reconnect* back to the base path immediately after the first vertex which is determined by the pixel location – all other vertices can stay the same (Figure 2.1). This strategy makes sense when the materials in question are rough (diffuse), but does not work for specular surfaces like ideal mirrors, as this shifting strategy modifies the reflection angles. To remedy this, the shift mapping in G-MLT tries to preserve *specular chains*, i.e., consecutive ideal reflection

or refraction angles, when dealing with very glossy materials. This is done with the *manifold perturbation* technique [33] which essentially amounts to applying Newton's method to fulfill the constraints of ideal reflection for the offset path.

After sampling the colors and gradients, G-MLT reconstructs the final image by solving a screened Poisson equation. This essentially happens by writing a system of equations: the colors of the result image should match the measured colors, and the gradients of the result image should match the measured gradients. This system of equations is overdetermined and has no exact solution, but a solution which minimizes a loss, e.g. mean-squared error (L_2) or absolute deviation (L_1), is used instead. More technically, the final image is defined as

$$I = \arg\min_x \|\alpha(x - c)\|_p^p + \|\nabla x - g\|_p^p \tag{2.14}$$

where x is a candidate image, c and g are the measured noisy colors and gradients, α determines the relative weight of the colors versus gradients, ∇ evaluates the discrete image gradients, i.e. the horizontal and vertical differences, and p is the order of the norm. The L_2 solution is unbiased, but the L_1 solution often produces more visually pleasing results.

Publications PI and PII extend gradient-domain rendering from Markov Chain Monte Carlo to standard Monte Carlo and analyze more closely why gradient-domain rendering is beneficial also in this context. The reader is referred to the recent survey by Hua and Gruson et al. [30] for a more encompassing introduction to gradient-domain rendering.

2.7 Neural Networks

Before delving into reconstruction and denoising, let us briefly sidestep to supervised learning via artificial neural networks, a machine learning technique to produce a mapping from inputs to outputs by learning from examples.

Essentially, a neural network is a parametric function $f(x;\Theta)$ constructed recursively from simple building blocks. In supervised learning the purpose is to *learn* the mapping from inputs to outputs from data, a collected set of examples $\{x_i\}$ which need to map to targets $\{y_i\}$. The neural network f is then *taught* by optimizing the parameters Θ to minimize a loss $L(f(x_i,\Theta),y_i)$, e.g. the mean squared error, over a dataset by a stochastic optimization method such as Stochastic Gradient Descent [57] or Adam [42].

The basic building blocks of neural networks are matrix products, adding constants (*bias terms*), and *activation functions* such as rectified linear units, $\mathrm{ReLU}(x) = \max(x,0)$. Activation functions make it possible to present nonlinear behavior: without them, a combination of matrix products and biases would still be an affine model with limited expression power.

When working with images, the matrix multiplications are often restricted to implement convolutions instead of general products: for example, a single densely stored matrix product for an image of resolution 1000×1000 would require a matrix of size $10^6 \times 10^6$ which is of course infeasible. The convolutional structure only requires storing the kernel weights, makes the model applicable to images of different sizes, and also encodes the often useful idea that the pixels of an image should, perhaps with the exception of edges, often be treated rather similarly.

Images often consist of red, green and blue channels which can be thought of as three scalar images. The internal layers of a neural network often consist of hundreds of such channels, often called *features* in the neural network context.

The common *convolutional layer* building block produces a multi-channel image by applying a sequence of convolutions to the channels of the input image. More precisely, each channel of the output image has a separate convolution kernel for each of the input image's channels. Adding up the convolution results over the input channels produces the output image's channel. For example, a 3×3 convolution which maps an image from 5 to 20 channels would thus consist of $3 \times 3 \times 5 \times 20 = 900$ scalar parameters.

2.8 U-Net

The combination of convolutional layers, biases, ReLUs, and downsampling and upsampling layers results in powerful neural network architectures such as U-Net [58]. The U-Net architecture, which maps images to images, was originally used in biomedical image segmentation but has since become a common starting point for many applications.

The U-Net processes the image with a large number of convolutional layers in different resolutions. The use of multiple resolutions makes it easier for the network to study patterns on multiple scales.

In the first half of execution U-Net repeatedly processes the image with convolutional layers and downsamples the result to half-resolution. In the second half, it upsamples the results back and further processes them with more convolutional layers.

In addition, it stores the results right before each downsampling and adds them back as extra channels (features) to supplement the upsampled half-resolution results. These *skip connections* make the neural network's task easier as it does not need to transmit all useful data through the downsampled layers.

Convolutional neural networks are an important tool for many modern denoising and reconstruction methods. Publication PIV also uses a novel network architecture based on the ideas of U-Net and Densely Connected Convolutional Networks [31]. The reader is referred to Goodfellow et

al. [17] and Oshea and Nash [53] for a more encompassing introduction to convolutional neural networks.

2.9 Reconstruction and Denoising

In the realistic image synthesis context, *reconstruction* is the process of building the final virtual photograph from the samples. The traditional method for reconstruction consists merely of averaging the color samples in each pixel. Gradient-domain rendering reconstructs the final image by solving a screened Poisson equation.

Denoising is the process of removing noise from an existing image, and it is a subset of reconstruction. Although many methods exist to remove noise – such as projection to a truncated SVD basis [76] – the most evident way of removing noise is blurring. Blurring, like other methods, however causes loss of detail if done carelessly, and thus context-aware methods like the bilateral [66] and non-local means filters [8] are often used. They attempt to avoid loss of detail by carefully selecting which pixels to average over. In other words, the methods use heuristics on pixel similarity to predict a separate smoothing kernel – weights for a weighted average – for each pixel in the image.

The *bilateral filter* [66] replaces the color of each pixel by a weighted average of its *nearby* pixels. The weights are typically evaluated as a product of Gaussians of the pixel distance and color difference. If also other images are used for cues on expected pixel similarity, the filter is called *cross-bilateral*. Xu et al. [77] use a slightly modified bilateral filter for denoising Monte Carlo renderings.

The *non-local means* filter [8] is a variation of the bilateral filter. Unlike the bilateral filter, the non-local means filter replaces a pixel's color by an average over pixels whose *neighborhoods* are similar to the current pixel's neighborhood, with no limitation on pixel location. Rousselle et al. [61] build on the non-local means filter for denoising Monte Carlo renderings.

The denoising process can be helped by utilizing *auxiliary information* such as albedo, surface normal and depth of the first scene intersection behind the pixels (Figure 2.2). This data can be collected during sampling for almost no extra cost and can help in identifying which pixels should have similar colors. However, for example shadows cannot be inferred from the typical auxiliary buffers, and as such many denoisers tend to reconstruct them less accurately than many other effects.

The auxiliary buffers are widely used in denoising Monte Carlo renderings. For instance, Li et al. [47] and Rousselle et al. [62] use the auxiliary features with a combination of non-local means, cross-bilateral filtering, and Stein's Unbiased Risk Estimate [64] to denoise Monte Carlo renderings. Bitterli et al. [7] combine non-local means with first-order regression,

and Moon et al. [51] use the auxiliary buffers with polynomial regression of varying order.

Neural networks have also recently gained much popularity in Monte Carlo denoising due to their tendency to produce excellent results. For instance, Kalantari et al. [39] use neural networks to estimate parameters for the cross-bilateral filter, Bako et al. [2] use a convolutional neural network to directly predict the smoothing kernel for each pixel, and Vogels et al. [72] further improve the kernel prediction method of Bako et al. and extend it to animations. Chaitanya et al. [10] use the U-Net architecture with recurrent connections to denoise very low sample count animation rendering.

Many denoisers efficiently remove high-frequency noise from images. However, the images are typically left with some amount of slowly varying low-frequency noise. Different low-frequency noise patterns in consecutive frames of an animation may result in distracting flickering. This problem can be alleviated by cross-frame reconstruction, i.e., using the data of multiple animation frames to reconstruct each animation frame. Typical solutions use *motion vectors* to track the motion of objects between successive animation frames in pixel-space [28, 78, 72]. The link established by the motion vectors allows extending the smoothing kernels to encompass several frames.

The reader is referred to e.g. Zwicker et al. [79] for more information on denoising and reconstruction.

2.10 Summary

The transportation of light in a virtual environment is described by the rendering equation. The rendering equation is often solved by Monte Carlo methods such as Path Tracing and Bidirectional Path Tracing, which

Figure 2.2. Auxiliary inputs. Example of a Monte Carlo sampled image (“Color”) and the albedo, depth and normal auxiliary inputs which are often used to guide in the reconstruction.

estimate the pixel colors by sampling random paths that connect the lights to the sensor.

Lehtinen et al. [45] presented gradient-domain rendering in the Markov Chain Monte Carlo context. They supplement the standard color samples by directly evaluating image gradients. They then reconstruct the final image from the color and gradient samples by solving a screened Poisson equation.

The objective of this thesis is to study the uncharted intersection of gradient-domain rendering and standard Monte Carlo methods. The individual research questions were presented in Section 1.4, and each question is answered by a separate publication. The following chapter presents the most important results from the publications.

3. Publication Summaries

Answering the research questions in Section 1.4 led to many interesting findings: an improved understanding of gradient-domain rendering and why it is beneficial, several new gradient-domain Monte Carlo rendering methods for realistic image synthesis, and a machine learning based reconstruction for gradient-domain rendering. This chapter summarizes the most important results from the publications.

3.1 Gradient-Domain Path Tracing

Publication PI formulates a gradient-domain adaptation of the de facto industry standard Path Tracing [38] rendering algorithm. The big picture of the new method is quite similar to Gradient-Domain Metropolis Light Transport: The method samples base paths by Path Tracing and evaluates a standard color image with them. The base paths are also shifted into offset paths in the adjacent pixels to evaluate gradients. The colors and gradients are joined into a final image by solving a screened Poisson equation.

The most important distinction between the two methods is, of course, that in the new method base paths are sampled by Path Tracing, not by a Markov chain. A sample produced by Path Tracing consists of recursively constructed paths that share the same beginning and essentially form a tree. Each vertex in the tree defines a path that needs to be shifted. Shifting the whole tree with the manifold perturbation shift of G-MLT (Section 2.6) would be very inefficient. Gradient-Domain Path Tracing (G-PT) uses a novel shift mapping based on local decisions, which allows shifting of the whole tree of paths by visiting each vertex only once.

The new shift mapping uses the reconnection shift from G-MLT for the simple case (Figure 2.1), but handles specular materials by copying the half-vector (the surface normal that would transform between the incoming and outgoing directions in ideal reflection or refraction) from the base path to the offset path. The new offset direction is generated from the half-vector

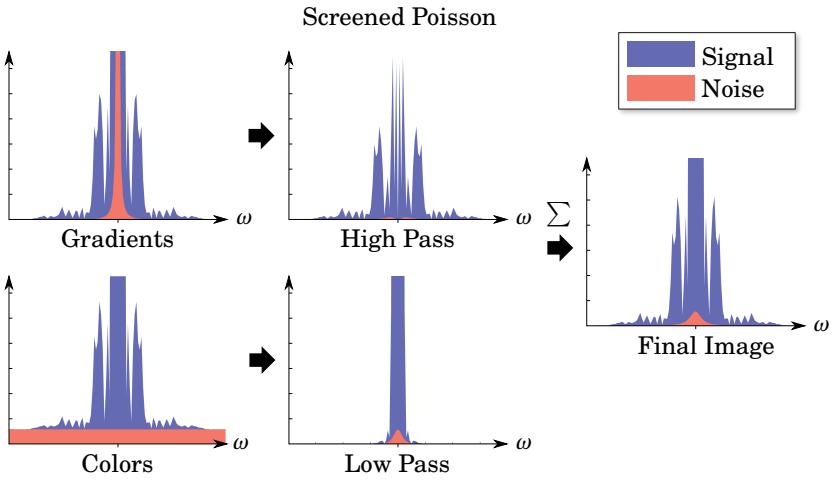


Figure 3.1. Frequency-domain analysis of gradient-domain rendering. Gradients (top left) capture the high frequencies of the image with little error but do not contain much useful information about the low frequencies (notice the noise singularity near the zero frequency). The standard color samples (bottom left) capture all frequencies uniformly well but are not as effective for the high frequencies. The screened Poisson reconstruction implicitly filters out the noisy low frequencies from the integrated gradients (top middle) and the noisy high frequencies from the color samples (bottom middle) and sums the filtered signals to form an image with less error than either of the inputs.

by applying the ideal reflection or refraction to the offset path.

The reason why gradient-rendering rendering is beneficial also in the traditional Monte Carlo context can be approximately simplified into the following: Most energy in natural images is concentrated to the low frequencies which are strongly suppressed by the (discrete) gradients. Since Monte Carlo noise is relative to the signal's total energy, the sampled gradients typically have much less noise than the color samples. Integrating the image from only the gradients would result in a high amount of low-frequency error, but the screened Poisson reconstruction takes the low frequencies from the sampled colors and the high frequencies from the gradients, essentially taking the best of both worlds. See Figure 3.1.

The test scenes suggest a typical five- to twelve-fold improvement in rendering time compared to standard Path Tracing. However, there are also cases for which Gradient-Domain Path Tracing does not work very well: sub-pixel scale geometric detail is not well captured by the one-pixel finite differences, and some complex materials may not obey the assumptions behind the shift mapping. See Figure 3.2. Both cases result in magnified noise in the unbiased gradients, which increases reconstruction error. Problematic cases may include for example hair, fur, and dense foliage.

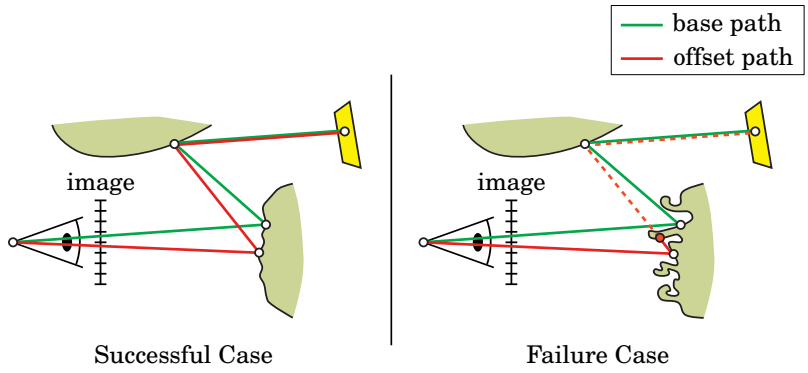


Figure 3.2. Subpixel-scale geometric detail. *Left*: A small amount of subpixel-scale geometric detail does not prevent constructing useful offset paths. *Right*: Too much subpixel-scale detail may break the correlation between the base and offset paths. Here the shift is inefficient for two reasons: The offset path is occluded, and the very different surface orientations at the first intersections may cause differences in the BSDF and the dot product in Equation 2.2. The Jacobian determinant of the shift in Equation 2.13 could also become very large or very small.

3.2 Gradient-Domain Bidirectional Path Tracing

The unidirectional sampling method of Path Tracing is not always efficient at finding the relevant light paths that contribute to the image. In complex scenes this may result in very slowly vanishing noise and impractically long rendering times. Gradient-Domain Path Tracing uses the path sampler from Path Tracing and inherits the problem. The more advanced path sampler of Bidirectional Path Tracing [44, 69, 70] often produces significantly less noise, and a bidirectional adaptation of Gradient-Domain Path Tracing would often be useful. Publication PII presents the Gradient-Domain Bidirectional Path Tracing (G-BDPT) algorithm.

Gradient-Domain Bidirectional Path Tracing differs from the unidirectional method in two ways: the bidirectional path sampler, and the corresponding shift mapping. The frequency analysis of the previous publication still holds.

Bidirectional Path Tracing works by sampling a sub-path from the sensor and a sub-path from a light and connecting these sub-paths in all possible ways. The key insight in Gradient-Domain Bidirectional Path Tracing is that discarding connections with a very glossy vertex on either side only very slightly increases per-sample variance, but allows an efficient shifting strategy which shifts each vertex only once – even with the original shift mapping from Gradient-Domain Metropolis Light Transport.

The new method brings the benefits of Gradient-Domain Path Tracing to scenes which require the bidirectional sampler for good convergence. A typical case is when the scene is mostly lit indirectly and most light sampling connections to the light sources fail.

3.3 Temporal Gradient-Domain Path Tracing

While the previous Monte Carlo gradient-domain methods often significantly reduce rendering time of still images compared to standard Path Tracing, the images retain similar low-frequency noise. While this noise is hard to notice in still images, in animation it displays as flickering. Publication PIII extends the gradient computation to the time dimension and reconstructs the animation by solving a three-dimensional spatiotemporal screened Poisson equation. This significantly reduces the flickering and shortens the rendering time to equivalent quality.

The time-components of the gradients are evaluated by a method of random seed sharing: Each frame is rendered in two buckets, with the first bucket sharing the random seeds with the previous frame, and the second bucket sharing the random seeds with the next frame. More precisely, denoting by $I(k,\ s)$ the frame k rendered with seed s, the frame k is rendered in buckets $I(k,\ k)$ and $I(k,\ k+1)$, and the temporal difference is given by $I(k+1,\ k+1) - I(k,\ k+1)$. A temporal shift mapping based on random seed sharing enables independent rendering of the video frames. Subtracting the colors of the corresponding pixels in the successive frames, evaluated with exactly the same random seeds, then provides the time-component of the gradient.

The two-dimensional screened Poisson reconstruction removes much of the spatial high-frequency noise from the image compared to the input colors. Similarly, the spatiotemporal screened Poisson reconstruction removes much of the temporal high-frequency noise, flickering.

The publication further improves the rendering quality in two ways. First, it shows that adaptive sampling for gradient-domain rendering should distribute samples according to the variance of the gradient samples instead of the color samples. Second, it shows that it is beneficial to evaluate the time-dimension component of the gradients by following the motion vectors which track the objects as they move between frames.

3.4 Deep Convolutional Reconstruction for Gradient-Domain Rendering

While Gradient-Domain Path Tracing often produces improved quality compared to equal-time Path Tracing, modern denoising methods applied to the outputs of Path Tracing have typically achieved even better quality in equal time. Denoising methods exploit the heuristic correlations of the unknown ground-truth image and the auxiliary buffers such as albedo, depth and normals which are captured for free during the sampling. The screened Poisson reconstruction used by earlier gradient-domain rendering methods does not make use of this data.

Publication PIV replaces the screened Poisson reconstruction by a method similar to modern-day denoising. More specifically, it presents a novel deep convolutional neural network trained to map the sampled colors, gradients, and auxiliary buffers to estimates of the ground-truth image.

The new reconstruction method usually produces much cleaner images than the screened Poisson reconstruction. It also does not result in energy loss for low sample counts like the L_1 screened Poisson reconstruction. However, similarly to the screened Poisson solver, high variance of the gradients samples e.g. due to sub-pixel scale geometric detail or some complex materials may hurt reconstruction quality.

The novel reconstruction method also typically improves over primal-domain methods used with denoisers, especially in the lower sample count regime. The improvement is most prominent in shadows, since shadows are not captured by the usual auxiliary buffers, but shadow edges and penumbra are captured by the gradient samples.

The publication also presents a novel neural network architecture which results in a small but consistent improvement over the standard U-Net [58] architecture. The network is trained with the neural perceptual image similarity metric E-LPIPS [41] which tends to result in slighty sharper and more natural images than the classic L_1 loss. The increased sharpness may, however, sometimes be undesired if the image is actually supposed to look blurry e.g. due to a strong depth of field effect. In this case it might be preferable to train the network with the L_1 loss.

3.5 Summary

The publications in this thesis resulted in many general purpose rendering methods that are often faster than earlier methods. The publications extend the theory and understanding of gradient-domain rendering and demonstrate that it is a widely applicable, general-purpose paradigm for realistic image synthesis in the path sampling context. The first publication shows that gradient-domain rendering is beneficial also with standard Monte Carlo, and the other publications extend it to animation, improved reconstruction and improved sampling, thereby advancing the state of the art in realistic image synthesis.

Perhaps most interestingly, however, the publications show that the previously uncharted intersection of gradient-domain rendering and standard Monte Carlo is rich in interesting theory and applications.

The publications have already led to additional research which is summarized in the next chapter. The next chapter also further discusses the scientific and practical implications of the publications and lists ideas for future work.

4. Discussion

This chapter discusses the nature of gradient-domain rendering, its scientific and practical significance, reliability of the conducted research and future research topics in gradient-domain rendering.

4.1 Gradient-Domain Rendering as Path Space Denoising

The relationship between gradient-domain reconstruction and image-space denoising is a relatively common topic in discussions. Both are reconstruction methods and share the same goal: to produce an image with little noise. Both classes result in images with decreased high-frequency error, while the low-frequency error is harder to remove. Image-space denoisers often remove noise by heuristics on which pixels to average over, while gradient-domain reconstruction can resort to unbiased estimates of pixel differences.

The usefulness of gradient-domain rendering stems from the ability to construct correlated path pairs for nearby pixels. Much of the noise of these individual samples cancels out in the finite difference due to the correlation. Nothing would be gained by gradient-domain rendering without this correlation, e.g. by using independent samples. It is thus possible to say that gradient-domain rendering is based on noise cancellation already in path space.

This argument can be extended to temporal reconstruction for animation: while traditional cross-frame denoising methods extend smoothing kernels to multiple frames, temporal gradient-domain rendering obtains unbiased cross-frame finite differences.

However, one does not need to choose between noise cancellation in path space and image-space denoising. Publication PIV shows that it is often beneficial to use them together.

4.2 Recent Research by Others

The publications in this thesis demonstrate that gradient-domain rendering is a general-purpose rendering paradigm that is useful also in the more common traditional Monte Carlo context. Following this observation, many researchers have since continued the work on gradient-domain rendering, with most publications primarily targeting the newly enabled traditional Monte Carlo context.

Hua et al. [29] extend photon density estimation methods, e.g. [21, 43] to gradient-domain, and Gruson et al. [19] extend gradient-domain photon density estimation to participating media. Sun et al. [65] merge gradient-domain photon density estimation into a hybrid framework with gradient-domain path sampling. Petitjean et al. [55] use gradient-domain rendering in the wavelength dimension to reconstruct the spectrum of incident radiance for image pixels. Bauszat et al. [3] improve Path Reusing [5] by the half-vector shift from Publication PI and hide distracting pattern artifacts by transforming the method into gradient-domain.

Manzi et al. [49] regularize the screened Poisson reconstruction by a soft constraint that the final image is locally representable as linear combinations of the auxiliary buffers. Rousselle et al. [59] formulate gradient-domain reconstruction with control variates and propose an improved reconstruction method. Back et al. [1] use bootstrap aggregation and inverse variance weighted Poisson reconstruction from the gradient samples to approximate an ideal feature for local regression based reconstruction. Ha et al. [20] identify probable gradient outliers by a heuristic based on non-local means filtering, and remove the corresponding constraints from the screened Poisson reconstruction.

The recent survey by Hua and Gruson et al. [30] provides an excellent in-depth introduction to gradient-domain rendering and many of the recent advances described above.

Much of the recent gradient-domain rendering research transforms existing image synthesis methods into gradient-domain, improves the reconstruction from gradients and colors, or extends the scope of gradient-domain rendering in some other way. These improvements are important also for the adoption of gradient-domain rendering in production environments, but some challenges remain.

4.3 Recommendations for Future Research

Many production renderers cover large numbers of techniques for different purposes and consist of large codebases. Adapting such large renderers to a new rendering paradigm can be a challenging task. To ease in the adoption of gradient-domain rendering in existing renderers, the gradient-domain

rendering component should likely be made somewhat separated from the existing code. For instance, instead of modifying the path tracer to evaluate gradients on the fly, the path tracer could store its paths as a tree and give the tree to a gradient-domain component for processing. The gradient-domain component would then shift the tree while reusing existing computation as much as possible and evaluate the gradients. While this might be a good starting point, many questions remain unanswered.

The following treatment outlines some of the most important challenges and open questions in gradient-domain rendering from the perspective of practical use. The treatment follows a rough order of urgency – not necessarily importance – as decided by the author.

4.3.1 Volumetric Gradient-Domain Path Tracing

One of the more challenging tasks in adapting a production path tracer to gradient-domain is related to designing an efficient shift mapping for volumes, especially heterogeneous participating media. Gruson et al. [19] extend gradient-domain rendering to homogenous participating media by volumetric photon density estimation. They also present results for volumetric gradient-domain path tracing in homogenous media. Hua and Gruson et al. [30] report in their survey about promising initial results for gradient-domain path tracing in heterogeneous participating media, but do not present a full algorithm. An extension of Gradient-Domain Path Tracing to heterogeneous participating media would be very important since it is a common element in many production scenes.

4.3.2 Color – Gradient Adaptive Sampling

Gradients might not be efficient in some pixels that for example contain dense fur or foliage, but might be efficient on others. A robust rendering system should autodetect on which pixels gradients are worth the effort, and enable or disable gradient computation accordingly. The reconstruction should produce a very similar noise pattern for all pixels regardless of the ratio of color and gradient samples. Optimally, the reconstruction should guide the sampler on optimal placement of color and gradient samples.

This kind of a renderer would make it much easier to approach gradient-domain rendering, as even a partial implementation might provide concrete benefits.

4.3.3 Temporal Reconstruction

Gradient-domain rendering, like single-frame denoising, often removes a large amount of high-frequency noise from the image. The low-frequency noise remains and may result in flickering in animation. Publication PIII

extends gradient-domain rendering to cross-frame reconstruction by introducing temporal gradients – without requiring the renderer to keep multiple frames in memory at once.

Publication PIV presents a new machine learning based reconstruction method for gradient-domain rendering. This method should be extended to cross-frame reconstruction to output flicker-free animations. Possible directions include utilizing temporal gradients and/or reprojection methods like Vogels et al. [72].

4.3.4 Simple Shift Mappings

Hua and Gruson et al. [30] suggest in their survey a simple shift mapping based on a combination of random number replay and the reconnection shift. They report that this shift mapping often produces comparable results to the somewhat more complicated half-vector shift from Publication PI. The simplicity and generality of this simple shift mapping could make the adoption of gradient-domain rendering to production renderers more alluring, especially if the method can be extended to work well with heterogeneous participating media. This approach, however, still relies on knowing when to apply the reconnection shift, which is discussed in the following.

4.3.5 Vertex Classification

Contemporary gradient-domain rendering methods require the classification of vertices into connectable and non-connectable, often by comparing the roughness of the material against a threshold. This information is used for choosing the shift mapping. However, materials are often modeled with layers, and may not have a single roughness parameter that could satisfactorily be used. Roughness can also be direction-dependent, and programmable materials present another complication. A more satisfactory way of choosing the shift mapping would make the adoption of gradient-domain rendering easier.

As a partial solution, the roughness threshold can be made probabilistic, as long as the used random numbers can be treated as constants selected before sampling the path. The shift mapping also needs to be reversible[1] with each value of the constant.

4.3.6 Combining Shift Mappings

In some cases, it is unclear which shift mapping should be used. In these cases, the shift mapping can be chosen randomly with the same conditions

[1] If path x maps to path y when shifting right, then path y needs to map to path x when shifting left.

as in Section 4.3.5.

It would be good to be able to combine the different shift mappings by giving them weights that sum to one like in multiple importance sampling. One idea is to output the unbiased results of the different shift mappings into separate buffers and combine them with inverse-variance weighting.

4.3.7 Generic Shift Mappings

Contemporary shift mappings are based on invariant properties of the paths: The reconnection shift keeps the reconnection vertex invariant, the half-vector shift keeps the half-vector invariant, and random number replay keeps the random numbers that define the path invariant. These invariants make the shifts reversible, as all critical information is stored and retained in both paths. Is there a more generic principle, or perhaps another invariant, that would replace the need for case-specific heuristics? One interesting idea is to somehow use the gradient of the path contribution, perhaps evaluated in a half-vector parametrization which encodes that materials often approximately follow the law of ideal reflection. One of the challenges is that the gradient does not stay invariant in this shift, and something else is required to make the shift reversible.

4.3.8 Path Guiding

Path guiding methods, e.g. [74, 26, 52, 27], train the path importance sampler on the fly and are becoming increasingly common in today's production renderers [73]. They often provide a significant variance reduction in equal time. Path guiding could probably be used to provide better gradient samples.

4.3.9 Kernel Prediction

Many primal-domain denoisers such as Bako et al. [2] and Vogels et al. [72] use the color and auxiliary buffers to predict an averaging kernel for each pixel. The gradient-domain reconstruction presented in Publication PIV does a direct prediction: the network directly outputs a new color for each pixel based on the input colors, gradients, and auxiliary buffers. Extending kernel prediction to gradient-domain rendering might provide some benefits. The kernels should give weights to the gradient data in addition to the primal colors since the gradients contain unbiased information which is not present in the color buffer.

4.3.10 Sample-Based Reconstruction

Instead of working with averaged samples in image-space, Gharbi et al. [15] use a neural network to reconstruct an image directly from the individual color samples by predicting a splatting kernel for each sample. This approach could benefit from an extension to include gradients.

4.4 Practical Implications

The publications in this thesis present methods that often speed up realistic image synthesis. The previous section describes some of the remaining challenges in implementing Monte Carlo gradient-domain rendering in contemporary production renderers. A successful implementation could potentially result in significantly shorter rendering times in many scenes.

Movie and advertisement companies use realistic image synthesis for animations, which requires the synthesis of a large number of animation frames, often by supercomputers. Shorter rendering times could allow faster production or more iterations on the content by content artists. The presented rendering methods could potentially also advance the feasibility of rendering to slightly more complex scenes. Alternatively, the shorter rendering times could lead to electricity savings in the rendering cluster.

Following an implementation in a widely used renderer, e.g. product design and architecture companies could benefit from faster depictions of products and buildings. This could allow more iterations, and in some cases lead to improved design decisions. Somewhat more realistic visualizations of future buildings and infrastructure projects might also become feasible.

4.5 Reliability and Validity

The following sections of the thesis describe the details of the novel methods in the form of peer-reviewed scientific publications.

The research methods in the publications follow standard scientific practices of the field. The results seem consistent and agree with our theoretical understanding. All publications in this thesis have gone through rigorous peer review processes, and any known inaccuracies or confusions have been corrected.

The empirical tests in the publications use the same publicly available test scenes that are used by the vast majority of realistic image synthesis research. The scenes are technically challenging for various reasons, and although not as filled in detail as some industrial-grade scenes, they are quite representative of many of the challenges present.

The consistent success of the presented methods speaks for the generality

of the results. Gradient-domain rendering in the Monte Carlo context seems to do well for many kinds of applications, but it also has some limitations which are more closely explained in the publications. Such problematic cases, e.g. large amounts of sub-pixel scale geometric detail and some complex materials, might be better rendered without gradients.

A robust rendering system (Section 4.3.2) should select the appropriate sampling method for each pixel and get the best of both worlds. This way the renderer could fall back to primal-domain when there is little correlation in the light transport between adjacent pixels, and benefit from gradient-domain rendering when it is most useful.

References

[1] Jonghee Back, Sung-Eui Yoon, and Bochang Moon. Feature generation for adaptive gradient-domain path tracing. *Computer Graphics Forum*, 37(7):65–74, 2018.

[2] Steve Bako, Thijs Vogels, Brian Mcwilliams, Mark Meyer, Jan NováK, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.*, 36(4):97:1–97:14, July 2017.

[3] Pablo Bauszat, Victor Petitjean, and Elmar Eisemann. Gradient-domain path reusing. *ACM Trans. Graph.*, 36(6):229:1–229:9, November 2017.

[4] Philippe Bekaert, László Neumann, Attila Neumann, Mateu Sbert, and Yves D Willems. Hierarchical Monte Carlo radiosity. In *Rendering Techniques' 98*, pages 259–268. Springer, 1998.

[5] Philippe Bekaert, Mateu Sbert, and John Halton. Accelerating path tracing by re-using paths. In *Proceedings of the 13th Eurographics Workshop on Rendering*, EGRW '02, pages 125–134, Aire-la-Ville, Switzerland, 2002. Eurographics Association.

[6] Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. Reversible jump Metropolis light transport using inverse mappings. *ACM Trans. Graph.*, 37(1):1:1–1:12, October 2017.

[7] Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, José A Iglesias-Guitián, David Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novák. Nonlinearly weighted first-order regression for denoising Monte Carlo renderings. In *Computer Graphics Forum*, volume 35, pages 107–117. Wiley Online Library, 2016.

[8] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[9] Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. The design and evolution of Disney's Hyperion renderer. *ACM Trans. Graph.*, 37(3):33:1–33:22, July 2018.

[10] Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph.*, 36(4):98:1–98:12, July 2017.

[11] Per Christensen, Julian Fong, Jonathan Shade, Wayne Wooten, Brenden Schubert, Andrew Kensler, Stephen Friedman, Charlie Kilpatrick, Cliff Ramshaw, Marc Bannister, Brenton Rayner, Jonathan Brouillat, and Max Liani. RenderMan: An advanced path-tracing architecture for movie rendering. *ACM Trans. Graph.*, 37(3):30:1–30:21, August 2018.

[12] Luca Fascione, Johannes Hanika, Mark Leone, Marc Droske, Jorge Schwarzhaupt, Tomáš Davidovič, Andrea Weidlich, and Johannes Meng. Manuka: A batch-shading architecture for spectral path tracing in movie production. *ACM Trans. Graph.*, 37(3):31:1–31:18, August 2018.

[13] Iliyan Georgiev, Thiago Ize, Mike Farnsworth, Ramón Montoya-Vozmediano, Alan King, Brecht Van Lommel, Angel Jimenez, Oscar Anson, Shinji Ogaki, Eric Johnston, Adrien Herubel, Declan Russell, Frédéric Servant, and Marcos Fajardo. Arnold: A brute-force production path tracer. *ACM Trans. Graph.*, 37(3):32:1–32:12, August 2018.

[14] Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.*, 31(6):192:1–192:10, November 2012.

[15] Michaël Gharbi, Tzu-Mao Li, Miika Aittala, Jaakko Lehtinen, and Frédo Durand. Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Trans. Graph.*, 38(4):125:1–125:12, July 2019.

[16] Andrew S. Glassner. A model for fluorescence and phosphorescence. In *Photorealistic Rendering Techniques*, pages 60–70. Springer, 1995.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[18] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *SIGGRAPH Comput. Graph.*, 18(3):213–222, January 1984.

[19] Adrien Gruson, Binh-Son Hua, Nicolas Vibert, Derek Nowrouzezahrai, and Toshiya Hachisuka. Gradient-domain volumetric photon density estimation. *ACM Trans. Graph.*, 37(4):82:1–82:13, July 2018.

[20] Saerom Ha, Sojin Oh, Jonghee Back, Sung-Eui Yoon, and Bochang Moon. Gradient outlier removal for gradient-domain path tracing. *Computer Graphics Forum*, 38(2):245–253, 2019.

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

[22] Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. Multiplexed Metropolis light transport. *ACM Trans. Graph.*, 33(4):100:1–100:10, July 2014.

[23] Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Trans. Graph.*, 27(5):130:1–130:8, December 2008.

[24] Toshiya Hachisuka, Jacopo Pantaleoni, and Henrik Wann Jensen. A path space extension for robust light transport simulation. *ACM Trans. Graph.*, 31(6):191:1–191:10, November 2012.

[25] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

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

[27] Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik P. A. Lensch, and Jaroslav Křivánek. Volume path guiding based on zero-variance random walk theory. *ACM Trans. Graph.*, 38(3):25:1–25:19, June 2019.

[28] Robert Herzog, Elmar Eisemann, Karol Myszkowski, and H.-P. Seidel. Spatio-temporal upsampling on the GPU. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, pages 91–98, New York, NY, USA, 2010. ACM.

[29] Binh-Son Hua, Adrien Gruson, Derek Nowrouzezahrai, and Toshiya Hachisuka. Gradient-domain photon density estimation. In *Computer Graphics Forum*, volume 36, pages 31–38. Wiley Online Library, 2017.

[30] Binh-Son Hua, Adrien Gruson, Victor Petitjean, Matthias Zwicker, Derek Nowrouzezahrai, Elmar Eisemann, and Toshiya Hachisuka. A survey on gradient-domain rendering. In *Computer Graphics Forum*, volume 38, pages 455–472. Wiley Online Library, 2019.

[31] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[32] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *SIGGRAPH Comput. Graph.*, 20(4):133–142, August 1986.

[33] Wenzel Jakob and Steve Marschner. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Trans. Graph.*, 31(4):58:1–58:13, July 2012.

[34] Adrian Jarabo and Diego Gutierrez. Bidirectional rendering of polarized light transport. In *Proceedings of the XXVI Spanish Computer Graphics Conference*, CEIG '16, pages 15–22, Goslar, Germany, 2016. Eurographics Association.

[35] Henrik Wann Jensen. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, pages 21–30, London, UK, 1996. Springer-Verlag.

[36] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[37] Herman Kahn and Andy W Marshall. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

[38] James T. Kajiya. The rendering equation. In *Computer Graphics*, pages 143–150, 1986.

[39] Nima Khademi Kalantari, Steve Bako, and Pradeep Sen. A machine learning approach for filtering Monte Carlo noise. *ACM Trans. Graph.*, 34(4):122–1, 2015.

[40] Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the Metropolis light transport algorithm. In *Computer Graphics Forum*, volume 21, pages 531–540. Wiley Online Library, 2002.

[41] Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. E-LPIPS: Robust perceptual image similarity via random transformation ensembles. *CoRR*, abs/1906.03973, 2019.

[42] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[43] Claude Knaus and Matthias Zwicker. Progressive photon mapping: A probabilistic approach. *ACM Trans. Graph.*, 30(3):25:1–25:13, May 2011.

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

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

[46] Jaakko Lehtinen, Matthias Zwicker, Emmanuel Turquin, Janne Kontkanen, Frédo Durand, François X. Sillion, and Timo Aila. A meshless hierarchical representation for light transport. *ACM Trans. Graph.*, 27(3):37:1–37:9, August 2008.

[47] Tzu-Mao Li, Yu-Ting Wu, and Yung-Yu Chuang. SURE-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph.*, 31(6):194:1–194:9, November 2012.

[48] Marco Manzi, Fabrice Rousselle, Markus Kettunen, Jaakko Lehtinen, and Matthias Zwicker. Improved sampling for gradient-domain Metropolis light transport. *ACM Trans. Graph.*, 33(6):178:1–178:12, November 2014.

[49] Marco Manzi, Delio Vicini, and Matthias Zwicker. Regularizing image reconstruction for gradient-domain rendering with feature patches. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics*, EG '16, pages 263–273, Goslar, Germany, 2016. Eurographics Association.

[50] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[51] Bochang Moon, Steven McDonagh, Kenny Mitchell, and Markus Gross. Adaptive polynomial rendering. *ACM Trans. Graph.*, 35(4):40:1–40:10, July 2016.

[52] Thomas Müller, Markus Gross, and Jan Novák. Practical path guiding for efficient light-transport simulation. *Comput. Graph. Forum*, 36(4):91–100, July 2017.

[53] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[54] Jacopo Pantaleoni. Charted Metropolis light transport. *ACM Trans. Graph.*, 36(4):75:1–75:14, July 2017.

[55] Victor Petitjean, Pablo Bauszat, and Elmar Eisemann. Spectral gradient sampling for path tracing. In *Computer Graphics Forum (Proceedings of EGSR)*, 2018.

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

[57] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[59] Fabrice Rousselle, Wojciech Jarosz, and Jan Novák. Image-space control variates for rendering. *ACM Trans. Graph.*, 35(6):169:1–169:12, November 2016.

[60] Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.*, 30(6):159:1–159:12, December 2011.

[61] Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. Adaptive rendering with non-local means filtering. *ACM Trans. Graph.*, 31(6):195:1–195:11, November 2012.

[62] Fabrice Rousselle, Marco Manzi, and Matthias Zwicker. Robust denoising using feature and color information. In *Computer Graphics Forum*, volume 32, pages 121–130. Wiley Online Library, 2013.

[63] Cecie Starr, Christine Evers, and Lisa Starr. Biology: Concepts and applications, 2006.

[64] Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, pages 1135–1151, 1981.

[65] Weilun Sun, Xin Sun, Nathan A. Carr, Derek Nowrouzezahrai, and Ravi Ramamoorthi. Gradient-domain vertex connection and merging. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*, EGSR '17, pages 83–92, Goslar, Germany, 2017. Eurographics Association.

[66] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, Jan 1998.

[67] Joran Van de Woestijne, Roald Frederickx, Niels Billen, and Philip Dutré. Temporal coherence for metropolis light transport. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*, EGSR '17, pages 55–63, Goslar, Germany, 2017. Eurographics Association.

[68] Eric Veach. *Robust Monte Carlo methods for light transport simulation*, volume 1610. Stanford University PhD thesis, 1997.

[69] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*, pages 145–167. Springer, 1995.

[70] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 419–428, New York, NY, USA, 1995. ACM.

[71] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 65–76, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[72] Thijs Vogels, Fabrice Rousselle, Brian Mcwilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. Denoising with kernel prediction and asymmetric loss functions. *ACM Trans. Graph.*, 37(4):124:1–124:15, July 2018.

[73] Jiří Vorba, Johannes Hanika, Sebastian Herholz, Thomas Müller, Jaroslav Křivánek, and Alexander Keller. Path guiding in production. In *ACM SIGGRAPH 2019 Courses*, SIGGRAPH '19, pages 18:1–18:77, New York, NY, USA, 2019. ACM.

[74] Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. *ACM Trans. Graph.*, 33(4):101:1–101:11, July 2014.

[75] Alexander Wilkie and Andrea Weidlich. How to write a polarisation ray tracer. In *SIGGRAPH Asia 2011 Courses*, SA '11, pages 8:1–8:36, New York, NY, USA, 2011. ACM.

[76] Yodchanan Wongsawat, Kamisetty R. Rao, and Soontorn Oraintara. Multichannel SVD-based image de-noising. In *2005 IEEE International Symposium on Circuits and Systems*, pages 5990–5993 Vol. 6, May 2005.

[77] Ruifeng Xu and Sumanta N. Pattanaik. A novel Monte Carlo noise reduction operator. *IEEE Computer Graphics and Applications*, 25(2):31–35, 2005.

[78] Henning Zimmer, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Path-space motion estimation and decomposition for robust animation filtering. In *Computer Graphics Forum*, volume 34, pages 131–142. Wiley Online Library, 2015.

[79] Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and S.-E. Yoon. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Comput. Graph. Forum*, 34(2):667–681, May 2015.

Errata

Publication I

The results and comparisons use a slightly different formula for relative MSE than described in Section 6: the errors are averaged over color channels instead of summing and the formula uses epsilon 0.01 instead of 0.001.

Realistic image synthesis, that is, the creation of realistic-looking virtual photographs of virtual environments by computational simulation of light, is present in the daily lives of most people in the form of movies and advertising. This thesis presents four new methods for synthesizing such virtual photographs. The novel methods work in the gradient-domain: In addition to sampling pixel colors, the methods directly evaluate finite differences between adjacent pixels and reconstruct the image as an integration problem. The presented methods may lead to cheaper production costs, improved image quality, and smaller environmental footprint.

Aalto-DD 5/2020

ISBN 978-952-60-8907-2 (printed)
ISBN 978-952-60-8908-9 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS