UNIVERSITY OF CALIFORNIA,
IRVINE

Microscale-based Macro-appearance Rendering and Its Inverse Problem

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Yu Guo

Dissertation Committee:
Professor Shuang Zhao, Chair
Professor Gopi Meenakshisundaram
Professor Charless Fowlkes

2021

DEDICATION

To Myself and My Family

TABLE OF CONTENTS

Page

LIST OF FIGURES **vi**

LIST OF TABLES **viii**

ACKNOWLEDGMENTS **ix**

VITA **x**

ABSTRACT OF THE DISSERTATION **xii**

1 Introduction **1**

2 Background **6**
2.1 Light Transport 6
2.1.1 Surface rendering equation 7
2.1.2 Volume rendering equation 7
2.2 Scattering Distribution Function 8
2.2.1 Bidirectional reflectance distribution function (BRDF) 9
2.2.2 Phase function 11
2.3 Maxwell Equations 11
2.3.1 Basic operators and notation 11
2.3.2 Derivation 12
2.4 Bayesian Inference 13
2.4.1 Maximum a Posteriori (MAP) 14
2.4.2 Markov Chain Monte Carlo (MCMC) 14

3 Layered Materials Rendering **17**
3.1 Introduction 18
3.2 Related Work 21
3.2.1 Discretized layered BSDFs 21
3.2.2 Analytic layered BSDFs 22
3.2.3 Microfacet models for interfaces 22
3.2.4 Capability comparison 23
3.3 Background and Overview 24
3.3.1 Assumptions 24

3.3.2 Review of Veach's path integral formulation . . . 25
3.3.3 Overview . . . 26
3.4 Position-Free Path Formulation . . . 27
3.4.1 Notation . . . 27
3.4.2 Position-free path integral . . . 28
3.4.3 Derivation . . . 31
3.4.4 Normal mapping . . . 33
3.4.5 Note about reciprocity . . . 34
3.4.6 Multiple slabs . . . 34
3.5 Our Estimators . . . 35
3.5.1 BSDF sampling . . . 35
3.5.2 BSDF evaluation . . . 36
3.5.3 Pdf estimation . . . 40
3.6 Applications and Results . . . 44
3.6.1 Validations . . . 44
3.6.2 Main Results . . . 45
3.6.3 Performance . . . 50
3.7 Conclusion . . . 50

4 Bulk Scattering in Volumetric Rendering **52**
4.1 Introduction . . . 53
4.2 Related Work . . . 55
4.3 Preliminaries . . . 57
4.3.1 Electromagnetic Scattering . . . 57
4.3.2 Foldy-Lax Equations . . . 59
4.4 Scattering from Clusters of Particles . . . 63
4.4.1 Relationship with the Radiative Transfer Theory . . . 66
4.4.2 Relationship with Independent Scattering . . . 68
4.5 Computing the Bulk Scattering Parameters . . . 69
4.6 Experiments . . . 70
4.6.1 Validation . . . 70
4.6.2 Main Results . . . 71
4.7 Conclusion . . . 79

5 Latent Representation for Materials **81**
5.1 Introduction . . . 82
5.2 Related work . . . 85
5.3 MaterialGAN: A Generative SVBRDF Model . . . 87
5.3.1 Overview of StyleGAN and its latent spaces . . . 88
5.3.2 MaterialGAN training . . . 89
5.4 SVBRDF Capture using MaterialGAN . . . 90
5.4.1 Incorporating the MaterialGAN prior . . . 92
5.4.2 Latent space . . . 93
5.4.3 Optimization strategy . . . 95
5.4.4 Initialization . . . 97

5.5 Results . . . 99
5.5.1 Comparison with prior work on real data . . . 102
5.5.2 Additional comparisons . . . 104
5.6 Conclusion . . . 108

6 Procedural Parameters for Materials **109**
6.1 Introduction . . . 110
6.2 Related Work . . . 112
6.3 Preliminaries . . . 115
6.4 Summary Functions . . . 117
6.5 Bayesian Inference of Material Parameters . . . 119
6.5.1 Bayesian formulation . . . 119
6.5.2 Point estimate of parameter values . . . 121
6.5.3 Markov-Chain Monte Carlo Sampling of the Posterior . . . 121
6.6 Material Models and Results . . . 123
6.6.1 Similarity Relations in Translucency . . . 124
6.6.2 Procedural Material Models . . . 125
6.6.3 Additional Comparisons . . . 130
6.7 Conclusion . . . 134

7 Conclusion and Future work **135**

Bibliography **138**

Appendix A Appendix for Chapter 3 **149**
A.1 Detailed Derivations . . . 149
A.2 Efficient Weight Computation . . . 150
A.3 MIS with stochastic function and weight evaluation . . . 153

LIST OF FIGURES

Page

1.1 Examples of complex materials 2

3.1 Teaser of LayeredBSDF 17
3.2 Equal-time comparisons 20
3.3 Comparison to previous work 23
3.4 Small displacement assumption 24
3.5 Outgoing lobes of a layered BSDF 27
3.6 Example paths 29
3.7 Our Monte Carlo estimators 38
3.8 Validation of our pdf estimates 43
3.9 Multiple importance sampling 43
3.10 White furnace tests 45
3.11 Top vs. bottom height variation 46
3.12 Reflection and transmission A 47
3.13 Reflection and transmission B 47
3.14 Anisotropic media within layers 49
3.15 Comparison to volumetric cloth 49
3.16 Multi-layer BSDF 50

4.1 Teaser 52
4.2 Schematical representation of the particles scattering geometry 61
4.3 Comparison against Lorenz-Mie theory 70
4.4 Comparison against Lorenz-Mie theory 2 71
4.5 Comparison for different parameters 72
4.6 Renderings of homogeneous Lucy models 73
4.7 Comparison with different radius distribution 75
4.8 Multi-spectral results 76
4.9 Multi-spectral rendering of homogeneous Lucy model 77
4.10 Visualizations of phase function 77
4.11 Renderings of homogeneous Lucy models 78
4.12 Correlated particles 79

5.1 Teaser 81
5.2 Materials generated by MaterialGAN 84
5.3 Interpolation in MaterialGAN latent space 90

5.4 Inverse rendering pipeline . . . 91
5.5 Embedding SVBRDFs into different latent spaces . . . 94
5.6 Optimization strategy . . . 96
5.7 Noise optimization vs. post-refinement . . . 98
5.8 Visualization of our constant initializations . . . 99
5.9 SVBRDF reconstruction on synthetic data . . . 100
5.10 SVBRDF reconstruction on real data . . . 101
5.11 Performance statistics . . . 103
5.12 SVBRDF results with different initialization . . . 104
5.13 Per-pixel post-refinement . . . 105
5.14 Performance using different numbers of input images (synthetic data) . . . 106
5.15 Performance using different numbers of input images (real data) . . . 107
5.16 Material interpolation . . . 108

6.1 Teaser . . . 109
6.2 Pipeline . . . 111
6.3 L2 norm difference . . . 117
6.5 Synthetic results . . . 126
6.6 Synthetic results with discrete parameters . . . 127
6.7 Real results . . . 128
6.8 Comparison with mismatched model . . . 130
6.9 Comparison with Aittala et al . . . 131
6.10 Comparison to Hu et al . . . 131
6.11 Initialization . . . 132
6.12 Comparison to Deschaintre et al . . . 133
6.13 Quantitative evaluation . . . 134

LIST OF TABLES

Page

2.1 List of radiometry quantities 6
2.2 List of Maxwell notations 12

3.1 Notation used in §3 28
3.2 Render times of all our results 51

4.1 Notation used in §4 58
4.2 Performance statistics for our simulation 71

5.1 Accuracy of the novel-view renderings 102

6.1 Performance 125

ACKNOWLEDGMENTS

I would like to thank my advisor Shuang Zhao for his patient guidance and unconditional support.

I am sincerely grateful to Miloš Hašan for his inspiration and offering me opportunities for my internships in Autodesk and Adobe. I would also like to thank my other committee members Gopi Meenakshisundaram and Charless Fowlkes for the constructive advice they have provided.

I want to thank all the collaborator from my publications. And my intership mentors.

I would like to thank my labmates and other friends.

Finally, I thank my wife and my parents.

This work was supported in part by NSF grant IIS-1813553, Autodesk Inc., Adobe Inc., University of Zaragoza (Spain) and Department of Computer Science, UC Irvine.

Chapter 3 is based on the material as it appears in ACM Transactions on Graphics, 2018 ("Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs", Yu Guo, Miloš Hašan and Shuang Zhao). The dissertation author was the primary investigator and author of this paper.

Chapter 4 is based on the material as it appears in ACM Transactions on Graphics, 2021 ("Beyond Mie Theory: Systematic Computation of Bulk Scattering Parameters based on Microphysical Wave Optics", Yu Guo, Adrian Jarabo and Shuang Zhao). The dissertation author was the primary investigator and author of this paper.

Chapter 5 is based on the material as it appears in ACM Transactions on Graphics, 2020 ("MaterialGAN: Reflectance Capture using a Generative SVBRDF Model", Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli and Shuang Zhao). The dissertation author was the primary investigator and author of this paper.

Chapter 6 is based on the material as it appears in Computer Graphics Forum, 2020 ("A Bayesian Inference Framework for Procedural Material Parameter Estimation", Yu Guo, Miloš Hašan, Lingqi Yan and Shuang Zhao). The dissertation author was the primary investigator and author of this paper.

This dissertation is based on a LaTeX template for thesis and dissertation documents at UC Irvine [105].

VITA

Yu Guo

EDUCATION

Doctor of Philosophy in Computer Science **2016 – 2021**
University of California, Irvine *Irvine, CA, US*

Master of Science in Computational Sciences **2010 – 2013**
University of Chinese Academy of Sciences *Beijing & Shenzhen, China*

Bachalar of Science in Applied Mathematics **2006 – 2010**
Central South University *Changsha, China*

RESEARCH EXPERIENCE

Research Associate **2013 – 2016**
Nanyang Technological University *Singapore*

REFEREED PUBLICATIONS

Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs **2018**
ACM Transactions on Graphics

MaterialGAN: Reflectance Capture using a Generative SVBRDF Model **2020**
ACM Transactions on Graphics

A Bayesian Inference Framework for Procedural Material Parameter Estimation **2020**
Computer Graphics Forum

Beyond Mie Theory: Systematic Computation of Bulk Scattering Parameters based on Microphysical Wave Optics **2021**
ACM Transactions on Graphics

ABSTRACT OF THE DISSERTATION

Microscale-based Macro-appearance Rendering and Its Inverse Problem

By

Yu Guo

Doctor of Philosophy in Computer Science

University of California, Irvine, 2021

Professor Shuang Zhao, Chair

Physically-based rendering has become mature and commonplace in recent decades. However, the rendered results look artificial and overly perfect. Better realism needs higher fidelity detailed geometry or model complexity, which substantially increases computational power and human works. To achieve higher physical realism and enable more effective material content creation, many techniques are developed in material reflection and scattering models. We put emphasis on accurately representing and reproducing the rich visual world from *micro*-level details to overall (*macro*) appearance.

The first half of the dissertation focuses on building the bridge from *micro* to *macro* world: we present an accurate appearance model for layered materials derived from microstructures to define their optical behavior and a general framework of bulk scattering in participating medium which considers the microscale effects. Contradictory, in the second half, we discuss the inverse problem of retrieving the micro parameters from photo-captured materials.

Our first work introduces a new unbiased layered BSDF model based on Monte Carlo simulation, whose only assumption is the layer assumption itself. Our novel position-free path formulation is fundamentally more powerful at constructing light transport paths than generic light transport algorithms applied to the particular case of flat layers. We introduce two techniques for sampling the position-free path integral, a forward path tracer with next-event

estimation and a full bidirectional estimator. We show several examples featuring multiple layers with surface and volumetric scattering, surface and phase function anisotropy, and spatial variation in all parameters.

Our second work presents a generalized framework capable of systematically and rigorously computing bulk scattering parameters beyond the far-field assumption of the Lorenz-Mie theory. Our technique accounts for microscale wave-optics effects such as diffraction and interference and interactions between nearby particles. Our framework is general, can be plugged in any renderer supporting Lorenz-Mie scattering, and allows arbitrary packing rates and particle correlation; we demonstrate this generality by computing bulk scattering parameters for many materials, including anisotropic materials and correlated media.

Finally, we present *MaterialGAN*, a deep generative convolutional network based on StyleGAN2, trained to synthesize realistic SVBRDF parameter maps. We show that MaterialGAN can be used as a robust material prior in an inverse rendering framework: we optimize its latent representation to generate material maps that match the appearance of the captured images when rendered.

Furthermore, we explore the inverse rendering problem of procedural material parameter estimation from photographs, presenting a unified view of the problem in a Bayesian framework. In addition to computing point estimates of the parameters by optimization, our framework uses a Markov Chain Monte Carlo approach to sample the space of plausible material parameters.

Chapter 1

Introduction

Rendering is one of the three fundamental problems in computer graphics accompany by modeling and simulation. It builds the bridge between the 3D virtual world and the images showed on display. In recent decades, path-tracing-based photorealistic rendering became a general and standard technique in the movie and animation industries. Most commercial Ads using rendered images instead of captured photos. We know how light interacts with surfaces and participating media from the principles of physical rules (especially in optics). As a result, an indistinguishable virtual world could be established using computers.

As we enjoy the realism of the virtual world from pioneers' works, more challenges await us. The real world contains many types of materials, while only a few can be modeled in renderings, such as diffuse, specular, transparent, and so on. In most cases, people can quickly tell the object in an image is rendered since it looks *too perfect to be true*. The over-perfect issue is because of details missing for complex materials (See Figure 1.1). Artists always hack the macro appearance with texture mapping or mix the existing appearance models, which is ad-hoc and unrealistic since microstructures of a surface or medium bring undesirable light interaction and further affect the macro appearance. In this dissertation, we

first address a more general but efficient way to handle complex surface reflectance (layered material) and volumetric scattering with micro details.

Figure 1.1: Examples of complex materials in real life. **Left:** car with 'mermaid' paint; **Right:** cloud iridescence phenomena.

To better represent the real world, another challenge is to acquire the geometry of the objects, the physical properties of the materials, and the scene illumination from observed measurements more accurately and efficiently. This inverse process of rendering (inverse rendering) becomes a popular topic in Computer Graphics and Computer Vision. In industry, artists use *photoshop* to create material maps or use some heavy capturing systems to acquiring them. In the second half of this dissertation, we will focus on material properties estimation by just giving a small number of input cellphone captures.

To summarize, we develop a smart technique to render layered materials, a framework to compute scatterings in participating media based on wave optics, an optimization-based method for SVBRDF (Spatially Varying Bidirectional Reflectance Distribution Functions, as will be introduced in Chapter 2) reconstruction and then extend it to posterior estimation using Bayesian inference. These techniques were presented at multiple ACM SIGGRAPH (Asia) conferences [49, 50, 51] and Pacific Graphics [48]. Our specific contributions include:

Position-free Monte Carlo simulation for arbitrary layered BSDFs. Real-world materials are often layered: metallic paints, biological tissues, and many more. Variation in the interface and volumetric scattering properties of the layers leads to a rich diversity of material appearances from anisotropic highlights to complex textures and relief patterns. However, simulating light-layer interactions is a challenging problem. Past analytical or numerical solutions either introduce several approximations and limitations, or rely on expensive operations on discretized BSDFs, preventing the ability to freely vary the layer properties spatially. In Chapter 3, we introduce a new unbiased layered BSDF model based on Monte Carlo simulation, whose only assumption is the layer assumption itself. Our novel position-free path formulation is fundamentally more powerful at constructing light transport paths than generic light transport algorithms applied to the special case of flat layers, since it is based on a product of solid angle instead of area measures, so does not contain the high-variance geometry terms needed in the standard formulation. We introduce two techniques for sampling the position-free path integral, a forward path tracer with next-event estimation and a full bidirectional estimator. We show a number of examples, featuring multiple layers with surface and volumetric scattering, surface and phase function anisotropy, and spatial variation in all parameters.

Beyond Mie theory: systematic computation of bulk scattering parameters based on microphysical wave optics. Light scattering in participating media and translucent materials is typically modeled using the radiative transfer theory. Under the assumption of independent scattering between particles, it utilizes several bulk scattering parameters to statistically characterize light-matter interactions at the macroscale. To calculate these parameters based on microscale material properties, the Lorenz-Mie theory has been considered the gold standard. In Chapter 4, we present a generalized framework capable of systematically and rigorously computing bulk scattering parameters beyond the far-field assumption of Lorenz-Mie theory. Our technique accounts for microscale wave-optics effects

such as diffraction and interference as well as interactions between nearby particles. Our framework is general, can be plugged in any renderer supporting Lorenz-Mie scattering, and allows arbitrary packing rates and particles correlation; we demonstrate this generality by computing bulk scattering parameters for a wide range of materials, including anisotropic and correlated media.

MaterialGAN: reflectance capture using a generative SVBRDF model. We address the problem of reconstructing spatially-varying BRDFs from a small set of image measurements. This is a fundamentally under-constrained problem, and previous work has relied on using various regularization priors or on capturing many images to produce plausible results. In Chapter 5, we present *MaterialGAN*, a deep generative convolutional network based on StyleGAN2, trained to synthesize realistic SVBRDF parameter maps. We show that MaterialGAN can be used as a powerful material prior in an inverse rendering framework: we optimize in its latent representation to generate material maps that match the appearance of the captured images when rendered. We demonstrate this framework on the task of reconstructing SVBRDFs from images captured under flash illumination using a hand-held mobile phone. Our method succeeds in producing plausible material maps that accurately reproduce the target images, and outperforms previous state-of-the-art material capture methods in evaluations on both synthetic and real data. Furthermore, our GAN-based latent space allows for high-level semantic material editing operations such as generating material variations and material morphing.

A Bayesian Inference Framework for Procedural Material Parameter Estimation. Procedural material models have been gaining traction in many applications thanks to their flexibility, compactness, and easy editability. In Chapter 6, we explore the inverse rendering problem of procedural material parameter estimation from photographs, presenting a unified view of the problem in a Bayesian framework. In addition to computing point estimates of

the parameters by optimization, our framework uses a Markov Chain Monte Carlo approach to sample the space of plausible material parameters, providing a collection of plausible matches that a user can choose from, and efficiently handling both discrete and continuous model parameters. To demonstrate the effectiveness of our framework, we fit procedural models of a range of materials—wall plaster, leather, wood, anisotropic brushed metals and layered metallic paints—to both synthetic and real target images.

The dissertation is organized as follows. We first introduce the basic background on light transport, wave-optics and BRDF representations in Chapter 2. From Chapters 3 to 6, we present technical details of our layered rendering, wave-optics bulk scattering, SVBRDF reconstruction and procedure model estimation, respectively. Finally, we present our conclusion and discuss future research directions in Chapter 7.

Chapter 2

Background

In this chapter, we briefly review some background knowledge closely related to this dissertation. Firstly, we recap the fundamental light transport theory and Bidirectional Reflectance Distribution Function (BRDF). Then we introduce Maxwells' equation in wave optics. Finally we talk about some concepts in Markov Chain Monte Carlo (MCMC) methods.

2.1 Light Transport

Radiometry is a set of techniques for measuring electromagnetic radiation, and we use is to measure the energy of visible lights in nowadays renderings. First we list some important Radiometric quantities and then describe rendering equations in the following sections.

Table 2.1: List of radiometry quantities.

Quantity	Symbol	Unit	Notes
Flux(Power)	Φ	W	Radiant energy emitted, reflected, transmitted or received, per unit time.
Irradiance	E	W/m^2	Flux received by a surface per unit area.
Radiance	L	$W/(Sr \cdot m^2)$ *	Flux per unit solid angle per unit projected area.

* watt per steradian per square meter.

2.1.1 Surface rendering equation

To render photorealistic image, a key concept is to simulate light transport, which models the light interaction between camera/eyes, scene objects and lightsource. For any point in the scene, we want to know its spectral radiance $L_{\mathrm{o}}(\boldsymbol{r}, bmomegao, \lambda, t)$ of wavelength λ directed outward along direction $\boldsymbol{\omega}_{\mathrm{o}}$ at time t, from a particular position $\boldsymbol{r}$. For simplisity, the commonly used *rendering equation* (RE) [71] for surface have two assumptions: geometric optics only and steady state. Therefore, we reformulate the light radiance as a $5D$ function of position ($\boldsymbol{r}$) and direction ($\boldsymbol{\omega}_{\mathrm{o}}$), the outgoing radiance (L_{o}) is the sum of the emitted radiance (L_{e}) and the reflected radiance (L_{r}). The reflected radiance itself is the sum of all directions of incoming radiance (L_{i}) weighted by the surface reflection (f_{r}) and cosine of incident angle.

$$L_{\mathrm{o}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}}) = L_{\mathrm{e}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}}) + L_{\mathrm{r}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}}) \tag{2.1}$$

$$= L_{\mathrm{e}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}}) + \int_{\mathbb{S}^2} L_{\mathrm{i}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{i}}) f_{\mathrm{r}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) \langle \boldsymbol{n}(\boldsymbol{r}), \boldsymbol{\omega}_{\mathrm{i}} \rangle \, \mathrm{d}\boldsymbol{\omega}_{\mathrm{i}} \tag{2.2}$$

Note that, $\boldsymbol{\omega}_{\mathrm{o}}$ is the direction of the outgoing light, and $\boldsymbol{\omega}_{\mathrm{i}}$ is the negative direction of the incoming light.

The rendering equation can fully model the light transport in a space without any participating media. It is popular to expand this integral equation to *path integral formulation* and solve it using Monte Carlo methods (see Veach's thesis [125]).

2.1.2 Volume rendering equation

When light travels in a participation medium (e.g., smoke, marble and skin), we use *radiative transfer equation* (RTE) [18] to describe how the radiance changes by four types of interaction

events: emission, absorption, out-scattering, and in-scattering.

$$(\boldsymbol{\omega}_{\mathrm{o}} \cdot \nabla) L_{\mathrm{o}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}}) = \overbrace{\sigma_{\mathrm{s}}(\boldsymbol{r}) \int_{\mathbb{S}^2} L_{\mathrm{i}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{i}}) f_{\mathrm{p}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) \, \mathrm{d}\boldsymbol{\omega}_{\mathrm{i}}}^{\text{a) In-scattering}} \tag{2.3}$$

$$\overbrace{-\sigma_{\mathrm{s}}(\boldsymbol{r}) L_{\mathrm{o}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}})}^{\text{b) Out-scattering}} \overbrace{-\sigma_{\mathrm{a}}(\boldsymbol{r}) L_{\mathrm{o}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}})}^{\text{c)Absorption}} \overbrace{+L_{\mathrm{e}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{o}})}^{\text{d)Emission}} \tag{2.4}$$

The RTE is a integro-differential equation which can be derived via conservation of energy. Briefly, the RTE states that a beam of light loses energy through divergence and extinction (including both absorption (c) and scattering (b) away from the beam) and gains energy from light sources (d) in the medium and scattering (a) directed towards the beam. Same as RE, coherence, polarization and light speed are neglected. Optical properties such as refractive index (m), absorption coefficient (σ_{a}), scattering coefficient (σ_{s}) are taken as time-invariant but may vary spatially. In addition, we define the extinction coefficient $\sigma_{\mathrm{t}} = \sigma_{\mathrm{a}} + \sigma_{\mathrm{s}}$, and the ratio between σ_{s} and σ_{t} controls the fraction of radiant energy not being absorbed at each scattering and is also known as the single-scattering albedo (a). We use phase functions $f_{\mathrm{p}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}})$ to describe the directional distribution of light scattered in a medium.

It is desirable to rewrite the RTE as an integral equation, which can them be solved numerically using Monte Carlo methods (see Veach's thesis [125]).

2.2 Scattering Distribution Function

In RE and RTE, an important term is still missing. When light hit a surface or a particle in the medium, how does the light scatter, or in other words, redistribute both in energy and direction? To model this scattering effect, we use *bidirectional reflectance distribution function* (BRDF) for surface interaction and *phase function* (PF) for light scattering in a medium.

2.2.1 Bidirectional reflectance distribution function (BRDF)

The BRDF is a 4-D function that defines how light is reflected at an opaque surface.

$$f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) = \frac{\mathrm{d}L_{\mathrm{o}}(\boldsymbol{\omega}_{\mathrm{o}})}{\mathrm{d}E_{\mathrm{i}}(\boldsymbol{\omega}_{\mathrm{i}})} = \frac{\mathrm{d}L_{\mathrm{o}}(\boldsymbol{\omega}_{\mathrm{o}})}{L_{\mathrm{i}}(\boldsymbol{\omega}_{\mathrm{i}})\langle \boldsymbol{n}, \boldsymbol{\omega}_{\mathrm{i}} \rangle \, \mathrm{d}\boldsymbol{\omega}_{\mathrm{i}}} \tag{2.5}$$

The function takes an incoming light direction $\boldsymbol{\omega}_{\mathrm{i}}$, and outgoing direction $\boldsymbol{\omega}_{\mathrm{o}}$, and returns the ratio of reflected radiance (L_{o}) exiting along $\boldsymbol{\omega}_{\mathrm{o}}$ to the irradiance incident (L_{i}) on the surface from direction $\boldsymbol{\omega}_{\mathrm{i}}$. Each direction $\boldsymbol{\omega}$ is itself parameterized by azimuth angle φ and polar angle θ. $\boldsymbol{n}$ is the (macro) surface normal.

Physically based BRDFs have several properties, including,

Positivity:

$$f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) \geq 0 \tag{2.6}$$

Reciprocity:

$$f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) = f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{o}} \to \boldsymbol{\omega}_{\mathrm{i}}) \tag{2.7}$$

Conserving energy:

$$\forall \boldsymbol{\omega}_{\mathrm{o}}, \int_{\mathbb{S}}^{2} f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) \langle \boldsymbol{n}, \boldsymbol{\omega}_{\mathrm{i}} \rangle \leq 1 \tag{2.8}$$

Some basic BRDFs and the BRDFs used in this dissertation are listed below:

Lambertian BRDF distribute the incident energy equally towards all the outgoing directions and give a diffuse appearance.

$$f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) = k_{\mathrm{d}} \tag{2.9}$$

where k_{d} is the albedo or absorption of light which will introducing the color.

Phong and Blinn-Phong BRDF adds a specular component to introduce glossy effect.

$$f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) = k_{\mathrm{d}} + k_{\mathrm{s}}(\boldsymbol{\omega}_{\mathrm{ir}} \cdot \boldsymbol{\omega}_{\mathrm{o}})^n \tag{2.10}$$

where $\boldsymbol{\omega}_{\mathrm{ir}}$ is the reflection of incident light and larger n will increase the glossiness of the material.

Microfacet BRDF is the state-of-the-art model which is widely used in all kinds of renderers. The microfacet theory assumes that all surfaces are formed by tiny microfacets that are perfectly specular that reflect rays like perfectly smooth mirrors.

$$f_{\mathrm{r}}(\boldsymbol{\omega}_{\mathrm{i}} \to \boldsymbol{\omega}_{\mathrm{o}}) = \frac{F(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{h}) G(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}, \boldsymbol{h}) D(\boldsymbol{h})}{4\langle \boldsymbol{n}, \boldsymbol{\omega}_{\mathrm{i}} \rangle \langle \boldsymbol{n}, \boldsymbol{\omega}_{\mathrm{o}} \rangle} \tag{2.11}$$

where $\boldsymbol{h}$ is the half vector that $\boldsymbol{h} = (\boldsymbol{\omega}_{\mathrm{i}} + \boldsymbol{\omega}_{\mathrm{o}})/2$. The first component F is Fresnel term, G is the geometry term (shading factor) and D is *normal distribution function* (NDF) which indicate the distribution of microfacets normals. With the change of statistics of the micro-geometry, the macro-appearance changes accordingly. All NDF should follow:

$$\int_{\mathbb{S}}^{2} D(\boldsymbol{h}) \langle \boldsymbol{n}, \boldsymbol{h} \rangle \, \mathrm{d}\boldsymbol{h} = 1 \tag{2.12}$$

There're two forms of NDF we used in most of the papers, *Beckmann* and *GGX*.

BRDF is a special case for opaque surface with reflection only. It can be extend to *bidirectional transmittance distribution function* (BTDF) for opposite side of the surface, and *bidirectional scattering distribution function* (BSDF), a superset and generalization of BRDF and BTDF.

Spatially varying BRDF The *spatially varying BRDF* (SVBRDF) is a 6-D function, $f_{\mathrm{r}}(\boldsymbol{r}, \boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}})$, where $\boldsymbol{r}$ describes a 2D location over an object's surface.

2.2.2 Phase function

Phase function is usually parameterized as a function of the angle (θ) between $\boldsymbol{\omega}_{\mathrm{i}}$ and $\boldsymbol{\omega}_{\mathrm{o}}$, to model how light scattered in medium. A common phase function is *Henyey-Greenstain* (HG) phase function with parameter $-1 < g < 1$:

$$f_{\mathrm{p}}(\theta, g) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g\cos\theta)^{3/2}} \tag{2.13}$$

2.3 Maxwell Equations

2.3.1 Basic operators and notation

The differential operator given in Cartesian coordinates $\{x, y, z\}$:

$$\nabla = \frac{\partial}{\partial x}\mathbf{i} + \frac{\partial}{\partial y}\mathbf{j} + \frac{\partial}{\partial z}\mathbf{k} \tag{2.14}$$

For a scalar function $f(x, y, z)$ and a vector field $\mathbf{F}(x, y, z) = f_1(x, y, z)\mathbf{i} + f_2(x, y, z)\mathbf{j} + f_3(x, y, z)\mathbf{k}$, we have,

Gradient:

$$\nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j} + \frac{\partial f}{\partial z}\mathbf{k} \tag{2.15}$$

Divergence:

$$\nabla \cdot \mathbf{F} = \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} \tag{2.16}$$

Curl:

$$\nabla \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ f_1 & f_2 & f_3 \end{vmatrix} \tag{2.17}$$

Laplace operator: $\nabla^2 f = \nabla \cdot (\nabla f)$

Curl of Curl: $\nabla \times (\nabla \times \mathbf{F}) = \nabla(\nabla \cdot \mathbf{F}) - \nabla \cdot (\nabla \mathbf{F}) = -\nabla \cdot (\nabla \mathbf{F}) = -\nabla^2 \mathbf{F}$

2.3.2 Derivation

Table 2.2: List of Maxwell notations.

Symbol	Unit	Notes
$\mathbf{E}$	V/m	Electric field
$\mathbf{H}$	A/m	Magnetic field
$\mathbf{D}$	C/m^2	Electric displacement
$\mathbf{B}$	Wb/m^2	Magnetic induction
$\mathbf{J}$	A/m^2	Electric current density
$\mathbf{M}$		Magnetic current density
$\mathbf{P}$		Electric polarization
ρ	C/m^3	Electric charge density
q	Wb/m^3	Magnetic charge density
ε_0	F/m	Electric permittivity of free space ($= 8.854187817 \times 10^{-12}$)
μ_0	H/m	Magnetic permeability of free space ($= 4\pi \times 10^{-7}$)

The mathematical description of Maxwell's equations are [16]:

$$\begin{aligned} \nabla \cdot \mathbf{D} &= \rho & \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} & \nabla \times \mathbf{H} &= \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \end{aligned} \tag{2.18}$$

where, $\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P}$ and $\mathbf{H} = \frac{1}{\mu_0}\mathbf{B} - \mathbf{M}$.

In free space, the polarization ($\mathbf{P}$) and magnetization ($\mathbf{M}$) vanish identically. And if there is no Electric charge density (ρ) and Electric current density ($\mathbf{J}$), we rewrite *Maxwell equation*

in the form of $\mathbf{E}$ and $\mathbf{H}$,

$$\begin{aligned} \nabla \cdot \mathbf{E} &= 0 & \nabla \cdot \mathbf{H} &= 0 \\ \nabla \times \mathbf{E} &= -\mu_0 \frac{\partial \mathbf{H}}{\partial t} & \nabla \times \mathbf{H} &= \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \end{aligned} \tag{2.19}$$

To consider Electric and Magnetic field as as time-harmonic (time variation is sinusoidal) fields with angular frequency of ω, which has the form of $\hat{\mathbf{u}} = \mathbf{u}e^{-i\omega t}$, the *Maxwell equation* become,

$$\begin{aligned} \nabla \cdot \mathbf{E} &= 0 & \nabla \cdot \mathbf{H} &= 0 \\ \nabla \times \mathbf{E} &= i\omega\mu_0 \mathbf{H} & \nabla \times \mathbf{H} &= -i\omega\varepsilon_0 \mathbf{E} \end{aligned} \tag{2.20}$$

Take the curl of (2.20),

$$\begin{aligned} \nabla \times (\nabla \times \mathbf{E}) &= i\omega\mu_0(\nabla \times \mathbf{H}) = \omega^2 \mu_0 \varepsilon_0 \mathbf{E} \\ \nabla \times (\nabla \times \mathbf{H}) &= -i\omega\varepsilon_0(\nabla \times \mathbf{E}) = \omega^2 \mu_0 \varepsilon_0 \mathbf{H} \end{aligned} \tag{2.21}$$

If we use the rule *Curl of Curl*, the *Maxwell equations* reduce to the Helmholtz equations,

$$\nabla^2 \mathbf{E} + k^2 \mathbf{E} = 0 \qquad \nabla^2 \mathbf{H} + k^2 \mathbf{H} = 0 \tag{2.22}$$

where $k = \omega/c$, and $c = \frac{1}{\sqrt{\mu_0 \varepsilon_0}}$ is the light speed in vacuum.

2.4 Bayesian Inference

Bayesian inference is a paradigm for constructing statistical models based on Bayes' Theorem

$$p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})} \propto p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{2.23}$$

Generally speaking, the goal of Bayesian inference is to estimate the posterior distribution ($p(\boldsymbol{\theta}|\mathbf{X})$) given the likelihood ($p(\mathbf{X}|\boldsymbol{\theta})$) and the prior distribution ($p(\boldsymbol{\theta})$). The likelihood is something that can be estimated from the training data.

2.4.1 Maximum a Posteriori (MAP)

In most of cases, we actually seek to maximize the posterior distribution which takes the existing data as fixed and determines the probability of any parameter setting $\boldsymbol{\theta}$ given that data $\mathbf{X}$. We call this process *Maximum a Posteriori* (MAP), an iterative process which updates the model's parameters in an attempt to maximize the probability of matching data to its distribution. Which is exactly the training pocess in a regular machine learning model.

MAP estimates can be computed via numerical optimization such as the conjugate gradient method or Newton's method. This usually requires first or second derivatives, which have to be evaluated analytically or numerically.

2.4.2 Markov Chain Monte Carlo (MCMC)

While MAP is the first step towards fully Bayesian inference, it's still only computing what statisticians called a *point estimate*. The downside of point estimates is that they don't tell you much about a parameter other than its optimal setting. In reality, we often want to know other information, like how certain we are that a parameter's value should fall within this predefined range. Therefore, a number of fascinating Bayesian methods have been devised that can be used to sample (i.e. draw sample values) from the posterior distribution. The most famous of these is an algorithm called *Markov Chain Monte Carlo* (MCMC).

In statistics, MCMC methods comprise a class of algorithms for sampling from a probability distribution. By constructing a Markov chain that has the desired distribution as its equi-

librium distribution, one can obtain a sample of the desired distribution by recording states from the chain. The more steps are included, the more closely the distribution of the sample matches the actual desired distribution.

MCMC is used to simulate physical systems with Gibbs canonical distribution (**we will start to use x instead of $\boldsymbol{\theta}$ from here**):

$$p(\mathbf{x}) \propto \exp\left(-\frac{U(\mathbf{x})}{T}\right) \tag{2.24}$$

Probability $p(\mathbf{x})$ of a system to be in the state $\mathbf{x}$ depends on the energy of the state $U(\mathbf{x})$ and temperature T. Any distribution can be rewritten as Gibbs canonical distribution, but for many problems such energy-based distributions appear very naturally. The goal becomes learning to sample from the canonical distribution. System has higher probability of staying in the states with lower energies, so minimize energy is the same as maximum a posteriori.

Metropolis-Hastings (MH) algorithm for MCMC is the simplest Markov Chain process that can sample from the distribution picks the neighbour of the current state and either accepts it or rejects depending on the change in energy. Algorithm produces a chain of states: $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$. Each time a candidate from a neighborhood of the last state is selected $\mathbf{x}'_n = \mathbf{x}_n + \varepsilon$ (ϵ is usually taken to be Gaussian with some spread σ). With probability $p = \min\left[1, \exp\left(\frac{U(\mathbf{x}_n)-U(\mathbf{x}'_n)}{T}\right)\right]$, system accepts new state (jumps to the new state): $\mathbf{x}_{n+1} = \mathbf{x}'_n$ and with probability $1-p$ new state is rejected: $\mathbf{x}_{n+1} = \mathbf{x}_n$. Note, when energy is lower in new state $U(\mathbf{x}'_n) < U(\mathbf{x}_n)$, it is always accepted: $p = 1$. In this way, we give preference to the states with lower energies, while not restricting the algorithm to always decrease the energy. The lower temperature, the lower probability to increase energy.

Sampling high-dimensional distributions with MH becomes very inefficient in practice. A more efficient scheme is

Hamiltonian Monte Carlo (HMC) algorithm, is also known as Hybrid Monte Carlo. Velocity v is added to the parameters describing the system. Energy of the system consists of potential and kinetic parts: $E(\mathbf{x}, \mathbf{v}) = U(\mathbf{x}) + K(\mathbf{v})$. Thus, velocities $\mathbf{v}$ and positions $\mathbf{x}$ have independent canonical distributions:

$$p(\mathbf{x}, \mathbf{v}) \propto \exp\left(\frac{-E(\mathbf{x}, \mathbf{v})}{T}\right) = \exp\left(\frac{-U(\mathbf{x})}{T}\right) \exp\left(\frac{-K(\mathbf{v})}{T}\right) \propto p(\mathbf{x})\, p(\mathbf{v}). \tag{2.25}$$

So once it can be sampled from joint distribution $p(\mathbf{x}, \mathbf{v})$, $\mathbf{x}$ can be also sampled by ignoring computed velocities $\mathbf{v}$. After initializing the system parameters $\mathbf{x}, \mathbf{v}$, it could be evolved using physics equations: $\dot{x}_i = v_i$, $m\dot{v}_i = -\frac{\partial U(\mathbf{x})}{\partial x_i}$. During a long period of time, it will not get a canonical distribution by collecting system states, because energy E is conserved in the system. At some points, velocity is resampled from $p(\mathbf{v})$, thus changing the total energy and resample the parameters. Sampling from $p(\mathbf{v})$ is very simple, because $\mathbf{v}$ is normally distributed.

HMC uses not only energy $U(\mathbf{x})$, but also it's gradient. So the 'price' of a single iteration is higher, but HMC is still significantly more efficient than MH. In most cases HMC accepts new states, but still, it has problems with sampling from distributions with isolated local minimum and discrete parameters (no gradient provided).

Metropolis-Adjusted Langevin Algorithm (MALA) is based on *Langevin Monte Carlo* (LMC). Different from gradient-based HMC, LMC uses a discrete Markov chain, which is equivalent to a gradient ascent procedure with injected Gaussian noise [88]. The injected noise prevents the chain from collapsing to just the (local) maximum. Due to discretization error, the Markov chain is not guaranteed to converge to the same stationary distribution as the continuous process. This can be corrected by using the Metropolis-Hasting rule to accept or reject states of the chain. This approach, known as the *Metropolis-adjusted Langevin algorithm* (MALA).

Chapter 3

Layered Materials Rendering

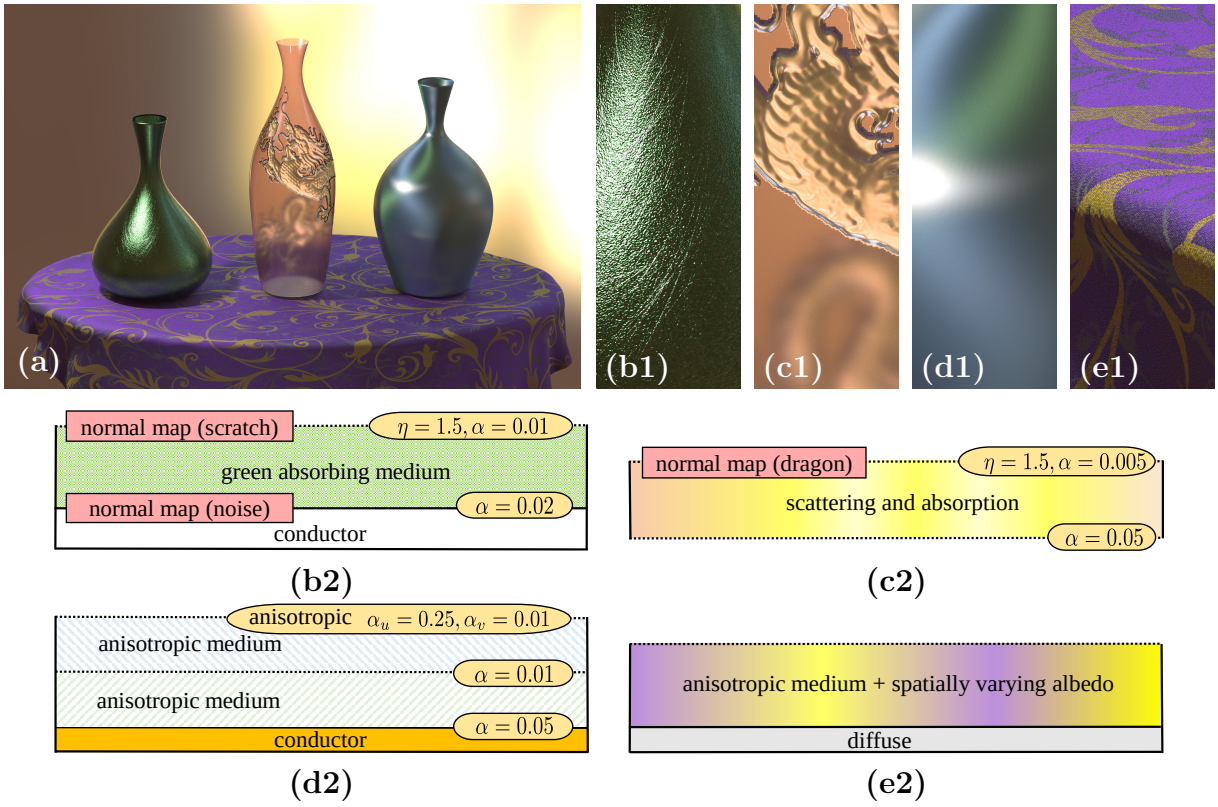


Figure 3.1: We introduce a new BSDF model leveraging an efficient Monte Carlo simulation algorithm applied locally to layered geometries. Our model enjoys the flexibility of using arbitrary layer interfaces and internal media and is capable of reproducing a wide variety of appearances. This example contains three vases on a tablecloth, all described using our BSDF model (see the insets for layer configurations).

3.1 Introduction

Physically-based shading models have become mature and commonplace in recent years across a number of rendering applications, within entertainment, architecture, and industrial design. However, we are seeing constant progress in the area of material reflection and scattering models, aiming to achieve higher physical realism and to enable more effective material content creation.

Many real world materials are comprised of thin layers with varying compositions. For example, metallic paint is a dielectric coating covering a metallic substrate composed of randomly oriented aluminum flakes; the absorption and scattering properties of the dielectric layer give the material its color and modify its directional scattering properties as well. Many biological materials (e.g. plant leaves) are also layered, and their appearance is a complex combination of the absorption properties, scattering phase function, air-material interface roughness, and thickness variation. Different characteristics of such interfaces and volumetric scattering properties can produce richly diverse material appearances from anisotropic highlights to complex textures. Furthermore, detailed layer thickness variations, scratches and bumps on the layer interfaces give these materials additional richness. Accurately understanding and simulating these interactions is therefore key to further progress in the rendering of materials.

However, explicitly simulating light-layer interactions by modeling the full geometry of these layers would be very expensive and cumbersome. The complex and spatially varying interface and internal microgeometries are much too costly to describe and simulate using standard 3D scene modeling tools such as triangle meshes and volumetric grids. Furthermore, due to the presence of multiple refractive interfaces, it can be very challenging to correctly construct light transport paths that connect light scattering locations to light sources, a key operation in most practical Monte Carlo rendering systems. Cheap approximations to these light transport problems (e.g. ignoring refraction, or composing layers using simple blending) are

not sufficient to achieve true realism.

A few techniques have been developed to address this problem. Weidlich and Wilkie [131] construct a simple and flexible analytical model. However, significant approximations are necessary; interface roughness is not fully handled for transmission, and no volumetric scattering is supported. The work of Belcour [12] recently introduced a more advanced approach based on tracking low-order moments of the BSDF lobes; however, it still introduces some approximations and limitations. On the other hand, Jakob et al. [66] (with a recent follow-up [139]) introduce a solution that is very accurate, but expensive: it represents BSDFs as discretized datasets and relies on expensive Fourier-domain operations on these to implement layer composition and thickness adjustment. This makes free spatial variation of the layer properties prohibitively expensive: a significant limitation in practice.

In this chapter, we introduce a new layered BSDF model without the above limitations. Our model provides an accurate, unbiased solution; to our knowledge, it is the only such model. Unlike previous work, we do not attempt to derive an analytic model for the BSDF lobe shapes. Instead, inside the evaluation and sampling routines of the layered BSDF, we run a Monte Carlo simulation of light transport within flat slabs. This is substantially faster than explicitly constructing the layer geometry, because no expensive scene ray tracing is required. Our model computes an accurate solution of the layered light transport problem. It is based on physical interface and volume scattering models, conserves energy and is reciprocal when possible. It can also be easily integrated into standard Monte Carlo rendering systems. This requires no precomputation and thus can efficiently handle spatially varying appearances. It also supports the full range of editability of the layer properties, both interface and volumetric, and allows anisotropy in both interface BSDFs and phase functions. In fact, the only limiting assumption of our model is the layer assumption itself.

Our solution is fundamentally more powerful at constructing light transport paths than generic transport algorithms (e.g standard path tracing, bidirectional or Metropolis trans-

port); see Figure 3.2.

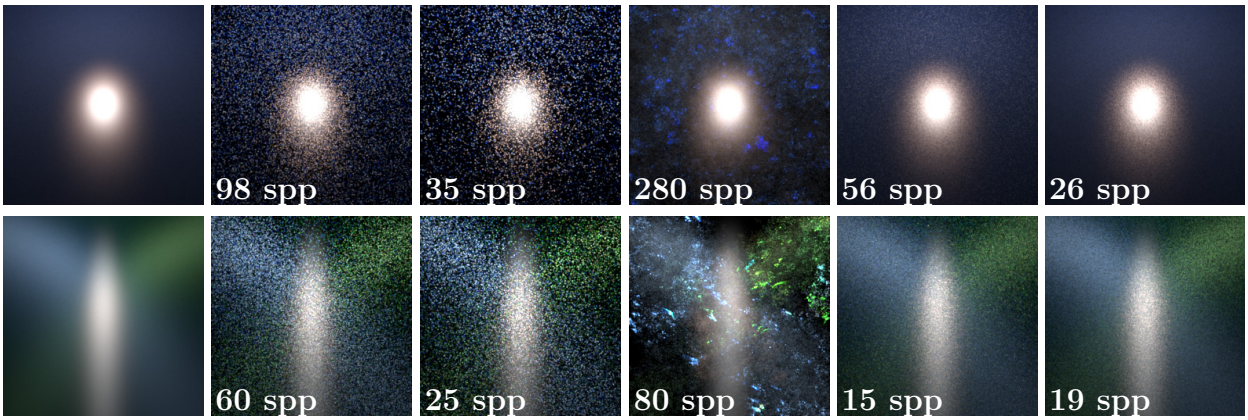


Figure 3.2: Equal-time comparisons of our unidirectional and bidirectional approach to standard transport algorithms, on a simple flat layered configuration lit by a small area light. For standard PT, BDPT and MLT, results are all generated using 3D tracing by applying these algorithms in a simple 3D scene containing a very large slab with flat interfaces. **Top**: A single slab with Henyey-Greenstein scattering between two interfaces, where our estimators perform similarly, but both significantly outperform path tracing, bidirectional and Metropolis transport. **Bottom**: A more complex configuration with two slabs and three interfaces; both media are using an anisotropic microflake phase function [65]. Our bidirectional estimator is a clear winner in this case. The references are generated using standard PT with 100K spp, and all the other images are rendered in 10 seconds.

We introduce a modified path integral framework for light transport in flat slabs, superior to the standard path formulation in this setting. Because it is based on a product of solid angle instead of area measures, it does not contain the high-variance geometry terms needed in standard algorithms. We introduce two simulation techniques within this formulation: the first is analogous to a forward path tracer with next event estimation through layer boundaries and multiple importance sampling; the second is a fully bidirectional estimator. We show the capabilities of this solution on a number of examples, featuring multiple layers with surface and volumetric scattering. Our examples show spatial variation in all parameters: surface BSDF, volume and phase function parameters, layer thickness and surface normal. See Figure 3.1.

3.2 Related Work

3.2.1 Discretized layered BSDFs

Previously, a number of BSDF models have been proposed to describe layers with various assumptions on the interface and subsurface scattering.

An early analytical model by Hanrahan and Krueger [53] already supported multiple layers, but only single scattering, and without supporting arbitrary BSDFs at interfaces. They also proposed to add multiple scattering by Monte Carlo simulation, but their simulation approach only considers volume scattering events (as opposed to a combination of volume and rough interface events). Furthermore, it uses binning on the outgoing direction, as opposed to an efficient BSDF evaluation method for a given outgoing direction, which is provided by our approach.

A model by Stam [118] introduces a solution for rendering skin as a layered material consisting of rough dielectric interfaces bounding a volumetric scattering slab. The solution is based on discretization of the BSDF into a directional basis, on which the light transport problem is solved. The model introduced by Jakob et al. [66] can be seen as a significant extension of Stam's discretization approach, working in the Fourier domain. It handles arbitrary layer stacks, supporting subsurface scattering within thin layers using the adding-doubling method, in addition to microfacet rough interfaces. The work of Zeltner extends this approach to anisotropic surface reflectance [139]. These models are highly accurate and efficient to render with, once the discretized BSDF has been constructed. However, as the BSDF construction in the discretized basis is relatively expensive, they are best suited for homogeneous BSDFs. A small number of such BSDFs can be spatially blended with varying weights, but this has strict limitations, compared to our support for arbitrary spatial texturing of all parameters.

3.2.2 Analytic layered BSDFs

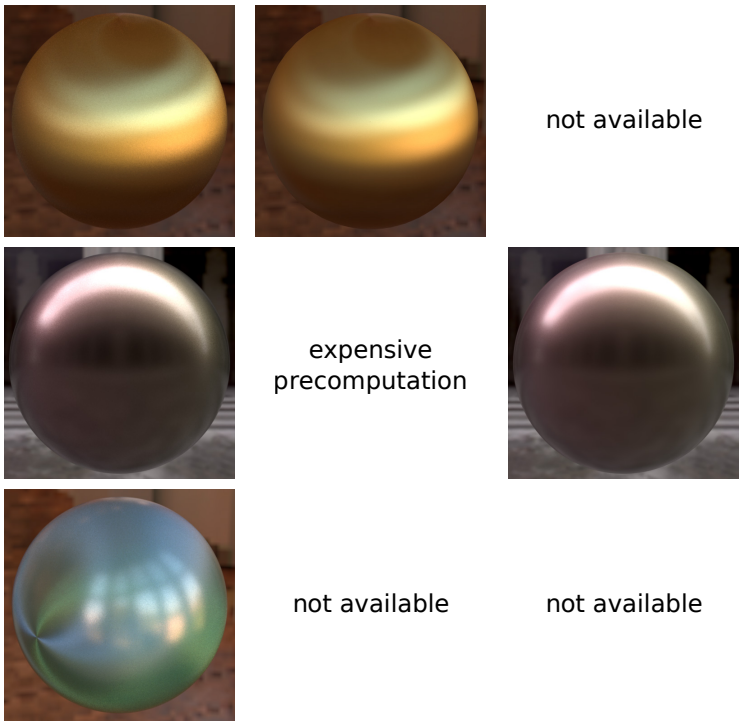
The model by Weidlich and Wilkie [131] takes a different approach. They focus on layers where subsurface scattering is absent (though absorption is allowed), by analytically combining microfacet BSDFs from the interfaces into a single, potentially multi-lobe, microfacet-like BSDF. There are significant approximations in this approach, carefully chosen so that integration (Monte Carlo or otherwise) is never required within a single BSDF query. This makes the model fast and flexible. Another recent model [47] also takes the approach of avoiding Monte Carlo integration during queries, by introducing extended normal distribution functions (ENDFs), analogous to microfacet NDFs but capturing multiple reflection or scattering events. In the most recent work, Belcour [12] introduced an approach based on tracking low-order moments of the BSDF lobes. This is a very fast and practical solution, but still introduces some approximations and limitations (e.g. no surface or volume anisotropy). In contrast, our method offers unbiased accuracy and even more flexibility, at the cost of some additional computation and variance. Several previous techniques model light scattering in layered materials like human skin [27], but these are focused on lateral light spreading in BSSRDFs, and are orthogonal to our focus on the directional properties of BSDF models.

3.2.3 Microfacet models for interfaces

BSDF models based on the microfacet theory are commonly used in computer graphics to capture how light reflects and refracts when interacting with specular surfaces with rough microstructure. The model by Walter et al. [127] extends the microfacet model of Cook and Torrance [20] to handle light reflection and transmittance through rough dielectric interfaces, and is currently seen as standard in physically-based rendering. We use this model to describe our layer interfaces.

The microfacet model recently developed by Heitz et al.[57] is capable of capturing interreflections between the facets and better conserves energy. Schüssler [113] introduced a solution to the energy loss common in normal mapping techniques, caused by a mismatch between the shading and geometric normal. These models (or any future improved microfacet models) could be combined with our approach.

3.2.4 Capability comparison



Ours Zeltner 2018 [139] Belcour 2018 [12]

Figure 3.3: Comparison to previous work. The **top row** shows an example with anisotropic surface reflectance, where our solution closely matches Zeltner's, but Belcour's approach does not support anisotropy. The **middle row** shows an example with spatial variation in the parameters; here our method closely matches Belcour's, but Zeltner's approach does not naturally support spatial variation. The **bottom row** shows a two-layer configuration with anisotropic microflake phase functions, which is only supported by our method.

In Figure 3.3, we compare the capabilities of our approach to recent work [139, 12]. We consider three features supported by our approach: surface anisotropy, spatial variation, and

volumetric medium anisotropy. Only one of these is supported in the compared systems: spatial variation in Belcour's approach and surface anisotropy in Zeltner's.

3.3 Background and Overview

In this section, we explicitly state the assumptions of our method, provide background on the standard path formulation of light transport, and provide a quick overview of the rest of the chapter.

3.3.1 Assumptions

Although light generally enters and leaves the layer from different locations, we note that when the layers are thin and the lighting is comparably distant, the entrance and departure locations will be close enough to each other. We assume it is acceptable to ignore this displacement, allowing us to describe the light transport in the layers using BSDFs, rather than BSSRDFs (Figure 3.4).

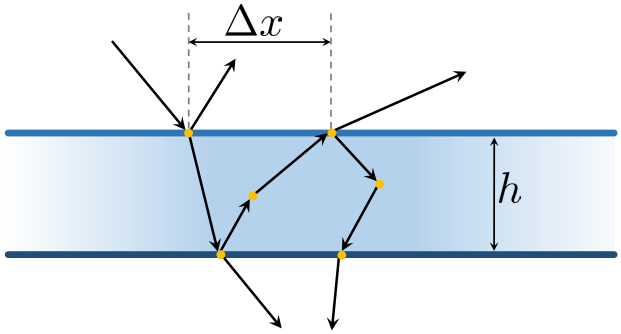


Figure 3.4: Small displacement assumption: when light hits a thin layer, it gets reflected and refracted by the interfaces and scattered and absorbed internally. Since the geometric thickness h of the layer is small, we assume the displacements (e.g., Δx) of light's entrance and departure locations can be neglected.

Furthermore, we assume that the spatial variation of layer properties is slow enough that a

BSDF evaluation at a single surface point can locally approximate them as spatially uniform. This is related to the above in assuming that the horizontal spreading of light is small enough to be negligible.

In fact, these are the *only* approximating assumptions of our approach, which otherwise offers unbiased accuracy and full flexibility in setting the layer properties and varying them spatially.

3.3.2 Review of Veach's path integral formulation

In the Veach formulation of light transport [124], light paths are defined as sequences of vertices connected by segments. The value of a light transport integral (for example, but not necessarily limited to, a pixel value) is written as

$$I = \int_{\Omega} f(\bar{x})\, \mathrm{d}\mu(\bar{x}), \tag{3.1}$$

where $\bar{x} = (x_0, \ldots, x_k)$ is a path with k segments and $k+1$ vertices on the surfaces or within the participating media of a scene. Ω is the space of all paths and is defined as the union of Ω_k for $k \geq 0$, where Ω_k indicates the set of paths of length k. Furthermore, $f(\bar{x})$ is the path contribution to the integral, and $\mu(\bar{x})$ is a special measure on the path space, defined as the product of area measures on the vertices x_i. The contribution $f(\bar{x})$ is a product of vertex terms (normally BSDFs and phase functions) and geometry terms corresponding to path segments. The geometry terms contain the squared distance between the two vertices in the denominator; this is a significant source of variance when trying to connect independently sampled vertices on thin layer configurations.

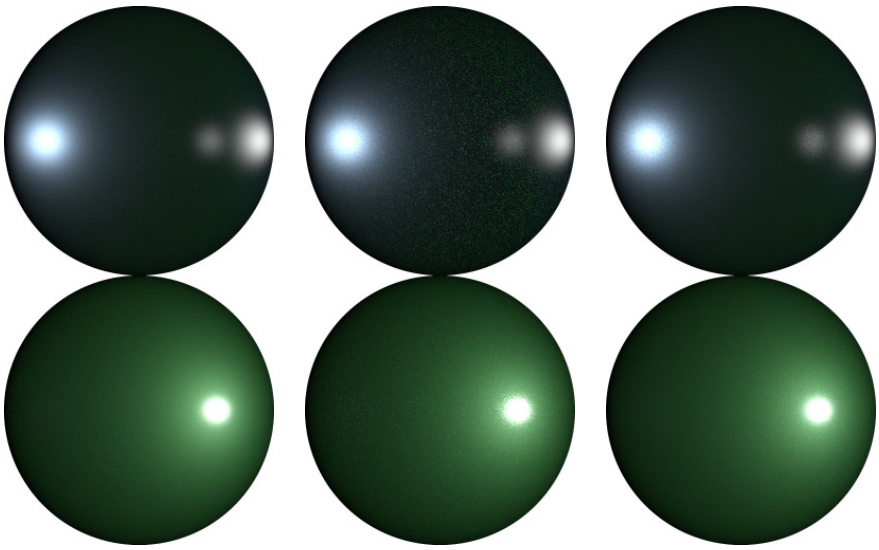
3.3.3 Overview

In Section 3.4, we describe our path formulation of layered light transport. Our path integral differs from Veach's formulation in that it is *position-free*. The key idea is that on an infinite flat slab, the horizontal positions of vertices do not matter: it is only the vertical position (depth) of a vertex, and the *directions* between vertices, that are relevant to a light transport integral. The vertices are defined by their depth in the layer, as opposed to a full 3D position, and the segments have variable unit directions.

It is important to note that our position-free formulation is not just a simplified specialization of the standard formulation to the flat slab setting, but in fact a new approach that achieves much superior variance to the standard formulation. The key benefit of this new formulation is that it does not contain the inverse square distance falloff terms that are required between any two vertices with full positional information. The leads to high variance, even in advanced estimators such as bidirectional and Metropolis transport, which in fact perform even worse in this setting than unidirectional; see Figure 3.2 for examples.

In contrast, our approach leads to an efficient estimator based on unidirectional sampling with next event estimation, and an even more efficient bidirectional estimator. The unidirectional performs similarly (though usually not better) in simpler cases, but in challenging cases with sharp and/or anisotropic BSDFs and phase functions, the bidirectional version is clearly more efficient (Figure 3.2, bottom). Figure 3.5 demonstrates the performance of the estimators through BSDF lobe visualization, also showing a close match to ground truth. In Section 3.5, we describe these two estimators in detail, and also focus on the two additional operations critical for integrating a BSDF into a practical renderer: importance sampling and pdf evaluation.

Finally, we present results in Section 3.6, and summarize in Section 3.7.



Ground truth Our unidir. Our bidir.

Figure 3.5: Outgoing lobes of a layered BSDF (reflection and transmission) visualized as projected hemispheres. **Left:** ground truth computed by sampling and binning the light paths. **Middle:** Our unidirectional estimator. **Right:** Our bidirectional estimator (same time).

3.4 Position-Free Path Formulation

In this section, we theoretically define the value of a layered BSDF due to a given layer stacking, for given query directions $\boldsymbol{\omega}_\mathrm{i}$ and $\boldsymbol{\omega}_\mathrm{o}$, as a path integral. Given such a definition, any Monte Carlo method can be used to evaluate the BSDF by randomly sampling paths, evaluating their contributions and dividing by the corresponding probability density values.

3.4.1 Notation

We will use the notation $\cos\boldsymbol{\omega}$ to denote the z-component of the unit vector $\boldsymbol{\omega}$. We will also use $\mathbb{I}(x)$ to denote an indicator function, returning 1 if the boolean condition x is true and 0 if false. A bold font is used to denote unit vectors (directions) on $\mathbb{S}^2$. Please refer to Table 3.1 for the notation used in this section.

Table 3.1: Notation used in §3.

$\boldsymbol{\omega}_\mathrm{i}$	light direction
$\boldsymbol{\omega}_\mathrm{o}$	camera direction
$\cos \boldsymbol{\omega}$	z-component of the unit vector $\boldsymbol{\omega}$
$\mathbb{I}(x)$	binary indicator function
$f_\mathrm{l}(\boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$	layered BSDF (our goal)
$f_\mathrm{s}(z, \boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$	interface BSDF at depth z
$f_\uparrow, f_\downarrow$	BSDFs f_s at top and bottom interface
$f_p(\boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$	phase function (normalized as a pdf)
$\sigma_\mathrm{s}, \sigma_\mathrm{t}$	scattering and aborption coefficient
$\hat{f}_p$	reduced phase function, $\hat{f}_p = \sigma_\mathrm{s} f_p$
z_i	depth of i-th path vertex
$\boldsymbol{d}_i$	direction of i-th path segment
$\bar{x}$	light path $(\boldsymbol{d}_0,\ z_1,\ \boldsymbol{d}_1,\ \ldots,\ z_k,\ \boldsymbol{d}_k)$
v_i	i-th vertex contribution
s_i	i-th segment contribution
$\tau(z, z', \boldsymbol{\omega})$	transfer through segment
α_i	i-th segment cosine term exponent
$\mu(\bar{x})$	path space measure
$\sigma(\boldsymbol{\omega})$	solid angle measure on unit directions
$\lambda(z)$	line (Lebesgue) measure on real numbers
$p(\bar{x})$	pdf of path $\bar{x}$ in measure $\mu(\bar{x})$
$L_v(z, \boldsymbol{\omega}_\mathrm{o})$	volume radiance
$L_s(z, \boldsymbol{\omega}_\mathrm{o})$	outgoing surface radiance
$L_s^i(z, \boldsymbol{\omega}_\mathrm{i})$	incoming surface radiance
$S(z, \boldsymbol{\omega})$	source term in radiative transfer eq.

3.4.2 Position-free path integral

To develop the theory, we will first assume a single infinite flat slab with a BSDF $f_\uparrow$ on the top interface and a BSDF $f_\downarrow$ on the bottom interface, combined with a homogeneous scattering volume inside the slab to produce a resulting layered BSDF. The volumetric medium is defined by a phase function f_p, scattering coefficient σ_s and extinction coefficient σ_t; we will use the notation $\hat{f}_\mathrm{p} = \sigma_\mathrm{s} f_\mathrm{p}$.

For simplicity, we will drop the depth dependence of the volume parameters (though they

could vary) and we will assume constant scattering / extinction coefficients, though they can vary with direction for fully anisotropic phase functions, which we also support. We will further assume that the slab has unit thickness; the formulation can be easily adjusted for any thickness.

A **vertex** $z_i \in [0, 1]$ is a single real number indicating the depth within the layer. A value of 0 or 1 indicates a surface reflection or refraction event on the bottom or top interface, respectively. Fractional values indicate volume scattering events at the specified depth. Note again that the horizontal positions of vertices on the infinite flat interfaces are not needed.

A **direction** $\boldsymbol{d}_i$ is a unit vector on $\mathbb{S}^2$ denoting the light flow between vertices. In our convention (inherited from Veach), the vectors point in the direction of light flow (i.e. from light source to camera), and the vertex/direction indexing follows this as well.

A **light path** $\bar{x}$ is a sequence of directions and vertices: $\bar{x} = (\boldsymbol{d}_0,\ z_1,\ \boldsymbol{d}_1,\ \ldots,\ z_k,\ \boldsymbol{d}_k)$. The first and last directions are aligned with the input and output directions of the layered BSDF query, i.e. $\boldsymbol{d}_0 = -\boldsymbol{\omega}_\mathrm{i}$ and $\boldsymbol{d}_k = \boldsymbol{\omega}_\mathrm{o}$. In contrast to Veach's formulation, the path interleaves directions with vertices, and the two ends of the path are defined by directions (not vertices). See Figure 3.6 for some example paths.

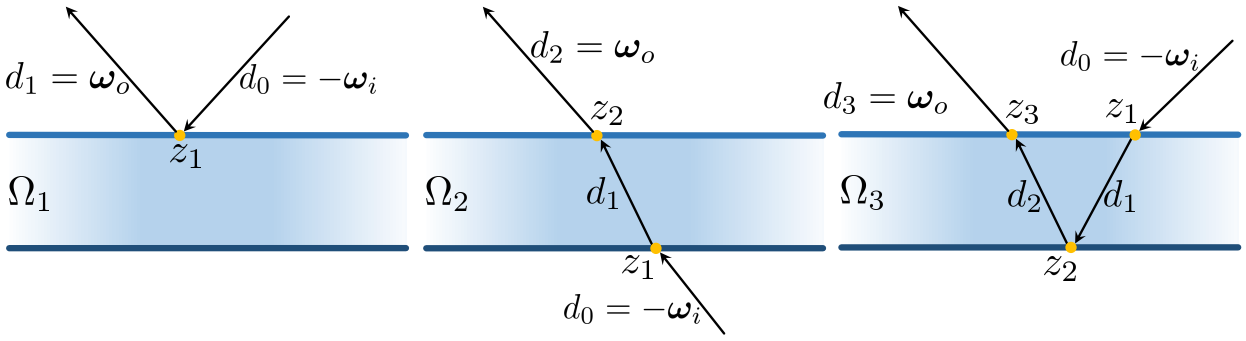


Figure 3.6: Example paths of lengths 1, 2 and 3. In our formulation, the exact positions of the vertices do not matter: the z_i only carry information about which interface the vertex occurs on. The first and last in the sequence of directions $\boldsymbol{d}_i$ map to the incoming and outgoing directions of the underlying BSDF query.

The **path contribution** $f(\bar{x})$ of a light path is the product of vertex terms v_i (on each

vertex) and segment terms s_i (on all internal segments):

$$f(\bar{x}) = v_1 s_1 v_2 s_2 \dots s_{k-1} v_k. \tag{3.2}$$

The vertex term consists of the BSDF or phase function value:

$$v_i = v(z_i, -\boldsymbol{d}_{i-1}, \boldsymbol{d}_i) = \begin{cases} f_\uparrow(-\boldsymbol{d}_{i-1}, \boldsymbol{d}_i) & \text{if } z_i = 0, \\ f_\downarrow(-\boldsymbol{d}_{i-1}, \boldsymbol{d}_i) & \text{if } z_i = 1, \\ \hat{f}_p(-\boldsymbol{d}_{i-1}, \boldsymbol{d}_i) & \text{if } 0 < z_i < 1. \end{cases} \tag{3.3}$$

Define the transfer term $\tau(z_1, z_2, \boldsymbol{\omega})$ as follows:

$$\tau(z, z', \boldsymbol{\omega}) := \exp\left(\frac{-\sigma_\mathrm{t}|z' - z|}{|\cos \boldsymbol{\omega}|}\right) \cdot \mathbb{I}\left(\frac{z' - z}{\cos \boldsymbol{\omega}} > 0\right). \tag{3.4}$$

The purpose of the exponential term is to compute the transmittance when going from depth z to z' following direction $\boldsymbol{\omega}$. The indicator term checks the validity of the configuration (i.e. if the direction points up, then z' should be greater than z, and vice versa). The segment term for internal segments can now be defined as:

$$s_i = s(z_i, z_{i+1}, \boldsymbol{d}_i) := \tau(z_i, z_{i+1}, \boldsymbol{d}_i) \cdot |\cos \boldsymbol{d}_i|^{\alpha_i}, \tag{3.5}$$

where

$$\alpha_i = \mathbb{I}(z_i \in \{0, 1\}) + \mathbb{I}(z_{i+1} \in \{0, 1\}) - 1. \tag{3.6}$$

This definition encapsulates the subtle behavior of cosine terms along the path segments. For a detailed derivation, please refer to Appendix A.1.

The **path space** $\Omega(\boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$ is the set of all paths of one or more vertices, such that the first direction of the path is equal to $-\boldsymbol{\omega}_\mathrm{i}$ and the last to $\boldsymbol{\omega}_\mathrm{o}$. It can be seen as the union of the

spaces of such paths of all lengths $k \geq 1$, that is, $\Omega = \cup_{k \geq 1} \Omega_k$.

The **path space measure** $\mu(\bar{x})$ is a product of solid angle measures σ on the internal directions of the path, times the product of line measures λ on volumetric scattering vertices. That is, for a k-vertex path,

$$\mu(\bar{x}) = \prod_{i=1}^{k-1} \sigma(\boldsymbol{d}_i) \cdot \prod_{i \in V(\bar{x})} \lambda(z_i). \tag{3.7}$$

Here $V(\bar{x})$ is the set of indices of volumetric vertices on $\bar{x}$, and λ is the line measure (i.e. standard Lebesgue measure on the real numbers).

Finally, we can define the **layered BSDF value** $f_{\mathrm{l}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}})$ as an integral over the set of paths $\Omega(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}})$:

$$f_{\mathrm{l}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = \int_{\Omega(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}})} f(\bar{x}) \, \mathrm{d}\mu(\bar{x}). \tag{3.8}$$

As usual, any Monte Carlo method can be used to compute this integral. As long as the probability density $p(\bar{x})$ with respect to measure $\mu(\bar{x})$ of generated sample paths is known, we simply average a number of samples of the form $f(\bar{x})/p(\bar{x})$.

3.4.3 Derivation

Here we sketch the derivation of the path formulation. Like in Veach's version (and its volumetric extension), the derivation proceeds by recursively expanding the surface and volume rendering equation (the latter also commonly known as the radiative transfer equation). Denote the surface radiance by $L_s(z, \boldsymbol{\omega})$ (for $z \in \{0, 1\}$) and the volume radiance $L_v(z, \boldsymbol{\omega})$ (for $z \in [0, 1]$).

The volume radiance will satisfy the standard radiative transfer equation, specialized to our

position-free setting:

$$L_v(z,\boldsymbol{\omega}) = S(z,\boldsymbol{\omega}) + \int_0^1 \frac{\tau(z',z,\boldsymbol{\omega})}{|\cos\boldsymbol{\omega}|} \int_{\mathbb{S}^2} \hat{f}_p(\boldsymbol{\omega}',\boldsymbol{\omega})\, L_v(z',\boldsymbol{\omega}')\, \mathrm{d}\boldsymbol{\omega}'\, \mathrm{d}z', \tag{3.9}$$

where the source term $S(z,\boldsymbol{\omega})$ gives illumination from the boundary of the slab:

$$S(z,\boldsymbol{\omega}) = \tau(0,z,\boldsymbol{\omega})L_s(0,\boldsymbol{\omega}) \;+\; \tau(1,z,\boldsymbol{\omega})L_s(1,\boldsymbol{\omega}). \tag{3.10}$$

Notice that, although the source term has two components, only one of them will be non-zero for any given query. This formulation is valid even with no scattering within the layer, in which case $\hat{f}_p = 0$ and the second term of Eq. (3.9) vanishes. Further, the $1/|\cos\boldsymbol{\omega}|$ factor is due to a change of variable (from free-flight distance to depth). For more details, please refer to Appendix A.1.

The surface radiance $L_s(z,\boldsymbol{\omega})$ satisfies the standard rendering equation:

$$L_s(z,\boldsymbol{\omega}) = \int_{\mathbb{S}^2} f_{\mathrm{s}}(z,\boldsymbol{\omega},\boldsymbol{\omega}')\; |\cos\boldsymbol{\omega}'|\; L_s^i(z,\boldsymbol{\omega}')\, \mathrm{d}\sigma(\boldsymbol{\omega}'), \tag{3.11}$$

where $L_s^i(z,\boldsymbol{\omega}')$ is the incoming surface radiance. In case the incoming radiance query points back into the layer, we have

$$L_s^i(z,\boldsymbol{\omega}) = L_v(z,-\boldsymbol{\omega}). \tag{3.12}$$

The BSDF value is defined as the radiance leaving the surface in direction $\boldsymbol{\omega}_{\mathrm{o}}$, under unit irradiance from a directional light in direction $\boldsymbol{\omega}_{\mathrm{i}}$. This is equivalent to evaluating $L_s(1,\boldsymbol{\omega}_{\mathrm{o}})$ under the boundary condition

$$L_s^i(z,\boldsymbol{\omega}) = \frac{\delta(\boldsymbol{\omega}-\boldsymbol{\omega}_{\mathrm{i}})}{|\cos\boldsymbol{\omega}_{\mathrm{i}}|}. \tag{3.13}$$

One can easily check that the irradiance under this illumination is unit. Thus the incoming surface radiance L_s^i is given by Eq. (3.13) when $\boldsymbol{\omega}$ points out of the layer and Eq. (3.12) when it points back into the layer.

The path formulation can now be obtained by recursively expanding the desired value $L_s(1, \boldsymbol{\omega}_\mathrm{o})$ using the above equations for L_s and L_v, terminating the paths using the boundary condition. Note that:

- Each recursive expansion of Eqs. (3.9) or (3.11) will contribute an $\hat{f}_p$ or f_s term, respectively, to the path vertex.
- Each volumetric segment will introduce a $\tau(z, z', \boldsymbol{\omega})$ term, whether the first or second term in Eq. (3.9) is taken.
- Expanding the rendering equation contributes a cosine term to the *next* segment, while expanding the radiative transfer equation contributes a 1/cosine term to the *previous* segment. A combination of these contributions explains the α_i term above.
- The last surface cosine is canceled out when using the boundary condition, due to the denominator cosine in Eq. (3.13).

3.4.4 Normal mapping

An important feature of our method is the mapping of normals of the layer interfaces, introducing mismatches between geometric (flat) normals and shading (mapped) normals. The definition of the segment term (Eq. (3.5)) changes with the presence of shading normals. Precisely, it becomes

$$s_i = \tau(z_i, z_{i+1}, \boldsymbol{d}_i)\ \frac{|\langle \boldsymbol{n}(z_i), \boldsymbol{d}_i \rangle|\ |\langle \boldsymbol{n}(z_{i+1}), \boldsymbol{d}_i \rangle|}{|\cos \boldsymbol{d}_i|}, \tag{3.14}$$

where $\boldsymbol{n}(z)$ denotes the local shading normal at z (for $z \in \{0, 1\}$). This term is no longer symmetric, which implies that BSDFs with mapped normals will in general not be reciprocal. When sampling paths from the light, it is important to handle such BSDF using the correction term introduced by Veach [124] (Eq. 5.19).

3.4.5 Note about reciprocity

Our layered BSDF will be reciprocal whenever the path contribution $f(\bar{x})$ is symmetric with respect to the reversal of the path. Assuming normal mapping is not used, the segment term s_i will be symmetric, so the reciprocity boils down to the symmetry of the vertex terms v_i. This will certainly hold if all phase functions and BSDFs are reciprocal.

Note, however, that crossing an interface between regions of different index of refraction (whether smooth or rough) is not reciprocal in the usual sense. Instead, a physical refractive BSDF should obey a modified reciprocity relation $f_{\mathrm{s}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}) = \eta_o^2/\eta_i^2 \cdot f_{\mathrm{s}}(\boldsymbol{\omega}_{\mathrm{o}}, \boldsymbol{\omega}_{\mathrm{i}})$ [127], where η_i and η_o are the refractive indices of the corresponding media. In the common case where the layered BSDF's incoming and outgoing directions are both assumed to be in air, the final layered BSDF will still be reciprocal, because there will be an equal number of η^2 and $1/\eta^2$ terms along the path for each medium with index η.

3.4.6 Multiple slabs

Finally, we support extending the framework to multiple slabs. This is relatively straightforward theoretically, and simply requires explicitly keeping track of the interface or volume that a vertex/segment belongs to. We also need to modify the transfer term $\tau(z, z'\boldsymbol{\omega})$ to return zero in cases when the segment crosses an internal interface.

Another option to obtain a multi-layer BSDF is by recursively nesting the BSDFs. To construct the layered BSDF due to a layer stacking of n slabs, we define the layered BSDF due to the stacking of the bottom $n-1$ slabs, and use this BSDF as the bottom interface's BSDF in adding the top layer according to the above theory. We have found that this approach works in practice, but its performance is worse than the explicit implementation above.

3.5 Our Estimators

We now describe our specific layered BSDF method, by presenting our Monte-Carlo solutions to enable the three key operations needed to fully define a BSDF model: sampling (§3.5.1), evaluation (§3.5.2) and pdf computation (§3.5.3). Sampling produces the outgoing direction $\boldsymbol{\omega}_\mathrm{o}$ given the incoming one $\boldsymbol{\omega}_\mathrm{i}$ (or the reverse), while evaluation answers the BSDF query for given $\boldsymbol{\omega}_\mathrm{i}$ and $\boldsymbol{\omega}_\mathrm{o}$. Note that the values returned from sampling, evaluation and pdf procedures are themselves stochastic, and are equal to the true BSDF value, pdf value or sampling weight only in expectation. Stochastic evaluation was also used in some recent BSDF models [57].

Multiple importance sampling (MIS) is commonly used to combine multiple techniques to produce a given path, and key to obtaining low-noise results under complex lighting conditions. This technique typically uses the sampling pdfs of the techniques being combined to derive the weights, which requires the pdf values of the layered BSDFs. We introduce two solutions: an unbiased solution for estimating the exact pdf values in expectation, as well as a fast and approximate version which we demonstrate is sufficient for MIS (§3.5.3). In a supplementary document, we show that the estimators are still unbiased in the presence of approximate pdfs for MIS weighting and stochastic evaluation of both weights and function values.

3.5.1 BSDF sampling

Sampling a BSDF is the problem of drawing the outgoing direction $\boldsymbol{\omega}_\mathrm{o}$ given the incoming one $\boldsymbol{\omega}_\mathrm{i}$ (or the reverse), with a pdf proportional, exactly or approximately, to the value $f_\mathrm{l}(\boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$ (times the cosine term, if possible). This is straightforward: we draw $\boldsymbol{\omega}_\mathrm{o}$ simply by following the stochastic process given by light interacting with the layered configuration.

That is, we utilize a pure forward path tracing process that starts with a ray with direction $-\boldsymbol{\omega}_\mathrm{i}$ and explicitly simulates interactions between the ray and the layer's interfaces and internal media by sampling the corresponding BSDFs and phase functions, accumulating a throughput value along the way. When the ray eventually leaves the layer, its direction gives $\boldsymbol{\omega}_\mathrm{o}$ and the throughput of the full light transport path gives the stochastic sample weight. Formally, this weight is an estimate of the BSDF value, times the exitant cosine direction, divided by the sampling pdf in solid angle measure.

Although this simulation is analogous to standard Monte Carlo path tracing, it is usually much more efficient than tracing paths in the global scene thanks to the simplicity of the flat slab configuration (under which ray tracing becomes simple numerical computation, not requiring any acceleration structures).

3.5.2 BSDF evaluation

To evaluate our BSDF f_l at given incoming and outgoing directions $\boldsymbol{\omega}_\mathrm{i}$ and $\boldsymbol{\omega}_\mathrm{o}$, we introduce two Monte Carlo based methods to evaluate the path integral from Eq. (3.8). The first one (§3.5.2) is analogous to a unidirectional path tracer with next-event estimation (NEE), while the second (§3.5.2) uses a bidirectional scheme.

Unidirectional simulation

In standard path tracing, a shading point would be directly connected to a light source in a process often called *direct illumination* or *next event estimation* (NEE), which is crucial for low-variance rendering. In an analogy to this technique, consider a shading point inside a single layer slab (whether on the bottom interface or a scattering point within the medium). We would like to create a path ending with $\boldsymbol{\omega}_\mathrm{i}$, intuitively connecting it to an external

directional light source with direction $\boldsymbol{\omega}_\mathrm{i}$. However, direct connection between the shading point and the desired external direction is usually invalid due to the layer's top refractive interface.

To address this problem, we introduce our NEE scheme that directly connects scattering events across potentially rough refractive interfaces. Assume without loss of generality that our path tracing starts with direction $\boldsymbol{\omega}_\mathrm{o}$. At each scattering event, we need to find a direction $\boldsymbol{\omega}'_\mathrm{i}$ so that $\boldsymbol{\omega}_\mathrm{i} \to \boldsymbol{\omega}'_\mathrm{i}$ follows the BSDF at the interface. To this end, we draw $\boldsymbol{\omega}'_\mathrm{i}$ by sampling the interface BSDF backwards, given $\boldsymbol{\omega}_\mathrm{i}$. Finally, we simply multiply the accumulated throughput by the weight returned from the sampling routine, and the BSDF (or phase function) value at the scattering event.

Furthermore, this NEE connection can be combined with a path continuation (by sampling the phase function or interface BSDF), using MIS for the weighting. This is analogous to the MIS direct illumination used in many practical path tracers, with the difference that the path can cross a refractive boundary. Note the distinction between this *local* MIS, and the *global* MIS used by the scene-level transport algorithm (a standard path tracer in our results). An illustration of these two techniques, applied to a transmit-reflect-transmit (TRT) configuration, can be found in Figure 3.7-ab.

Previous work on next-event estimation in scattering volumes through refractive interfaces [128, 77] is related to our scheme, but focuses on arbitrary geometries, which is not necessary in the flat layer setting.

Extending this NEE scheme to cross multiple layer interfaces is somewhat tedious to implement, as care must be taken not to double-count light paths. We instead use the recursive nesting approach to multiple layers (§3.4.6) when using the unidirectional estimator, which handles these issues automatically.

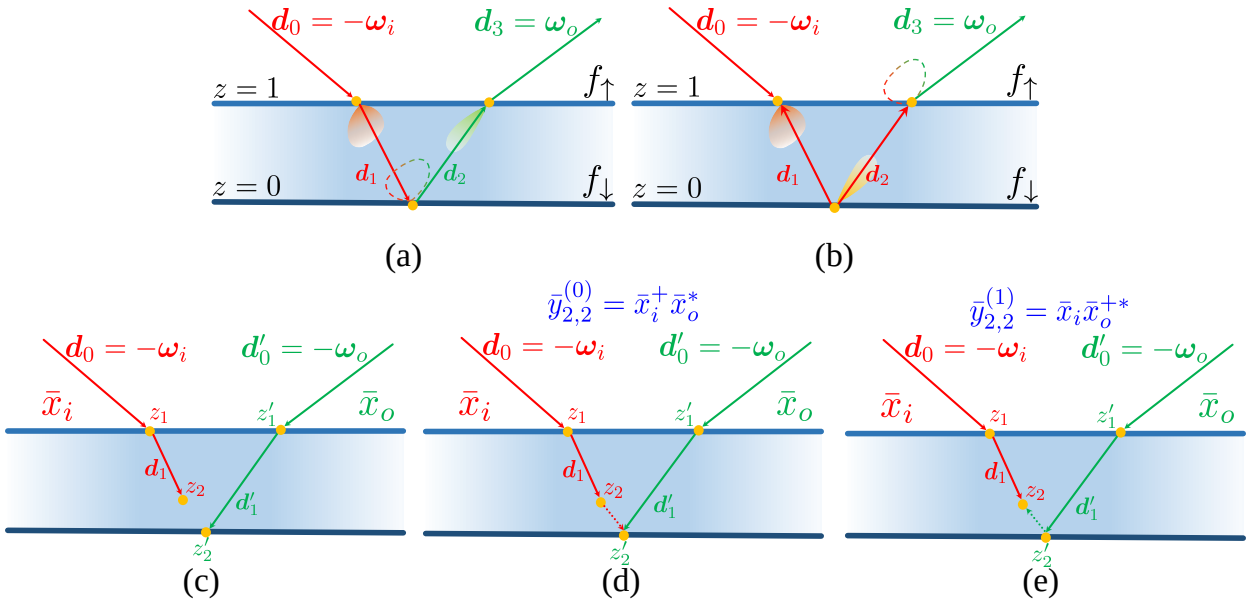


Figure 3.7: Our Monte Carlo estimators for BSDF values. **(ab)** Unidirectional estimator uses two path sampling strategies for "shading" a vertex on the bottom layer: (a) sampling the BSDF $f_\uparrow$ of the top interface and connecting at the bottom (next event estimation); or (b) sampling $\boldsymbol{d}_2$ using $f_\downarrow$ and connecting at the top (path continuation). These strategies are combined using local MIS. **(cde)** Bidirectional estimator: (c) Two subpaths with initial directions $\boldsymbol{\omega}_\mathrm{i}$ and $\boldsymbol{\omega}_\mathrm{o}$. (de) Two full light paths constructed by sampling an additional direction from each sub-path.

Bidirectional simulation

Although our unidirectional solution works well in many cases, we introduce a new bidirectional approach that performs even better. Our approach is conceptually similar to bidirectional path tracing (BDPT) but is technically different in several ways due to our position-free path formulation.

Given the incoming and outgoing directions $\boldsymbol{\omega}_\mathrm{i}$ and $\boldsymbol{\omega}_\mathrm{o}$, consider two light transport paths, generated from the light and camera, respectively.

$$\begin{aligned} \bar{x}_i &= (\boldsymbol{d}_0, z_1, \boldsymbol{d}_1, \ldots, z_s) \\ \bar{x}_o &= (\boldsymbol{d}'_0, z'_1, \boldsymbol{d}'_1, \ldots, z'_t), \end{aligned} \tag{3.15}$$

where $\boldsymbol{d}_0 = -\boldsymbol{\omega}_\mathrm{i}$ and $\boldsymbol{d}'_0 = -\boldsymbol{\omega}_\mathrm{o}$ (Figure 3.7-c). Now we can construct a full light path $\bar{y}_{s,t}$ connecting the s-th vertex in $\bar{x}_i$ and the t-th vertex in $\bar{x}_o$ (assuming the connection between z_s and z'_t does not cross any layer boundary):

$$\bar{y}_{s,t} = (\boldsymbol{d}_0, \ldots, z_{s-1}, \boldsymbol{d}_{s-1}, z_s, \tilde{\boldsymbol{d}}, z'_t, -\boldsymbol{d}'_{t-1}, z'_{t-1}, \ldots, -\boldsymbol{d}'_0). \tag{3.16}$$

Unlike traditional BDPT, where the connection term between two given subpaths endpoints is fixed, there exists infinitely many valid directions $\tilde{\boldsymbol{d}}$ connecting z_s and z'_t in our case, which gives us freedom to importance-sample the direction. In practice, we choose $\tilde{\boldsymbol{d}}$ in two ways by sampling additional directions $\boldsymbol{d}_s$ and $\boldsymbol{d}'_t$ by extending the two subpaths with an extra importance sampling step. We set $\tilde{\boldsymbol{d}}$ to $\boldsymbol{d}_s$ and $-\boldsymbol{d}'_t$ respectively. This yields two light paths $\bar{y}^{(0)}_{s,t}$ and $\bar{y}^{(1)}_{s,t}$ (Figure 3.7-de), thus providing two samples of the path integral. Denote the extended subpaths by

$$\bar{x}^+_i := (\boldsymbol{d}_0, z_1, \boldsymbol{d}_1, \ldots, z_s, \boldsymbol{d}_s), \tag{3.17}$$

$$\bar{x}^+_o := (\boldsymbol{d}'_0, z'_1, \boldsymbol{d}'_1, \ldots, z'_t, \boldsymbol{d}'_t), \tag{3.18}$$

and let $\bar{x}^*$ denote the adjoint (reversed) version of a light path $\bar{x}$, e.g.,
$\bar{x}^{+*}_o = (-\boldsymbol{d}'_t, z'_t, -\boldsymbol{d}'_{t-1}, \ldots, z'_1, -\boldsymbol{d}'_0)$.

Let $v(z, \boldsymbol{\omega}, \boldsymbol{\omega}')$ and $s(z, z', \boldsymbol{\omega})$ be the vertex and segment contributions defined in eqs. (3.3) and (3.5). We can easily verify that

$$f(\bar{y}^{(0)}_{s,t}) = f(\bar{x}^+_i)\, f(\bar{x}^*_o)\, s(z_s, z'_t, \boldsymbol{d}_s)\, v(z'_t, -\boldsymbol{d}_s, -\boldsymbol{d}'_{t-1}), \tag{3.19}$$

$$f(\bar{y}^{(1)}_{s,t}) = f(\bar{x}_i)\, f(\bar{x}^{+*}_o)\, v(z_s, -\boldsymbol{d}_{s-1}, -\boldsymbol{d}'_t)\, s(z_s, z'_t, -\boldsymbol{d}'_t). \tag{3.20}$$

It follows that the two Monte Carlo estimates will be:

$$\frac{f(\bar{y}^{(0)}_{s,t})}{p(\bar{y}^{(0)}_{s,t})} = \frac{f(\bar{x}^{+}_{i})}{p(\bar{x}^{+}_{i})} \frac{f(\bar{x}^{*}_{o})}{p(\bar{x}_{o})}\, s(z_s, z'_t, \boldsymbol{d}_s)\, v(z'_t, -\boldsymbol{d}_s, -\boldsymbol{d}'_{t-1}) \tag{3.21}$$

$$\frac{f(\bar{y}^{(1)}_{s,t})}{p(\bar{y}^{(1)}_{s,t})} = \frac{f(\bar{x}_{i})}{p(\bar{x}_{i})} \frac{f(\bar{x}^{+*}_{o})}{p(\bar{x}^{+}_{o})}\, v(z_s, \boldsymbol{d}_{s-1}, \boldsymbol{d}'_t)\, s(z_s, z'_t, -\boldsymbol{d}'_t), \tag{3.22}$$

Note that in general $f(\bar{x}_o) \neq f(\bar{x}^*_o)$ and $f(\bar{x}^+_o) \neq f(\bar{x}^{+*}_o)$ due to non-reciprocal operations such as shading normals; care must be taken to compute correct throughputs of light subpaths, as detailed in Chapter 5 of Veach [124].

The above discussion assumed a single light and single camera subpath. In practice, we combine all prefixes of the sampled subpaths. In particular, we sample subpaths of length n_i and n_o from the light and camera respectively (the lengths are chosen through Russian roulette):

$$\begin{aligned} \bar{x}_i &= (\boldsymbol{d}_0, z_1, \boldsymbol{d}_1, \ldots, z_{n_i}, \boldsymbol{d}_{n_i}), \\ \bar{x}_o &= (\boldsymbol{d}'_0, z'_1, \boldsymbol{d}'_1, \ldots, z'_{n_o}, \boldsymbol{d}'_{n_o}). \end{aligned} \tag{3.23}$$

For all s and t combinations, Eqs. (3.21) and (3.22) provide $2n_in_o$ estimators of $f_{\mathrm{l}}(\boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}})$. Combining them using MIS gives us our bidirectional estimator for paths of length 2 or more vertices. We handle single vertex paths separately. The details of MIS weighting are discussed in the supplementary document.

3.5.3 Pdf estimation

Another important operation for practical BSDF models is to evaluate the probability density for sampling provided incoming and outgoing directions. That is, to evaluate $p(\boldsymbol{\omega}_{\mathrm{o}} \mid \boldsymbol{\omega}_{\mathrm{i}})$, the probability density of $\boldsymbol{\omega}_{\mathrm{o}}$ given $\boldsymbol{\omega}_{\mathrm{i}}$ (assuming that the sampling process draws $\boldsymbol{\omega}_{\mathrm{o}}$ and fixes $\boldsymbol{\omega}_{\mathrm{i}}$). This operator is used for weight computation in multiple importance sampling (using balance or power heuristics), a crucial technique for generating low-noise results using scene-

level Monte Carlo rendering techniques. Note that this pdf is in the solid angle measure; it is a marginal pdf distinct from the path pdf $p(\bar{x})$.

Although $p(\boldsymbol{\omega}_\mathrm{o} \mid \boldsymbol{\omega}_\mathrm{i})$ is usually easily available for traditional analytical BSDFs, no closed-form pdf exists in our case. Instead, the pdf evaluation has comparable form to the BSDF evaluation itself. It can be expressed using another position-free path integral:

$$p(\boldsymbol{\omega}_\mathrm{o} \mid \boldsymbol{\omega}_\mathrm{i}) = \int_{\Omega(\boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})} \mathcal{P}(\bar{x}) \, \mathrm{d}\mu(\bar{x}), \tag{3.24}$$

where

$$\mathcal{P}(\bar{x}) := \left(\prod_{j=1}^{k} p(\boldsymbol{d}_j \mid z_j, \boldsymbol{d}_{j-1}) \right) \left(\prod_{j=1}^{k-1} p(z_{j+1} \mid z_j, \boldsymbol{d}_{j-1}) \right), \tag{3.25}$$

with k denoting the number of vertices in $\bar{x}$. Note that $\boldsymbol{d}_k = \boldsymbol{\omega}_\mathrm{o}$.

We introduce two nondeterministic methods, an unbiased and a fast approximate approach, to estimate $p(\boldsymbol{\omega}_\mathrm{o} \mid \boldsymbol{\omega}_\mathrm{i})$. These operations are not used in standard Monte Carlo light transport and are new, to our knowledge. In practice, the approximate approach can be used when exact estimations are unnecessary (as is the case for a global path tracer with MIS, which we use for our results). Note that the estimated $p(\boldsymbol{\omega}_\mathrm{o} \mid \boldsymbol{\omega}_\mathrm{i})$ is only ever used for MIS weight computation. We never use approximate path pdfs for Monte Carlo estimates, as this would introduce bias. Our BSDF value estimators directly return path throughput with accurate pdf factored in.

Unbiased pdf estimation

Both our unidirectional and bidirectional Monte Carlo estimators introduced in §3.5.2 can be adapted to estimate the path integral in Eq. (3.24) in an unbiased manner. For instance, the estimators given by Eqs. (3.21) and (3.22) simply require a replacement of f by $\mathcal{P}$, and

become:

$$\frac{\mathcal{P}(\bar{y}_{s,t}^{(0)})}{p(\bar{y}_{s,t}^{(0)})} = \frac{\mathcal{P}(\bar{x}_o^*)}{p(\bar{x}_o)} p(\boldsymbol{d}_s \mid z'_t, -\boldsymbol{d}'_{t-1})\, p(z'_t \mid z_s, \boldsymbol{d}_s), \tag{3.26}$$

$$\frac{\mathcal{P}(\bar{y}_{s,t}^{(1)})}{p(\bar{y}_{s,t}^{(1)})} = \frac{\mathcal{P}(\bar{x}_o^{+*})}{p(\bar{x}_o^{+})} p(\boldsymbol{d}'_t \mid z_s, \boldsymbol{d}_{s-1})\, p(z_s \mid z'_t, \boldsymbol{d}'_t). \tag{3.27}$$

Note that some cancellation occurs because $p(x_i) = \mathcal{P}(x_i)$, but in general $p(x_o) \neq \mathcal{P}(x_o^*)$.

When jointly estimating the path integrals for the BSDF value (3.8) and the conditional probability (3.24), the light transport paths $\bar{x}$ need to be sampled *independently* to ensure unbiasedness. Please refer to the supplemental document for a proof.

Approximate pdf estimation

Although the adapted estimators defined in 3.5.3 provide unbiased pdf estimations, they introduce computational overhead comparable to the BSDF evaluation itself. Thus, for applications where unbiased pdfs are unnecessary, we introduce an approximation to accelerate the pdf estimation process. The key idea is to only consider short paths reflecting/refracting from interfaces, as these events have the largest effect on the pdf lobe shape, and add a constant (Lambertian) term to approximate the effect of volume scattering and longer paths.

In practice, we run Monte Carlo simulation on a simplified layer configuration where all volumetric media are removed. We further limit the maximal number of vertices on the light paths to $(2L + 1)$ when $\boldsymbol{\omega}_\mathrm{i} \cdot \boldsymbol{\omega}_\mathrm{o} > 0$ (i.e., $f_\mathrm{l}(\boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$ captures reflection) and $(L + 1)$ when $\boldsymbol{\omega}_\mathrm{i} \cdot \boldsymbol{\omega}_\mathrm{o} < 0$ (i.e., $f_l(\boldsymbol{\omega}_\mathrm{i}, \boldsymbol{\omega}_\mathrm{o})$ captures transmission) where L denotes the number of layers. Lastly, we add a small constant term to the estimation result. The exact scaling of this term is not important for MIS weighting (as it will be overwhelmed by the pdfs of sharply peaked lobes) and we found setting it to 0.1 works well.

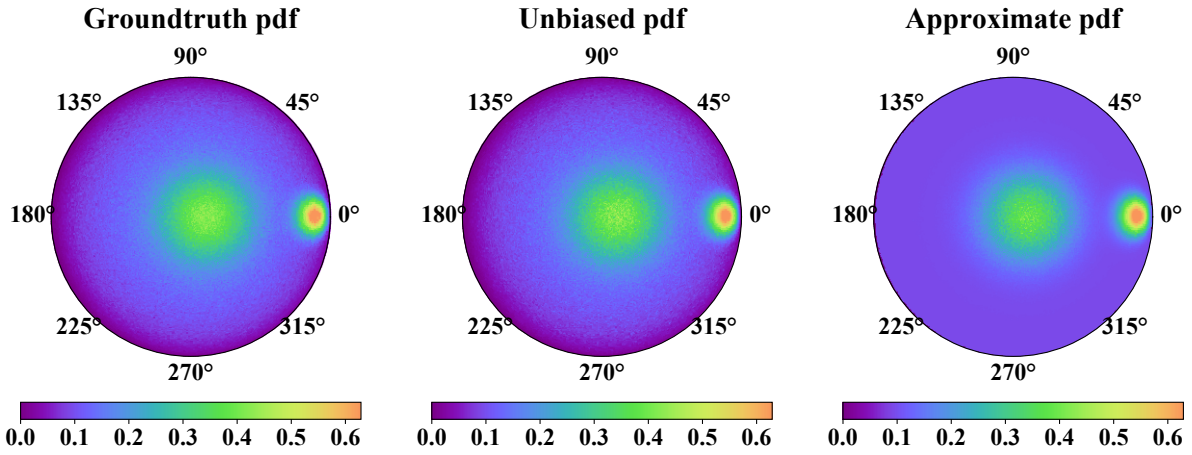
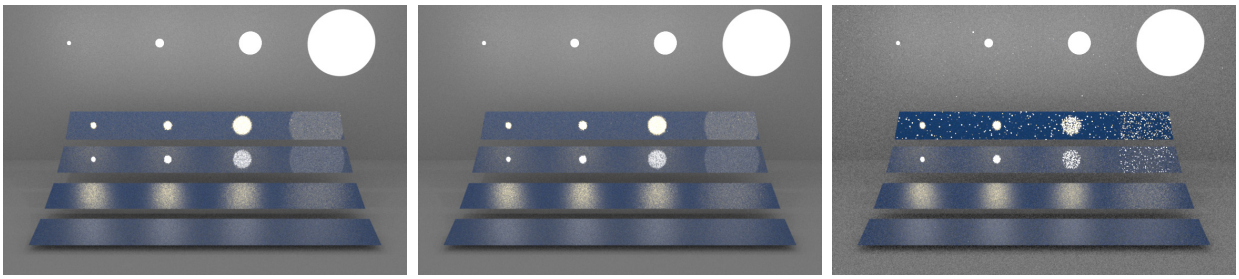


Figure 3.8: Validation of our pdf estimates. The visualization applies a $\log(1+x)$ map for better shape perception. **Left:** Ground truth by sampling and binning. **Middle:** Using the unbiased pdf from §3.5.3. **Right:** Using the approximate pdf from §3.5.3 matches the shape of the most important features and approximates longer paths and volume scattering as diffuse.

See Figure 3.8 for validation of the above pdf approaches against ground truth, and Figure 3.9 for a comparison between renderings using the unbiased and approximated pdf estimation results. All the other results in this chapter are using approximated PDF for MIS. Unbiased PDF is much slower, because it requires long light paths, and has to be computed twice per shading event.



MIS + unbiased pdf (14 min) MIS + approx. pdf (4 min) no MIS (4 min)

Figure 3.9: Multiple importance sampling using our BSDFs. The slabs in this figure use a layered material with rough dielectric on the top, rough gold conductor on the bottom, and blueish homogeneous scattering medium in between. **Left:** Using the unbiased pdf for MIS in a traditional global path tracer. **Middle:** Using the approximate pdf is faster and gives equivalent quality. **Right:** Using no MIS is clearly inferior.

3.6 Applications and Results

In this section, we first provide experimental validations (§3.6.1) and then showcase our method on a number of applications and demonstrate its effectiveness (§3.6.2). All the renderings are generated using the Mitsuba physically based renderer [64] with our layered model implemented as a BSDF plugin. Please see the accompanying video for animated versions of several results.

All the multi-layer results in the chapter use our bidirectional estimator with the explicit implementation (although our BSDF plugin also supports nesting BSDFs). This is because the former runs faster, as seen in Figure 3.16-(c).

3.6.1 Validations

Cross validation In Figures 3.5 and 3.8 as well as the supplemental material, we cross-validate our Monte Carlo estimators depicted in §3.5.2 by comparing our estimated BSDFs/pdfs to references generated using forward sampling (§3.5.1) and binning. Notice that the sampling procedure is a straightforward process that requires none of the complexity introduced by our path formulation and estimators.

White furnace tests We conducted a few "white furnace tests" to demonstrate the energy conservation of our layered BSDFs (Figure 3.10). For all these examples, the BSDFs are constructed such that no energy is lost due to light-layer interactions. Under constant lighting (where identical amount of light comes from all directions), the object becomes invisible, demonstrating that our layered BSDFs indeed conserve energy properly.

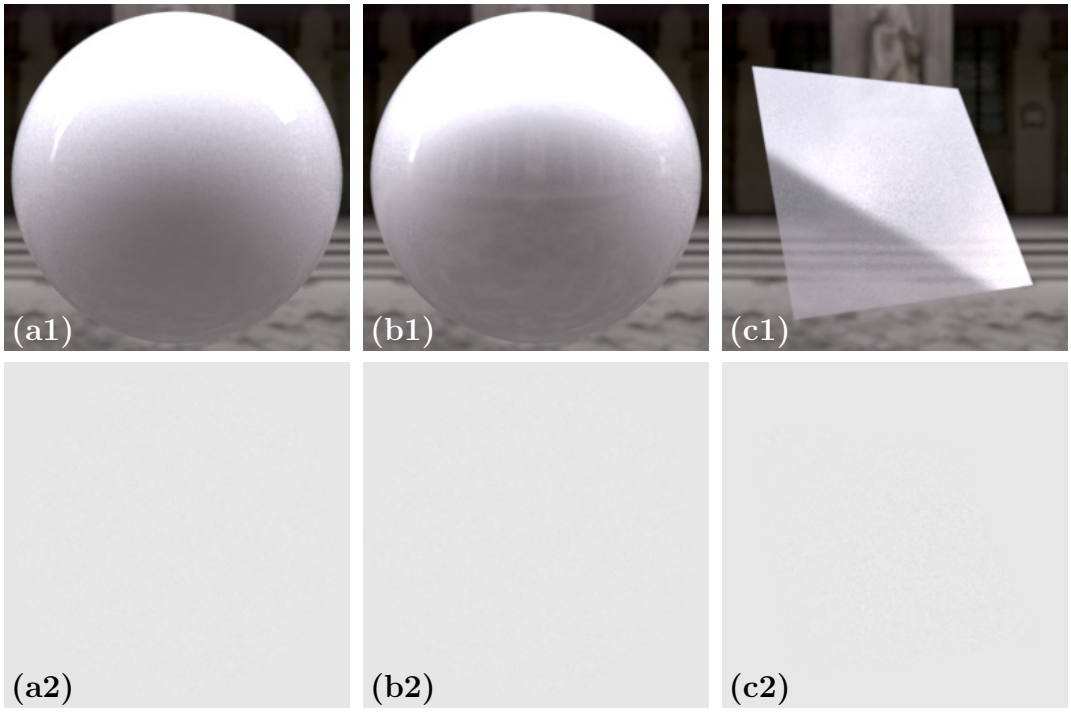


Figure 3.10: White furnace tests. We demonstrate that our BSDFs conserve energy properly via three layered BSDF examples respectively given by **(a)** a dielectric and a diffuse interface; **(b)** a dielectric and a conductor interface and participating medium; **(c)** two dielectric interfaces and participating medium in between. All the interfaces and media have albedo one (so no energy is lost due to light-layer interactions). For each example, a simple object is rendered under both environmental (1) and constant (2) illuminations.

3.6.2 Main Results

(Only a small subset of our results fits into the thesis. Please see our supplemental material and video for more results. [Click here])

Application: Coating thickness/normal variation

Figure 3.11 shows renderings of a globe with a dielectric coating on top of a metallic substrate. In this example, both interfaces are colorless and the layer medium has a blue tint. In Figure 3.11-(a), both interfaces are smooth, creating two overlapped reflections of the environment map with different amounts of blur. In Figure 3.11-(b), the top interface of

the globe is smooth, leading to one clear reflection. On the bottom (metallic) interface, we use a detailed height field to drive the normal variation as well as the medium thickness. The high-frequency variation of normal direction has resulted in detailed highlights on the bottom surface. Further, due to varying amounts of attenuation at different thickness, these highlights exhibit different colors: reflections from greater depths become darker and more saturated. In Figure 3.11-(c), the height variation is instead applied to the top dielectric interface, causing the clear reflection of the environment to be replaced by a blurred one. Further, since the areas under the continents now have larger thickness, their colors become more saturated. Our layered BSDF model is capable of producing all these appearances using a simple set of parameters (thickness, roughness and medium absorption) in conjunction with spatial variation.

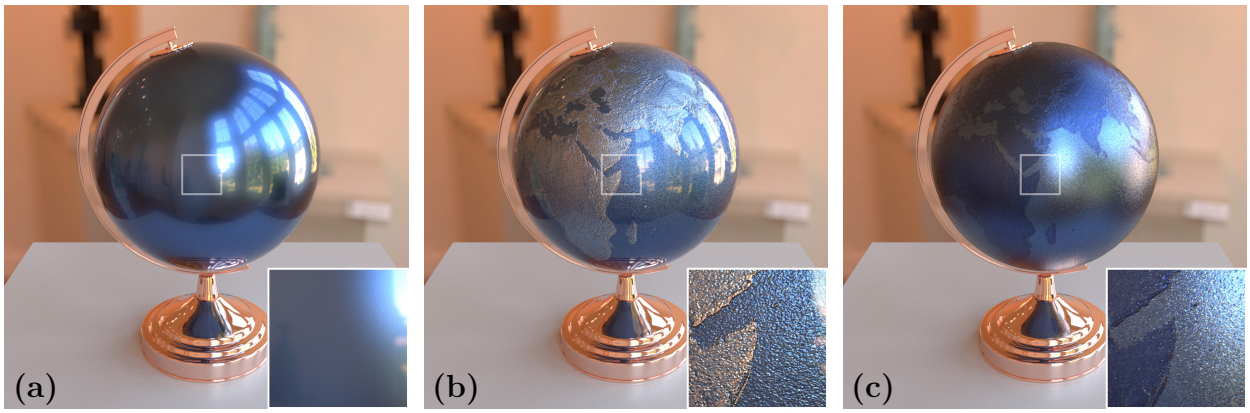


Figure 3.11: Top vs. bottom height variation. Thanks to the physically-based nature of our layered BSDF model, manipulating heights on its top and bottom interfaces has greatly varying effects on the final appearance. The height variation drives both normals and thickness differences (and thus medium absorption). **(a)** No height variation. **(b)** Height variation applied to the bottom interface. **(c)** Height variation applied to the top interface.

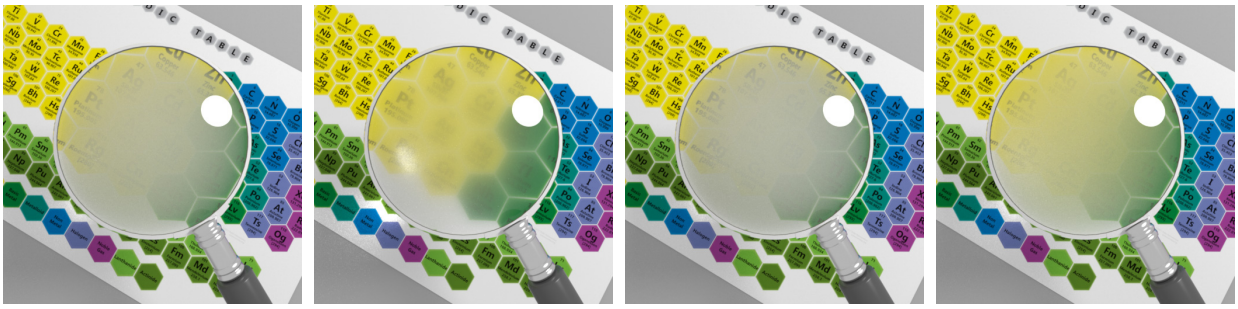
Application: Complex thin sheet transmission

Our physically based BSDF is capable of accurately modeling not only reflection but also transmission. Figure 3.12 contains an example flat surface rendered with our layered BSDF under varying illuminations. This model involves dielectric interfaces with spatially varying

roughnesses and a normal map applied to the front surface. The optical thickness at each location is obtained by multiplying a base density, which varies across the color channels, by the geometric height field matching the normal map. In other words, the optical densities (mean free paths) are spectrally varying, which results in subtle color variations across the surface (especially for transmitted light), a phenomenon that would be challenging to model accurately using existing BSDF models. Note again that all of these effects come from the BSDF model, as the scene geometry is a simple flat polygon.

Back-lit Front-lit Back-lit Front-lit

Figure 3.12: Reflection and transmission: A flat surface rendered with our layered BSDF under varying illuminations. This model involves dielectric interfaces with spatially varying roughnesses, normal maps, and thickness. The optical densities (mean free paths) are spectrally varying, which results in subtle color variations across the surface. Note that the color (albedo) is not varying.



HG ($g = 0.9$) HG ($g = 0.99$) vMF ($\kappa = 10$) vMF ($\kappa = 100$)

Figure 3.13: Reflection and transmission: A flat surface with a layered BSDF of spatially varying thickness (which captures the shape of real convex lens). A range of spatially varying and physically plausible blurring effects can be achieved by varying phase functions.

Figure 3.13 shows renderings of a magnifying lens filled with scattering media with spatially varying thickness (which captures the shape of real convex lens). Note that the scene geometry is still just a flat surface. When coupled with different phase functions (Henyey-Greenstein and von-Mises-Fisher, with different forward scattering parameters), a range of spatially varying and physically plausible blurring effects can be achieved.

Please see the supplemental images and video for more variations with similar configurations.

Application: Anisotropic layer media for fabrics

Our layered BSDF allows any phase functions within volumetric scattering layers, including anisotropic microflake phase functions [65, 141, 56] capable of representing fabrics. Figure 3.14 shows three fabrics modeled using our model with "null" top and bottom interfaces (ones that allows light to travel through without reflecting or refracting it) and anisotropic layer media with spatially varying albedo and flake orientations (the optical density does not vary in these examples, though it could). The satin weave shows well aligned yarns have created smooth and strongly anisotropic highlights. The twill pattern has warp and weft yarns in different colors, leading to dual colored highlights. The velvet exhibits strong grazing-angle highlights, an effect that is challenging to model using traditional BSDF models. Our model successfully captures all the diverse appearances from all three fabrics and produces convincing impressions of these materials.

Figure 3.15 shows a fabric rendered using fiber orientation data acquired by micro-CT imaging [141]. Our rendering uses a fiber orientation map derived from the full data, and matches the full volumetric simulation fairly closely, while being 40 times faster. The speedup is because ours is still a flat BSDF model with parameter mapping, as opposed to full volumetric tracing that requires expensive ray marching through massive data.

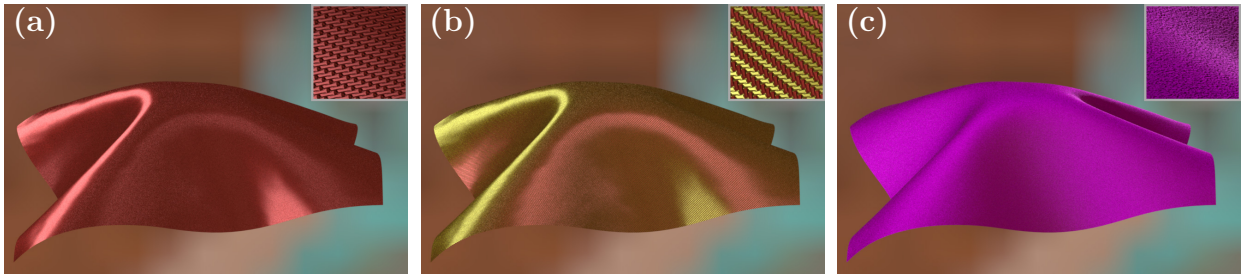
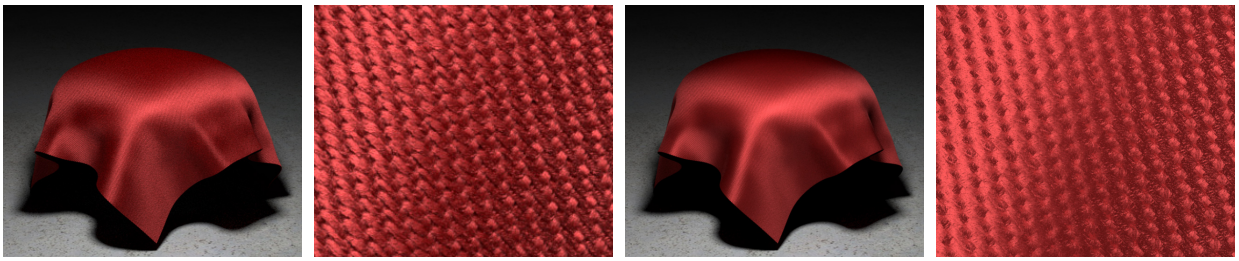


Figure 3.14: Anisotropic media within layers. Our layered BSDF offers the generality to use anisotropic layer media with microflake phase functions. This example shows three fabrics modeled with our BSDF model with anisotropic layer media: **(a)** satin; **(b)** twill; and **(c)** velvet.



(a) Volume rendering **(b)** Our BSDF + fiber-direction map

Figure 3.15: Comparison to volumetric cloth. (a) Images rendered from micro-CT volumetric data, using the microflake phase function. **(b)** Renderings using our approach using a single microflake volumetric layer, where we are using fiber direction maps extracted from the volumetric data. Our rendering is 40× faster than the volumetric simulation.

Application: Multiple layers

Lastly, in Figure 3.16, we show rendered results of a kettle with varying layer configurations. In column (a), the material has a single transparent water layer with a dielectric interface on the top and a metallic surface on the bottom. Both interfaces are normal mapped to capture the water drops and the scratches, respectively. In column (b), the material shares the same bottom surface as in (a) but has a smooth top interface and a translucent coating layer with spatially varying optical thickness and albedo, making only part of the bottom surface directly visible. Lastly, in column (c), the material has a dual-layer configuration by stacking the layers from (a) on top of (b). Our method offers the flexibility to model all three cases with the last one described using the explicit implementation depicted in §3.4.6.

Figure 3.16: Multi-layer BSDF. This result shows renderings of a kettle described with: **(a)** a single transparent layer with a dielectric top interface capturing the water drops over a conducting bottom surface with scratches; **(b)** a single translucent layer with spatially varying optical thicknesses and albedo over the same bottom surface of (a); **(c)** a dual layer configuration created by stacking the transparent layer (a) over the translucent one (b).

3.6.3 Performance

The Monte Carlo processes for sampling and evaluating our BSDFs do introduce computational overhead. Table 3.2 lists the performance numbers of all our results. Further, we provide baseline timings using "trivial" BSDFs (that require no stochastic evaluation) to the same scene geometries. Our performance does degrade with the presence of optically thick and highly scattering media. However, as already demonstrated in Figure 3.2, rendering using our model is still significantly faster than explicitly simulating light transport in layered geometries.

3.7 Conclusion

In this chapter, we introduced a new BSDF model to capture the appearance of layered materials. Inside the evaluation and sampling routines of the layered BSDF, we run a Monte Carlo simulation of light transport within flat slabs. This is substantially faster than

Table 3.2: Render times of all our results (using our "unidir." and "bidir." estimators) as well as baseline models with "trivial" BSDFs (that require no stochastic evaluation). All the multi-layer models are described using nesting BSDFs for the unidirectional estimator and the explicit implementations for the bidirectional one. The baseline models exhibit different appearances and are created solely for performance comparison. All the timings are converted to a 6-core Intel i7-6800K CPU time, and those between parentheses indicate render time per mega-pixel. The numbers in bold correspond to methods used for creating the figures. Please refer to the supplemental material for all the other renderings.

	Image size	spp	Render time					
			Unidir.		Bidir.		Trivial	
Fig. 3.1a	3000×2000	1024	2.5 h	(25 m)	**2.2 h**	**(22 m)**	38 m	(6.3 m)
Fig. 3.11b	1024×1024	256	**2.2 m**	**(2.1 m)**	2.6 m	(2.5 m)	1.3 m	(1.2 m)
Fig. 3.12	800×1200	512	15.2 m	(7.9 m)	**24 m**	**(12.5 m)**	2.4 m	(1.3 m)
Fig. 3.13	512×512	1024	6.4 m	(6.1 m)	**13 m**	**(12.6 m)**	1.6 m	(1.5 m)
Fig. 3.14a	876×584	256	**1.1 m**	**(2.2 m)**	1.4 m	(2.7 m)	0.6 m	(1.1 m)
Fig. 3.14b	876×584	256	**1.1 m**	**(2.2 m)**	1.4 m	(2.7 m)	0.5 m	(0.9 m)
Fig. 3.14c	876×584	256	**2.5 m**	**(4.9 m)**	5.4 m	(10.5 m)	0.5 m	(0.9 m)
Fig. 3.15b	640×540	256	1.5 m	(4.3 m)	**1.9 m**	**(5.5 m)**	0.5 m	(1.4 m)
Fig. 3.16a	1200×1400	256	**6.7 m**	**(4.0 m)**	12 m	(7.1 m)	3.7 m	(2.2 m)
Fig. 3.16b	1200×1400	256	**7.0 m**	**(4.2 m)**	13 m	(7.7 m)	3.7 m	(2.2 m)
Fig. 3.16c	1200×1400	256	67 m	(40 m)	**20 m**	(**12 m**)	4.7 m	(2.8 m)

explicitly constructing the layer geometry, but also allows constructing light transport paths that would not easily be available to a generic light transport algorithm, due to our new position-free path formulation.

Within this framework, we introduced unbiased Monte Carlo techniques analogous to a forward path tracer with next event estimation (NEE) and a fully bidirectional estimator. We demonstrated the capabilities of our solution on a number of examples, featuring multiple layers with surface and volumetric scattering, surface and phase function anisotropy, and spatial variation in all parameters. This leads to the first BSDF layering solution that offers unbiased accuracy and full flexibility in setting the layer properties.

Chapter 4

Bulk Scattering in Volumetric Rendering

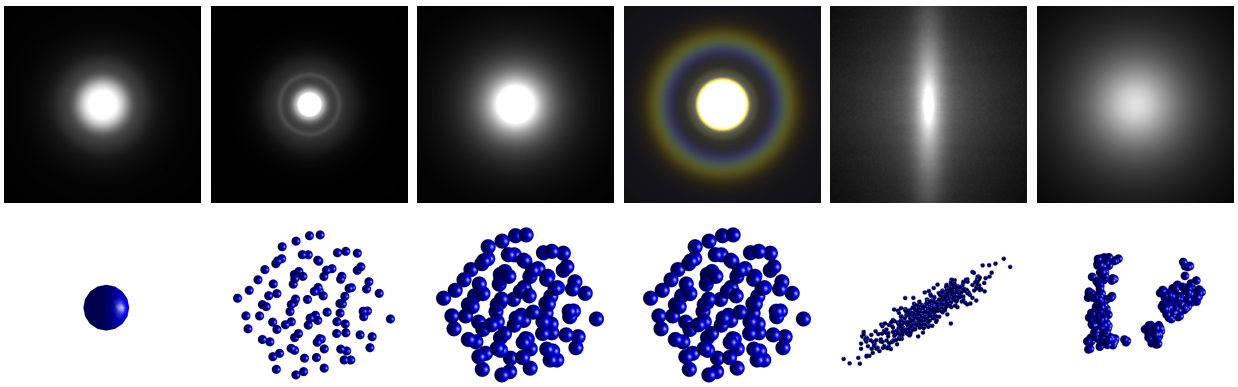


Figure 4.1: We introduce a new technique to compute bulk scattering parameters (i.e., the extinction and scattering coefficients as well as the single-scattering phase function) in a systematic fashion. By considering wave optical effects and particle (scatterer) interactions at the microscopic level, our technique enjoys the generality of supporting a wide range of media (e.g., isotropic, anisotropic, and correlated). In this figure, we show renderings of thin slabs lit with a small area light from behind (top). Additionally, we show visualizations of the corresponding particle distributions (middle) as well as per-cluster particle counts N^{cls} radii a_i (bottom).

4.1 Introduction

Participating media and translucent materials—such as marble, milk, wax, and human skin—are ubiquitous in the real world. These materials allow light to penetrate their surfaces and scatter in the interior.

In computational optics and computer graphics, how light interacts with participating media and translucent materials is typically modeled using the radiative transfer theory (RTT). Under this formulation, a participating medium consists of microscopic particles (*scatterers*) randomly dispersed in some homogeneous embedding medium. After entering a translucent material, light travels in straight lines in the embedding medium and occasionally collides with a particle and gets redirected into a new direction. To capture the macroscopic behavior of light, the RTT uses a statistical description of the particles (the medium bulk parameters), namely the extinction coefficient σ_{t} (aka. optical density), the scattering coefficient σ_{s}, and the phase function f_{p}.

While purely phenomenological in origin, the RTT has been demonstrated a corollary of Maxwell equations, under the assumption of far-field or independent scattering [94]. Therefore, these optical bulk parameters can be obtained from first principles, using e.g. Lorenz-Mie theory [123, 32]. However, although very successful in practice, this theory neglects the interactions occurring between particles in their near-field, including wave-optics effects such as diffraction and interference with neighbor particles. Consequently, Lorenz-Mie theory is largely limited to isotropic media with relatively low packing rates.

Previously, the classical radiative transfer theory has been generalized to handle materials with (statistically) organized microstructures. Anisotropic media [65], for instance, have bulk scattering parameters with stronger directional dependency compared to isotropic media. Additionally, media comprised of particles with correlated locations can exhibit non-exponential transmittance and characteristic scattering profiles [15, 67]. Although several

empirical models have been proposed to model these media, these still base on the very same far-field assumption of Lorenz-Mie scattering. Therefore, techniques capable of computing the bulk optical parameters of a material, based its microscopic properties, have been lacking.

In this chapter, we bridge this gap by introducing a new technique to systematically and rigorously compute the bulk scattering parameters. The elementary building block of our technique is *particle clusters* in which individual particles follow user-specified distributions. Within a cluster, we consider full near-field light transport effects; Between clusters, on the contrary, we use a far-field approximation to allow efficient modeling of macroscopic level light transport.

Our formulation is derived from first principles of light transport (i.e., Maxwell electromagnetism) and reduces to the Lorenz-Mie theory in the special case of single-particle scatterers. Based on this formulation, we demonstrate how the bulk parameters can be computed numerically. Using our technique, we systematically generate radiative transfer optical parameters capturing multi-spectral, anisotropic, and correlated scattering effects for particles with arbitrary distributions (Figure 4.1).

Concretely, our contributions include:

- Establishing a computational framework for modeling light scattering from clusters of particles (§4.4).
- Showing how radiatve transfer parameters can be computed numerically based on our formulation (§4.5).
- Demonstrating how our technique can be applied to systematically compute scattering parameters for a variety of participating media (§4.6).

4.2 Related Work

Radiative Transfer. Simulating the propagation of light in participating media has been widely studied in graphics [103], building upon the radiative transfer equation (RTE), introduced 125 years ago by von Lommel [126] (see [95] for a historical perspective).

This escalar radiative formulation has been extended in graphics accounting for anisotropic [65], refractive [8], bispectral [52], or spatially-correlated media [67, 15]. All these works assume a radiometric light transport model, establishing no connections with the electromagnetic behaviour governing light transport.

From a wave optics perspective, a few works have generalized light transport in media to account for wave-based properties, including polarized light transport [134, 68], or coherence [10]. This last work is of special relevance, given that it was able to simulate purely wave-based phenomena such as speckle or coherent back-scattering on top of a radiative model.

All these works build on the assumption of the far-field approximation and independent scattering, which largely simplifies computations. A notable exception is the near-field model proposed by Bar et al.[11], that renders speckle statistics in the near-field zone of the camera, although it still considers independent far-field scattering between particles. In contrast, in this work we explicitly relate the radiometric light transport modeled by the RTE with physics-based optics based on electromagnetism, and generalize the independent scattering approximation to account clusters of particles in the near field.

Modeling scattering in media The phase function models the average scattering distribution at an interaction with the medium. A common approach is to use simple phenomenological models, such as the Henyey-Greenstein phase function [58] or mixtures of von Mishes-Fisher distributions [40], as well as other functions modeling the scattering of ideal-

ized anistropic particles [141, 56]; however, these methods lack an explicit relationship with the underlying microscopic material properties. Under the assumption of geometric optics, several works have proposed to precompute the phase functions of more complex particles for granular materials [93, 99] or cloth fibers [7] using explicit path tracing, by neglecting wave effects.

A more rigorous phase function is based on the Lorenz-Mie theory [123], which provides closed-form solutions for the Maxwell's equations for spherical particles [63, 32]. Sadeghi et al. [112] generalized the Lorenz-Mie theory to larger non-spherical particles in the context of accurately modeling rainbows. To avoid the expensive sum series of the Lorenz-Mie theory, Guo et al. [46] proposed to use the geometric optics approximation [41], which give a good approximation to Lorenz-Mie theory for larger particles at significantly lower cost.

All these approaches provide accurate rigorous solutions to the far-field scattering of disperse particles.

Beyond Lorenz-Mie, several exact rigorous solutions have been proposed for computing electromagnetic scattering of particles in media, including the finite elements method (FEM), the finite difference time domain (FDTD) method, or the boundary elements method (BEM) [135], which solve the Maxwell's equations for arbitrary shapes. Xia et al. [136] proposed using BEM for accurately precomputing the far-field scattering of individual fibers. Unfortunately these methods are very slow as the number of particles increase, limiting its applicability to individual elements in problems with reduced dimensionality.

The T-matrix method [130] generalizes the Lorentz-Mie theory to particles of arbitrary shape in both the near- and far-fields, with the only assumption of the computed field being outside a sphere surrounding the particles. This method was later extended to clusters of multiple particles [106, 90]. We leverage the T-matrix method for computing the scattering of groups of particles.

Wave optics in surface scattering Inspired on the vast background on electromagnetic surface scattering in optics (see [34] for a general survey), several works in graphics have taken into account relevant wave effects including diffraction-aware BSDFs [55, 117, 21, 26, 59, 120, 132, 138], goniochromatic patterns due to thin-layer interference [116, 42, 13, 45] or birefringence [119]. These works assume single scattering, with no interaction between different particles with a few exceptions that assume full incoherence after single scattering [29, 45]. Notably, Moravec [98] and Musbach et al. [100] computed the full electromagnetic surface scattering by solving the full wave propagation using the FDTD.

4.3 Preliminaries

We now briefly revisit the basics on first principles of (classical) light transport theory based on Maxwell electromagnetism. Table 4.1 summarize the symbols using along this chapter.

4.3.1 Electromagnetic Scattering

The propagation of a time-harmonic monochromatic electromagnetic field with frequency ω is defined by the Maxwell curl equations as

$$\begin{aligned} \nabla \times \mathbf{E}(\mathbf{r}) &= \mathrm{i}\,\omega\,\mu(\mathbf{r})\,\mathbf{H}(\mathbf{r}), \\ \nabla \times \mathbf{H}(\mathbf{r}) &= \mathrm{i}\,\omega\,\varepsilon(\mathbf{r})\,\mathbf{E}(\mathbf{r}), \end{aligned} \tag{4.1}$$

where $\nabla \times \cdot$ is the curl operator; $\mathbf{E}(\mathbf{r})$ and $\mathbf{H}(\mathbf{r})$ indicate, respectively, the (vector-valued) electric and magnetic fields at $\mathbf{r}$; $\mu(\mathbf{r})$ and $\varepsilon(\mathbf{r})$ denote the (scalar-valued) magnetic permeability and electric permittivity at $\mathbf{r}$, respectively; and $\mathrm{i} := \sqrt{-1}$ is the imaginary unit.

Assuming a non-magnetic medium satisfying $\mu(\mathbf{r}) = \mu_0$ with μ_0 being the magnetic perme-

Table 4.1: Notation used in §4.

$\mathbf{r} \in \mathbb{R}^3$	Position
$\hat{\mathbf{r}} \in \mathbb{S}^2$	Direction to $\mathbf{r}$.
$r \in \mathbb{R}$	Distance.
$\varepsilon(\mathbf{r})$	Permittivity
$\mu(\mathbf{r})$	Permeability
ω	Wave angular frequency [1/s]
$\lambda = 2\pi\omega^{-1}$	Wavelength [m]
$k(\mathbf{r}) = \omega\sqrt{\varepsilon(\mathbf{r})\mu(\mathbf{r})}$	Wavenumber at $\mathbf{r}$
$m(\mathbf{r}) = k_2(\mathbf{r})/k_1$	Relative refractive index at $\mathbf{r}$
$\mathbf{H}(\mathbf{r})$	Magnetic field at $\mathbf{r}$
$\mathbf{E}(\mathbf{r})$	Electric field at $\mathbf{r}$ (4.4)
$\mathbf{E}^{\mathrm{inc}}(\mathbf{r})$	Incident electric field $\mathbf{r}$
$\mathbf{E}^{\mathrm{sca}}(\mathbf{r})$	Scattered electric field at $\mathbf{r}$ (4.4)
$\mathbf{E}_0$	Amplitude of a planar electric field
$\mathbf{E}_1^{\mathrm{sca}}(\hat{\mathbf{r}})$	Far-field angular distribution of the scattered radiation
$\overleftrightarrow{G}$	Free-space dyadic Green's function (4.5)
$\overleftrightarrow{T}$	Dyad transition operator (4.9)
$g(\hat{\mathbf{n}}, \mathbf{r})$	Planar field scalar propagator
V_i	Volume suspended by particle/cluster i
$\mathbf{R}_i \in \mathbb{R}^3$	Representative position of particle/cluster i
$\hat{\mathbf{R}}_{ij} \in \mathbb{S}^2$	Direction from $\mathbf{R}_j$ to $\mathbf{R}_i$
$R_{ij} \in \mathbb{R}$	Distance from $\mathbf{R}_j$ to $\mathbf{R}_i$
N^{cls}	Number of particles in a cluster
$\mathbf{E}_i^{\mathrm{sca}}(\mathbf{r})$	Scattered field of $\mathbf{r} \in V_i$ (4.8)
$\mathbf{E}_i(\mathbf{r})$	Exciting field in $\mathbf{r} \in V_i$
$\mathbf{E}_{ij}^{\mathrm{exc}}(\mathbf{r})$	Partial exciting field in $\mathbf{r} \in V_i$ from particle j (4.10)
$\overleftrightarrow{A}_i^{\mathrm{near}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \mathbf{r})$	Near-field scattering dyad of particle/cluster i (4.20).
$\overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}})$	Far-field scattering dyad of particle/cluster i (4.23).
$C_{\mathrm{t}}(\hat{\mathbf{n}}^{\mathrm{inc}}), C_{\mathrm{s}}(\hat{\mathbf{n}}^{\mathrm{inc}})$	Extinction (4.28) and scattering (4.29) cross-sections [m^2]
$f_{\mathrm{p}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}})$	Phase function (4.30)
ρ	Particles density [m^{-3}]
$\sigma_{\mathrm{t}}(\hat{\mathbf{n}}^{\mathrm{inc}}), \sigma_{\mathrm{s}}(\hat{\mathbf{n}}^{\mathrm{inc}})$	Extinction (4.31) and scattering (4.32) coefficients [m^{-1}]

ability of a vacuum, Equation (4.1) reduces to the electric field wave equation

$$\nabla^2 \times \mathbf{E}(\mathbf{r}) - k^2\,\mathbf{E}(\mathbf{r}) = 0, \tag{4.2}$$

where $k(\mathbf{r}) = \omega\sqrt{\varepsilon(\mathbf{r})\mu_0}$ is the medium's wave number at $\mathbf{r}$.

We now assume an infinite homogeneous isotropic medium with permittivity ε_1, filled with scatterers bounded by a finite disjoint region V, with potentially inhomogeneous permittivity $\varepsilon_2(\mathbf{r})$. Under this assumption, we can solve Equation (4.2) by expressing it as the *volume integral equation* (see §3.1 of Mishchenko's work [97] for a step-by-step derivation) as the sum of the incident field $\mathbf{E}^{\text{inc}}(\mathbf{r})$ and the scattered field $\mathbf{E}^{\text{sca}}(\mathbf{r})$ due to inhomogeneities in the medium in the form of scatterers:

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + \mathbf{E}^{\text{sca}}(\mathbf{r}) \tag{4.3}$$

$$= \mathbf{E}^{\text{inc}}(\mathbf{r}) + k_1^2 \int_V [m^2(\mathbf{r}') - 1]\, \overleftrightarrow{G}(\mathbf{r}, \mathbf{r}')\, \mathbf{E}(\mathbf{r}')\, \mathrm{d}\mathbf{r}', \tag{4.4}$$

with k_1 the wave number at the hosting medium, $m(\mathbf{r}) = k_2(\mathbf{r})/k_1$ the index of refraction of the interior regions V with respect to the hosting medium, and $\overleftrightarrow{G}(\mathbf{r}, \mathbf{r}')$ the free-space dyadic Green's function defined as:

$$\overleftrightarrow{G}(\mathbf{r}, \mathbf{r}') = \left(\overleftrightarrow{I} + k_1^{-2}\, \nabla \otimes \nabla\right) \frac{\exp(\mathrm{i}\, k_1\, |\mathbf{r} - \mathbf{r}'|)}{4\pi\, |\mathbf{r} - \mathbf{r}'|}, \tag{4.5}$$

where $\overleftrightarrow{I}$ is the identity dyad, and $.\otimes.$ denotes the dyadic product of two vectors. Intuitively, Equation (4.4) models the scattering field as the superposition of the spherical wavelets resulting from a change of permitivitty (i.e. with $m(\mathbf{r}') \neq 1$). Note also the recursive nature of Equation (4.4); we will deal with this recursivity in the following section, computing $\mathbf{E}^{\text{sca}}(\mathbf{r})$ as a function of the incident field $\mathbf{E}^{\text{inc}}(\mathbf{r})$.

4.3.2 Foldy-Lax Equations

We now consider a medium filled with N finite discrete particles with volume V_i and index of refraction $m_i(\mathbf{r})$. Considering an incident E-field $\mathbf{E}^{\text{inc}}(\mathbf{r})$, we can rewrite Equation (4.4)

as

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + \int_{\mathbb{R}^3} U(\mathbf{r}')\, \overleftrightarrow{G}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{E}(\mathbf{r}')\, \mathrm{d}\mathbf{r}', \tag{4.6}$$

where $\overleftrightarrow{G}(\mathbf{r}, \mathbf{r}')$ is the dyadic Green's function (4.5), and $U(\mathbf{r})$ the potential function given by

$$U(\mathbf{r}) = \sum_{i=1}^{N} U_i(\mathbf{r}) \quad \text{with} \quad U_i(\mathbf{r}) = \begin{cases} 0, & (\mathbf{r} \notin V_i) \\ k_1^2[m_i^2(\mathbf{r}) - 1]. & (\mathbf{r} \in V_i) \end{cases} \tag{4.7}$$

By combining Equations (4.6) and (4.7), we can express the field at any position $\mathbf{r} \in \mathbb{R}^3$ following the so-called *Foldy-Lax equation* [30, 81] as

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + \sum_{i=1}^{N} \overbrace{\int_{V_i} \overleftrightarrow{G}(\mathbf{r}, \mathbf{r}') \cdot \int_{V_i} \overleftrightarrow{T}_i(\mathbf{r}', \mathbf{r}'') \cdot \mathbf{E}_i(\mathbf{r}'')\, \mathrm{d}\mathbf{r}''\, \mathrm{d}\mathbf{r}'}^{=: \mathbf{E}_i^{\text{sca}}(\mathbf{r})} \tag{4.8}$$

with $\mathbf{E}_i^{\text{sca}}(\mathbf{r})$ and $\mathbf{E}_i(\mathbf{r})$ the scattered and partial field of particle i, and $\overleftrightarrow{T}_j(\mathbf{r}, \mathbf{r}')$ $\overleftrightarrow{T}_i(\mathbf{r}, \mathbf{r}')$ the dyad transition operator for particle i defined as [121]

$$\overleftrightarrow{T}_i(\mathbf{r}, \mathbf{r}') = U_i(\mathbf{r})\, \delta(\mathbf{r} - \mathbf{r}')\, \overleftrightarrow{I} + U_i(\mathbf{r}) \int_{V_i} \overleftrightarrow{G}(\mathbf{r}, \mathbf{r}'') \cdot \overleftrightarrow{T}(\mathbf{r}'', \mathbf{r}')\, \mathrm{d}\mathbf{r}'', \tag{4.9}$$

with $\delta(x)$ the Dirac delta. The partial field at particle i is defined as $\mathbf{E}_i(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + \sum_{j(\neq i)=1}^{N} \mathbf{E}_{ij}^{\text{exc}}(\mathbf{r})$, where the partial exciting field $\mathbf{E}_{ij}^{\text{exc}}(\mathbf{r})$ from particles j to i is

$$\mathbf{E}_{ij}^{\text{exc}}(\mathbf{r}) = \int_{V_j} \overleftrightarrow{G}(\mathbf{r}, \mathbf{r}') \int_{V_j} \overleftrightarrow{T}_j(\mathbf{r}', \mathbf{r}'') \mathbf{E}_j(\mathbf{r}'')\, \mathrm{d}\mathbf{r}''\, \mathrm{d}\mathbf{r}', \tag{4.10}$$

with $\mathbf{r} \in V_i$. Note that the scattered and exciting fields for particle j have essentially the same form. As shown by Mishchenko [94], the Foldy-Lax equation (4.8) solves exactly the volume integral equation (4.4) for multiple arbitrary particles in the medium, without any assumptions on their composition or packing rate, beyond the assumption of a homogeneous hosting medium.

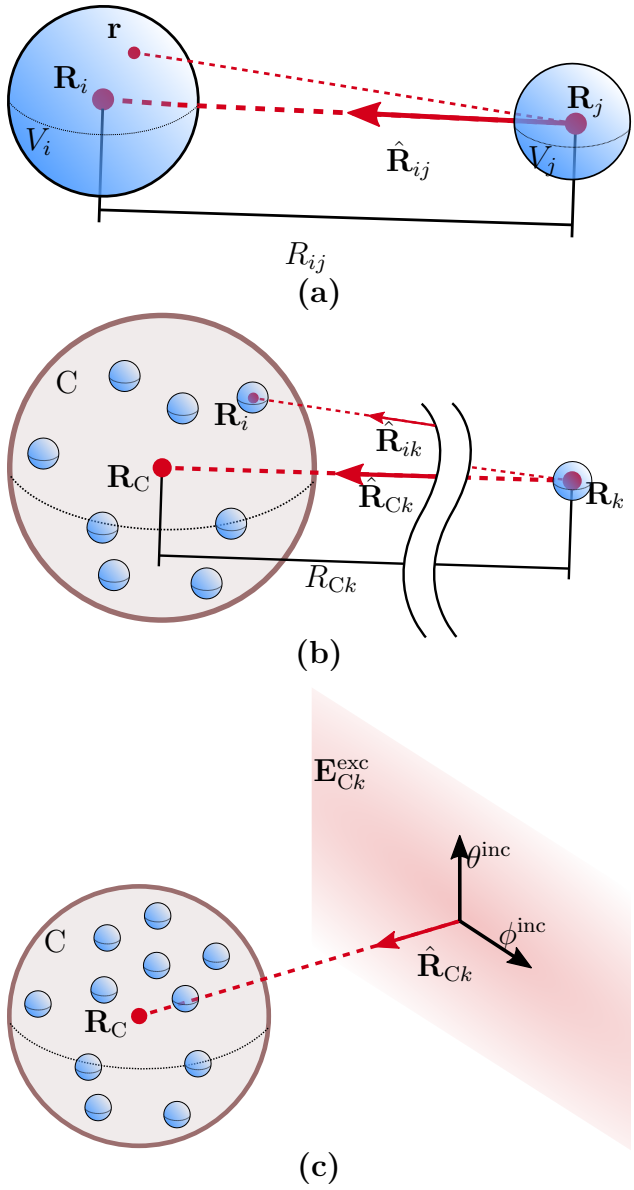


Figure 4.2: Schematical representation of the particles scattering geometry. (a) Previous methods, including Lorenz-Mie theory, assume independent scattering of particles, assuming that the distance R_{ij} between two particles i and j is very large (i.e. $R_{ij} \to \infty$), neglecting the potential interactions between particles. **(b)** In our work we differentiate between near field scattering of particles within a small region in space (cluster C centered at $\mathbf{R}_C$), and particles k on the far-field region of the cluster (distance $R_{Ck} \to \infty$). **(c)** For large values of R_{Ck}, the direction between particle k and any particle $j \in$ C is $dPx_{ik} \approx \hat{\mathbf{R}}_{Ck}$: Therefore, we can assume an planar exciting field $\mathbf{E}^{\text{exc}}(\mathbf{r})_{Ck}$ on the whole cluster C from particle k, with direction $\hat{\mathbf{R}}_{Ck}$.

Far-field Foldy-Lax Equations Equation (4.10) defines the exact exciting field resulting from the scattering by particle j on particle i. However, if the distance $R_{ij} := \|\mathbf{R}_i - \mathbf{R}_j\|$

between particles (with $\mathbf{R}_i$ denoting the center of particle i) is large, we can approximate the propagation distance between any point $\mathbf{r} \in V_i$ and $\mathbf{r}' \in V_j$ as

$$\|\mathbf{r} - \mathbf{r}'\| \approx R_{ij} + (\hat{\mathbf{R}}_{ij} \cdot \Delta\mathbf{r}) - (\hat{\mathbf{R}}_{ij} \cdot \Delta\mathbf{r}'), \tag{4.11}$$

with $\hat{\mathbf{R}}_{ij} := (\mathbf{R}_i - \mathbf{R}_j)/R_{ij}$, $\Delta\mathbf{r} := \mathbf{r} - \mathbf{R}_i$ and $\Delta\mathbf{r}' := \mathbf{r}' - \mathbf{R}_j$ (see Figure 4.2, left). With this approximation, we can now express $\mathbf{E}_{ij}^{\text{exc}}(\mathbf{r})$ for a point $\mathbf{r} \in V_i$ using its *far-field* approximation, as:[1]

$$\begin{aligned} &\mathbf{E}_{ij}^{\text{exc}}(\mathbf{r}) \\ &\approx \frac{\mathrm{e}^{\mathrm{i}k_1(R_{ij}+\hat{\mathbf{R}}_{ij}\cdot\Delta\mathbf{r})}}{4\pi R_{ij}} \int_{V_j} g(\hat{\mathbf{R}}_{ij}, \Delta\mathbf{r}') \int_{V_j} \overleftrightarrow{T}_j(\mathbf{r}', \mathbf{r}'') \cdot \mathbf{E}_j(\mathbf{r}'')\, \mathrm{d}\mathbf{r}''\, \mathrm{d}\mathbf{r}' \\ &= \frac{\exp(\mathrm{i}k_1 R_{ij})}{R_{ij}} g(\hat{\mathbf{R}}_{ij}, \Delta\mathbf{r})\, \mathbf{E}_{1ij}^{\text{exc}}(\hat{\mathbf{R}}_{ij}), \end{aligned} \tag{4.12}$$

where: $\mathbf{r} \in V_i$ is a point in particle i; $g(\hat{\mathbf{n}}, \Delta\mathbf{r}) = \exp(\mathrm{i}k_1 \hat{\mathbf{R}}_{ij} \cdot \Delta\mathbf{r})$; and $\mathbf{E}_{1ij}^{\text{exc}}$ is the far-field exciting field from particle j to particle i that is solely characterized by the propagation direction $\hat{\mathbf{R}}_{ij}$. In order for Equation (4.12) to be valid, the distance R_{ij} needs to hold the far-field criteria, which relates the R_{ij} with the radius of the particle a_j following the inequality [97]:

$$k_1 R_{ij} \gg \max\left(1, \frac{k_1^2 a_j^2}{2}\right). \tag{4.13}$$

This far-field assumption is both the basis for the Lorenz-Mie theory [123] (to model electromagnetic scattering from small spherical particles) and, as shown by Mishchenko [94], at the core of the radiative transfer theory.

In the following, we relax the assumption of near field scattering and compute the Foldy-Lax equations for clusters of particles for both the near- and far-field regions. Then, we use them to compute the scattering matrix to be used in the RTE to efficiently approximate light

[1] We note that, accordingly to Mishchenko [97], the product would require to multiply the integrand by the dyad $(\overleftrightarrow{I} - \hat{\mathbf{R}}_{ij} \otimes \hat{\mathbf{R}}_{ij})$ to ensure a transverse planar field; we remove it for clarity.

transport between clusters of particles.

4.4 Scattering from Clusters of Particles

In this section, we present our main theoretical result: the far-field approximated scattering dyad relating a field incoming at a particle, which will be shown in Equation (4.23). This dyad can then be used to compute a medium's bulk scattering parameters, which we will discuss in §4.4.1.

The two forms of computing the exciting field from particle j to i [Equations (4.10) and (4.12)] suggest that we can consider two subsets of particles j depending on their distance with respect to the point of interest $\mathbf{r}$: One set of N_{near} particles in the near field and another set of N_{far} particles in the far field. With that, we can now calculate the exciting field in particle i as:

$$\mathbf{E}_i(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + \sum_{j(\neq i)=1}^{N_{\text{near}}} \mathbf{E}_{ij}^{\text{exc}}(\mathbf{r}) + \sum_{k=1}^{N_{\text{far}}} \mathbf{E}_{ik}^{\text{exc}}(\mathbf{r}). \tag{4.14}$$

In what follows, we derive the far-field Foldy-Lax equations for groups of particles where a cluster of these particles are in their respective near-field region, while the other elements in the system are in the far field. For the simplicity of our derivations, we consider a single far-field incident field in the cluster, and assume that the far-field particles k do not have neighbor particles in their respective near field region.

More formally, we now consider a cluster C of N_{C} particles, where all particles $i \in \text{C}$ are in their respective near-field region, and that the particles of the cluster have a bounding sphere centered at $\mathbf{R}_{\text{C}}$ with radius a_{C} (see Figure 4.2, middle).

Since both the incident field $\mathbf{E}^{\text{inc}}(\mathbf{r})$ and the exciting field $\mathbf{E}_{Ck}^{\text{exc}}(\mathbf{r})$ from particle k are in the

far-field region, we can assume both fields to be planar waves defined as

$$\mathbf{E}^{\mathrm{inc}}(\mathbf{r}) = \mathbf{E}_0^{\mathrm{inc}} \exp(\mathrm{i}k_1 \hat{\mathbf{n}} \cdot \Delta\mathbf{r}) = \mathbf{E}_0^{\mathrm{inc}}\, g(\hat{\mathbf{n}}, \Delta\mathbf{r}), \tag{4.15}$$

$$\mathbf{E}_{\mathrm{C}k}^{\mathrm{exc}}(\mathbf{r}) = \mathbf{E}_{0\mathrm{C}k}^{\mathrm{exc}} \exp(\mathrm{i}k_1 \hat{\mathbf{R}}_{\mathrm{C}k} \cdot \Delta\mathbf{r}) = \mathbf{E}_{0\mathrm{C}k}^{\mathrm{exc}}\, g(\hat{\mathbf{R}}_{\mathrm{C}k}, \Delta\mathbf{r}), \tag{4.16}$$

with $\mathbf{E}_0^{\mathrm{inc}}$ the amplitude of the planar incident field, $\hat{\mathbf{n}}$ its direction, and $\Delta\mathbf{r} = \mathbf{r} - \mathbf{R}_{\mathrm{C}}$. Equivalently, $\mathbf{E}_{0\mathrm{C}k}^{\mathrm{exc}} = \frac{\exp(\mathrm{i}k_1 R_{\mathrm{C}k})}{R_{\mathrm{C}k}} \mathbf{E}_{1\mathrm{C}k}^{\mathrm{exc}}(\hat{\mathbf{R}}_{\mathrm{C}k})$ is the amplitude of the exciting field at C from particle k, and $\hat{\mathbf{R}}_{\mathrm{C}k}$ its direction.

Now, let us slightly abuse the dot product notation, remove the dependency on the spatial dependency on each term, and use $(\varphi_1 \cdot \varphi_2) = \int \varphi_1(x)\, \varphi_2(x)\, \mathrm{d}x$ for scalar-valued functions φ_1 and φ_2. From the far-field assumptions, plugging Equation (4.14) into the definition of the scattered field from particle $i \in \mathrm{C}$ in Equation (4.8) (with $N_{\mathrm{near}} = N^{\mathrm{cls}}$) yields

$$\begin{aligned} \mathbf{E}_i^{\mathrm{sca}}(\mathbf{r}) &= \overleftrightarrow{G} \cdot \overleftrightarrow{T}_i \cdot \mathbf{E}_i \\ &= \overleftrightarrow{G} \cdot \overleftrightarrow{T}_i \cdot \left[\mathbf{E}^{\mathrm{inc}} + \sum_{k=1}^{N_{\mathrm{far}}} \mathbf{E}_{\mathrm{C}k}^{\mathrm{exc}} + \sum_{j(\neq i)=1}^{N^{\mathrm{cls}}} \mathbf{E}_{ij}^{\mathrm{exc}} \right]. \end{aligned} \tag{4.17}$$

By recursively expanding $\mathbf{E}_{ij}^{\mathrm{exc}}$ and some algebraic operations (see the supplemental for the full derivation), this results into

$$\begin{aligned} \mathbf{E}_i^{\mathrm{sca}}(\mathbf{r}) = \mathbf{E}_0 \overleftrightarrow{G} \cdot \overleftrightarrow{T}_i \cdot \left[g(\hat{\mathbf{n}}) + \sum_{j(\neq i)=1}^{N^{\mathrm{cls}}} [...]_j^{g(\hat{\mathbf{n}})} \right] \\ + \sum_{k=1}^{N_{\mathrm{far}}} \mathbf{E}_{0\mathrm{C}j}^{\mathrm{exc}} \left[\overleftrightarrow{G} \cdot \overleftrightarrow{T}_i \cdot \left[g(\hat{\mathbf{R}}_{\mathrm{C}k}) + \sum_{j(\neq i)=1}^{N^{\mathrm{cls}}} [...]_j^{g(\hat{\mathbf{R}}_{\mathrm{C}k})} \right] \right]. \end{aligned} \tag{4.18}$$

where the "$[...]_l^{\varphi}$" term represents the recursivity as

$$[...]_j^{\varphi} = \overleftrightarrow{G} \cdot \overleftrightarrow{T}_j \cdot \left[\varphi + \sum_{l(\neq j)=1}^{N^{\mathrm{cls}}} [...]_l^{\varphi} \right]. \tag{4.19}$$

Note that each element in the sum in Equation (4.18) above is the result of the amplitude of the far-field incident or exciting fields, and a series that encode all the near-field scattering in the cluster C. We can thus define the scattering dyad $\overleftrightarrow{A}_i^{\text{near}}(\hat{\mathbf{n}}^{\text{inc}}, \mathbf{r})$ relating a unit-amplitude planar incident field at particle i from direction $\hat{\mathbf{n}}^{\text{inc}}$ with the scattered field at point $\mathbf{r}$ as

$$\overleftrightarrow{A}_i^{\text{near}}(\hat{\mathbf{n}}^{\text{inc}}, \mathbf{r}) = \overleftrightarrow{G} \cdot \overleftrightarrow{T}_i \cdot \left[g(\hat{\mathbf{n}}^{\text{inc}}) + \sum_{j(\neq i)=1}^{N^{\text{cls}}} [...]_j^{g(\hat{\mathbf{n}}^{\text{inc}})} \right]. \tag{4.20}$$

By considering constant $\mathbf{E}_0^{\text{inc}}$ and $\mathbf{E}_{0\text{C}k}^{\text{exc}}$ for the whole cluster C, we can compute the cluster's scattering dyad as:

$$\overleftrightarrow{A}_{\text{C}}^{\text{near}}(\hat{\mathbf{n}}^{\text{inc}}, \mathbf{r}) = \sum_{i=1}^{N_{\text{C}}} \overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\text{inc}}, \mathbf{r}), \tag{4.21}$$

which defines the scattered field for a unit-amplitude incoming planar field in a scene consisting of the particles forming cluster C. In practice, the scattering dyad $\overleftrightarrow{A}_{\text{C}}^{\text{near}}(\hat{\mathbf{n}}^{\text{inc}}, \mathbf{r})$ can be computed numerically using standard methods from computational electromagnetics [96].

Far-field approximation Equation (4.20) represents the general form of the scattering dyad for particle i, which results into a five-dimensional function. Assuming that $\mathbf{r}$ is in the far-field region of a particle $i \in \text{C}$, by using the far-field approximation of the scattered or exciting field (4.12) (we refer to the supplemental document for the derivation), we get the scattered field by particle i as

$$\mathbf{E}_i^{\text{sca}}(\mathbf{r}) \approx \frac{\mathrm{e}^{\mathrm{i}k_1 R_i}}{R_i} \Big(\mathbf{E}_0 \overleftrightarrow{A}_i(\hat{\mathbf{n}}, \hat{\mathbf{R}}_i) + \sum_{k=1}^{N_{\text{far}}} \mathbf{E}_{0\text{C}k}^{\text{exc}} \overleftrightarrow{A}_i(\hat{\mathbf{R}}_{\text{C}k}, \hat{\mathbf{R}}_i) \Big), \tag{4.22}$$

with $R_i = |\mathbf{r} - \mathbf{R}_i|$ and $\hat{\mathbf{R}}_i = \frac{\mathbf{r}-\mathbf{R}_i}{R_i}$, and

$$\boxed{\overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}}) = \frac{g(\hat{\mathbf{n}}^{\text{sca}}) \cdot \overleftrightarrow{T}_i}{4\pi} \cdot \left[g(\hat{\mathbf{n}}^{\text{inc}}) + \sum_{j(\neq i)=1}^{N^{\text{cls}}} [...]_j^{g(\hat{\mathbf{n}}^{\text{inc}})} \right].} \tag{4.23}$$

Finally, since $\hat{\mathbf{R}}_i \approx \hat{\mathbf{R}}_{\mathrm{C}}$ for all particles $i \in \mathrm{C}$, we can approximate the far-field scattered field of cluster C as

$$\mathbf{E}_{\mathrm{C}}^{\mathrm{sca}}(\mathbf{r}) = \frac{\mathrm{e}^{\mathrm{i}k_1 R_{\mathrm{C}}}}{R_{\mathrm{C}}} \Big(\mathbf{E}_0 \overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{n}}, \hat{\mathbf{R}}_{\mathrm{C}}) + \sum_{k=1}^{N_{\mathrm{far}}} \mathbf{E}_{0\mathrm{C}k}^{\mathrm{exc}} \overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{R}}_{\mathrm{C}k}, \hat{\mathbf{R}}_{\mathrm{C}}) \Big), \tag{4.24}$$

where

$$\overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}}) = \sum_{i=1}^{N_{\mathrm{C}}} \overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}}), \tag{4.25}$$

is the far-field scattering dyad of cluster C.

Thus, by grouping the individual particles into N^{cls} near-field clusters, and assuming that all clusters and observation point $\mathbf{r}$ lay in their respective far field, we can approximate the Foldy-Lax equation (4.8) as

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{\mathrm{inc}}(\mathbf{r}) + \sum_{\mathrm{C}_j=1}^{N^{\mathrm{cls}}} \mathbf{E}_{\mathrm{C}_j}^{\mathrm{sca}}(\mathbf{r}), \tag{4.26}$$

with $\mathbf{E}_{\mathrm{C}_j}^{\mathrm{sca}}(\mathbf{r})$ the scattered field at cluster C_j.

4.4.1 Relationship with the Radiative Transfer Theory

The scattering dyad $\overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}})$ given by Equation (4.25) models how a particle cluster C scatters a planar unit-amplitude incident field in the far field region. However, for rendering we are generally interested on the average field intensity (i.e. radiance).

As shown by Mishchenko [94], the radiative transfer equation (RTE) directly derives from the far-field Foldy-Lax equations under three additional assumptions: (i) The amount of coherent backscattering is negligible; (ii) The particles are randomly distributed according to some distribution $p(R_i, \xi_i)$, with R_i and ξ denoting, respectively, the position and properties of a particle i; and (iii) We are interested on the average field $\langle \mathbf{E}(\mathbf{r}) \rangle$.

Following these assumptions, and after a lengthy derivation, Mishchenko demonstrates that the bulk scattering properties can be obtained from the far-field Foldy-Lax form, and in particular from the scattering dyad $\overleftrightarrow{A}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})$. Let us first assume that the distribution of particle properties ξ are independent of the particles position, and compute the average scattering dyad $\langle\overleftrightarrow{A}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})\rangle = \int_\Omega \overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}}) p(\xi_i)\, \mathrm{d}\xi_i$.

Then, note that the Foldy-Lax equation for clusters of particles (4.26), we derived above has the same form as the original Foldy-Lax equation (4.8). Thus, by the same derivation followed by Mishchenko we get to an equivalent RTE based on the scattering dyad of clusters.

Computing the scattering parameters By taking the vectors of the parallel and perpendicular polarization $\hat{\boldsymbol{\theta}}^{\text{inc}}$ and $\hat{\boldsymbol{\varphi}}^{\text{inc}}$ of the incident field as shown in Figure 4.2 (right), and equivalently for the scattered field $\hat{\boldsymbol{\theta}}^{\text{sca}}$ and $\hat{\boldsymbol{\varphi}}^{\text{sca}}$, we can compute the polarized scattering components $\boldsymbol{S}_\theta$ and $\boldsymbol{S}_\varphi$ from the cluster's scattering dyad $\overleftrightarrow{A}_{\text{C}}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})$ as

$$\begin{aligned}
\boldsymbol{S}_\theta(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}}) &= \hat{\boldsymbol{\theta}}^{\text{sca}} \cdot \langle\overleftrightarrow{A}_{\text{C}}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})\rangle \cdot \hat{\boldsymbol{\theta}}^{\text{inc}}, \\
\boldsymbol{S}_\varphi(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}}) &= \hat{\boldsymbol{\varphi}}^{\text{sca}} \cdot \langle\overleftrightarrow{A}_{\text{C}}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})\rangle \cdot \hat{\boldsymbol{\varphi}}^{\text{inc}}.
\end{aligned} \tag{4.27}$$

Then, based on the two scattering components $\boldsymbol{S}_\theta$ and $\boldsymbol{S}_\varphi$, we can obtain the optical parameters of the medium as

$$C_{\text{t}}(\hat{\mathbf{n}}^{\text{inc}}) = 4\pi\Re\left[\frac{\boldsymbol{S}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{inc}})}{k_i^2}\right], \tag{4.28}$$

$$C_{\text{s}}(\hat{\mathbf{n}}^{\text{inc}}) = \int_{\mathbb{S}}^{2} \frac{|\boldsymbol{S}_\theta(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})|^2 + |\boldsymbol{S}_\varphi(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})|^2}{2k_1^2}\, \mathrm{d}\hat{\mathbf{n}}^{\text{sca}}, \tag{4.29}$$

$$f_{\text{p}}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}}) = \frac{|\boldsymbol{S}_\theta(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})|^2 + |\boldsymbol{S}_\varphi(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{sca}})|^2}{2k_1^2 C_{\text{s}}}, \tag{4.30}$$

with $\boldsymbol{S}(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{inc}}) = \boldsymbol{S}_\varphi(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{inc}}) = \boldsymbol{S}_\theta(\hat{\mathbf{n}}^{\text{inc}}, \hat{\mathbf{n}}^{\text{inc}})$, and $\Re[x]$ returning the real part of a complex number x. Lastly, assuming a uniform distribution of clusters, we can compute the

extinction and scattering coefficients as

$$\sigma_{\mathrm{t}}(\hat{\mathbf{n}}^{\mathrm{inc}}) = C_{\mathrm{t}}(\hat{\mathbf{n}}^{\mathrm{inc}})\frac{\rho}{\langle N^{\mathrm{cls}}\rangle}, \tag{4.31}$$

$$\sigma_{\mathrm{s}}(\hat{\mathbf{n}}^{\mathrm{inc}}) = C_{\mathrm{s}}(\hat{\mathbf{n}}^{\mathrm{inc}})\frac{\rho}{\langle N^{\mathrm{cls}}\rangle}, \tag{4.32}$$

with ρ the number of particles per differential volume, and $\langle N^{\mathrm{cls}}\rangle$ the average number of particles per cluster. Note that the optical properties defined in Equations (4.28)–(4.32) are directionally dependent, so they are general and can represent both isotropic and anisotropic media.

4.4.2 Relationship with Independent Scattering

Most previous works rendering light transport in media [103] build on the assumption of independent scattering—that is, particles are in their respective far-field region. It is easy to verify that this is a special case of Equation (4.14) with $N^{\mathrm{cls}} = 1$, causing the scattering dyad $\overleftrightarrow{A}_{\mathrm{C}}$ of Equation (4.25) to reduce to

$$\overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}}) = \overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}}) = \frac{g(\hat{\mathbf{n}}^{\mathrm{sca}}) \cdot \overleftrightarrow{T}_i \cdot g(\hat{\mathbf{n}}^{\mathrm{inc}})}{4\pi}, \tag{4.33}$$

which encodes the scattered field in the far-field region of a particle when excited by an incident unit-amplitude planar field.

The Lorenz-Mie theory [123] provides closed-form expressions for $\overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}})$ for spheres and cylinders, while numerical solutions of $\overleftrightarrow{A}_i(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}})$ have been proposed for scatterers of arbitrary shapes via, for example, the T-matrix method [130], or more recently based on the BEM for cylindrical fibers [136]. Our work is therefore a generalization of these works to particles in the near field.

4.5 Computing the Bulk Scattering Parameters

We now detail our numerical computations of the scattering dyad $\overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}})$ of Equation (4.25), which in turn determines the bulk scattering parameters that can be directly used in any renderer supporting participating media.

Computing $\overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}})$ essentially boils down to solving the time-harmonic Maxwell equations for an incident unit-amplitude planar field with direction $\hat{\mathbf{n}}^{\mathrm{inc}}$. While several different methods exist for that purpose (see §16 of [96] for an overview), we opt for the superposition T-matrix method [89] that has been demonstrated efficient for moderately large N^{cls}, can handle scatterers with arbitrary geometry, and is based on the principles of the Foldy-Lax equations, making it particularly appealing for our work.

In practice, we use the open-source CUDA-based `CELES` solver [28], which implements the T-matrix method proposed by Mackowski and Mishchenko [90] for spherical or randomly rotated particles. In our implementation, we focus on clusters of spherical particles. Since the Lorenz-Mie theory also assumes spherical particles, this allows us to directly compare our results with those computed using the Lorenz-Mie theory.

To compute the average scattering dyad $\langle \overleftrightarrow{A}_{\mathrm{C}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \hat{\mathbf{n}}^{\mathrm{sca}}) \rangle$, we average the scattered field of several random realizations of the clusters (each of which obtained by randomly sampling the position of the particles inside the cluster's bounding sphere). As we will demonstrate in §4.6, we use a wide array of distributions including particles uniformly distributed over the volume of the cluster, positively-correlated particles following Shaw et al. [114], negatively-correlated particles using Poisson sampling of the sphere, and anisotropic distributions by uniformly sampling the particles on a oriented 2D disk.

Lastly, we represent the resulting phase function as well as the extinction and scattering cross sections as tabulated (i.e., piecewise constant) functions that can be used for rendering.

4.6 Experiments

In this section, we first validate our technique by comparing bulk scattering parameters computed with our method and the Lorenz-Mie theory (§4.6.1). Then, we apply our technique described in §4.4 and §4.5 to compute bulk scattering parameters for a wide range of participating media (§4.6.2).

4.6.1 Validation

To validate our technique, we compare computed bulk scattering parameters provided by our implementation and `MiePlot` [80], a free software based on the Lorenz-Mie theory. We focus on the configuration where a cluster contains only one (spherical) particle as this is a fundamental assumption of the Lorenz-Mie theory.

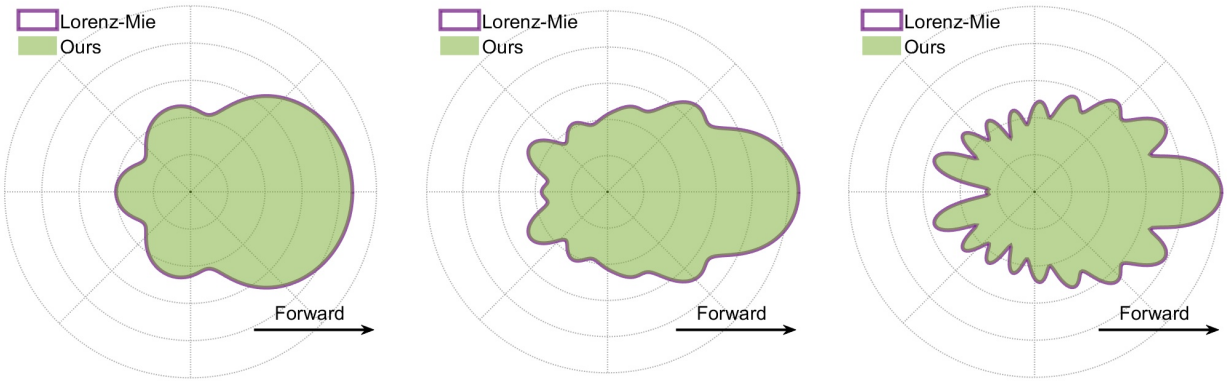


Figure 4.3: Comparison against Lorenz-Mie theory. We compare our method with clusters with a single particle $N^{\text{cls}} = 1$ against a reference solution based on Lorenz-Mie theory for three different particles radii $a_i \in \{300\text{nm}, 600\text{nm}, 900\text{nm}\}$. As expected, for a single particle our method reduces to the same results as Lorenz-Mie theory. The wavelength is $\lambda = 600\text{nm}$, while the refractive index of the particle is $m = 1.5 + 0.1\text{i}$.

In Figure 4.3, we visualize computed single-scattering phase functions at the wavelength 600 nm with three particle radii (300, 600, and 900 nm). We set the refractive index of the embedding medium to $(1.5 + 0.1\text{i})$.

Additionally, we show in Figure 4.4 the corresponding extinction and scattering cross sections C_{t} and C_{s} given by Equations (4.28) and (4.29), respectively. In all these examples, our computed scattering parameters match those predicted by the Lorenz-Mie theory perfectly.

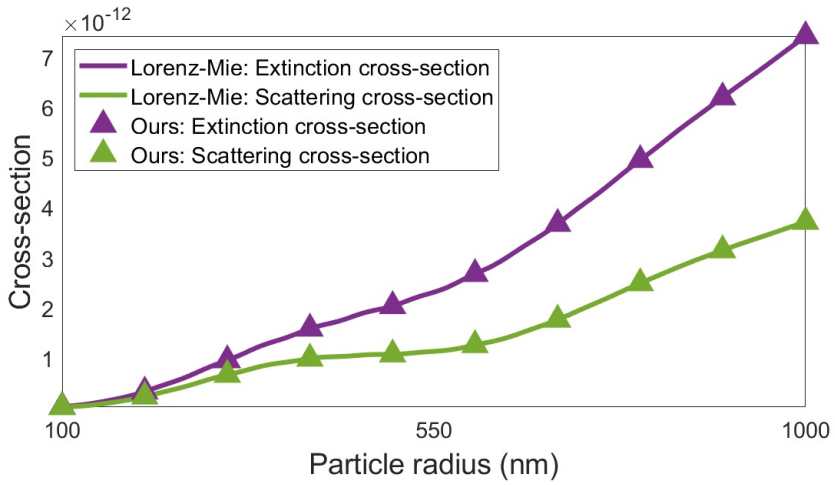


Figure 4.4: Comparison against Lorenz-Mie theory. We compare the extinction and scattering cross sections computed with our method for $N^{\mathrm{cls}} = 1$ against the results obtained using Lorenz-Mie theory. As in Figure 4.3 our results show perfect agreement.

4.6.2 Main Results

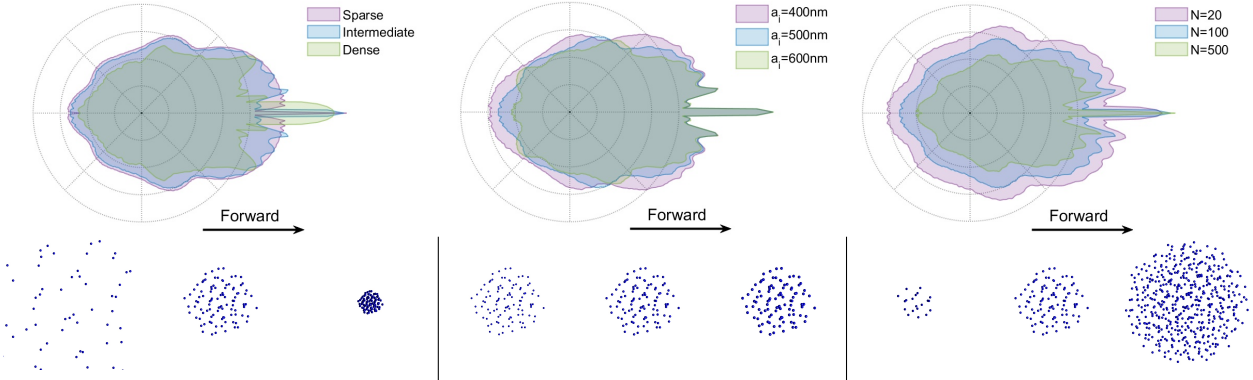
We now demonstrate the versatility of our technique by computing bulk scattering parameters for a range of participating media. Please see Table 4.2 for the performance statistics of our experiments.

Table 4.2: Performance statistics for our simulation. The numbers are collected using a workstation equipped with an Intel i7-6800K six-core CPU and an Nvidia GTX 1080 GPU. To average the randomness of the particle position, we run 50 times for each simulation, so all the number should times 50 for the results in this chapter.

	N	f_{p} res.	time
Regular (Fig 4.6)	1-500	180x360	3-16s
Multi-spectral (Fig 4.8)	100	180x360x50	35mins
Anisotropic (Fig 4.11)	100	180x360x90	13mins
Correlated (Fig 4.12)	100	180x360	98s

Isotropic media In computer graphics, volumetric light transport effects are typically simulated using *isotropic media* where the extinction and scattering coefficients σ_t, σ_s are directionally independent, and the single-scattering phase function f_p is formulated as a 1D function on the angle between the incident and scattered directions.

Our technique can produce bulk scattering parameters for isotropic media using particles distributed in radically symmetric densities. We conduct a few ablation studies to demonstrate how different particle arrangements in a cluster affects the resulting parameters. We use a wavelength of 600 nm for all these studies and represent the 1D phase functions as tabulated (i.e., piecewise constant) functions using 180 equal-sized bins.



(a) Varying particles spacing **(b)** Varying particles radius **(c)** Varying particles count

Figure 4.5: Comparison of the resulting phase function for different cluster parameters, for a planar incident field at $\lambda = 700$nm. Unless mentioned otherwise, the clusters have $N^{cls} = 100$ particles, and each particle has radius $a_i = 500$nm. For each phase function, we vary: (a) The distance between particles within the cluster; (b) The particle size a_i; and (c) The number of particles N^{cls}.

In our first study, we use a cluster of 100 particles with radii 500 nm. Then, we vary the distances between particles (by using bounding spheres with different sizes and distributing particles uniformly in these spheres). As shown in Figure 4.5 (a), the closer the particles are to each other, the more forward the resulting phase function is. This is expected: With sparsely distributed particles, it is simpler for light to pass straightly through.

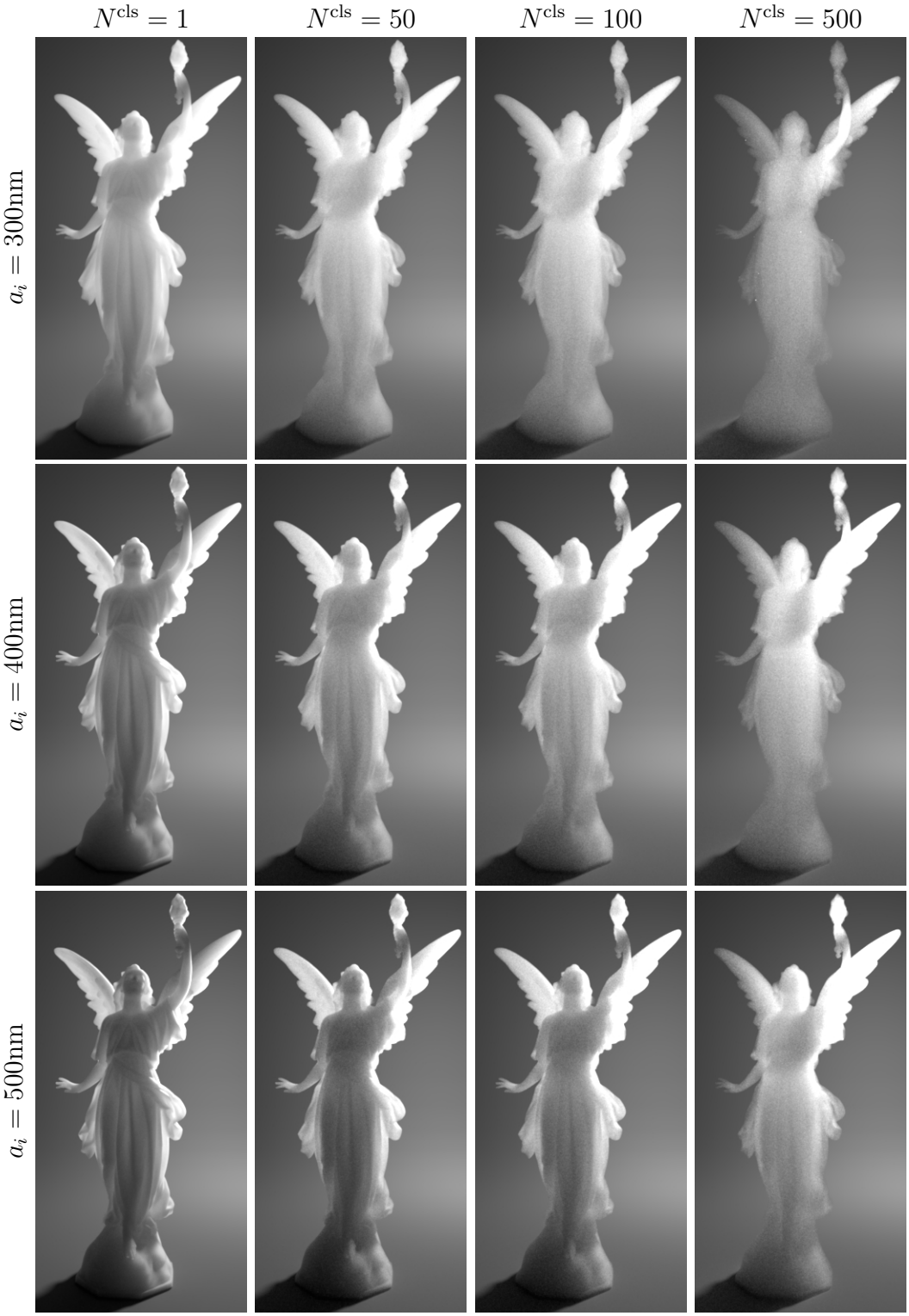


Figure 4.6: Renderings of homogeneous Lucy models at $\lambda = 700\text{nm}$. The bulk scattering parameters are computed using our method with different combinations of particle radius a_i and per-cluster particle count N^{cls}.

Our second ablation study examines the effect of particle size. With 100 uniformly distributed particles, we apply our technique to three particle sizes (a_i= 400, 500, and 600 nm). As shown in Figure 4.5 (b), as we increase the particles radius, the phase function becomes more forward and increases its frequency. This agrees with the behaviour of single particles predicted by Lorenz-Mie theory.

In our third study, we vary the number of particles in a cluster while keeping the particle size fixed to a_i=500 nm. Figure 4.5 (c) shows that as we increase the number of particles, the phase function gets more forward and of higher-frequency, in a behaviour somewhat correlated with the particles size. This is the result of the increasing number of diffractive elements on the cluster, that instead of making scattering more diffuse (as predicted by geometric optics) increases its forward frequency.

Lastly, we show in Figure 4.6 monochrome renderings using bulk scattering parameters obtained with varying combinations of particle count and radius.

Multi-spectral results Since our technique is derived using microphysical wave optics, it allows systematic generation of multi-spectral parameters based on a single (monochrome) configuration of particle cluster.

To demonstrate this, we use a configuration of 100 uniformly distributed particles (per cluster) with radius 500 nm and compute bulk scattering parameters at 50 wavelengths ranging from 400 nm to 700 nm.

In Figure 4.8, we visualize the computed phase functions at three wavelengths as well as multi-spectral renderings of a backlit thin slab. The smooth changes in scattering parameters across wavelength have resulted in a characteristic rainbow-like effect. When using the single-particle configuration (with identical overall particle density per unit volume), the rainbow effect is missing.

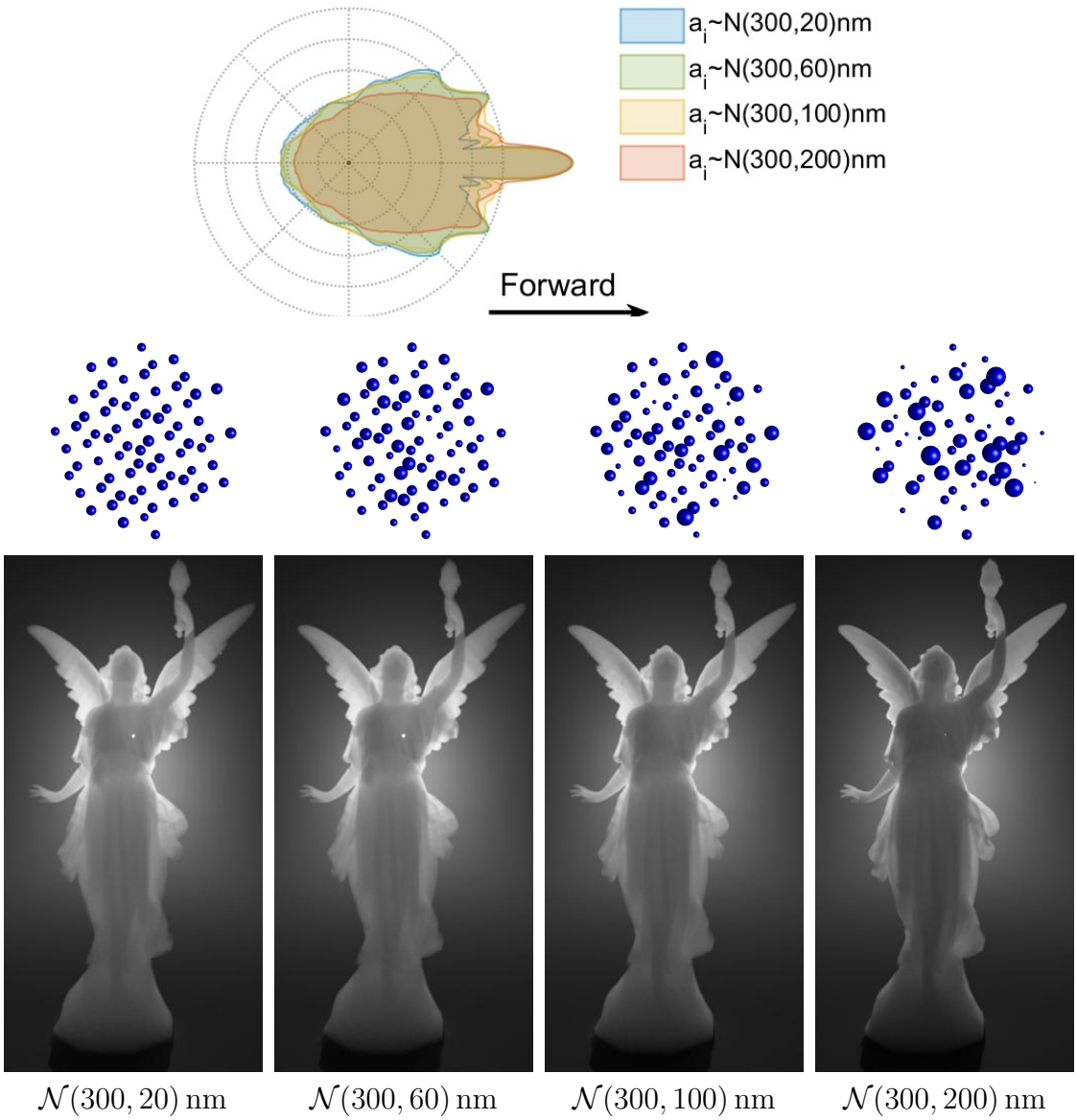
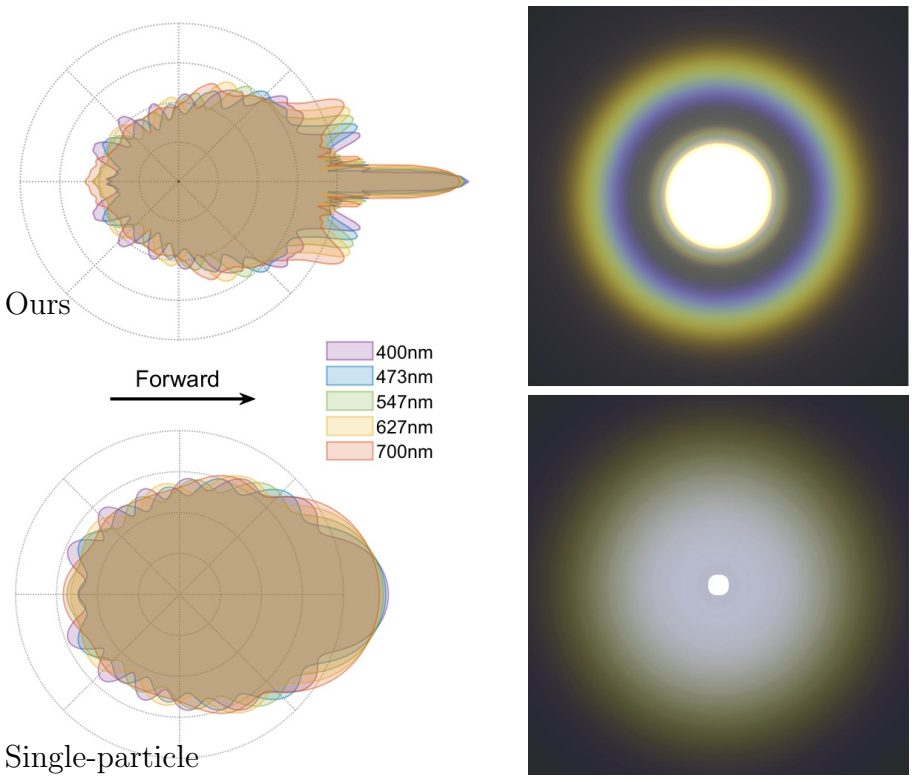


Figure 4.7: Comparison of the resulting phase function and rendering for different particle radius distribution. They have the same mean radius but different variations.

Figure 4.9 shows renderings of the Lucy model using these scattering parameters.

Anisotropic media Anisotropic media allow the extinction and scattering coefficients σ_t, σ_s to be directionally dependent, and have full 4D phase functions f_p. Previously, although the scattering parameters of anisotropic media can be devised based on the microflake models [65, 56], equivalences of the Lorenz-Mie theory, to our knowledge, have been lacking.



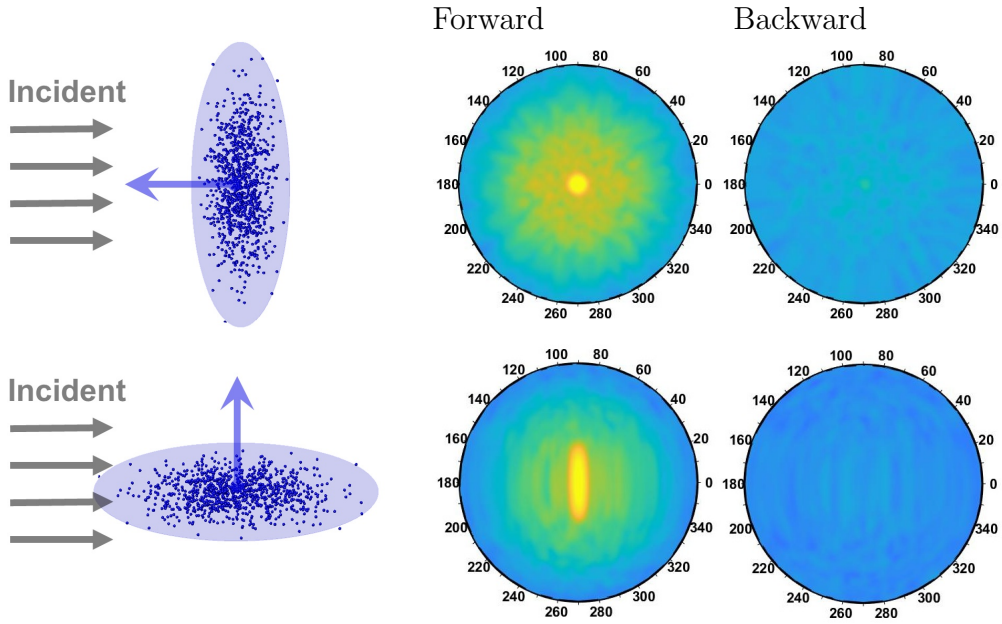
(a) Phase function **(b)** Thin-slab rendering

Figure 4.8: Multi-spectral results: (a) visualizations of phase functions; **(b)** corresponding multi-spectral renderings of a thin slab lit by a small area light from behind. Results on the top are generated using a cluster of 100 particles with radii 500nm. Results on the bottom are obtained using a conventional single-particle setting. We used identical particle counts per differential volume for both configurations.

By using anisotropic particle distributions, our technique can generate bulk scattering parameters for anisotropic media. To demonstrate this, we use a configuration where the cluster contains $N^{\text{cls}} = 100$ particles following an anisotropic Gaussian distribution, as illustrated in Figure 4.10(a). We tabulate the extinction and scattering cross sections using the latitude-longitude parameterization with a resolution of 180×360. Due to the symmetry of the disc, the resulting phase function f_{p} is three-dimensional, and we tabulated it with the resolution $90 \times 180 \times 360$. In Figure 4.10(b), we visualize slices of the computed single-scattering phase function f_{p} with two incident directions $\hat{\mathbf{n}}^{\text{inc}}$. In Figure 4.11, we show renderings of the Lucy model with three (spatially invariant) orientations.

(a) Multi. **(b)** 400nm **(c)** 550nm **(d)** 700nm

Figure 4.9: Multi-spectral rendering of Lucy model. **(a)** Multi-spectral rendering of a homogeneous Lucy model using identical bulk scattering parameters as the top row of Figure 4.8. **(b–d)** Monochrome renderings of the same model at three wavelengths.



(a) Incident direction **(b)** Phase function slice

Figure 4.10: Visualizations of slices $f_{\mathrm{p}}(\hat{\mathbf{n}}^{\mathrm{inc}}, \cdot)$ of a phase function for two incident directions $\hat{\mathbf{n}}^{\mathrm{inc}}$ at $\lambda = 700\mathrm{nm}$. This phase function is computed using a configuration where 100 particles with radii 500nm follow an anisotropic Gaussian distribution.

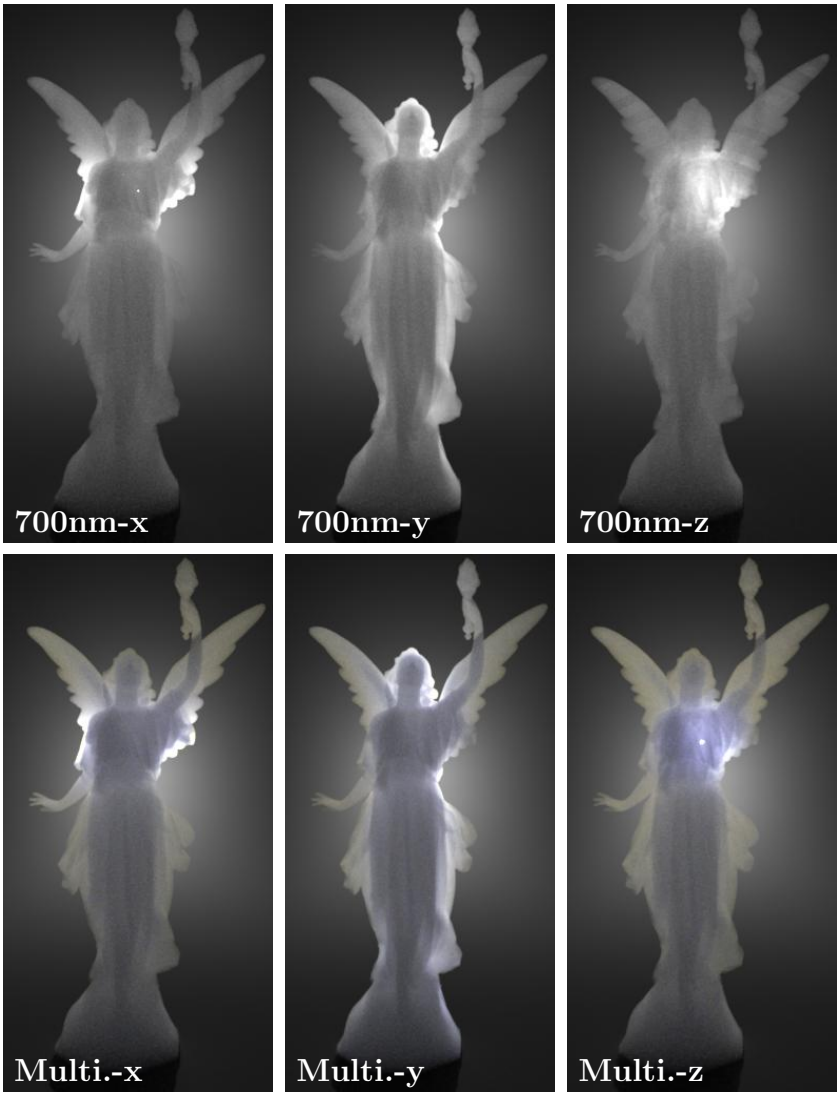


Figure 4.11: Renderings of homogeneous Lucy models with the same anisotropic medium as in Figure 4.10. With the medium's orientation – which determines the axis of the disk – aligned with the x-, y-, and z-axis, respectively, the Lucy model exhibit distinct appearances.

Correlated particles Finally, in Figure 4.12 we demonstrate the effect of particles correlation within the cluster, by analyzing particles distributed using both negative (Poisson sampled) and positive correlation [67]. We compare the effect of introducing microscopic correlation on media where the clusters position is itself correlated, compared with uniformly distributed particles inside the clusters. These two levels of correlation have significant effect on the final appearance of the translucent materials.

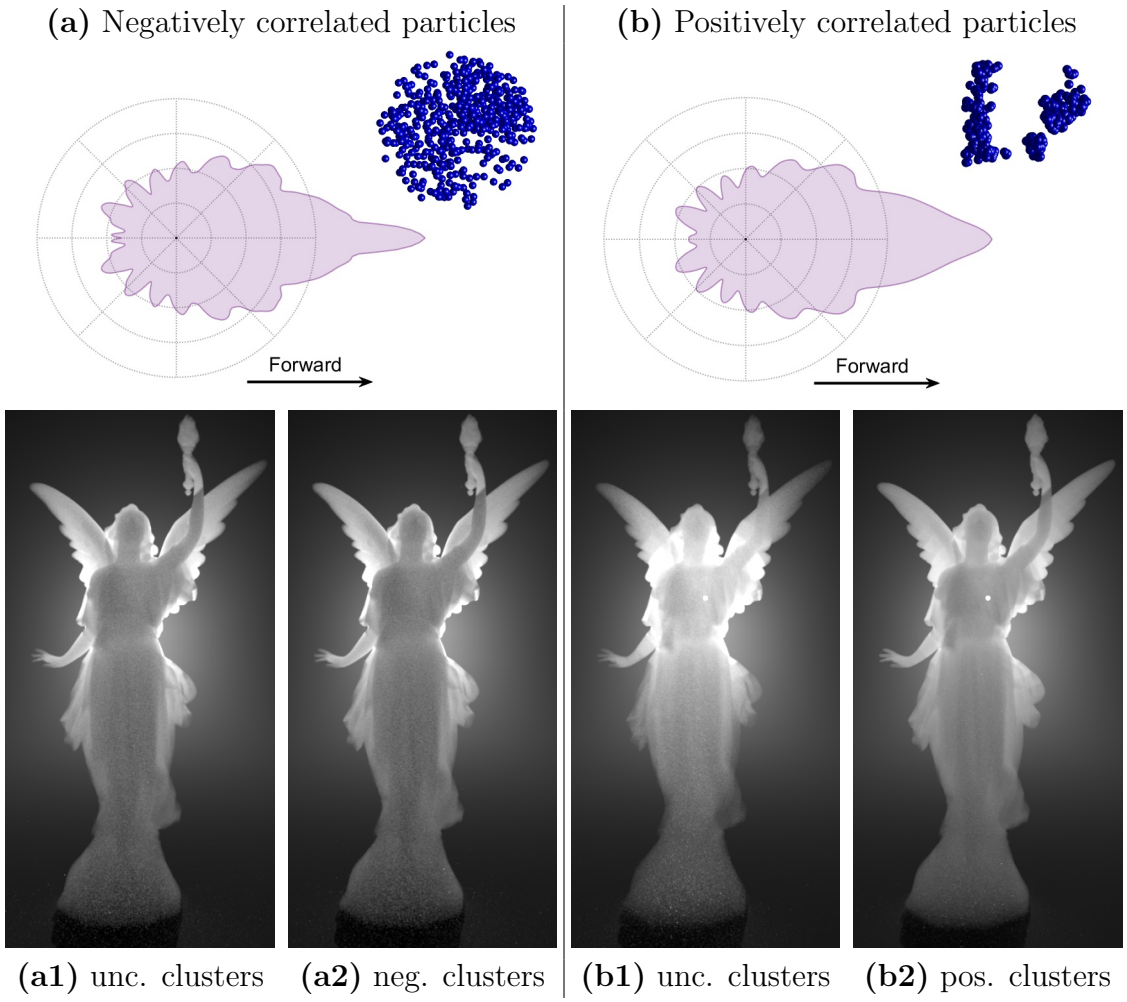


Figure 4.12: Correlated particles: By correlating particle positions in negatively (a) or positively (b), our method can produce bulk scattering parameters for correlated media. In this example, we use $\lambda = 400$nm, particle radius $a_i = 500$nm, and per-cluster particle count $N^{\text{cls}} = 100$. Additionally, we can further correlate particle clusters themselves, a variety of appearances can be achieved (**a1**–**b2**). (The bright dot in **(b1)** and **(b2)** emerges from unscattered light from the area source.)

4.7 Conclusion

In this chapter, we introduce a new technique to systematically compute bulk scattering parameters for participating media. Built upon first principles of light transport (i.e., Maxwell electromagnetism), our technique models a translucent material as clusters of particles randomly distributed in embedding media. Our work generalizes the widely-used Lorenz-Mie theory for rigorously deriving optical properties of scattering media, and can be readily used

in any radiative-based light transport simulator.

We have demonstrated the significant effects of departing from the underlying assumptions of Lorenz-Mie theory, and the versatility for modeling a wide range of participating media by modifying the arrangement of particles within each cluster, including isotropic, anisotropic, and correlated media.

Chapter 5

Latent Representation for Materials

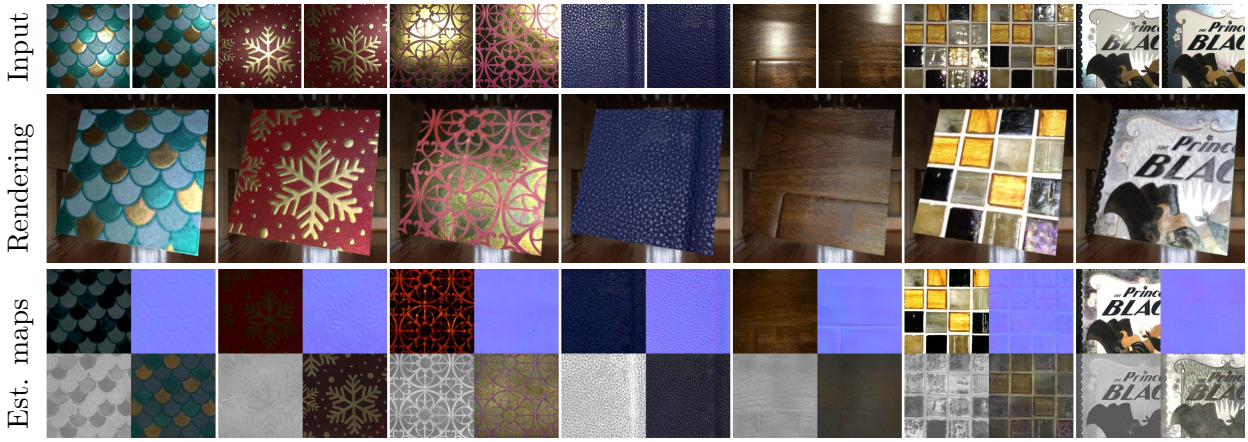


Figure 5.1: We introduce a method to capture SVBRDF material maps from a small number of mobile flash photographs, achieving high quality results both on original and novel views. Our key innovation is optimization in the latent space of MaterialGAN, a generative model trained to produce plausible material maps; MaterialGAN thus serves as a powerful implicit prior for result realism. Here we show re-rendered views for several different materials under environment illumination. We use 7 inputs for these results (with 2 of them shown). (Please use Adobe Acrobat and click the renderings to see them animated.)

5.1 Introduction

Despite a few decades of effort in computer graphics and vision, capturing spatially-varying reflectance of real-world materials remains a challenging and actively researched task. Measurement methods have traditionally used custom hardware systems to densely sample illumination and viewing directions [91, 92], followed by post-processing such as fitting parametric BRDF models [102]. However, such approaches are restricted to laboratory conditions.

Recent work has explored methods for casual capture of spatially-varying BRDFs (SVBRDFs) using commodity hardware and in less constrained environments [31, 109, 5, 6, 62]. These methods usually follow an *inverse-rendering* approach: they define a forward rendering model and optimize reflectance parameters so that the simulated appearance matches physical measurements under certain image metrics. With a small number of measured images, this approach is fundamentally under-constrained: there usually exist many material estimates capable of producing renderings that match the measurements, but many of these estimates can be unrealistic and may not generalize to novel illumination and viewing conditions. The solution to this problem has been to *regularize* the optimization using pre-determined *material priors* such as linear low-dimensional BRDF models [109, 62] or stationary stochastic textures [6, 4]. However, such hand-crafted priors do not generalize to a wide range of real-world materials.

More recently, learning-based approaches have demonstrated remarkable results for reconstructing SVBRDFs from one [22, 86] or more images [23]. While these methods use rendering-based losses (similar to the inverse rendering approaches) during training, at test time they predict SVBRDFs from images using a single feed-forward pass through a deep network. As a result, the reconstructed material parameters may not accurately reproduce the measured appearance. In contrast, Gao et al. [35] propose using an optimization-based approach in conjunction with a learned material prior. Specifically, they train a fully-

convolutional auto-encoder on a large material dataset and optimize in the latent space of this auto-encoder. This ensures that the reconstructed SVBRDF parameters both reproduce the measurements and are plausible real-world materials. However, while this learned material prior is a significant improvement over hand-crafted priors, it still produces a relatively localized and highly flexible latent space that requires a good initialization (for example, from single image methods [22, 86]) and even then can fail to produce good results.

In this chapter, we propose a different material prior that builds on the remarkable progress in image synthesis using deep Generative Adversarial Networks (GANs) [43, 72, 73]. We train *MaterialGAN*—a StyleGAN2-based deep convolutional neural network [74]—to generate plausible materials from a large-scale, spatially-varying material dataset [22]. MaterialGAN learns *global* correlations in material parameters, both spatially (thus encoding texture patterns) as well as across parameters (for example, relationships between diffuse and specular parameters). As illustrated in Figure 5.2, sampling from the MaterialGAN latent space produces plausible, realistic materials with complex variations and diverse appearance.

While GANs have traditionally been used to synthesize images, we demonstrate a very different application, using MaterialGAN as a powerful prior in an inverse rendering-based material capture framework. We append a rendering layer to MaterialGAN, setting up a differentiable pipeline from the learned latent space, through generating material maps, to rendering images under specified views and lighting. This allows us to optimize the MaterialGAN latent vector(s) to minimize the error between the rendered and measured images and reconstruct the corresponding material maps. Doing so ensures that the reconstructed SVBRDFs lie on the "manifold of realistic materials", while at the same time accurately reproducing the captured images.

We demonstrate that our GAN-based optimization framework produces high-quality SVBRDF reconstructions from a small number (3-7) images captured under flash illumination using hand-held mobile phones, and improves upon previous state-of-the-art methods [35, 23].

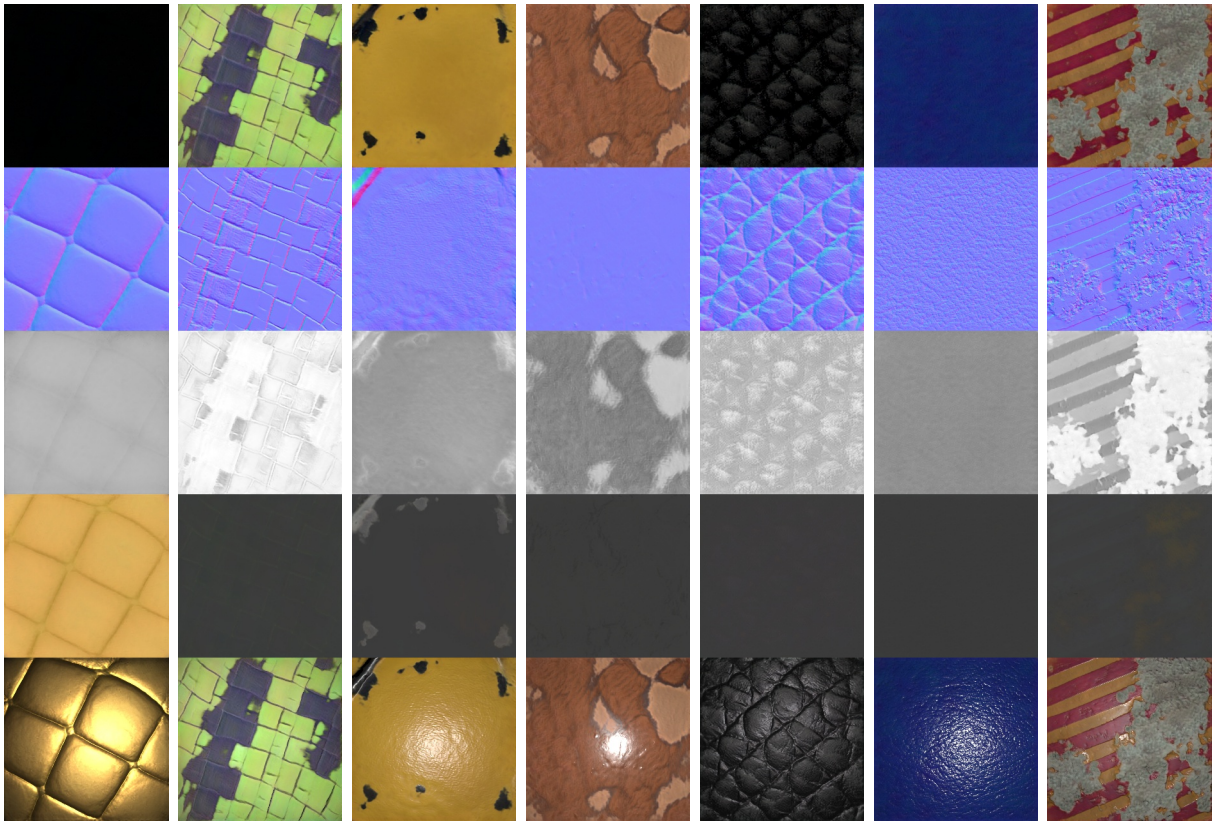


Figure 5.2: Seven materials generated by randomly sampling MaterialGAN. Top to bottom: diffuse albedo, normal, roughness, specular albedo and renderings under flash illumination. As can be seen, the material maps are high-quality with meaningful correlations both spatially and across materials parameters, and visually look like plausible real-world materials.

In particular, it produces cleaner, more realistic material maps that better reproduce the appearance of the captured material under both input and novel lighting. Moreover, as illustrated in Figure 5.10, MaterialGAN adapts to a wide range of SVBRDF samples ranging from diffuse to specular materials and near-stochastic textures to structured patterns with multiple distinct, complex materials.

Furthermore, our GAN-based latent space offers the ability to edit the latent vector in semantically meaningful ways (via operations like interpolation in the latent space) and generate realistic materials that go beyond the captured images. This is not possible with current material capture methods that do not afford any control over their per-pixel BRDF

estimates.

5.2 Related work

Reflectance capture. Acquiring material data from physical measurements is the goal of a broad range of methods. Please refer to surveys [133, 44, 25] for more comprehensive introduction to the related works.

Most reflectance capture approaches observe a material sample under varying viewing and lighting configurations. They differ in the number of light patterns required and their types such as moving linear light [36, 109], Gray code patterns [44], spherical harmonic illumination [39], and Fourier patterns [5].

Methods have also been proposed for material capture "in the wild", i.e., under uncontrolled environment conditions with commodity hardware, typically captured with a hand-held mobile phone with flash illumination. Some of these methods impose strong priors on the materials, such as linear combinations of basis BRDFs [62, 137] (where the basis BRDFs can come from the measured data [92]). Later work by Aittala et al. [6, 4] estimated per-pixel parameters of stationary spatially-varying SVBRDFs from two-shot and one-shot photographs. In the latter case, the approach used a neural Gram-matrix texture descriptor based on the texture synthesis and feature transfer work of Gatys [37, 38] to compare renderings with similar texture patterns but without pixel alignment.

More recently, deep learning-based approaches have demonstrated remarkable progress in the quality of SVBRDF estimates from single images (usually captured under flash illumination) [84, 22, 86]. These methods train deep convolutional neural networks with large datasets of artistically created SVBRDFs, and with a combination of losses that evaluate the difference in material maps and renderings from the dataset ground truth.

Deschaintre [23] extended the single-shot approach to multiple images. The key idea is to extract features from the input images with a shared encoder, max-pooling the features and decoding the final maps from the pooled features. This architecture has the benefit of being independent of the number of inputs, while also not requiring explicit light position information. In our experience, this approach produces smooth, plausible maps with low artifacts; however, re-rendering the maps tends to be not as close to the target measurements because the network cannot "check" its results at runtime. Moreover, we find that especially on real data, this method also has strong biases such as dark diffuse albedo maps and exaggerating surface normals (especially along strong image gradients that might be caused by albedo variations). We believe this is not due to any technical flaw; the method may be reaching the limit of what is possible using current feed-forward convolutional architectures and currently available datasets.

Gao et al. [35] introduced an inverse rendering-based material capture approach that optimizes for material maps to minimize error with respect to the captured images. Since this is an under-constrained problem, they propose optimizing over the latent space of a learned material auto-encoder network to minimize rendering error. This approach has the benefit of explicitly matching the appearance of the captured image measurements, while also using the auto-encoder as a material "prior". Moreover, the encoder and decoder are fully convolutional, which has the advantage of resolution independence. However, we find that the convolutional nature of this model also has the disadvantage of only providing local regularization and not capturing global patterns in the material, such as the long-range spatial patterns and correlations between the different material parameter maps. As a result, this method relies on previous methods (for example, Deschaintre et al. [22]) to provide a good initialization, without which it can converge to poor results. In contrast, our MaterialGAN is a more globally robust latent space and produces higher quality reconstructions without requiring accurate initializations, though it is no longer resolution-independent.

Generative adversarial networks. GANs [43] have become extremely successful in the past few years in various domains, including images [108], video [122], audio [24], and 3D shapes [85]. A GAN typically consists of two competing networks; a generator, whose goal is to produce results that are indistinguishable from the real data distribution, and a discriminator, whose job is to learn to identify generated results from real ones. For generating realistic images (especially of human faces), there has been a sequence of improved models and training strategies, including ProgressiveGAN [72], StyleGAN [73] and StyleGAN2 [74]. StyleGAN2 in particular is the state-of-the-art GAN model and our work is based on its architecture, modified to output more channels.

Recently, GANs have also been used to solve inverse problems [17, 9, 104]. In computer graphics and vision, this work has focused on embedding images into the latent space, with the goal of editing the images in semantically meaningful ways via latent vector manipulations [144]. This embedding requires solving an optimization problem to find the latent vector. More recent work such as Image2StyleGAN [1] and Image2StyleGAN++ [2] has looked at problem of embedding images specifically into the the StyleGAN latent space. While these methods focus on projecting portrait images into face-specific StyleGAN models, we find their analysis can be adapted to our problem. We build on this to propose a GAN embedding-based inverse rendering approach.

5.3 MaterialGAN: A Generative SVBRDF Model

Generative Adversarial Networks [43] are trained to map an input from a latent space (often randomly sampled from a multi-variate normal distribution) to a plausible instance of the target distribution. In recent years, GANs have made remarkable progress in synthesizing high-resolution, photo-realistic images. Inspired by this progress, we propose MaterialGAN, a GAN that is trained to generate plausible materials, thus implicitly learning an SVBRDF

manifold. MaterialGAN is based on the architecture of StyleGAN2 [74].

5.3.1 Overview of StyleGAN and its latent spaces

StyleGAN2 [74] is an improvement of StyleGAN [73] and is the state-of-the-art generative adversarial network (GAN) for image synthesis, especially for human faces. The architecture has several advantages over previous models like ProgressiveGAN [72] and DCGAN [108]. For our purposes, the main advantage is that the model is not simply a black-box stack of convolution and upsampling layers, but has additional, more specific structure, allowing for much easier inversion (latent space optimization). The StyleGAN2 architecture starts with a learned constant $4 \times 4 \times 512$ tensor and progressively upsamples it to the final output target resolution via a sequence of convolutional and upsampling layers (7 in total to end with a final image resolution of 256×256). Given an input latent code vector $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^{512}$, StyleGAN2 transforms it through a non-linear mapping network of fully-connected layers into an intermediate latent vector $\boldsymbol{w} \in \mathcal{W} \subset \mathbb{R}^{512}$. The rationale for the introduction of the space of $\mathcal{W}$ is that while $\mathcal{Z}$ requires (almost) every latent $\boldsymbol{z} \in \mathcal{Z}$ to correspond to a realistic output, vectors $\boldsymbol{w} \in \mathcal{W}$ are free from this overly stringent constraint, which leads to a less "entangled" mapping, with more meaningful dimensions (see [73, 74] for more discussion). In the original StyleGAN, the vector $\boldsymbol{w} \in \mathcal{W}$ is mapped via a learned affine transformation to mean and variance "style" vectors that control adaptive instance normalizations (AdaIN) [61] that are applied before and after every convolution in the generation process (thus $7 \times 2 = 14$ times for a model of resolution 256×256). The statistics of the AdaIN normalizations caused the feature maps and output images of StyleGAN to suffer from droplet artifacts. StyleGAN2 removes the droplet artifacts entirely by replacing the AdaIn normalization layers with a demodulation operation which bakes the entire style block into a single layer while maintaining the same scale-specific control as StyleGAN. We construct a matrix $\boldsymbol{w}^+ \in \mathcal{W}^+ \subset \mathbb{R}^{512 \times 14}$ by replicating $\boldsymbol{w}$ 14 times.

During training and standard synthesis, the columns of $\boldsymbol{w}^+$ are identical, and correspond to $\boldsymbol{w}$. However, as we will discuss later (and similar to Abdal et al. [1]), we relax this constraint when optimizing for an embedding; $\mathcal{W}^+$ thus becomes an extended latent space, more powerful than $\mathcal{W}$ or $\mathcal{Z}$. Additionally, StyleGAN2 injects Gaussian noise, $\boldsymbol{\xi}$, into each of the 14 layers of the generator. This noise gives StyleGAN2 the ability to synthesize stochastic details at multiple resolutions. Abdal et al. [2] make the observation that one can also treat these noise inputs $\boldsymbol{\xi}$ as a latent space $\mathcal{N}$. Thus, combining these two spaces defines yet another latent space $\mathcal{W}^+\mathcal{N}$.

5.3.2 MaterialGAN training

MaterialGAN was trained with the dataset provided by Deschaintre et al. [22] (and also used in Gao et al. [35]). They generated this dataset by sampling the parameters of procedural material graphs from Allegorithmic Substance Share to create an initial set of 155 high-quality SVBRDFs at resolution 4096×4096. The dataset was augmented by blending multiple SVBRDFs and generating 256×256 resolution crops at random positions, scales and rotations. The final dataset consists of around 200,000 SVBRDFs. For detailed information about the curation of dataset we refer the reader to [22]. Since pairs of SVBRDFs in the dataset were the same with only a slight variation, we selected 100,000 SVBRDFs. The maps for each SVBRDF are stacked in 9 channels (3 for albedo, 2 for normals, 1 for roughness, and 3 for specular albedo). We account for this by adapting the MaterialGAN architecture to output 9-channel outputs. MaterialGAN is trained in TensorFlow (version 1.15) with the same loss functions and similar hyper-parameters from StyleGAN2 [74]. StyleGAN2 configuration F was used for all experiments. The generator and discriminator were trained using Adam optimizers. The learning rate was increased per resolution from 0.001 to 0.0025 for both the generator and the discriminator. The discriminator was shown 25 million images. Training on 8× Nvidia Tesla V100 takes about 5 days. Figure 5.2 shows materials generated

by randomly sampling the MaterialGAN latent space and images rendered from them. As can be seen here, MaterialGAN generates a wide variety of nearly photorealistic materials ranging from structured to stochastic, diffuse to specular, and with large-scale variations to fine detail. Furthermore, Figure 5.3 and the accompanying video show example interpolations between pairs of generated materials in the latent space, producing plausible non-linear morphing results.

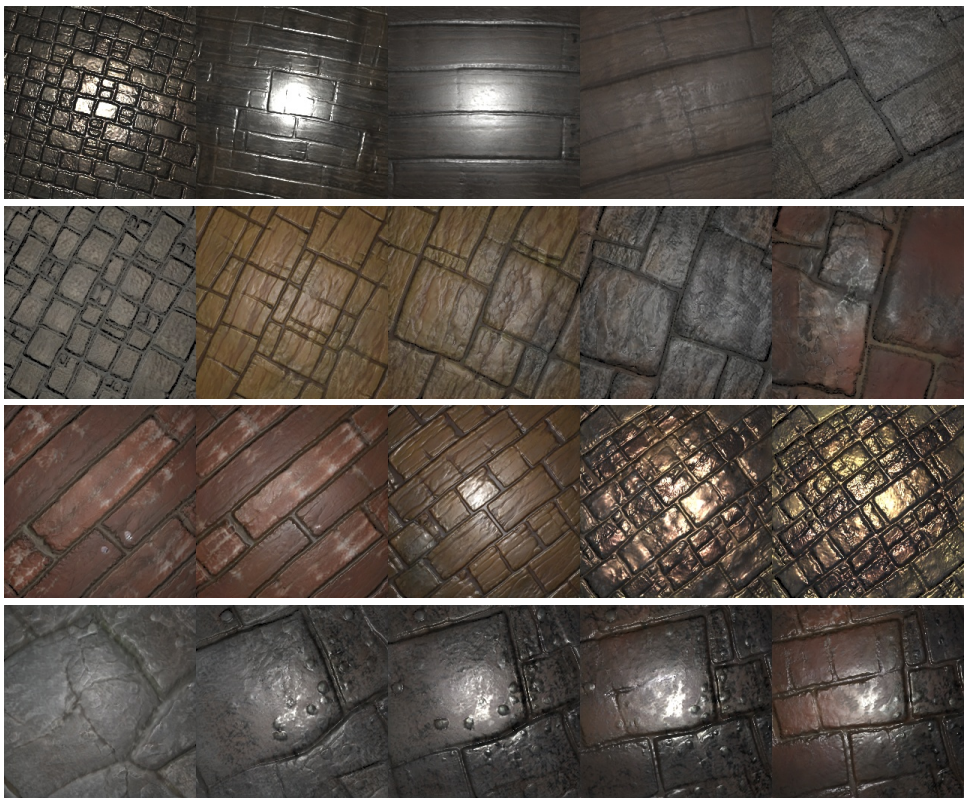


Figure 5.3: Interpolation in MaterialGAN latent space. Each row shows an example of interpolation between two randomly generated materials, demonstrating non-linear morphing behavior.

5.4 SVBRDF Capture using MaterialGAN

We utilize MaterialGAN, the powerful generative model described in the previous section, in a fundamentally new fashion: to *capture* SVBRDF maps. Specifically, we use MaterialGAN

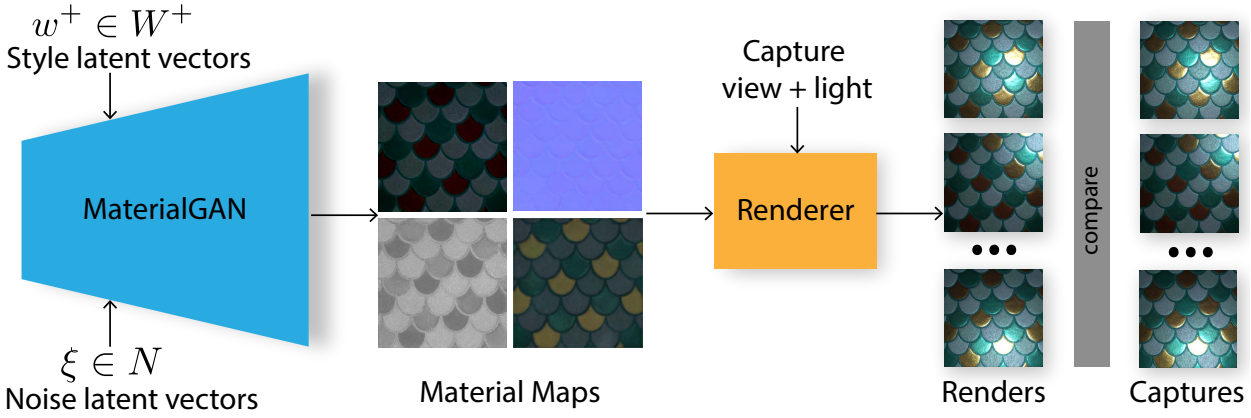


Figure 5.4: Our inverse rendering pipeline. We optimize for latent vectors $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$, that feed into the layers of the StyleGAN2-based MaterialGAN model. The MaterialGAN generator produces material maps (diffuse albedo, normal, roughness and specular albedo), that are rendered under the captured view/light settings. Finally, the renderings and measurements are compared using a combination of L2 and perceptual losses.

as a *material prior* for SVBRDF acquisition via an inverse rendering framework (See Figure. 5.4). Our goal is to estimate the SVBRDF parameter maps from one or a small number of photographs of a near-planar material sample. We utilize a common BRDF model that involves a diffuse and a specular component using the microfacet BRDF with the GGX normal distributions [127]. Our unknown parameter vectors $\boldsymbol{\theta} := (\boldsymbol{a}, \boldsymbol{n}, r, \boldsymbol{s})$ encode the four per-pixel parameter maps: diffuse albedo $\boldsymbol{a}$, surface normal $\boldsymbol{n}$, roughness r, and specular albedo $\boldsymbol{s}$. To recover the unknown parameter maps, we capture k images $\boldsymbol{I}_1, \cdots, \boldsymbol{I}_k$. We assume known viewing and lighting configurations for each image, which we denote as (L_i, C_i). Further, we assume that the material is lit by a single point source, collocated with the camera. [1] The images can be reprojected into a common frontal view (which is straightforward with a known viewing configuration).

We introduce a differentiable rendering operator $\mathcal{R}$ that takes as input the parameter maps as well as the viewing and lighting configurations, and synthesizes corresponding images of the material. Under this setup, our goal is to find values of the unknown parameters $\boldsymbol{\theta}$ so

[1] In theory, non-collocated lights, area lights or projection patterns (e.g. on an LCD or similar screen) can be used as well, and would require a straightforward modification to our forward rendering process.

that renderings with these parameters match the measurements $\boldsymbol{I}_i$. In other words, we focus on solving the following optimization problem:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{a},\boldsymbol{n},r} \sum_{i=1}^{k} \mathcal{L}(\mathcal{R}(\boldsymbol{\theta};\, L_i, C_i), \boldsymbol{I}_i), \tag{5.1}$$

where $\mathcal{L}$ is a loss function that measures the difference between the captured images, $\boldsymbol{I}_i$ and the renderings generated from the estimated SVBRDF parameters, $\mathcal{R}(\boldsymbol{\theta};\, L_i, C_i)$.

5.4.1 Incorporating the MaterialGAN prior

Eq. (5.1) is, in general, a challenging optimization to solve due to its under-constrained nature. Given a small number of input measurements, the optimization can overfit to the input, producing implausible maps that do not generalize to novel views and lighting. To overcome this challenge, we leverage the MaterialGAN prior: instead of directly optimizing for the parameter maps $\boldsymbol{\theta}$, we can optimize for a vector $\boldsymbol{u}$ in the MaterialGAN *latent space* and map (decode) this latent vector back into material maps $\boldsymbol{\theta}$. The optimization problem then becomes:

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \sum_{i=1}^{k} \mathcal{L}(\mathcal{R}(\mathcal{G}(\boldsymbol{u});\, L_i, C_i), \boldsymbol{I}_i), \tag{5.2}$$

where $\mathcal{G}$ is the learned MaterialGAN generator. Given that both $\mathcal{G}$ and $\mathcal{R}$ are differentiable operations, Eq. (5.2) can be optimized via gradient-based methods to estimate $\boldsymbol{u}^*$ and the corresponding SVBRDF maps $\mathcal{G}(\boldsymbol{u}^*)$.

The above operation is similar to recent work on embedding images in the StyleGAN latent space [1, 2]. The key difference is that we do not match material parameters directly, but evaluate their error through the rendering operator $\mathcal{R}(\cdot)$. To our knowledge, ours is the first approach to use a GAN latent space in combination with a rendering operator.

Loss function. We optimize Eq. 5.2 using a combination of a standard per-pixel L2 loss and a "perceptual loss" [70] that has been shown to produce sharper results in image synthesis tasks:

$$\mathcal{L}(\boldsymbol{I}, \boldsymbol{I}') = \lambda_1 \mathcal{L}_{\text{pixel}} + \lambda_2 \mathcal{L}_{\text{percept}}, \tag{5.3}$$

The perceptual loss is defined as:

$$\mathcal{L}_{\text{percept}}(\boldsymbol{I}, \boldsymbol{I}') = \textstyle\sum_{j=1}^{4} w_j^{\text{percept}} \left\| F_j(\boldsymbol{I}) - F_j(\boldsymbol{I}') \right\|_2^2, \tag{5.4}$$

where $F_1, \cdots, F_4$ are the flattened feature maps corresponding to the outputs of VGG-19 layers `conv1_1`, `conv1_2`, `conv3_2`, and `conv4_2` from a pre-trained VGG network [115]. See section 5.4.3 for more details.

Optimization details. We convert the TensorFlow-trained MaterialGAN model to PyTorch, in which our optimization framework is implemented. We optimize Eq. 5.2 using the Adam optimizer in PyTorch, with a learning rate of 0.01. We set all other hyper-parameters to default values. Now that our basic optimization framework is set up, there remain two key ingredients to implement our GAN-based optimization framework (Eq. (5.2)): (i) the choice of *latent space* that we optimize $\boldsymbol{u}$ over, and (ii) our optimization strategy to minimize the objective function. In the following sections, we describe our approach, along with an empirical analysis of these design choices.

5.4.2 Latent space

As discussed in Sec. 5.3.1, StyleGAN2 (and consequently, MaterialGAN) has a number of potential latent spaces. In particular, MaterialGAN uses three different *style* latent spaces: the input latent code $\boldsymbol{z} \in \mathcal{Z}$, the intermediate latent code $\boldsymbol{w} \in \mathcal{W}$ and per-layer styles $\boldsymbol{w}^+ \in \mathcal{W}^+$. StyleGAN2 also injects noise $\boldsymbol{\xi} \in \mathcal{N}$ into every layer of the network to generate

stochastic variations. The typical forward generation process of the GAN only uses $\boldsymbol{z}$, with $\boldsymbol{w}$ being generated from $\boldsymbol{z}$ via a mapping network, and $\boldsymbol{w}^+$ being generated from $\boldsymbol{w}$ via affine transformations. However, Abdal et al. [1] note that the space of $\mathcal{Z}$ is too restrictive for accurate embedding of faces or other content into the GAN space. In other words, given the image of a human face, it is generally impossible to find a single $\boldsymbol{z} \in \mathcal{Z}$ such that the generated image closely matches the target. This remains the case even when extending the space to $\mathcal{W}$, i.e., when searching for a $\boldsymbol{w}$ instead of a $\boldsymbol{z}$. The space $\mathcal{W}^+$, on the other hand, offers much stronger representative power. Our experiments on embedding material maps into MaterialGAN demonstrate that optimizing for $\mathcal{W}^+$ is also needed for MaterialGAN to accurately reproduce input maps. We demonstrate this in Figure 5.5, via an experiment where we embed a given material (with known material maps) into MaterialGAN. As shown in rows (2) and (3), maps generated by optimizing $\boldsymbol{w}^+ \in \mathcal{W}^+$ contain more detail compared to those using $\boldsymbol{w} \in \mathcal{W}$.

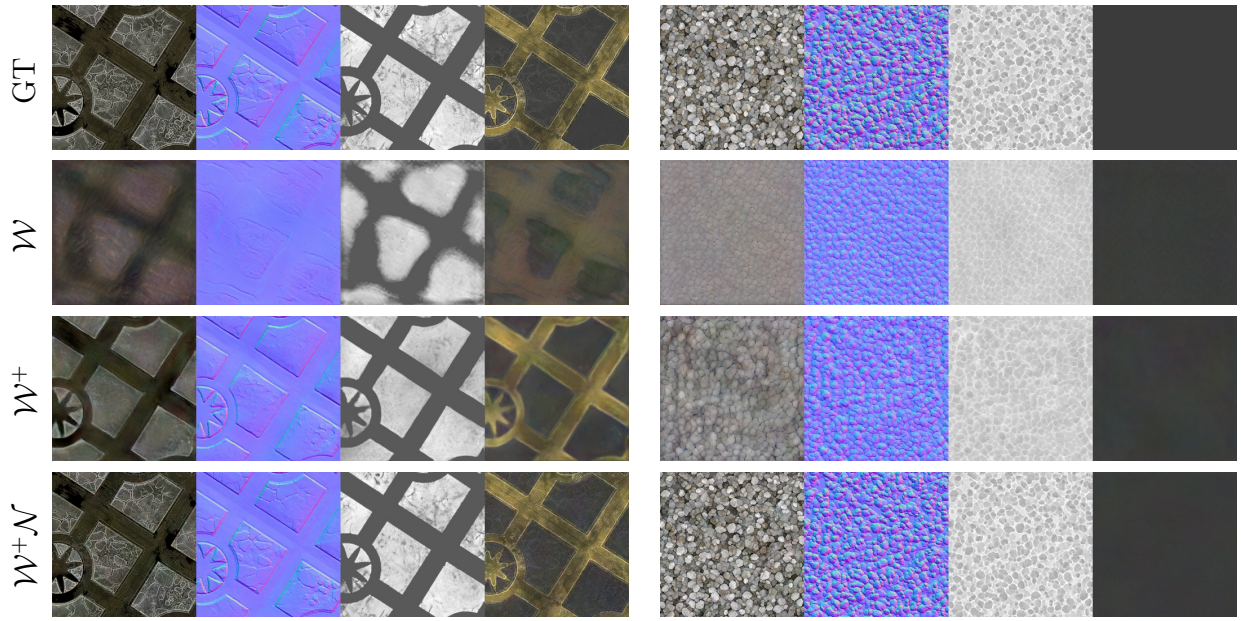


Figure 5.5: Embedding SVBRDFs into different latent spaces. We take two synthetic SVBRDF material maps (top) and embed them into different latent spaces with and without the noise space (second–fourth rows). For illustration, we also embed the maps into a pure-noise space *only*; this is unable to recover the color at all.

On the other hand, some small-scale details are still missing. In fact, according to our exper-

iments, only colors and large-scale features can be captured by the $\mathcal{W}^+$ space. For depicting high-frequency patterns, as demonstrated in the last row of Figure 5.5, we need to go even further and optimize the noise vector $\boldsymbol{\xi}$ (instead of drawing it from multi-variate normal distributions). We note that optimizing for the noise component is even more important in MaterialGAN, compared to embedding faces in StyleGAN or StyleGAN2. We suspect that this is because with human faces, the distinction between large-scale features (e.g., eyes, noise, and mouth) and small-scale features (e.g., winkles) is very prominent, allowing the $\mathcal{W}^+$ space to focus mostly on the large-scale features while leaving the small-scale ones to the noise vector $\boldsymbol{\xi} \in \mathcal{N}$. In our case, the boundary between large-scale and small-scale material features is much less distinct. The physical scales of real-world materials varies in a continuous fashion, making it virtually impossible to assign them to only one of the $\mathcal{W}^+$ and $\mathcal{N}$ spaces. We hypothesize that for this reason, we need to focus on both $\mathcal{W}^+$ and $\mathcal{N}$ to achieve high-quality reconstruction of SVBRDF maps. Based on these empirical observations, estimating SVBRDF parameter maps from photographs using our pre-trained MaterialGAN boils down to solving the following optimization:

$$\boldsymbol{u}^* = \underset{\boldsymbol{w}^+ \in \mathcal{W}^+, \boldsymbol{\xi} \in \mathcal{N}}{\arg\min} \sum_{i=1}^{k} \mathcal{L}(\mathcal{R}(\mathcal{G}(\boldsymbol{w}^+, \boldsymbol{\xi});\, L_i, C_i), \boldsymbol{I}_i). \tag{5.5}$$

Since there are two variables $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ that behave in a correlated fashion, a proper optimization strategy is crucial to achieve high-quality results. We now discuss our alternating two-step optimization method.

5.4.3 Optimization strategy

Abdal et al. [1, 2] recommended using a two-stage setting by first optimizing $\boldsymbol{w}^+$ (with $\boldsymbol{\xi}$ fixed) and then $\boldsymbol{\xi}$ (with $\boldsymbol{w}^+$ fixed). In our case, this approach does work in some cases but is not always the top-performing option. In addition to this strategy, we propose two

alternatives, leading to three different optimization schemes:

1. **Strategy 1**: Optimize $\boldsymbol{w}^+$ first, then optimize $\boldsymbol{\xi}$;
2. **Strategy 2**: Jointly optimize both $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$;
3. **Strategy 3**: Alternatively optimize $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ for a small number (for example, 10) of iterations each.

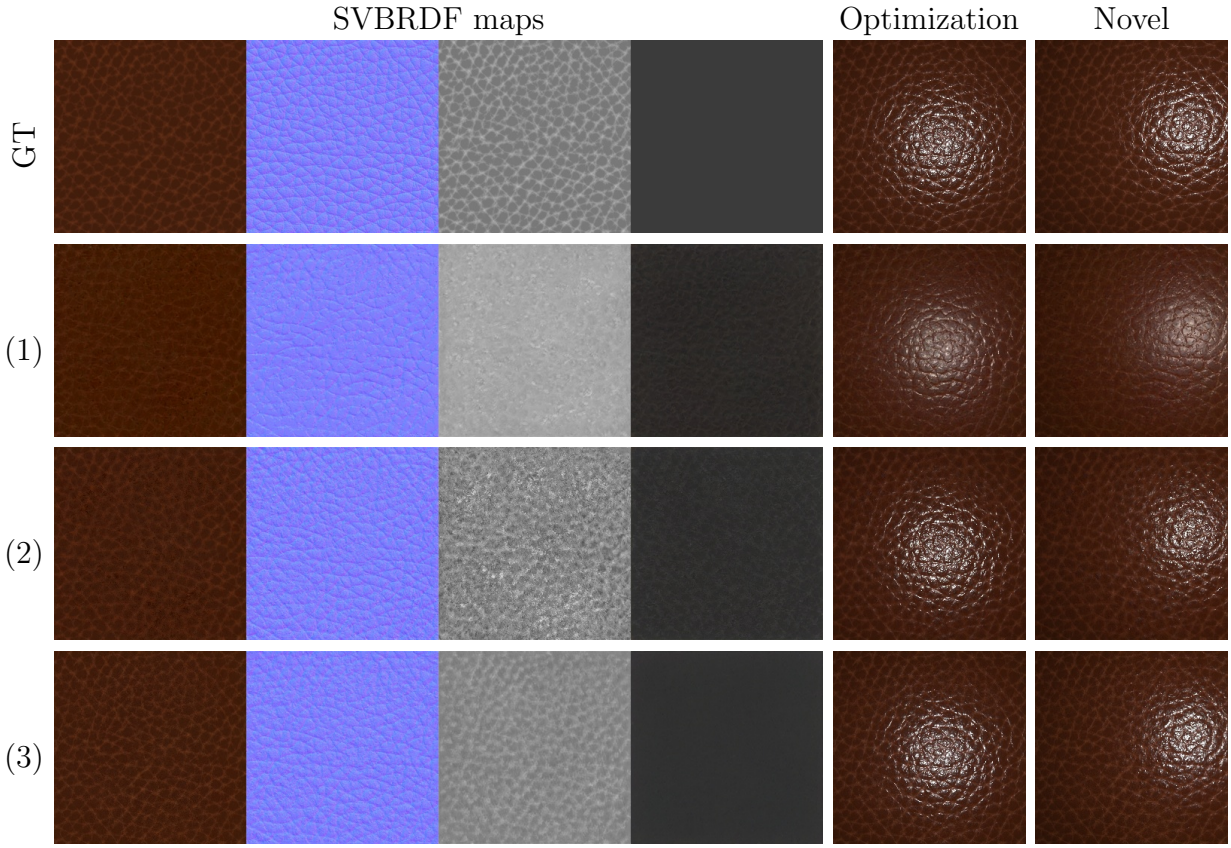


Figure 5.6: Optimization strategy. We evaluated three optimization strategies: (1) optimize $\boldsymbol{w}^+$ first, then $\boldsymbol{\xi}$; (2) jointly optimize $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$; (3) alternate between $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ every 10 iterations. Strategy (1) causes artifacts during the optimization, and (2) brings more noise into the maps. Particularly, for textures with small features, (1) and (2) may drive the optimization to bad local minima while the per-pixel loss could still be very low. Strategy (3) appears to be a good compromise, giving us better results in most cases. Note: "Optimization" means an optimized input view or its re-rendering, i.e. not a novel view.

Figure 5.6 shows a comparison of these strategies. All of them give reasonable results, but Strategy 1 is better suited for materials with strong large-scale features. Strategy 2 provides

the fastest convergence because it allows the noise vector $\boldsymbol{\xi}$ to be modified from the very beginning. This, however, generally causes the optimization to use $\boldsymbol{\xi}$ for encoding higher-level features and is prone to overfitting. Finally, Strategy 3—a hybrid of Strategies 1 and 2—behaves in a more robust fashion than either of the previous strategies in most cases. We use Strategy 3 for all the results in this chapter. Additionally, our experiments indicate that it is desirable to use different VGG layer weights for the optimization of $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$. The weights we are using are, for $\boldsymbol{w}^+$: [1/512,1/512,1/128,1/64]; for $\boldsymbol{\xi}$: [1/64,1/64,1/256,1/512].

Noise optimization vs. post-refinement. Instead of optimizing latent space $\boldsymbol{w}^+$ with noise $\boldsymbol{\xi}$, another option is to apply post-refinement (that is, pixel-space optimization without any latent space) after optimizing $\boldsymbol{w}^+$ only. However, the space $\boldsymbol{w}^+$ is too small to realistically match per-pixel detail: if optimizing $\boldsymbol{w}^+$ only, the resulting maps have significant artifacts. Adding post-refinement to such a result essentially becomes per-pixel optimization (with little regularization), which tends to to work poorly with a small number of inputs. Optimizing $\boldsymbol{\xi}$ offers more powerful regularization, as the noise is inserted into all layers of the generator, rather than just appended at the end (like post-refinement). We show two failure examples in Figure 5.7, where optimizing $\boldsymbol{w}^+$ leads to unsatisfactory texture maps.

5.4.4 Initialization

We find that our method is robust to the initialization of the latent vectors. We experimented with using the same initial configuration—represented by the material produced by the mean $\boldsymbol{w}$ of our GAN training data (see Figure 5.8(a))—and found that is works well for most of the materials we tried (both synthetic and real). However, this initialization represents a material with a high roughness (reflecting a bias in our training data) and sometimes leads to errors when fitting highly specular / low roughness materials. Therefore, we add an additional low roughness initialization (see Figure 5.8(b)). In practice, given the captured

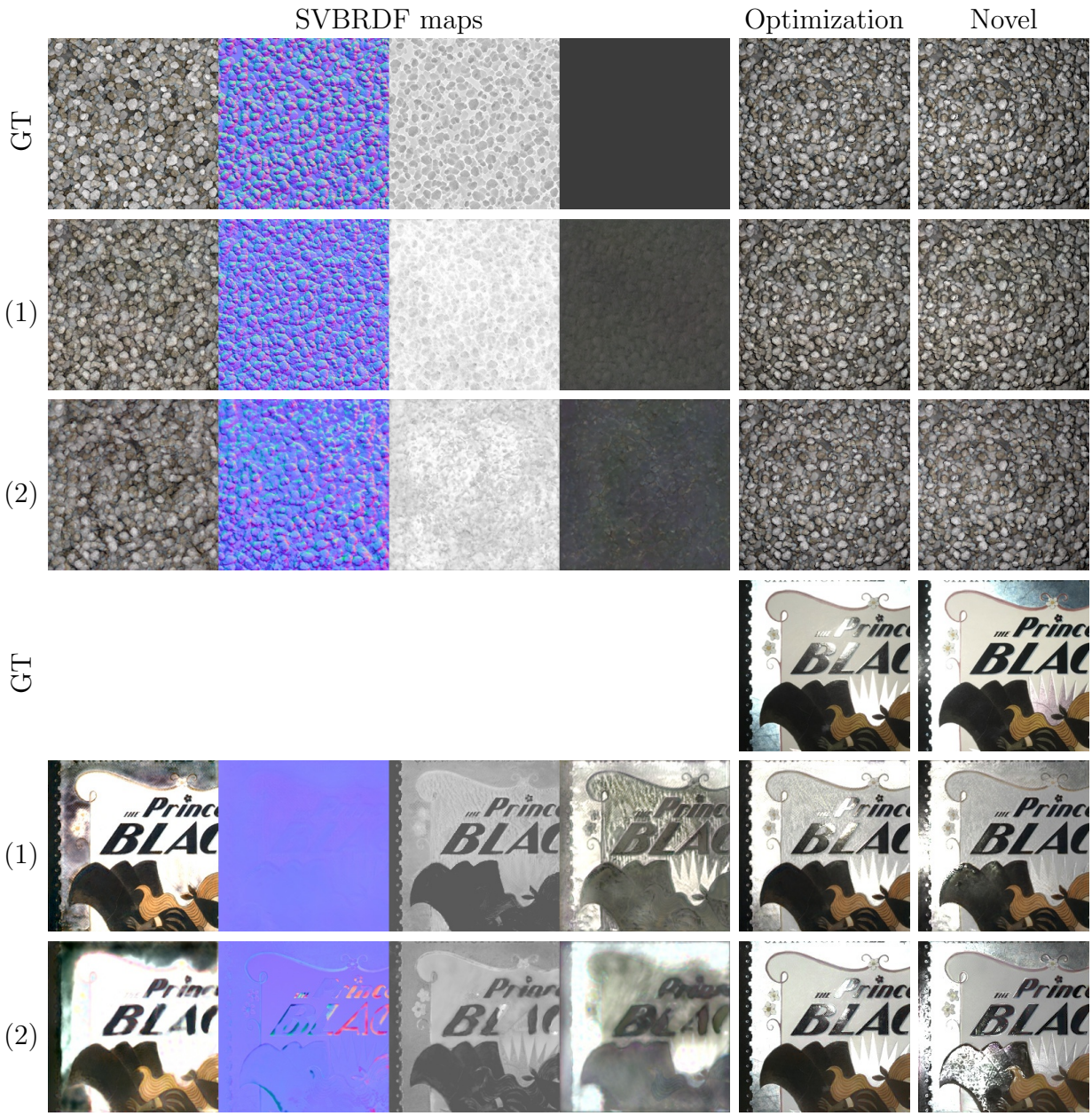


Figure 5.7: Noise optimization vs. post-refinement. (1) Optimize $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ but no post-refinement; (2) Optimize $\boldsymbol{w}^+$ only but with post-refinement. This shows that $\boldsymbol{\xi}$ takes an important role; optimizing only $\boldsymbol{w}^+$ has too little expressive power and converges to suboptimal solutions, which post-refinement cannot fix (see especially normal maps in (2)).

images, we run our MaterialGAN optimization starting from both initializations and retain the result with the lowest optimization error of Eq. (5.3). All of our results in this chapter followed this scheme.

(a) Mean $\boldsymbol{w}$ initialization (b) Low roughness initialization

Figure 5.8: Visualization of our constant initializations. We initialize our optimization with the two materials shown here and pick the result with the lowest final loss. (This applies in cases where we do not use the result from Deschaintre et al. as initialization, as detailed in the results section and supplementary materials.) Left: Material maps generated from the mean latent vector $\boldsymbol{w}$. Right: An additional low roughness, specular initialization.

5.5 Results

(Only a small subset of our results fits into the thesis. Please see our supplemental material and video for more results. [Click here])

Test data. For synthetic tests, we use several examples from the test set of Deschaintre [22], as well as some from the Adobe Stock dataset [86]. This gives a total of 30 synthetic results. For our real results, we use a hand-held mobile phone to capture images with flash, resulting in a collocated camera and point light illumination. Similar to previous work [62, 23], we use a paper frame to register the multiple images. We add markers to the frame to improve camera pose estimation. Using this process, we capture 40 physical samples with nine images per material, roughly covering the sample with 3×3 specular highlights. Unless otherwise specified, all our results use seven images for inverse-rendering optimizations and the remaining two (under novel lighting) for evaluating the results.

Inverse-rendering performance. Our optimization takes about 2 minutes to complete 2000 iterations on a Titan RTX GPU. In many cases, the results converge after 500 iterations,

but we use 2000 everywhere for simplicity.

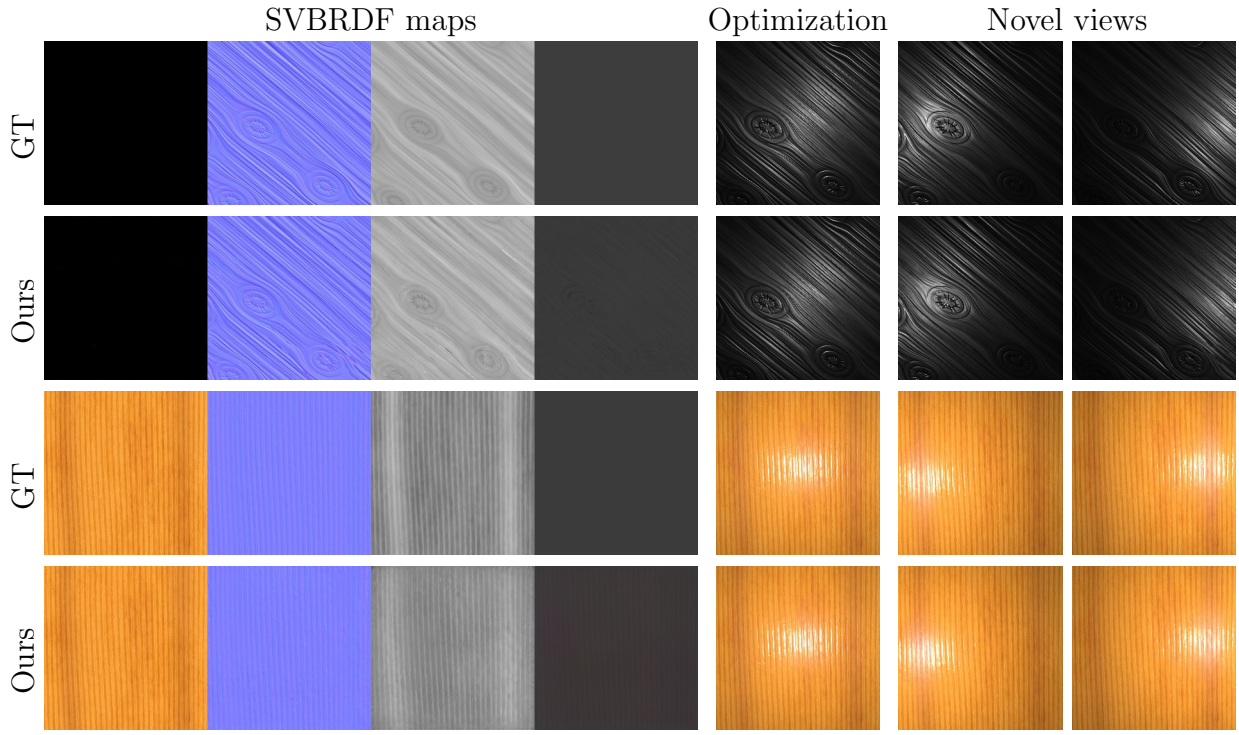


Figure 5.9: SVBRDF reconstruction on synthetic data. We demonstrate results on synthetic SVBRDFs, one from [23] (top) and one from the Adobe Stock Material dataset (bottom). We are able to accurately reconstruct these materials from 7 input images (one input shown). Many more synthetic results are available in supplementary materials.

Testing on synthetic data. Figure 5.9 contains two synthetic results using our method, showing a close match both in maps and in novel view renderings. For more results, please refer to supplemental materials. We note that all methods perform better on synthetic data than on real data, possibly because of the exact BRDF model match and perfect calibration, and also because the synthetic test set, while distinct from the training set, is relatively similar in style.

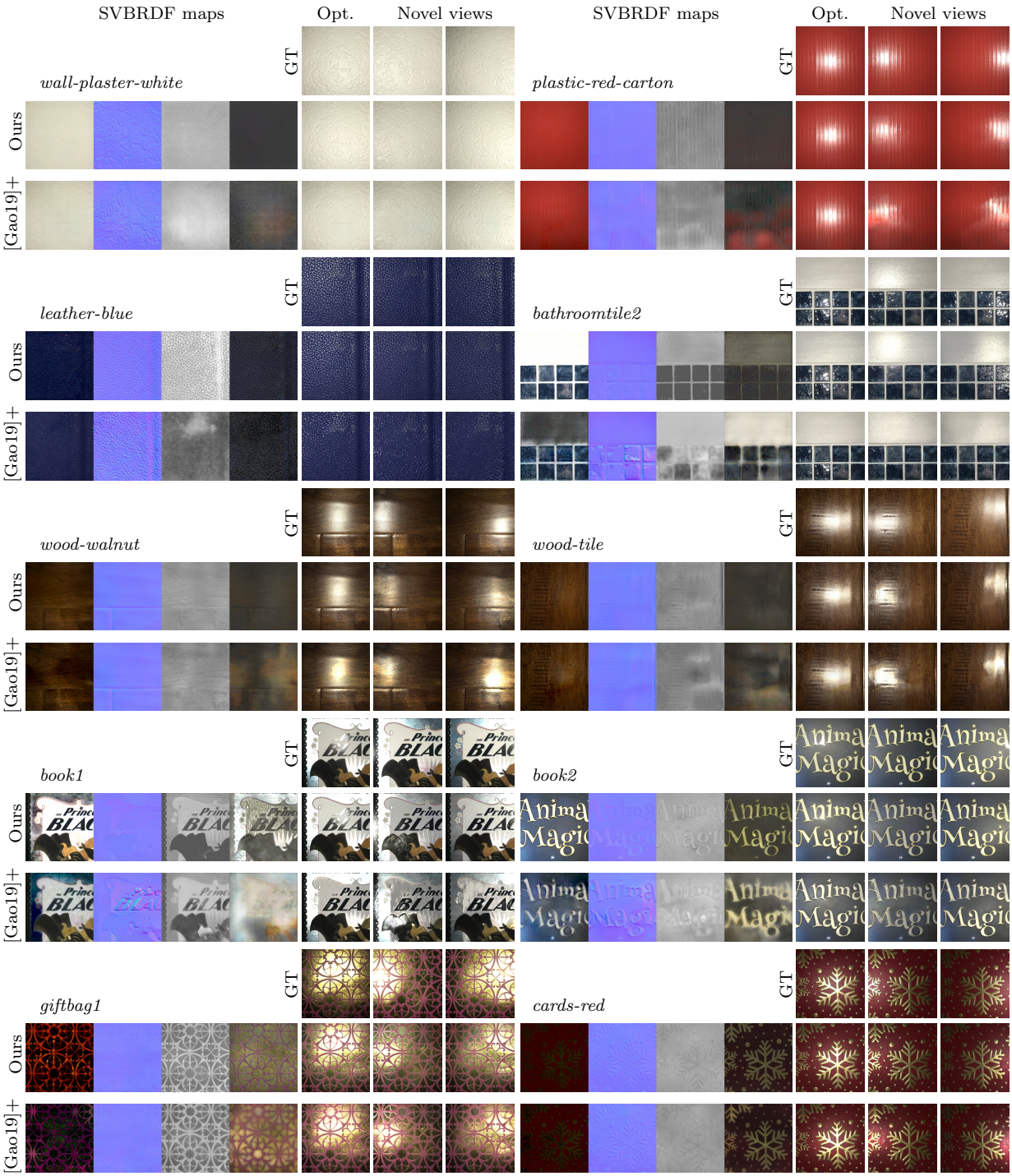


Figure 5.10: SVBRDF reconstruction on real data. We reconstruct SVBRDF maps from 7 inputs, and compare the resulting maps and images rendered under 2 novel views. Gao's method [35] initialized with Deschaintre's [23] direct predictions (denoted as "[Gao19]+") tends to have complex reflectance burnt into the specular albedo map, leading to inaccurate predictions under novel views. Our method with simple initializations, in contrast, is less prone to such burn-ins and generally produces more accurate renderings under novel views. Please refer to Table 5.1 for more information on the quality of these renderings.

Table 5.1: Accuracy of the novel-view renderings shown in Figure 5.10 measured using the Learned Perceptual Image Patch Similarity (LPIPS) metric where our method produces better predictions than Gao's [35] in most cases.

Material	Ours	[Gao19]+	Material	Ours	[Gao19]+
wall-plaster-white	**0.071**	0.132	*plastic-red-carton*	**0.095**	0.166
leather-blue	**0.146**	0.356	*bathroomtile2*	**0.225**	0.231
wood-walnut	**0.226**	0.252	*wood-tile*	0.202	**0.192**
book1	**0.147**	0.318	*book2*	**0.042**	0.122
giftbag1	**0.183**	0.218	*cards-red*	**0.059**	0.092

5.5.1 Comparison with prior work on real data

Here we compare our method and Gao et al. [35]. For more results and comparisons, including with Deschaintre et al. [23], and including with and without initialization for ours and Gao's method, please refer to supplemental materials. We show 10 real examples from our cell phone capture pipeline in Figure 5.10. Note that Gao's method is significantly dependent on initialization, while the same is not true for our method. Therefore, in this figure, we show Gao's result *with initialization* by Deschaintre et al. [23], while our result is shown *without initialization*. Furthermore, note that we are initializing Gao's method with the 2019 multi-input method by Deschaintre, which is a better initialization than the 2018 single-input method. Thus the baseline we are comparing against is, strictly speaking, even higher than what is published in Gao et al., and combines the two best methods published at this time. Generally, we find that our method produces cleaner maps and is less prone to overfitting (burn-in) than Gao's, while producing more accurate re-renderings under original and novel lighting. Table 5.1 shows a quantitative evaluation of the re-rendering quality on novel lighting. As these novel views would be hard to match pixel-wise using any method, as they have never been observed, we use a perceptual method, specifically the Learned Perceptual Image Patch Similarity (LPIPS) metric [140] (lower is better). Note that our method (without initialization by Deschaintre's method) produces better scores for novel views than Gao's method (with initialization) for most images; even in the case where our LPIPS score is worse, our maps still look more plausible overall. We also report quantitative

evaluations (histograms) for our entire set of results (see Figure 5.11). For synthetic data, we compare the RMSE of all predicted maps (diffuse albedo, normal, roughness, specular albedo), as we do know the ground truth for them. For both synthetic and real data, we compare the LPIPS scores on novel lighting. We use a + sign to indicate initialization by Deschaintre et al. In the top row, we compare both methods without initialization by Deschaintre's method, while in the middle row, both methods are initialized, and in the bottom row, we compare our method without initialization to Gao's with initialization. Generally, we find that if both methods are initialized the same way, our method outperforms Gao's. Even in the last row, our performance is comparable on synthetic data (worse on normal map and better on diffuse/specular maps) and still better on real data overall.

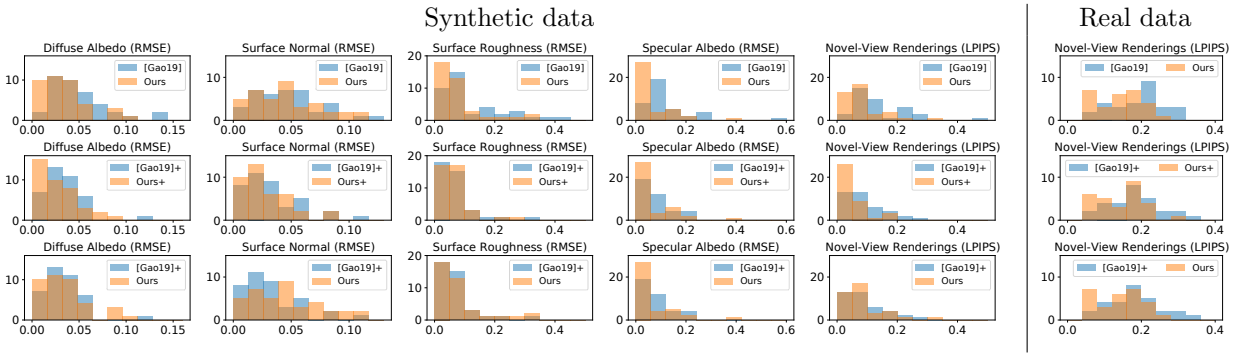


Figure 5.11: **Performance statistics** of Gao [35] and our method. For each technique, we compute (i) the Learned Perceptual Image Patch Similarity (LPIPS) metric between renderings of the output SVBRDF maps and the reference images for 40 *real* and 30 *synthetic* examples; and (ii) the root-measure-square error (RMSE) of the inferred maps for the *synthetic* examples. For both metrics, a lower score indicates a better accuracy. Using identical initializations, our technique ("Ours" and "Ours+") outperforms Gao's ("[Gao19]" and "[Gao19]+") consistently for both real and synthetic examples, as demonstrated in the top and the middle row. Furthermore, our technique with constant initializations ("Ours") has a similar performance with Gao's method initialized using Deschaintre's [23] direct predictions ("[Gao19]+") on the synthetic examples and outperforms the latter on the real examples, as shown on the bottom.

Note about Deschaintre et al. We find that the results from [23] have much less accurate re-rendering than either ours or Gao's method, as they are not doing any optimization to precisely fit the target images. The mismatches we observe are definitely not due to simple scaling or gamma correction issues, as that would be consistent across examples; rather, we find that the method performs much better on synthetic examples that match the visual

style of its training set. On the other hand, their method is fast and results tend to be clean and artifact-free, so they are very suitable for initialization of optimization methods.

5.5.2 Additional comparisons

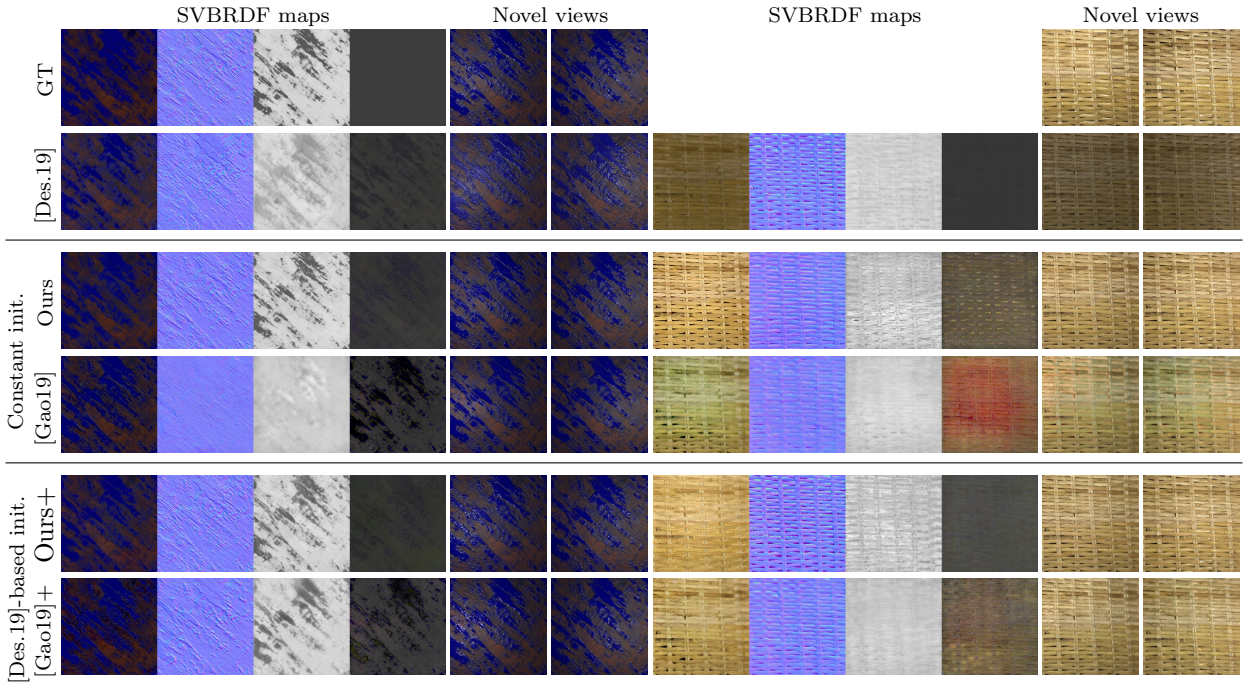


Figure 5.12: SVBRDF results with different initialization Unlike Gao's method, ours is less strongly dependent on a good initialization from Deschaintre's method [23]. In most of cases, starting from simple texture maps (given by our constant initializations) is already good enough to converge to a clean solution. We show all combinations (with and without good initializations) for both methods, for one synthetic and one real example, where techniques initialized with [Deschaintre] are denoted with the suffix "+" (i.e., "Ours+" and "[Gao19]+"). Note the failure of Gao's method without good initializations (i.e., "Gao19").

Optimization with different initializations. In Figure 5.12, we compare our method to Gao's with and without initialization by Deschaintre's method in all 4 combinations, on a synthetic and a real example. This shows that Gao's method more significantly dependent on good initialization that ours (even though our method can still occasionally benefit).

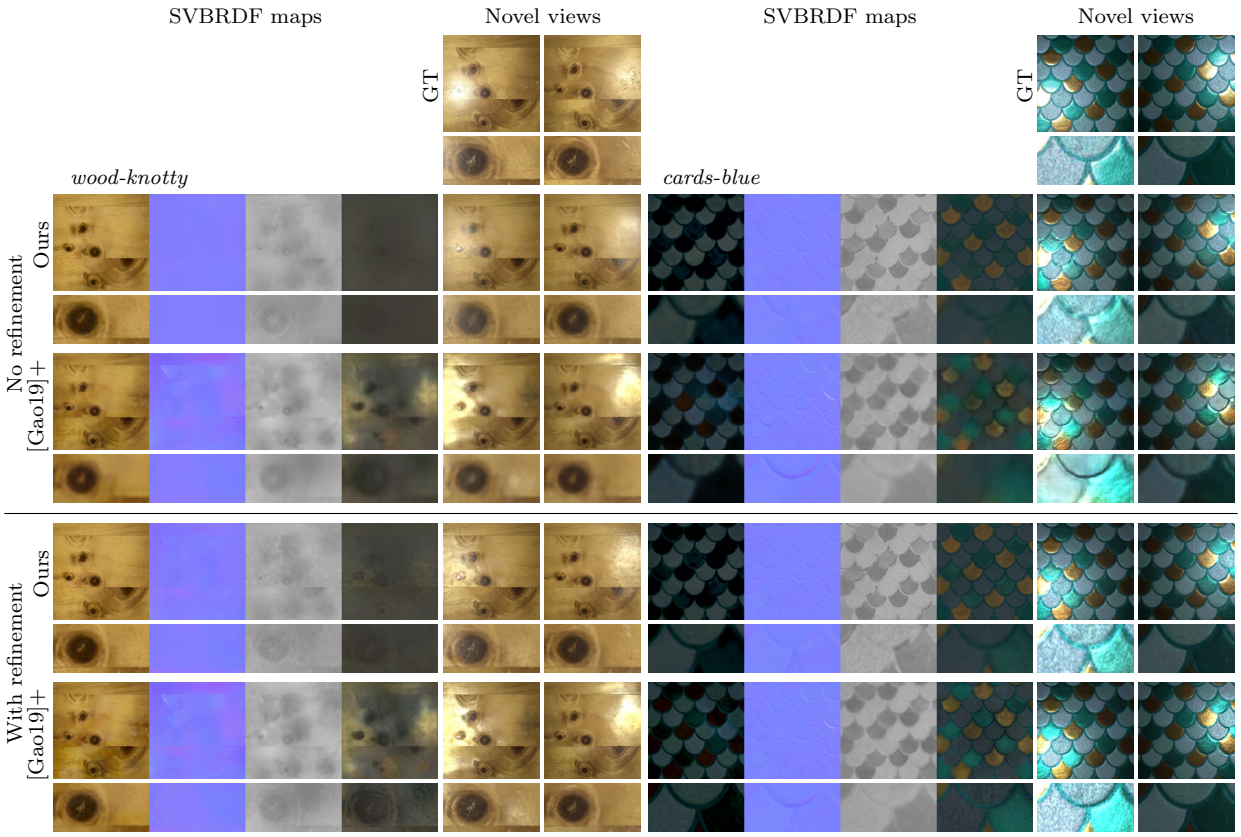


Figure 5.13: Per-pixel post-refinement. Unlike Gao's method, post-refinement via per-pixel optimization makes less of a difference in our method. Without post-refinement, [Gao19]+ (i.e., Gao's method initialized with Deschaintre's [23] direct predictions) usually produces blurry results, as shown in the row marked as "[Gao19]+ (NR)". Our method, on the contrary, does not rely nearly as heavily on post-refinement: Without it, our results are already quite sharp (see "Ours (NR)"), thanks to the generative power of our MaterialGAN. A zoomed-in version is attached below each SVBRDF map and novel-view image.

Post-refinement. In general, the quality of our maps is sufficient after using our MaterialGAN based optimization. However, Gao's method introduced a post-refinement step, where the maps are further optimized without any latent space, and with at most minor regularization. Therefore, we also implement a similar post-refinement step. However, like good initialization, this post-refinement makes less of a difference in our method, and Gao's method is more dependent on it, as it produces significantly blurry maps without it. This is shown in Figure 5.13; note the difference in sharpness of the maps.

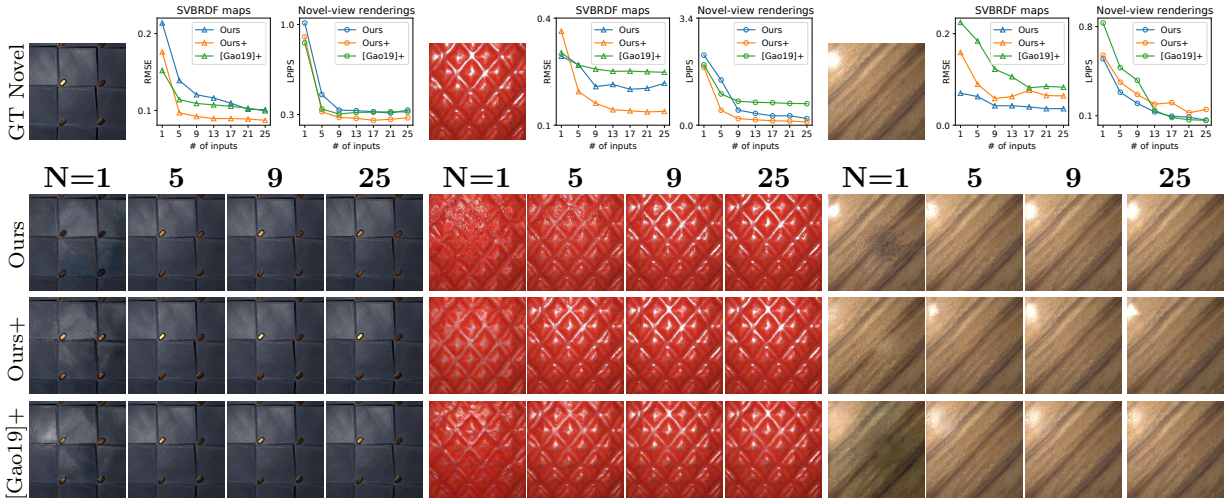


Figure 5.14: Performance using different numbers of input images (synthetic data). The quality of recovered SVBRDF maps, as demonstrated by the plots, generally improves with more input images for both our and Gao's [35] methods. Our method with constant (Ours) and neural (Ours+) initializations are comparable or better than Gao's ([Gao19]+) with neural initialization [23] . For a highly specular material shown on the right, although the LPIPS metric computed using renderings under 5 novel views of our results is similar to that of Gao's, ours better preserve the specular highlight. For each material, all the renderings including the references (GT Novel) are generated using one of the 5 novel views.

Optimization with different numbers of input images. While most of our results are shown with 7 inputs, using two additional inputs for novel lighting evaluation, our method does work with various numbers of input images. We show 3 synthetic examples in Figure 5.14, with different numbers of inputs from 1 to 25. All the three examples are the same as used in Gao's work. The errors of both reconstructed SVBRDF maps and novel-view renderings generally decrease with more input images, as is expected for an inverse-rendering method. In Figure 5.15, we compare real capture results with 1, 3, and 7 inputs, with and without initialization by Deschaintre's method, and also include Gao's results for 3 and 7 inputs (with initialization). Our result remains plausible with 3 inputs, though artifacts do get reduced with more inputs. For all numbers of inputs, our result (with or without initialization) tends to be cleaner than Gao's.

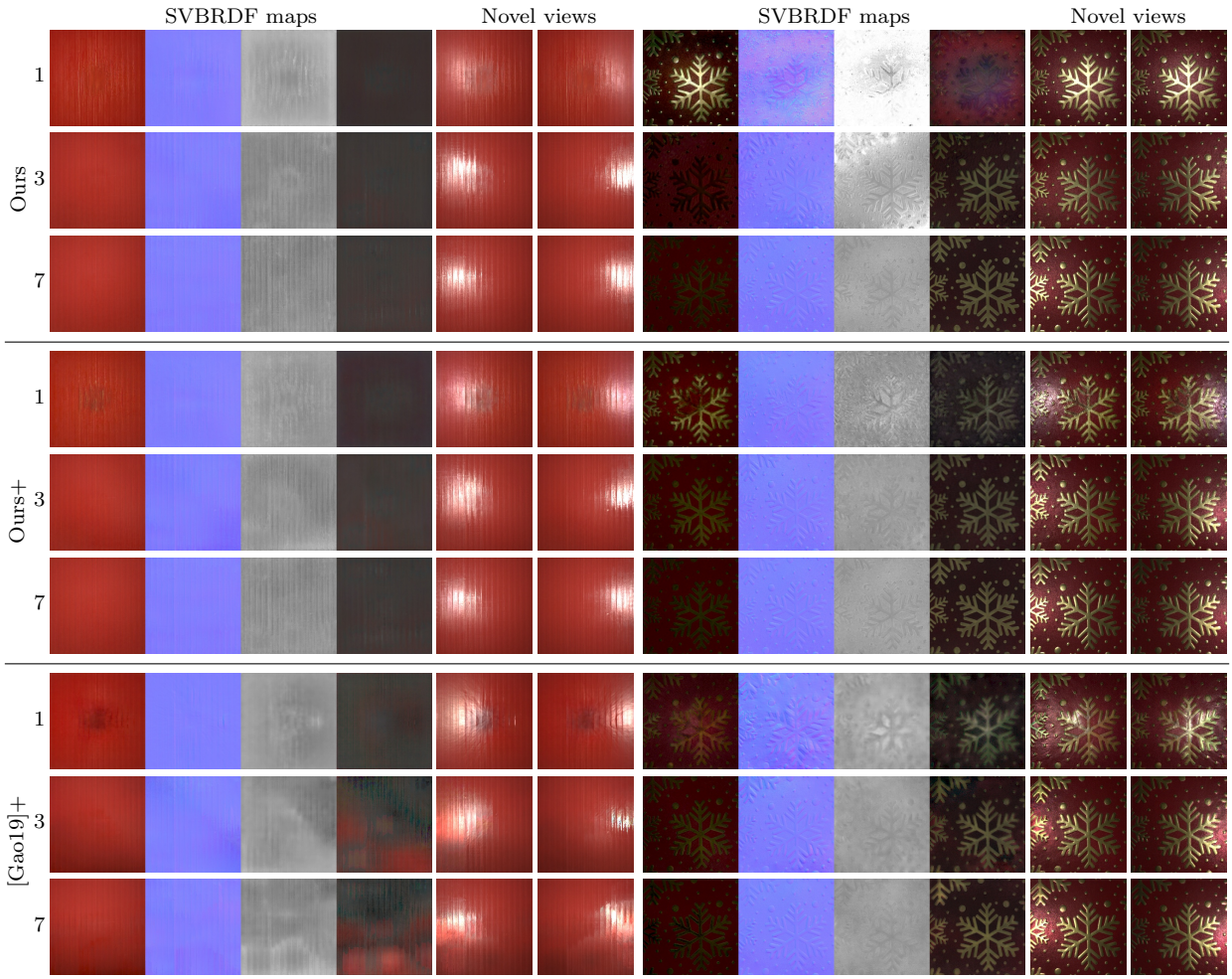


Figure 5.15: Performance using different numbers of input images (real data). The quality of SVBRDF maps recovered by our method generally improves with more input images under both constant initialization (see "Ours") and Deschaintre [23] initialization (see "Ours+").

Editing operations. An additional advantage of the StyleGAN-based latent space is the ability to achieve semantically meaningful operations such as morphing, by interpolating two or more parent latent codes to create a hybrid offspring material. Morphing in latent space often preserves semantic features qualitatively better than naive interpolation in pixel space. Figure 5.16 and the supplemental video show morphing of a few real materials using linear interpolation in latent space, compared to the corresponding naive interpolation (linear in pixel space).

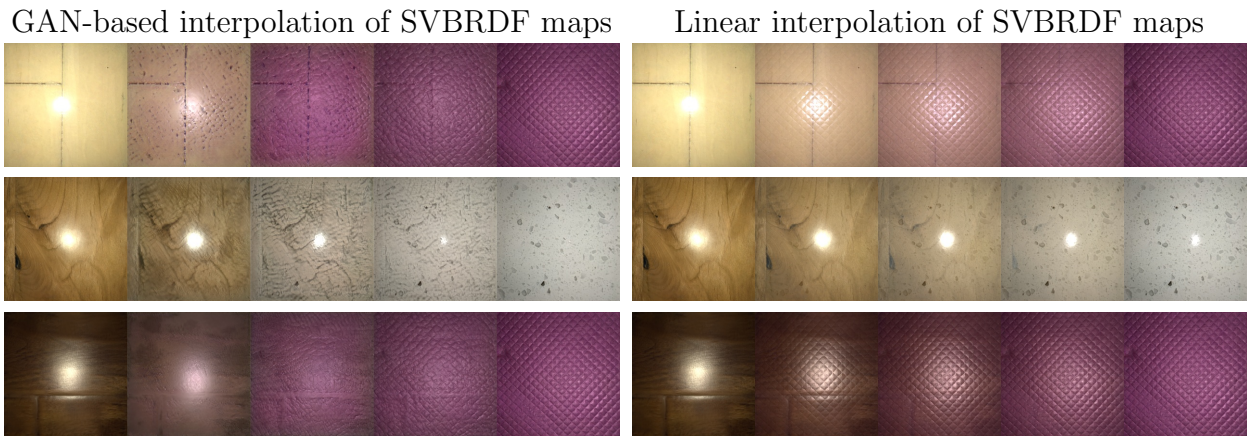


Figure 5.16: Material interpolation. Renderings of interpolations between two SVBRDFs recovered from real images using our method. Results on the left and right columns are obtained, respectively, using our GAN latent space and naïve linear interpolation.

5.6 Conclusion

We propose a novel method for acquiring SVBRDFs from a small number of input images, typically 3 to 7, captured using a hand-held mobile phone. We use an optimization framework that leverages a powerful material prior, based on a generative network, MaterialGAN, trained to synthesize plausible SVBRDFs. MaterialGAN learns correlations in SVBRDF parameters and provides local and global regularization to our optimization. This produces high-quality SVBRDFs that accurately reconstruct the input images, and because of our MaterialGAN prior, lie on a plausible material manifold. As a result, our reconstructions generalize better to novel views and lighting than previous state-of-the-art methods.

We believe that our work is only a first step toward GAN-based material analysis and synthesis and our experiments suggest many avenues for further exploration including improving material latent spaces and optimization techniques using novel architectures and losses, learning disentangled and editable latent spaces, and expanding beyond our current isotropic BRDF model.

Chapter 6

Procedural Parameters for Materials

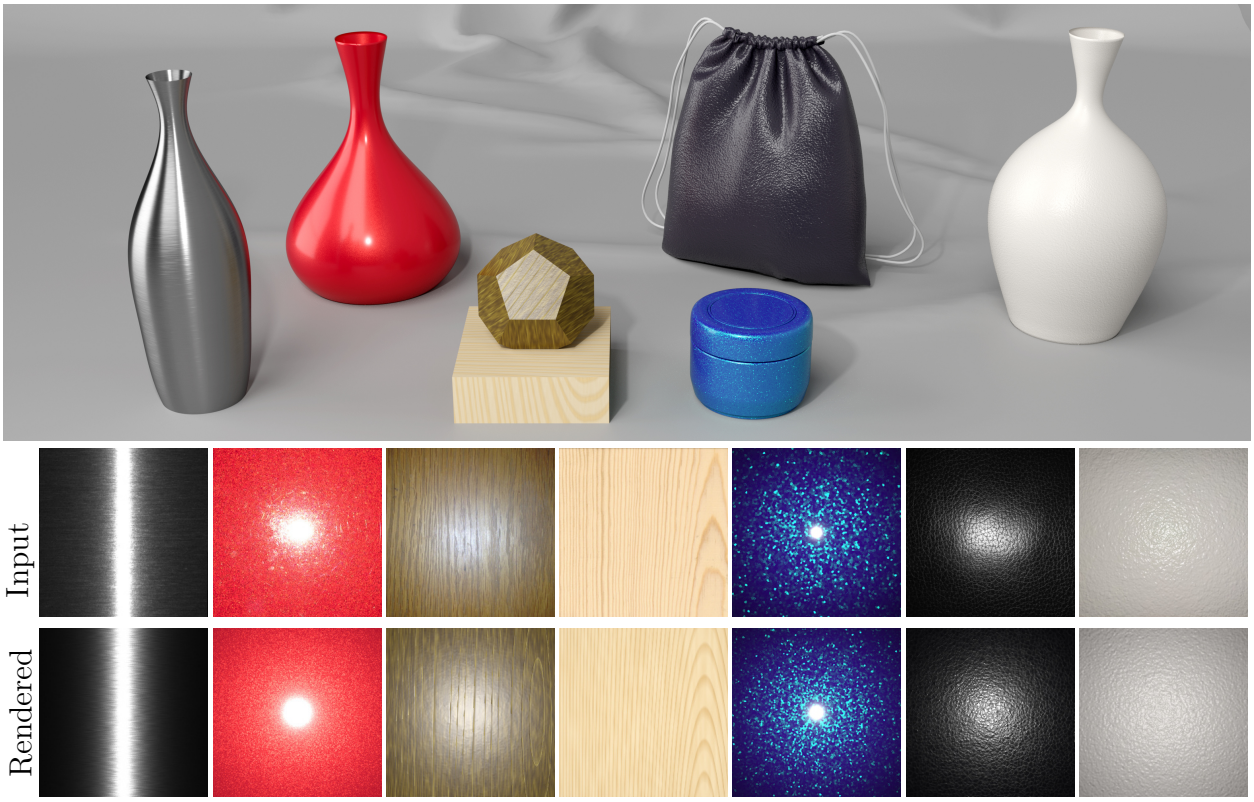


Figure 6.1: A scene rendered with material parameters estimated using our method: bumpy dielectrics, leather, plaster, wood, brushed metal, and metallic paint. The insets show a few examples of the input (target) images, and renderings produced using our procedural models with parameters found by Bayesian posterior sampling.

6.1 Introduction

Physically accurate simulation of material appearance is an important yet challenging problem, with applications in areas from entertainment to product design and architecture visualization. A key ingredient to photorealistic rendering is high-quality material appearance data. Acquiring such data from physical measurements such as photographs has been an active research topic in computer vision and graphics. Recently, *procedural* material models have been gaining significant traction in the industry (e.g., Substance [3]). In contrast to traditional texture-based spatially varying BRDFs that represent the variation of surface albedo, roughness, and normal vectors as 2D images, procedural models generate such information using a smaller number of user-facing parameters, providing high compactness, easy editability, and automatic seamless tiling.

The estimation of procedural model parameters faces several challenges. First, the procedural generation and physics-based rendering of materials is a complex process with a diverse set of operations, making the relationship between procedural model parameters and properties of the final renderings non-linear and non-trivial. Additionally, designing a suitable loss function (metric) to compare a synthesized image to a target image is not obvious. Finally, given the soft nature of the image matching problem, a single point estimate of the "best" match may be less informative than a collection of plausible matches that a user can choose from.

In this chapter, we introduce a new computational framework to estimate the parameters of procedural material models that focuses on these issues. Our framework enjoys high generality by not requiring the procedural model to take any specific form, and supporting any differentiable BRDF models, including anisotropy and layering.

To design the loss function, we consider neural summary functions (embeddings) based on Gram matrices of VGG feature maps [37, 38], as well as hand-crafted summary functions

(§6.4). The VGG feature map approach is becoming standard practice in computer vision, and was first introduced to material capture by Aittala et al. [4]; we extend this approach to procedural material estimation.

We make two main contributions. The first contribution is a unified view of the procedural parameter estimation problem in a *Bayesian framework* (§6.5), precisely defining the posterior distribution of the parameters given the captured data and priors, allowing for both maximization and sampling of the posterior. Four components (priors, procedural material model, rendering operator, summary function) together define our posterior distribution (outlined in Figure 6.2).

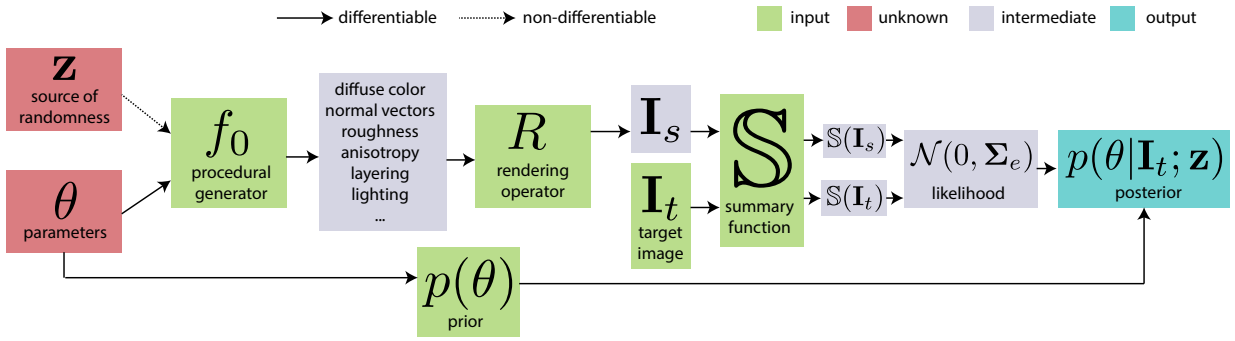


Figure 6.2: Our posterior computation combines priors, a procedural material model, a rendering operator, a summary function, and a target image. This posterior distribution can then be sampled to provide plausible values of the parameter vector. The value of the posterior is computed up to a normalization term, which does not effect MCMC sampling. The entire posterior definition is differentiable in the material parameters (excluding optional discrete model parameters).

Our second contribution is to introduce a *Bayesian inference* approach capable of drawing samples from the space of plausible material parameters. This provides additional information beyond single point estimates of material parameters (for example, though not limited to, discovering similarity structures in the parameter space). Further, due to an ability to combine multiple Markov-Chain Monte Carlo (MCMC) sampling techniques such as Metropolis-Hasting (MH), Hamiltonian Monte Carlo (HMC), and Metropolis-adjusted Langevin algorithm (MALA), our technique is capable of efficiently handling both discrete

and continuous model parameters. Posterior sampling is a well-studied area within statistics and has been used in computer vision and inverse rendering [79], but to our knowledge, it has not yet been applied to material appearance acquisition.

To demonstrate the effectiveness of our framework, we fit procedural models for a diverse set of materials from standard opaque dielectrics (e.g. plastics, leather, wall paint) to dielectrics with 3D structure (wood) to anisotropic brushed metals and layered metallic paints (see Figure 6.1, §6.6, and the supplemental materials).

6.2 Related Work

We review previous work on material parameter estimation in computer graphics and vision, as well as on Markov-Chain Monte Carlo (MCMC) methods in Bayesian inference.

SVBRDF capture. A large amount of previous work focuses on acquisition of material data from physical measurements. The methods generally observe the material sample with a fixed camera position, and solve for the parameters of a spatially-varying BRDF model such as diffuse albedo, roughness (glossiness) and surface normal. They differ in the number of light patterns required and their type; the patterns used include moving linear light [36], Gray code patterns [31] and spherical harmonic illumination [39]. In these approaches, the model and its optimization are specific to the light patterns and the optical setup of the method, as general non-linear optimization was historically deemed inefficient and not robust enough.

More recently, Aittala et al. [5] captured per-pixel SVBRDF data using Fourier patterns projected using an LCD screen; their algorithm used a fairly general, differentiable forward evaluation model, which was inverted in a maximum a-posteriori (MAP) framework.

Later work by Aittala et al. [6, 4] found per-pixel parameters of stationary spatially-varying SVBRDFs from two-shot and one-shot flash-lit photographs, respectively. In the latter case, the approach used a neural Gram-matrix texture descriptor based on the texture synthesis and feature transfer work of Gatys [37, 38] to compare renderings with similar texture patterns but without pixel alignment. We demonstrate that this descriptor makes an excellent summary function within our framework; in fact, the approach works well in our case, as the procedural nature of the model serves as an additional implicit prior, compared to per-pixel approaches. On the other hand, our forward evaluation process is more complex than Aittala et al., since it also includes the procedural material generation itself.

Recent methods by Deschaintre et al. [22], Li et al. [86] have been able to capture non-stationary SVBRDFs from a single flash photograph by training an end-to-end deep convolutional network. Gao et al. [35] introduced an auto-encoder approach, optimizing the appearance match in the latent space. All of these approaches estimate per-pixel parameters of the microfacet model (diffuse albedo, roughness, normal), and are not obviously applicable to estimation of procedural model parameters, nor to more advanced optical models (significant anisotropy, layering or scattering).

Procedural material parameter estimation. Focus on estimating the parameters of procedural models has been relatively rare. The dual-scale glossy parameter estimation work of Wang et al. [129] finds, under step-edge lighting, the parameters of a bumpy surface model consisting of a heightfield constructed from a Gaussian noise power spectrum and global microfacet material parameters. Their results provide impressive accuracy, but the solution is highly specialized for this material model and illumination.

Recently, Hu et al. [60] introduced a method for inverse procedural material modeling that treats the material as a black box, and trains a neural network mapping images to parameter vector predictions. The training data comes from evaluating the black box model for random

parameters. In our experiments, this approach was less accurate; our fully differentiable models can achieve higher accuracy fits and can be used to explore posterior distributions through sampling. In a sense, this neural prediction method could be seen as orthogonal to ours, as we could use it for initialization of our parameter vector, continuing with our MCMC sampling.

Optical parameters of fiber-based models. Several approaches for rendering of fabrics model the material at the microscopic fiber level [141, 142, 82]. However, the optical properties of the fibers (e.g. roughness, scattering albedo) have to be chosen separately to match real examples. Zhao et al. [141] use a simple but effective trick of matching the mean and standard deviation (in RGB) of the pixels in a well-chosen area of the target and simulated image. Khungurn et al. [76] have extended this approach with a differentiable volumetric renderer, combined with a stochastic gradient descent; however, their method is still specific to fiber-level modeling of cloth.

Bayesian inference and MCMC. A variety of methods used across the sciences are Bayesian in nature; in this chapter, we specifically explore Bayesian inference for parameter estimation through Markov-Chain Monte Carlo (MCMC) sampling of the posterior distribution. Provided a nonnegative function f, MCMC techniques can draw samples from the probability density proportional to the given function f without knowing the normalization factor. Metropolis-Hastings [54] is one of the most widely used MCMC sampling methods. If f is differentiable, the presence of gradient information leads to more efficient sampling methods such as Hamiltonian Monte Carlo (HMC) [101, 14] and Metropolis-adjusted Langevin algorithm (MALA) [111]. Our inference framework is not limited to any specific MCMC sampling technique. In practice, our implementation handles discrete model parameters using MH and continuous ones using MALA (with preconditioning [19]). We opt MALA for its simpler hyper-parameter tweaking (compared to HMC).

MCMC applications in graphics and vision. Markov chain Monte Carlo techniques have been heavily studied in rendering, though not for Bayesian inference, but rather for sampling light transport paths with probability proportional to their importance; notably Metropolis light transport [125] and its primary sample space variant [75]. Much further work has built on these techniques, including more recent work that uses a variant of Hamiltonian Monte Carlo [83]. However, all of these approaches focus on better sampling for traditional rendering, rather than parameter estimation in inverse rendering tasks.

In computer vision, Bayesian inference with MCMC has been used for the inverse problems of scene understanding. A notable previous work is Picture [79], a probabilistic system and programming language for scene understanding tasks, for example (though not limited to) human face and body pose estimation. The programming language is essentially used to specify a forward model (e.g., render a face in a given pose), and the system then handles the MCMC sampling of the posterior distribution through a combination of sampling (proposal) techniques. This is closely related to the overall design of our system. However, the Picture system does not appear to be publicly available, and our application is fairly distant from its original goals.

6.3 Preliminaries

Procedural model generation. We focus on *procedural material models* which utilize specialized procedures (pieces of code) to generate spatially varying surface reflectance information. Specifically, let $\boldsymbol{\theta}$ be the parameters taken by some procedural material generation process f_0. Then, $f_0(\boldsymbol{\theta})$ generates the material properties (e.g., albedo, roughness, surface normals, anisotropy, scattering, etc.), in addition to any other parameters required by rendering (e.g. light parameters), which can in turn be converted into a synthetic image $\boldsymbol{I}_{\mathrm{s}}$ via

a rendering operator R. This *forward evaluation* process can be summarized as

$$\boldsymbol{I}_{\mathrm{s}} = R(f_0(\boldsymbol{\theta})) = f(\boldsymbol{\theta}), \tag{6.1}$$

where f is the composition of R and f_0. When modeling real-world materials, it is desirable to capture naturally arising irregularities. In procedural modeling, this is usually achieved by making the model generation process f_0 to take extra random input $\boldsymbol{z}$ (e.g., random seeds, pre-generated noise textures, etc.) that is then used to create the irregularities. This also causes the full forward evaluation to become $f(\boldsymbol{\theta}; \boldsymbol{z}) := R(f_0(\boldsymbol{\theta}; \boldsymbol{z}))$.

Continuous and discrete parameters. While most procedural material parameters tend to be continuous, discrete parameters can be useful for switching certain components on and off, or for choosing between several discrete noise types. We model this by splitting the parameter vector into continuous and discrete components, $\boldsymbol{\theta} = (\boldsymbol{\theta}_c, \boldsymbol{\theta}_d)$. We assume the forward evaluator f to be differentiable with respect to $\boldsymbol{\theta}_c$ (but not $\boldsymbol{\theta}_d$ or the random input $\boldsymbol{z}$).

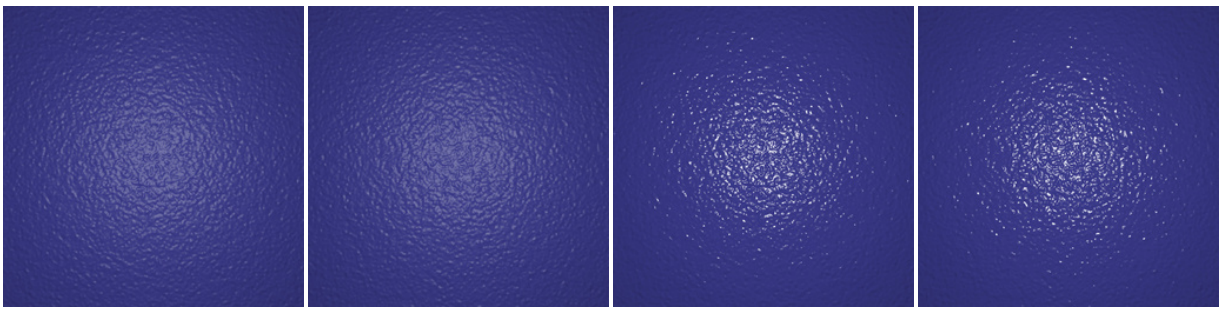
Inverse problem specification. We consider the problem of inferring procedural model parameters $\boldsymbol{\theta}$ given a target image $\boldsymbol{I}_{\mathrm{t}}$ (which is typically a photograph of a material sample under known illumination). This, essentially, requires inverting f in Eq. (6.1): $\boldsymbol{\theta} = f^{-1}(\boldsymbol{I}_{\mathrm{t}})$. Direct inversion of $f = R \circ f_0$ is intractable for any but the simplest material and rendering models. Instead, we aim to identify a collection of plausible values $\boldsymbol{\theta}$ such that $\boldsymbol{I}_{\mathrm{s}}$ has similar appearance to $\boldsymbol{I}_{\mathrm{t}}$:

$$\text{find examples of } \boldsymbol{\theta} \text{ s.t. } \boldsymbol{I}_{\mathrm{t}} \approx f(\boldsymbol{\theta}; \boldsymbol{z}), \tag{6.2}$$

for some (any) $\boldsymbol{z}$, where $\approx$ is an *appearance-match* relation that will be discussed in the next section.

6.4 Summary Functions

To solve the parameter estimation problem using Eq. (6.2), a key ingredient is the appearance-match relation. Unfortunately, we cannot use simplistic image difference metrics such as the L2 or L1 norms. This is because the features (bumps, scratches, flakes, yarns, etc.) in the images of real-world materials are generally misaligned, even when the two images represent the same material. In procedural modeling, as shown in Figure 6.3, with irregularities created differently using $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, the same procedural model parameters $\boldsymbol{\theta}$ can yield slightly different results $f(\boldsymbol{\theta}; \boldsymbol{z}_1)$ and $f(\boldsymbol{\theta}, \boldsymbol{z}_2)$.



(a1) (a2) (b1) (b2)

Figure 6.3: L2 norm difference. Each pair of images among (a, b) are generated using identical model parameters $\boldsymbol{\theta}$ but different irregularities $\boldsymbol{z}$. The pixel-wise L2 norm of the difference between these image pairs is large and not useful for estimating model parameters.

To overcome this challenge, we use the concept of a *summary function*, which abstracts away the unimportant differences in the placement of the features, and summarizes the predicted and target images into smaller vectors whose similarity can be judged with simple metrics like L2 distance. We define an image summary function $\mathcal{S}$ to be a continuous function that maps an image of a material ($\boldsymbol{I}_\mathrm{t}$ or $\boldsymbol{I}_\mathrm{s}$) into a vector in $\mathbb{R}^k$. An idealized summary function would have the property that

$$\mathcal{S}(f(\boldsymbol{\theta}_1, \boldsymbol{z}_1)) = \mathcal{S}(f(\boldsymbol{\theta}_2, \boldsymbol{z}_2)) \;\Leftrightarrow\; \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{6.3}$$

That is, applying the summary function would abstract away from the randomness $\boldsymbol{z}$ and

the difference between the two summary vectors would be entirely due to different material properties $\boldsymbol{\theta}$. Practical summary functions will satisfy the above only approximately. However, a good practical summary function will embed images of the same material close to each other, and images of different materials further away from each other. Below we discuss several techniques for constructing summary functions.

Neural summary function. Gatys et al. [37, 38] introduced the idea of using the features of an image classification neural network (usually VGG [115]) as a descriptor T_G of image texture (or style). Optimizing images to minimize the difference in T_G (combined with other constraints) allowed Gatys et al. to produce impressive, state-of-the art results for parametric texture synthesis and style transfer between images. While further work has introduced improvements [110], we find that the original version from Gatys et al. works already well in our case.

Aittala et al. [4] introduced this technique to capturing material parameter textures (albedo, roughness, normal and specular maps) of stationary materials. They optimized for a 256×256 stationary patch that matches the target image in various crops, using a combination of T_G and a number of special Fourier-domain priors. In our case (for procedural materials), we find that the neural summary function T_G works even more effectively; we can simply apply it to the entire target or simulated images (not requiring crops nor Fourier-domain priors).

Specifically, define the Gram matrix G of a set of feature maps $F_1, \cdots, F_n$ such that it has elements

$$G_{ij} = \text{mean}(F_i \cdot F_j), \tag{6.4}$$

where the product $F_i \cdot F_j$ is element-wise. T_G is defined as the concatenation of the flattened Gram matrices computed for the feature maps before each pooling operation in VGG19. Note that the size of the Gram matrices depends on the number of feature maps (channels),

not their size; thus T_G is independent of input image size.

Statistics and Fourier transforms of image bins. While the neural summary function performs quite well, we find that in some cases we can improve upon it. A simple idea for a summary function is to use the (RGB) mean of the entire image; an improvement is to subdivide the image into k bins (regions) and compute the mean of each region. We found concentric bins perform well for isotropic materials, and vertical bins are appropriate for anisotropic highlights (e.g. brushed metal). Furthermore, we can additionally use a fast Fourier transform of the entire image or within bins. Note that automatic computation of derivatives is possible with the FFT, and supported by the PyTorch framework. In our current results, we use a summary function that combines the means and 1D FFTs of 64 vertical bins for the brushed metal example; all other examples use the neural summary function combined with simple image mean.

6.5 Bayesian Inference of Material Parameters

In what follows, we first describe a Bayesian formulation of the estimation problem in terms of a posterior distribution. Next, we discuss how to use the posterior for point estimation in a maximum a-posteriori (MAP) framework, and how the Markov-Chain Monte Carlo (MCMC) methods for Bayesian inference extend the point estimate approach by sampling from the posterior.

6.5.1 Bayesian formulation

We treat the procedural model parameters $\boldsymbol{\theta}$ as random variables with corresponding probability distributions.

We first introduce a *prior* probability distribution $p(\boldsymbol{\theta})$ of the parameters, reflecting our pre-existing beliefs about the likelihood values of the unknown parameters. For example, for most material categories, we know what range the albedo color and roughness coefficients of the material should typically be in.

Further, we model the $\approx$ operator from Eq. (6.2) as an error distribution. Specifically, we postulate that the difference between the simulated image summary $\mathcal{S}(f(\boldsymbol{\theta}, \boldsymbol{z}))$ and the target image summary $\mathcal{S}(\boldsymbol{I}_\mathrm{t})$ follows a known probability distribution. In practice, we use a (multi-variate) normal distribution with zero mean and the covariance $\boldsymbol{\Sigma}_e$:

$$\mathcal{S}(f(\boldsymbol{\theta}, \boldsymbol{z})) - \mathcal{S}(\boldsymbol{I}_\mathrm{t}) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_e). \tag{6.5}$$

Our experiments indicate that this simple error distribution works well in practice, and we regard $\boldsymbol{\Sigma}_e$ as a hyper-parameter and set it manually.

We also have multiple options in handling the random vector $\boldsymbol{z}$. While it is certainly theoretically possible to estimate it, we are not really interested in its values; we find it simpler and more efficient to simply choose $\boldsymbol{z}$ randomly, fix it, and assume it known during the process of estimating the "interesting" parameters $\boldsymbol{\theta}$. Under these assumptions, according to the Bayes theorem, we can write down the posterior probability of parameters $\boldsymbol{\theta}$, conditional on the known values of $\boldsymbol{I}_\mathrm{t}$ and $\boldsymbol{z}$, as:

$$p(\boldsymbol{\theta}|\boldsymbol{I}_\mathrm{t}, \boldsymbol{z}) \propto \mathcal{N}\left[\mathcal{S}(f(\boldsymbol{\theta}, \boldsymbol{z})) - \mathcal{S}(\boldsymbol{I}_\mathrm{t}); 0, \boldsymbol{\Sigma}_e\right] p(\boldsymbol{\theta}), \tag{6.6}$$

where the right side does not need to be normalized; the constant factor has no effect on parameter estimates. For numerical stability, we compute the negative log posterior, viz. $-\log p(\boldsymbol{\theta}|\boldsymbol{I}_\mathrm{t}, \boldsymbol{z})$, in practice. Equation (6.6) also holds when $\boldsymbol{\theta}$ involves discrete parameters, as long as the prior is properly defined as a product of a continuous pdf $p(\boldsymbol{\theta}_c)$ and a discrete probability $p(\boldsymbol{\theta}_d)$.

6.5.2 Point estimate of parameter values

A point estimate of the parameter vector can be modeled in the maximum a-posteriori (MAP) framework. We simply estimate the desired parameter values $\boldsymbol{\theta}$ as the maximum of the posterior pdf $p(\boldsymbol{\theta}|\boldsymbol{I}_{\mathrm{t}}, \boldsymbol{z})$ given by Eq. (6.6). For continuous $\boldsymbol{\theta}$, this problem can be solved using standard non-linear optimization algorithms. In the presence of discrete parameters, there is no single accepted solution. While various heuristics could be used, our sampling approach described below provides a cleaner solution to discrete parameter estimation.

6.5.3 Markov-Chain Monte Carlo Sampling of the Posterior

Although the point estimate approach gives satisfactory results in many cases, it is not without problems. For example, since a perfect match between a procedural material and a photograph is generally impossible, it can be desirable to have a set of imperfect matches for the user to choose from. Further, there could be an entire subset of the parameter space giving solutions of approximately equivalent fit under the target view and lighting; however, these may look quite different from each other in other configurations, and a user may want to explore those differences. Lastly, with the presence of discrete parameters, it is not obvious how to solve the maximization in Eq. (6.6) efficiently.

In this chapter, we use the well-known technique of full Bayesian inference, sampling the posterior pdf defined in Eq. (6.6) using Markov-Chain Monte Carlo (MCMC) techniques, specifically Metropolis-Hasting (MH) [54], Hamiltonian Monte Carlo (HMC) [14], and Metropolis-adjusted Langevin algorithm (MALA) [111]. While well explored in statistics and various scientific fields, to our knowledge, this technique has not been used for the inference of material parameters.

The goal of the sampling is to explore the posterior with many (typically thousands or more)

samples, each of which represents a material parameter vector consistent with the target image. Plotting these samples projected into two dimensions (for a given pair of parameters) gives valuable insight into similarity structures. Furthermore, interactively clicking on samples and observing the predicted result can help a user to quickly zoom in on a preferred solution, which an automatic optimization algorithm is fundamentally incapable of.

Algorithm 1: MCMC sampling of material parameters $(\boldsymbol{\theta}_c, \boldsymbol{\theta}_d)$

```
1  samplePosterior(N, α, θ_c^(1), θ_d^(1))
   Input: Sample count N, probability α for sampling continuous parameters, initial
          continuous parameters θ_c^(1) and discrete ones θ_d^(1)
   Output: N material parameter estimates {(θ_c^(t), θ_d^(t)) : 1 ≤ t ≤ N}
2  begin
3      for t = 1 to (N − 1) do
4          Draw ξ ~ U[0, 1)
5          if ξ < α then                                  // Mutate continuous parameters
6              θ'_c ← ContinuousSample(θ_c^(t))
7              θ'_d ← θ_d^(t)
8          else                                           // Mutate discrete parameters
9              θ'_c ← θ_c^(t)
10             θ'_d ← DiscreteSample(θ_d^(t))
11         end
12         (θ_c^(t+1), θ_d^(t+1)) ← MetropolisHasting((θ'_c, θ'_d), (θ_c^(t), θ_d^(t)))
13     end
14     return {(θ_c^(1), θ_d^(1)), ..., (θ_c^(N), θ_d^(N))}
15 end
```

Algorithm 1 summarizes our MCMC sampling process. At each iteration, we mutate either the continuous parameters (with probability α) or the discrete ones (with probability $1-\alpha$). For the former case, we utilize the gradient of the log pdf with respect to $\boldsymbol{\theta}_c$ to efficiently obtain a new proposal $\boldsymbol{\theta}'_c$ (Line 6). Our implementation uses MALA for this process, although HMC could also work. For the latter case, we obtain a new proposal $\boldsymbol{\theta}'_d$ of the discrete parameters, currently by uniformly sampling their joint probability mass function (Line 10). Upon obtaining a full proposal, we use the standard Metropolis-Hasting rule (Line 12) to stochastically select the new sample $(\boldsymbol{\theta}_c^{(t+1)}, \boldsymbol{\theta}_d^{(t+1)})$ by either accepting the newly proposed

$(\boldsymbol{\theta}'_c, \boldsymbol{\theta}'_d)$ or (rejecting the proposal and) keeping the previous sample $(\boldsymbol{\theta}_c^{(t)}, \boldsymbol{\theta}_d^{(t)})$.

6.6 Material Models and Results

(Only a small subset of our results fits into the thesis. Please see our supplemental material and video for more results. [Click here])

We now demonstrate the effectiveness of our technique by fitting several procedural material models to a mix of synthetic and real target images.

Our forward evaluation process uses collocated camera and light. This configuration closely matches a mobile phone camera with flash (which is used for most of the real target images) and simplifies some BRDF formulations (because the incoming, outgoing, and half-way vectors are all identical). Further, we assume that the distance between camera and sample is known as it is generally easy to measure or estimate. The knowledge of the camera field of view allows us to compute the physical scale of the resulting pixels. Lastly, we treat light intensity and camera vignetting (expressed as an image-space Gaussian function) as (unknown) parameters of the forward evaluation process so that they do not need to be calibrated. Our parameter inference framework presented in §6.4 and §6.5 is not limited to this specific setup.

All the procedural material models we used, which will be detailed in §6.6.2, are implemented using PyTorch which automatically provides GPU acceleration and computes derivatives through backpropagation. For all material parameter inference tasks, our forward evaluation generates 512×512 images. Notice that the recovered parameters can then be used to generate results with much higher resolution because the procedural models are generally resolution-independent.

6.6.1 Similarity Relations in Translucency

As a motivating example, we first illustrate the behavior of the MCMC material parameter estimation process on the case of a homogeneous translucent material with two varying parameters. In this example, the shape of the posterior can be analytically derived (using the similarity theory) and easily plotted. This serves as a demonstration and validation of our approach.

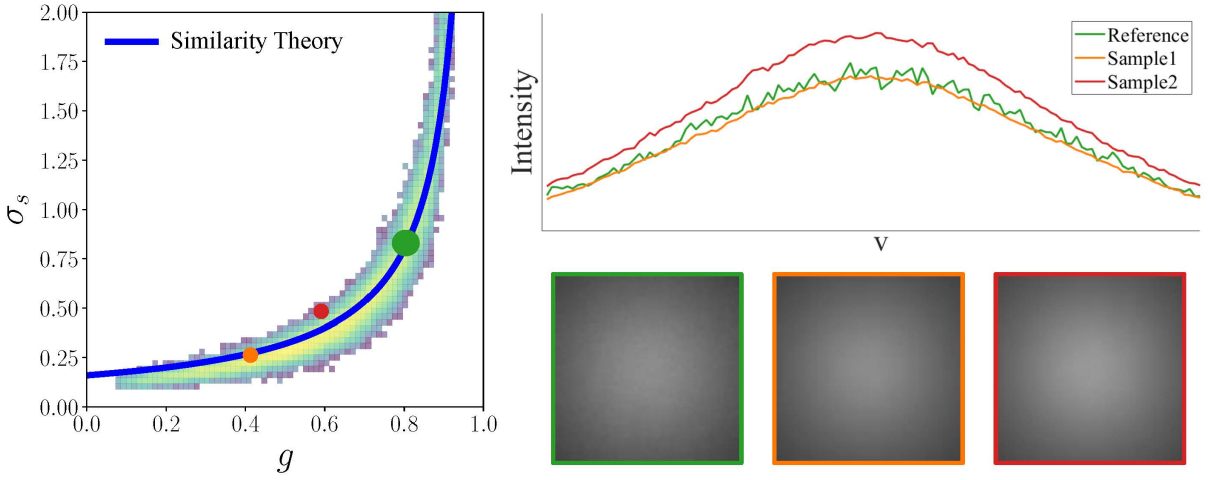


Figure 6.4: A motivating example of a scattering material with two estimated parameters (scattering coefficient and phase function parameter). The posterior distribution sampled with our method for three synthetic input images is able to detect the full structure of the parameter space, matching the predictions from similarity theory.

Specifically, the material parameter space of translucent materials under the radiative transfer framework [18] is known to be approximately over-complete [143]. Specifically, two sets of parameters $(\sigma_\mathrm{s}, \sigma_\mathrm{a}, g)$ and $(\sigma_\mathrm{s}^*, \sigma_\mathrm{a}^*, g^*)$ satisfying the following *similarity relation* usually yield similar final appearances:

$$\sigma_\mathrm{a} = \sigma_\mathrm{a}^*, \quad (1-g)\,\sigma_\mathrm{s} = (1-g^*)\,\sigma_\mathrm{s}^*, \tag{6.7}$$

where σ_a and σ_s are, respectively, the absorption and scattering coefficients, and g is the first

Legendre moment of the phase function. We show in Figure 6.4 that applying our Bayesian inference method to σ_s and g (with fixed σ_a) computes a posterior distribution that agrees well with the predicted similarity relation (6.7).

6.6.2 Procedural Material Models

We show results generated using synthetic images in Figures 6.5 and 6.6 as well as real photographs (taken with different cameras) in Figure 6.7. Please see the supplemental material for more results, including animations illustrating point estimations and sampling. Below we describe the six procedural models tested. Please refer to the supplement for additional detail and a PyTorch implementation. For each parameter, we define a reasonable truncated Gaussian distribution as its prior (also see supplement). In most cases, the MCMC sampling starts from the peak of the prior. In some examples (e.g wood), we first run posterior maximization and then switch to sampling from the optimized point. We drop some number (typically 200 to 1000) of initial MCMC samples due to burn-in.

Table 6.1: Performance statistics for our MCMC-based posterior sampling. The numbers are collected using a workstation equipped with an Intel i7-6800K six-core CPU and an Nvidia GTX 1080 GPU.

Material	bump	leather	plaster	flakes	metal	wood
# of parameters.	8	12	11	13	10	23
MCMC (1k iter.)	180s	194s	190s	187s	182s	290s

Bumpy microfacet surface. This model depicts an opaque dielectric surface with an isotropic noise heightfield. We use a standard microfacet BRDF with the GGX normal distribution [127] combined with a normal map computed from an explicitly constructed heightfield. We assume that the Fresnel reflectance at normal incidence can be computed from a known index of refraction (a value of 1.5 is a good estimate for plastics). We assume an unknown roughness r (GGX parameter $\alpha = r^2$) and a Lambertian diffuse term with unknown

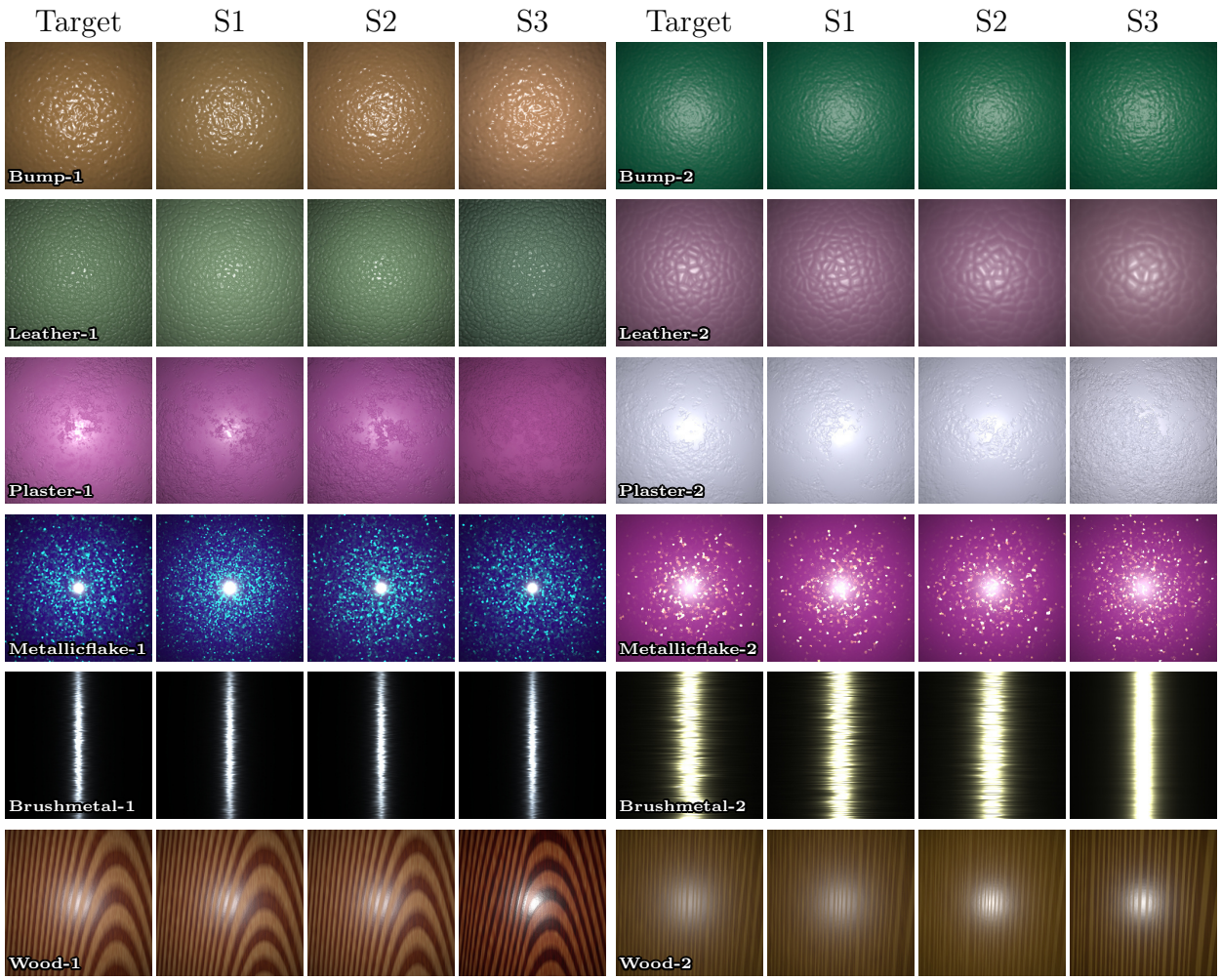


Figure 6.5: Results of our MCMC sampling on **synthetic** inputs. Each row corresponds to two examples of a different material model. For each example, the first column is the synthetic target image. We show MCMC samples in the other columns, where S1 and S2 are chosen closer to the peak of the posterior distribution, and S3 is further away. More results please refer to supplemental materials.

albedo ρ. This model is identical to Wang et al. [129], except using the GGX instead of Beckmann microfacet distribution. The main practical difference from the capture setup in that paper is that we use a point light, instead of step-edge illumination.

The bumpy heightfield is constructed using an inverse Fourier process including: (i) choosing a power spectrum in the continuous Fourier domain; (ii) discretizing it onto a grid of complex numbers; (iii) randomly choosing the phase of each texel on the grid (while keeping the chosen amplitude); and (iv) applying an inverse fast Fourier transform whose real component

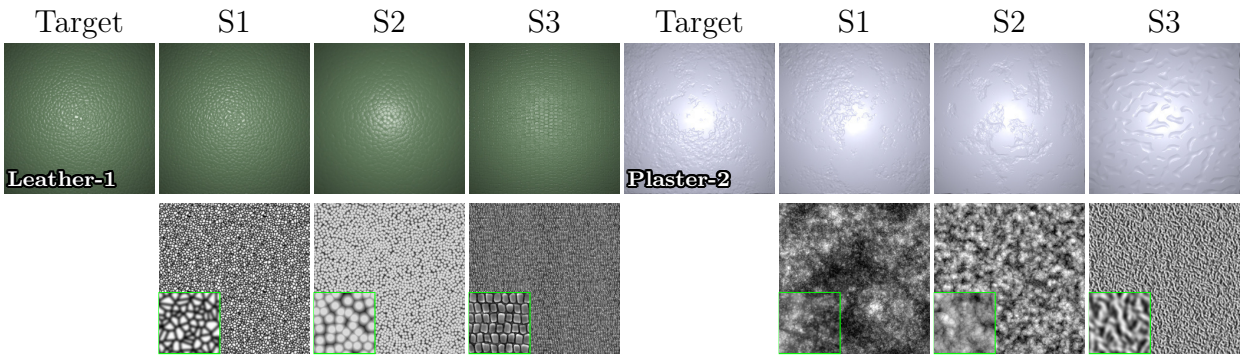


Figure 6.6: MCMC sampling with discrete parameters. In these examples, we illustrate the ability of our sampling to handle discrete parameters. In both examples, one noise inputs used in the procedural model can be switched between several different types of noise. Out of the thousands of sampled solutions, we pick three that have different settings of the discrete parameter where the (log) pdf values decrease from S1 to S3.

becomes the resulting heightfield. At render time, we use the normal map derived from this heightfield.

Leather and plaster. These materials can be modeled similarly as the aforementioned bumpy surfaces except for the computation of the heightfield and roughness. For plaster, a fractal noise texture is scaled (in space and intensity) and thresholded (controlled by additional parameters) to produce both the heightfield and a roughness variation texture. For leather, on the contrary, a Voronoi cell map is used to get the effect of leather-like cells (with parameters for scaling and thresholding), and additional small-scale fractal noise is added. Further, we use multiple (pre-generated) noise textures and Voronoi cell maps to diversify the micro-scale appearances that our models can produce. The choice of these textures and maps is captured using a discrete parameter. In Figure 6.6, we show a few example samples drawn from the posterior distributions using Algorithm 1.

Brushed metal. The brushed metal material extends the above bumpy surface, by introducing anisotropy to both the GGX normal distribution and the noise heightfield used to compute the normal map, while dropping the diffuse term. We make both the BRDF

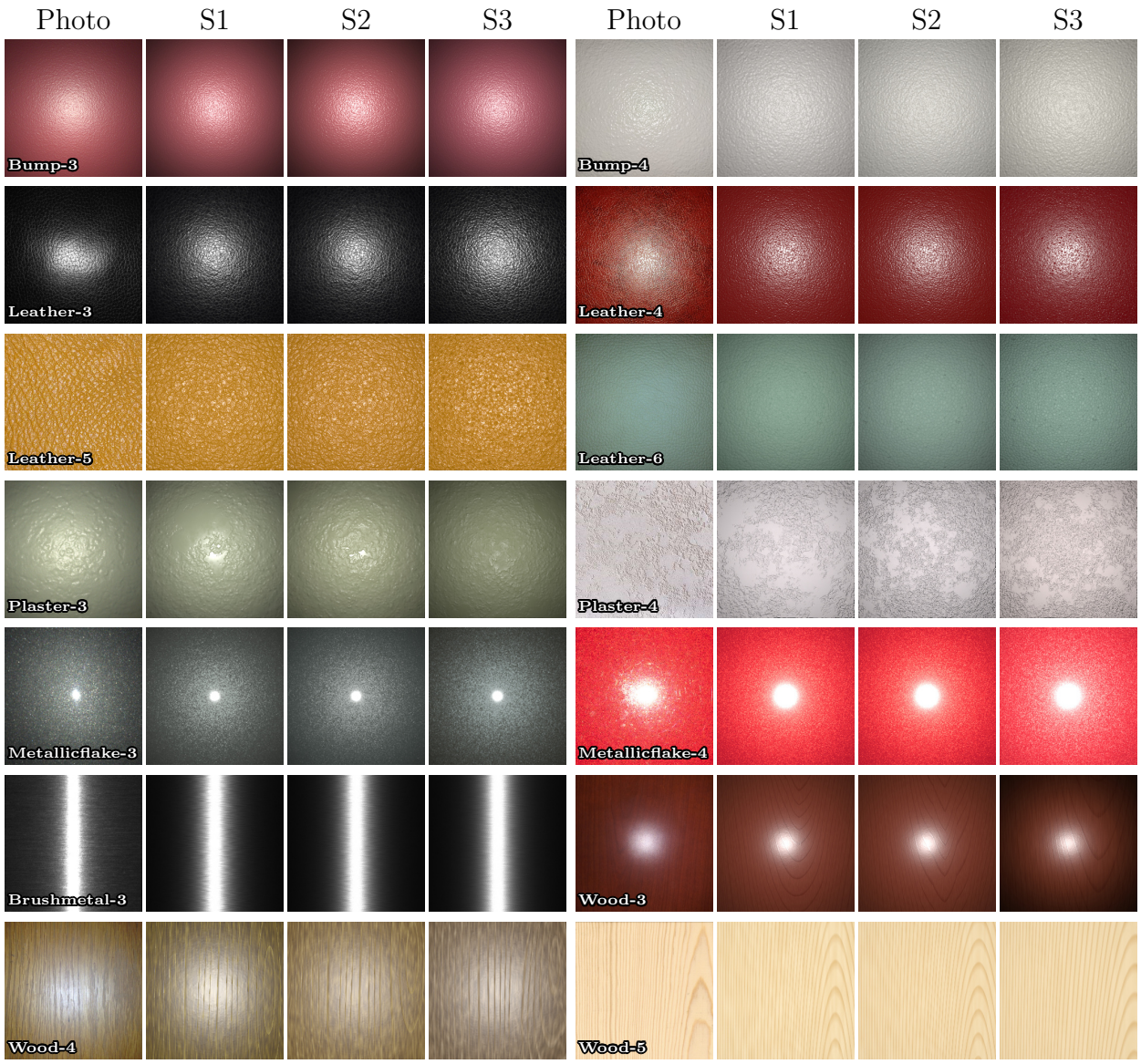


Figure 6.7: Results of our MCMC sampling on **real** inputs. For each example, the first column is the real target image (photo). We show MCMC samples in the other columns, where sample-1 and sample-2 are chosen closer to the peak of the posterior distribution, and sample-3 is further away. Note that the target images for Plaster-4 and Wood-5 are captured under natural illumination, while the corresponding synthetic images still assume collocated flash illumination; despite this mismatch, the estimated material parameters are still reasonable. Note, target images for Leather-4, Leather-6 and Wood-4 are from the publicly released dataset of [4]. For more results please refer to supplemental materials.

and the Fourier-domain Gaussian power spectrum anisotropic. The parameters of the model thus include two roughnesses, as well as two Fourier-domain standard deviations. We make the anisotropic highlight vertical and centered in the target image.

Metallic flakes. Metallic paint with flakes is a stochastic material with multiple BRDF lobes (caused by light reflecting off the flakes). Our model involves three components, each being an isotropic microfacet lobe, to describe top coating, flakes and glow, respectively. The top coating is usually highly specular, and we make its roughness a model parameter. We assume an index of refraction of 1.5, implying a Fresnel (Schlick) reflectivity at normal incidence of 0.04. The flakes are chosen as Voronoi cells of a random blue-noise point distribution; they have a roughness parameter and varying normals chosen from the Beckmann distribution with an unknown roughness, and with unknown Fresnel reflectivity. The scale of the cell map is itself a (differentiable) parameter. Lastly, the glow is a component approximating the internal scattering between the top interface and the flakes, and has its own roughness, Fresnel reflectivity and a flat normal. An extra weight parameter linearly combines the flakes and the glow.

Wood. We also created a partial PyTorch implementation of the comprehensive 3D wood model of Liu et al. [87]. This material is a 3D model of the growth rings of a tree, with a number of parameters controlling colors and widths of growth rings, as well as global distortions and small-scale noise features. The 3D wood is finally projected by a cutting plane to image space, defining diffuse albedo, roughness and height.

Mismatched models. Lastly, we demonstrate in Figure 6.8 the impact of forward procedural models. Since these model-generating procedures are essentially material-specific priors, using mismatched models generally leads to results that match overall image statistics but with incorrect patterns.

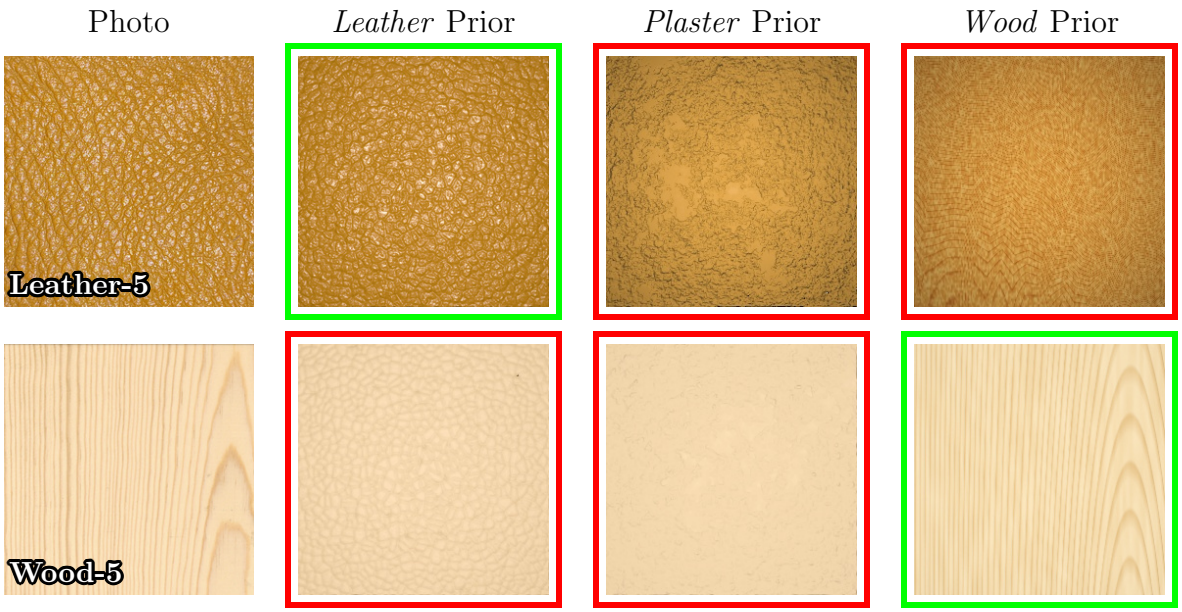


Figure 6.8: Comparison with mismatched forward models. With an inappropriate model as the prior, it would only match the global color but missing all the details.

6.6.3 Additional Comparisons

Comparison to Aittala et al. We first compare our technique to with one introduced by Aittala et al. [4] in Figure 6.9 using input photos published as supplemental materials from their work. Their work uses the same VGG-based loss (summary function), but optimizes directly in texture space. Both methods manage to reproduce the overall pattern and reflectance of the input photos. Thanks to the underlying procedural models, our method is able to synthesize larger results without visually obvious periodic patterns, and with more plausible global variation.

Comparison to neural methods. We also compare our method with the forward neural prediction method of Hu et al. [60]. Their method uses an AlexNet network structure [78], mapping an image of a material sample to the parameters of an appropriate procedural model. We apply their network structure with our BRDFs and lighting conditions, as their original implementation assumes Lambertian materials and outdoor sun/sky lighting. We

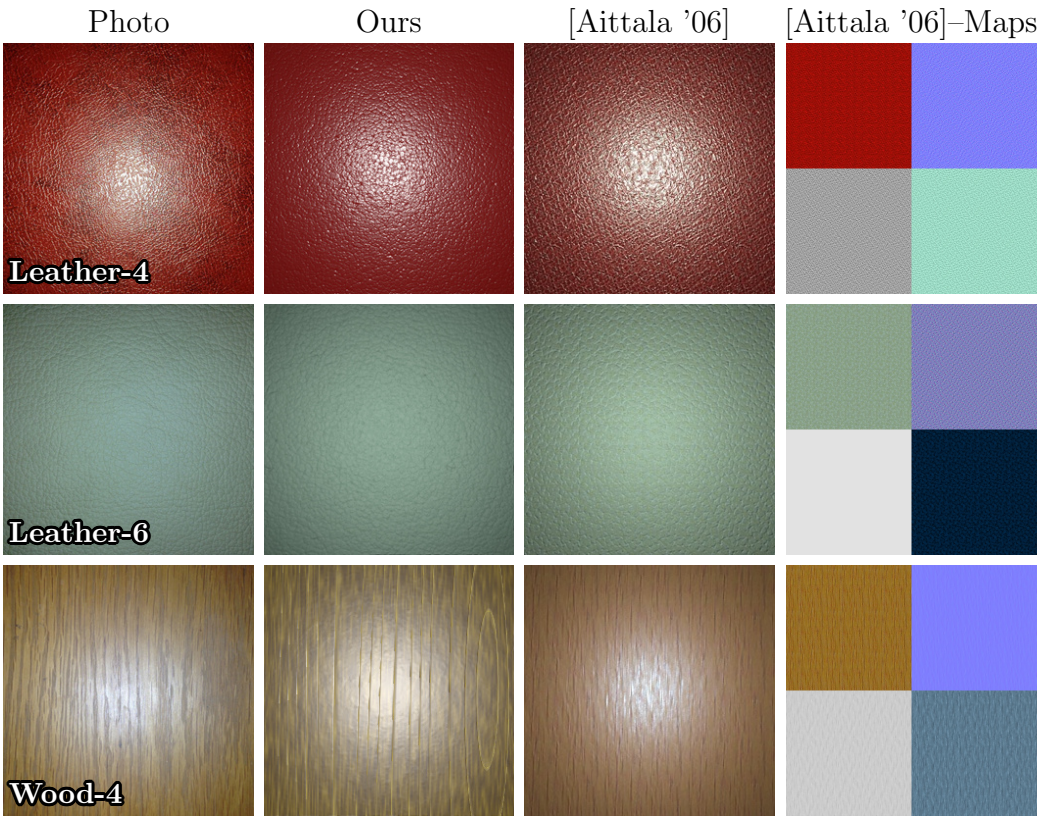


Figure 6.9: Comparison with the Aittala et al. [4]. Results in the third column are rendered using tiled versions of texture maps shown in the fourth column.

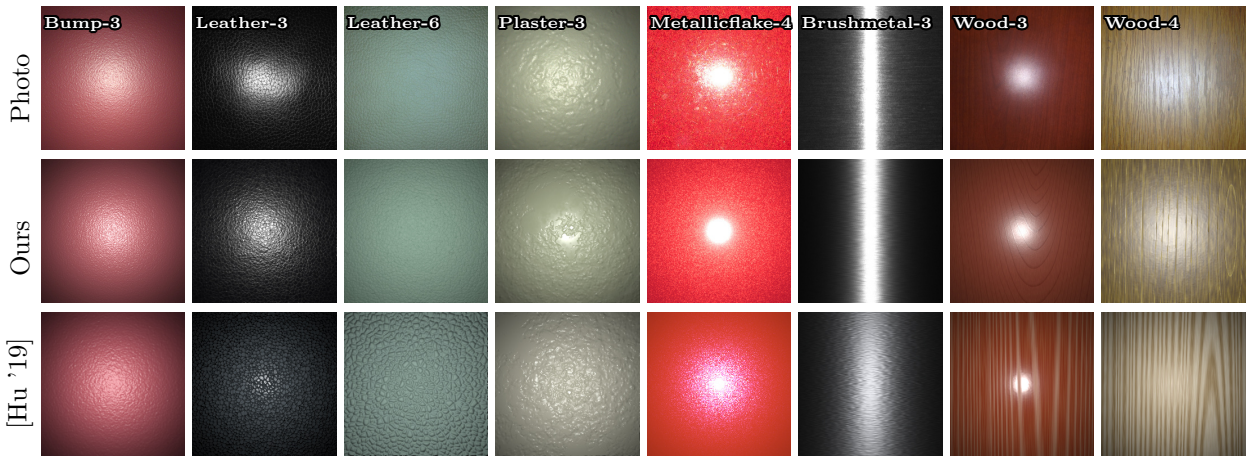


Figure 6.10: Comparison to the forward neural prediction method of Hu et al. [60], where we apply their network structure with our BRDFs and lighting conditions. The photo (top) is better matched by our MCMC sampling results (middle) than their prediction (bottom), which moreover tends to become worse for more complex BRDF models and with more parameters. On the other hand, [60] can be used as an efficient initialization of our sampling, as shown in Figure 6.11.

show the results in Figure 6.10. In general, we find the method gives moderately accurate results, which moreover tends to become worse for more complex BRDF models and with more parameters.

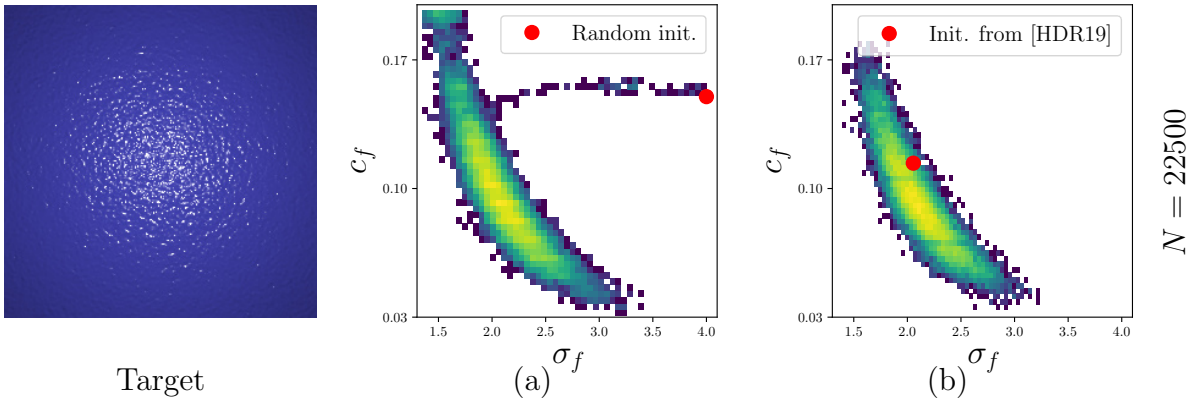


Figure 6.11: Initialization of our sampling with the method of Hu et al. [60] on a synthetic bumpy surface example. The figure shows joint posterior distributions over two parameters using different initializations: a random initialization drawn from the prior (a) and the prediction of [HDR19] (b). As we can see, starting from the result of [HDR19] can shorten the burn-in phase of the MCMC sampling process.

To some extent, the method of [60] is orthogonal to ours, as it can be used as an efficient initialization for our sampling. In Figure 6.11, we compare our MCMC sampling results with a random starting point to one using the result of Hu et al. for initialization. This reduces the burn-in period required by the MCMC method.

Finally, we also compare to the single input SVBRDF estimation method of Deschaintre et al. [22] (See Figure 6.12). This method takes a 256×256 target image, and produces material maps at the same resolution, pixel-wise aligned to the input. This pixel-wise alignment is not achievable with our (procedural) method. However, the overall perceptual appearance is usually worse than ours. In some cases (Plaster-4, Metallicflake-4), the method produces specular burn-in, as the strong highlight cannot be fully removed in the resulting maps. Advanced BRDF models like brushed metal and metallic flakes are not explicitly handled by their method and usually fail. Finally, their result does not support higher resolutions, seamless tiling, nor editability; these benefits come from the use of a procedural model.

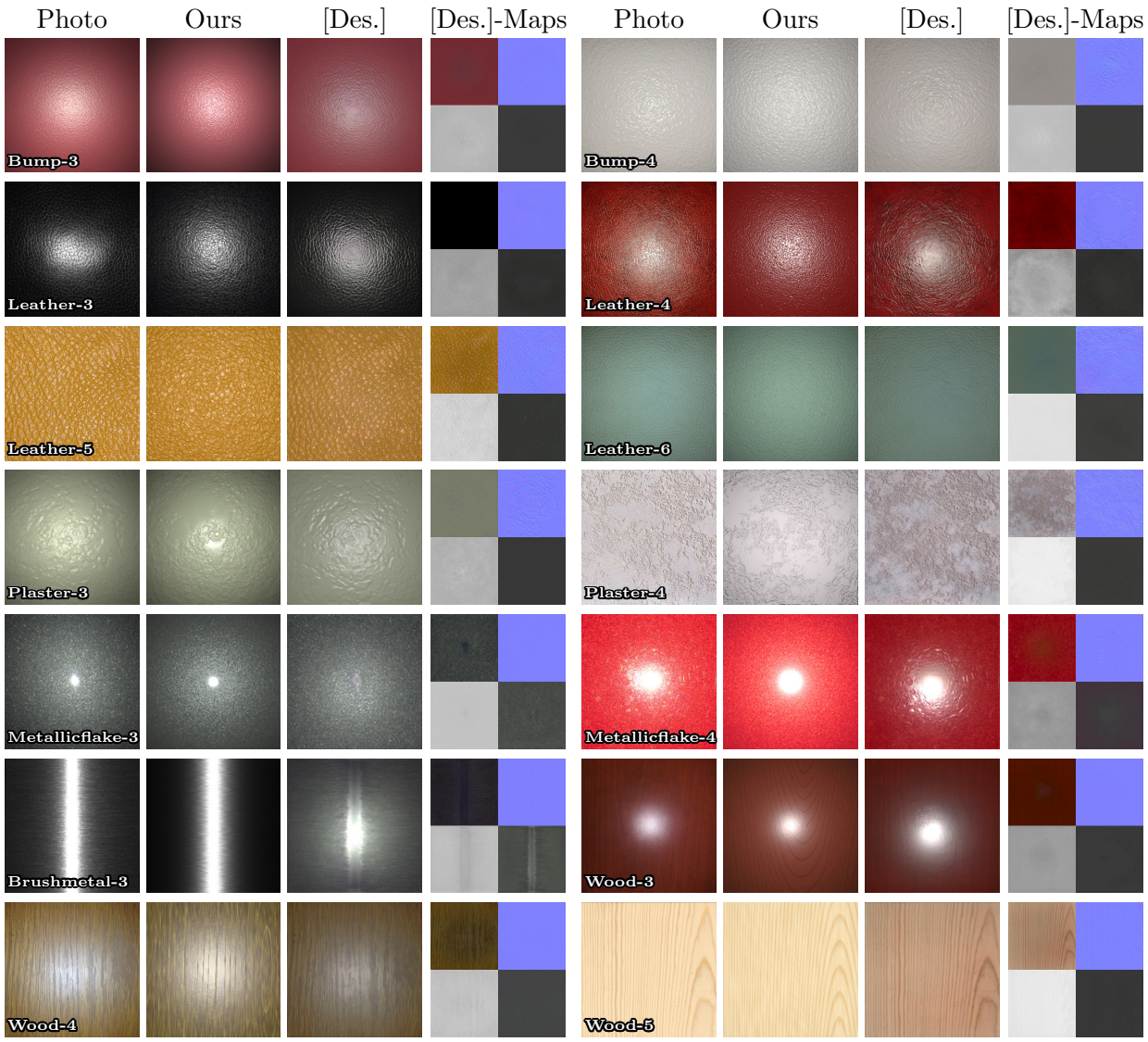


Figure 6.12: Comparison to the single input SVBRDF estimation method of Deschaintre et al. [22]. Due to the nature of the method, their texture patterns are closely aligned with the input image; however, the overall perceptual appearance match is usually worse than our method. In some cases, the method produces specular burn-in, as the strong highlight cannot be fully removed and causes holes in the resulting maps (Plaster-4, Metallicflake-4). Advanced BRDF models like brushed metal and metallic flakes are not explicitly handled by their method and usually fail.

Figure 6.13 shows a quantitative comparison of the Learned Perceptual Image Patch Similarity (LPIPS) metric [140] between the captured photos and the re-renderings using different methods.

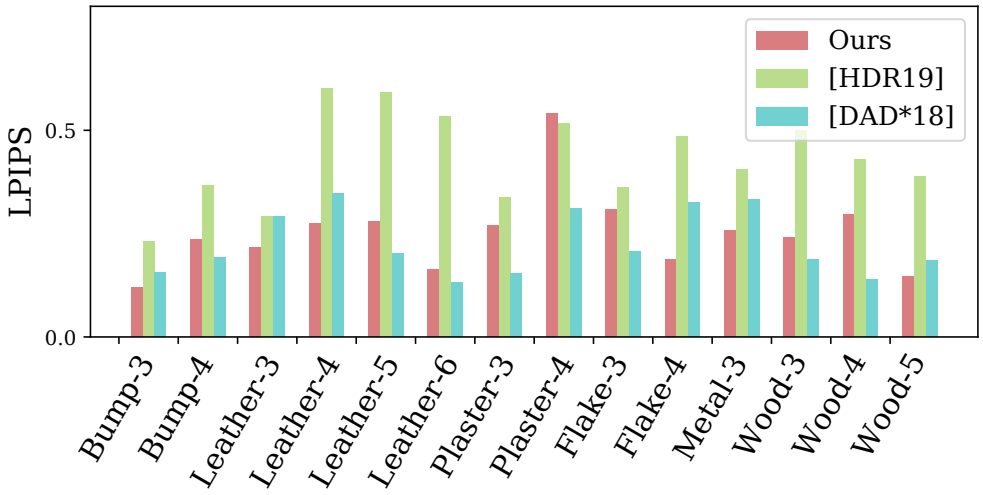


Figure 6.13: Quantitative evaluation. The LPIPS of our results are consistently better than [60]. Some LPIPS values from [22] are better than ours, since (as per-pixel methods) they can better match the noise patterns in the textures.

6.7 Conclusion

Procedural material models have become increasingly more popular in the industry, thanks to their flexibility, compactness, as well as easy editability. In this chapter, we introduced a new computational framework to solve the inverse problem: the inference of procedural model parameters based on a single input image.

The first major ingredient to our solution is a *Bayesian framework*, precisely defining the posterior distribution of the parameters, combining four components (priors, procedural material model, rendering operator, summary function). The second ingredient is an *Bayesian inference approach* that leverages MCMC sampling to sample posterior distributions of procedural material parameters. This technique enjoys the generality to handle both continuous and discrete model parameters and provides users additional information beyond single point estimates and allows a cleaner extension to handle discrete parameters.

Chapter 7

Conclusion and Future work

In the dissertation, we focus on material appearances modeling in both forward and inverse rendering. All the macro appearances are modeled from microscales or hyperparameter spaces.

First we have presented two scattering frameworks in forward rendering, one for layered materials (*thin planer surface*) and the other one for participating medium (*bulk,particles*). The first work **LayeredBSDF** provides a general solution to layered materials which is included in *Physically Based Rendering*, Fourth Edition [107]. It leads to the first BSDF layering solution that offers unbiased accuracy and full flexibility in setting the layer properties. Our second work **Beyond Mie Theory** generalizes the widely-used Lorenz-Mie theory for rigorously deriving optical properties of scattering media and can be readily used in any radiative-based light transport simulator.

Then we have estimated material properties using *latent space* and *procedural parameters*. Our third work **MaterialGAN** is the first step toward GAN-based material analysis by synthesis, and our experiments suggest many avenues for further exploration. Our last work **Bayesian Inference Sampling** handles both continuous and discrete model parameters

and provides users additional information beyond single-point estimates, and allows a cleaner extension to handle discrete parameters.

In the end, we will discuss the limitation of our work and future directions:

LayeredBSDF Our model relies on the assumption of thin flat layers (Figure 3.12) and cannot capture effects caused by geometric or optical variations at the global scale. Examples include internal caustics and shadowing arising from major normal variations and color bleeding caused by light scattering though media with varying colors. Generalizing our technique to include bidirectional subsurface scattering distribution functions (BSSRDFs) is an interesting further topic. In addition, as our model simulates subsurface scattering using Monte Carlo path tracing, the performance may degrade with the presence of optically thick layers with many scattering events. Using fast approximated solutions such as [69, 33] to capture multiple scattering may be a useful extension. Lastly, since we model light transport using traditional radiative transfer, wave effects such as thin film interference are not handled. An interesting challenge is to integrate wave optics into our model to accurately and efficiently handle light interference and phase shifts.

Beyond Mie Theory While taking into account the effect of the near-field on clusters, our work is still based on the RTT. Therefore it relies on the far-field approximation to represent a scattering dyad useful for rendering. Therefore, while we can handle near- and far-field scattering, we cannot accurately model the scattering in the intermediate region, which we treat as the far field. Using more accurate representations, that capture the effects at such near-field region could further enhance the generality of our theory and, thus, is an interesting future topic. This would however require exploring an alternative light transport framework beyond the RTT. Right now, our implementation requires precomputing the bulk optical properties of the media. This limits the applicability of our work to media with

homogeneous particle statistical properties. Finding faster approximations for our scattering functions, in the same spirit as the geometric optics approximation for Lorenz-Mie theory [41], is an interesting future research. Finally, our implementation is currently limited in practice to spherical particles with identical radii within a particle cluster. Allowing general and spatially varying particle shapes by using an alternative implementation of the T-matrix method would further improve the versatility of our technique.

MaterialGAN Our current BRDF model is shared by previous work, but certain common effects (layering on book covers, subsurface fiber scattering in woods, anisotropy in fabrics) are not correctly captured by it. An extension of our generative model and rendering operator would be possible, though the key challenge is finding high-quality training data for these effects. Our assumption of almost flat samples will fail for materials with strong relief patterns, and will produce blurring or ghosting if there are obvious parallax effects in the aligned captured images. Strong self-shadowing or inter-reflections are also not currently handled. Solving for height instead of normal, with a more advanced rendering operator, may be able to resolve parallax effects and to correctly predict (and undo) shadowing effects from strong height variations. Furthermore, more precise calibration may improve our accuracy. This would likely require knowledge of the cell phone hardware, and/or pre-calibration of its properties (e.g. flash light falloff, lens vignetting, and color processing properties). The resolution of our result can be increased with a coarse-to-fine post-process, since we have a fairly good result as the initialization of next level of resolution.

Bayesian Inference Sampling In the future, we would like to increase the complexity of the models supported even further, to handle materials like woven fabrics, transmissive BTDFs, and more. Finally, extensions to our approach could be used to estimate parameters of procedural models beyond materials, including geometry and lighting, as long as the parameters could be differentiated.

Bibliography

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to Embed Images into the StyleGAN Latent Space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to Edit the Embedded Images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020.

[3] Adobe. Substance. `www.substance3d.com`, 2019.

[4] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance Modeling by Neural Texture Synthesis. *ACM Transactions on Graphics*, 35(4):1–13, 2016.

[5] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Practical SVBRDF Capture in the Frequency Domain. *ACM Transactions on Graphics*, 32(4):1–12, 2013.

[6] Miika Aittala, Tim Weyrich, Jaakko Lehtinen, et al. Two-shot SVBRDF Capture for Stationary Materials. *ACM Transactions on Graphics*, 34(4):1–13, 2015.

[7] Carlos Aliaga, Carlos Castillo, Diego Gutierrez, Miguel A Otaduy, Jorge Lopez-Moreno, and Adrian Jarabo. An Appearance Model for Textile Fibers. *Computer Graphics Forum*, 36(4):35–45, 2017.

[8] Marco Ament, Christoph Bergmann, and Daniel Weiskopf. Refractive Radiative Transfer Equation. *ACM Transactions on Graphics*, 33(2):1–22, 2014.

[9] Muhammad Asim, Max Daniels, Oscar Leong, Ali Ahmed, and Paul Hand. Invertible Generative Models for Inverse Problems: Mitigating Representation Error and Dataset Bias. In *International Conference on Machine Learning*, pages 399–409, 2020.

[10] Chen Bar, Marina Alterman, Ioannis Gkioulekas, and Anat Levin. A Monte Carlo Framework for Rendering Speckle Statistics in Scattering Media. *ACM Transactions on Graphics*, 38(4):1–22, 2019.

[11] Chen Bar, Ioannis Gkioulekas, and Anat Levin. Rendering Near-field Speckle Statistics in Scattering Media. *ACM Transactions on Graphics*, 39(6):1–18, 2020.

[12] Laurent Belcour. Efficient Rendering of Layered Materials Using an Atomic Decomposition with Statistical Operators. *ACM Transactions on Graphics*, 37(4):1–15, 2018.

[13] Laurent Belcour and Pascal Barla. A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence. *ACM Transactions on Graphics*, 36(4):1–14, 2017.

[14] Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[15] Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. A Radiative Transfer Framework for Non-exponential Media. *ACM Transactions on Graphics*, 37(6):225, 2018.

[16] Craig F Bohren and Donald R Huffman. *Absorption and Scattering of Light by Small Particles*. John Wiley & Sons, 2008.

[17] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed Sensing Using Generative Models. In *International Conference on Machine Learning*, pages 537–546, 2017.

[18] Subrahmanyan Chandrasekhar. *Radiative Transfer*. Courier Corporation, 1960.

[19] Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the Gap Between Stochastic Gradient MCMC and Stochastic Optimization. In *Artificial Intelligence and Statistics*, pages 1051–1060, 2016.

[20] Robert L Cook and Kenneth E. Torrance. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.

[21] Tom Cuypers, Tom Haber, Philippe Bekaert, Se Baek Oh, and Ramesh Raskar. Reflectance Model for Diffraction. *ACM Transactions on Graphics*, 31(5):1–11, 2012.

[22] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF Capture with a Rendering-aware Deep Network. *ACM Transactions on Graphics*, 37(4):1–15, 2018.

[23] Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. Flexible SVBRDF Capture with a Multi-image Deep Network. *Computer Graphics Forum*, 38(4):1–13, 2019.

[24] Chris Donahue, Julian McAuley, and Miller Puckette. Synthesizing Audio with Generative Adversarial Networks. *arXiv preprint arXiv:1802.04208*, 1, 2018.

[25] Yue Dong. Deep Appearance Modeling: A Survey. *Visual Informatics*, 3(2):59–68, 2019.

[26] Zhao Dong, Bruce Walter, Steve Marschner, and Donald P Greenberg. Predicting Appearance from Measured Microgeometry of Metal Surfaces. *ACM Transactions on Graphics*, 35(1):1–13, 2015.

[27] Craig Donner, Tim Weyrich, Eugene d'Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz. A Layered Heterogeneous Reflectance Model for Acquiring and Rendering Human Skin. *ACM Transactions on Graphics*, 27(5):1–12, 2008.

[28] Amos Egel, Lorenzo Pattelli, Giacomo Mazzamuto, Diederik S Wiersma, and Uli Lemmer. CELES: CUDA-accelerated Simulation of Electromagnetic Scattering by Large Ensembles of Spheres. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 199:103–110, 2017.

[29] Viggo Falster, Adrian Jarabo, and Jeppe Revall Frisvad. Computing the Bidirectional Scattering of a Microstructure Using Scalar Diffraction Theory and Path Tracing. *Computer Graphics Forum*, 39(7):231–242, 2020.

[30] Leslie L Foldy. The Multiple Scattering of Waves. I. General Theory of Isotropic Scattering by Randomly Distributed Scatterers. *Physical review*, 67(3-4):107, 1945.

[31] Yannick Francken, Tom Cuypers, Tom Mertens, and Philippe Bekaert. Gloss and Normal Map Acquisition of Mesostructures using Gray Codes. In *International Symposium on Visual Computing*, pages 788–798, 2009.

[32] Jeppe Revall Frisvad, Niels Jørgen Christensen, and Henrik Wann Jensen. Computing the Scattering Properties of Participating Media Using Lorenz-Mie Theory. *ACM Transactions on Graphics*, 26(3):60–es, 2007.

[33] Jeppe Revall Frisvad, Toshiya Hachisuka, and Thomas Kim Kjeldsen. Directional Dipole Model for Subsurface Scattering. *ACM Transactions on Graphics*, 34(1):1–12, 2014.

[34] Jeppe Revall Frisvad, Søren Alkærsig Jensen, Jonas Skovlund Madsen, Antônio Correia, Li Yang, SØren Kimmer Schou Gregersen, Youri Meuret, and P-E Hansen. Survey of Models for Acquiring the Optical Properties of Translucent Materials. *Computer Graphics Forum*, 39(2):729–755, 2020.

[35] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep Inverse Rendering for High-resolution SVBRDF Estimation from an Arbitrary Number of Images. *ACM Transactions on Graphics*, 38(4):1–15, 2019.

[36] Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. Linear Light Source Reflectometry. *ACM Transactions on Graphics*, 22(3):749–758, 2003.

[37] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A Neural Algorithm of Artistic Style. *arXiv preprint arXiv:1508.06576*, 2015.

[38] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image Style Transfer using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[39] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A Wilson, and Paul Debevec. Estimating Specular Roughness and Anisotropy from Second Order Spherical Gradient Illumination. *Computer Graphics Forum*, 28(4):1161–1170, 2009.

[40] Ioannis Gkioulekas, Bei Xiao, Shuang Zhao, Edward H Adelson, Todd Zickler, and Kavita Bala. Understanding the Role of Phase Function in Translucent Appearance. *ACM Transactions on Graphics*, 32(5):1–19, 2013.

[41] Werner J Glantschnig and Sow-Hsin Chen. Light Scattering from Water Droplets in the Geometrical Optics Approximation. *Applied Optics*, 20(14):2499–2509, 1981.

[42] Jay S Gondek, Gary W Meyer, and Jonathan G Newman. Wavelength Dependent Reflectance Functions. In *SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 213–220, 1994.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[44] Darya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. BRDF Representation and Acquisition. *Computer Graphics Forum*, 35(2):625–650, 2016.

[45] Ibón Guillén, Julio Marco, Diego Gutierrez, Wenzel Jakob, and Adrian Jarabo. A General Framework for Pearlescent Materials. *ACM Transactions on Graphics*, 39(6):1–15, 2020.

[46] Jie Guo, Bingyang Hu, Yanjun Chen, Yuanqi Li, Yanwen Guo, and Ling-Qi Yan. Rendering Discrete Participating Media with Geometrical Optics Approximation. *arXiv preprint arXiv:2102.12285*, 2021.

[47] Jie Guo, Jinghui Qian, Yanwen Guo, and Jingui Pan. Rendering Thin Transparent Layers with Extended Normal Distribution Functions. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2108–2119, 2016.

[48] Yu Guo, Miloš Hašan, Lingqi Yan, and Shuang Zhao. A Bayesian Inference Framework for Procedural Material Parameter Estimation. *Computer Graphics Forum*, 39(7):255–266, 2020.

[49] Yu Guo, Miloš Hašan, and Shuang Zhao. Position-free Monte Carlo Simulation for Arbitrary Layered BSDFs. *ACM Transactions on Graphics*, 37(6):1–14, 2018.

[50] Yu Guo, Adrian Jarabo, and Shuang Zhao. Beyond Mie Theory: Systematic Computation of Bulk Scattering Parameters based on Microphysical Wave Optics. *ACM Transactions on Graphics*, 40(6):1–12, 2021.

[51] Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. MaterialGAN: Reflectance Capture Using a Generative SVBRDF Model. *ACM Transactions on Graphics*, 39(6):1–13, 2020.

[52] Diego Gutierrez, Francisco J Seron, Adolfo Munoz, and Oscar Anson. Visualizing Underwater Ocean Optics. *Computer Graphics Forum*, 27(2):547–556, 2008.

[53] Pat Hanrahan and Wolfgang Krueger. Reflection from Layered Surfaces Due to Subsurface Scattering. In *SIGGRPAH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 165–174, 1993.

[54] W. K. Hastings. Monte Carlo Sampling Methods using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

[55] Xiao D He, Kenneth E Torrance, Francois X Sillion, and Donald P Greenberg. A Comprehensive Physical Model for Light Reflection. *ACM SIGGRAPH Computer Graphics*, 25(4):175–186, 1991.

[56] Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. The SGGX microflake distribution. *ACM Transactions on Graphics*, 34(4):1–11, 2015.

[57] Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-scattering Microfacet BSDFs With the Smith Model. *ACM Transactions on Graphics*, 35(4):1–14, 2016.

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

[59] Nicolas Holzschuch and Romain Pacanowski. A Two-scale Microfacet Reflectance Model Combining Reflection and Diffraction. *ACM Transactions on Graphics*, 36(4):1–12, 2017.

[60] Yiwei Hu, Julie Dorsey, and Holly Rushmeier. A Novel Framework for Inverse Procedural Texture Modeling. *ACM Transactions on Graphics*, 38(6):1–14, 2019.

[61] Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time With Adaptive Instance Normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[62] Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, Jian Wang, and Aswin C Sankaranarayanan. Reflectance Capture using Univariate Sampling of BRDFs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5362–5370, 2017.

[63] Dietmar Jackel and Bruce Walter. Modeling and Rendering of the Atmosphere Using Mie-scattering. *Computer Graphics Forum*, 16(4):201–210, 1997.

[64] Wenzel Jakob. Mitsuba Renderer. `http://www.mitsuba-renderer.org`, 2010.

[65] Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. A Radiative Transfer Framework for Rendering Materials with Anisotropic Structure. *ACM Transactions on Graphics*, 29(4):1–13, 2010.

[66] Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. A Comprehensive Framework for Rendering Layered Materials. *ACM Transactions on Graphics*, 33(4):1–14, 2014.

[67] Adrian Jarabo, Carlos Aliaga, and Diego Gutierrez. A Radiative Transfer Framework for Spatially-correlated Materials. *ACM Transactions on Graphics*, 37(4):1–13, 2018.

[68] Adrian Jarabo and Victor Arellano. Bidirectional Rendering of Vector Light Transport. *Computer Graphics Forum*, 37(6):96–105, 2018.

[69] Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. A Practical Model for Subsurface Light Transport. In *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 511–518, 2001.

[70] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-time Style Transfer and Super-resolution. In *Proceedings of European conference on computer vision*, pages 694–711, 2016.

[71] James T Kajiya. The Rendering Equation. In *SIGGRAPH '86: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 143–150, 1986.

[72] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Proceedings of International Conference on Learning Representations*, 2018.

[73] Tero Karras, Samuli Laine, and Timo Aila. A Style-based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[74] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of Stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[75] Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A Simple and Robust Mutation Strategy for the Metropolis Light Transport Algorithm. *Computer Graphics Forum*, 21(3):531–540, 2002.

[76] Pramook Khungurn, Daniel Schroeder, Shuang Zhao, Kavita Bala, and Steve Marschner. Matching Real Fabrics with Micro-Appearance Models. *ACM Transactions on Graphics*, 35(1):1–26, 2015.

[77] David Koerner, Jan Novák, Peter Kutz, Ralf Habel, and Wojciech Jarosz. Subdivision Next-event Estimation for Path-traced Subsurface Scattering. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*, pages 91–96, 2016.

[78] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[79] Tejas D Kulkarni, Pushmeet Kohli, Joshua B Tenenbaum, and Vikash Mansinghka. Picture: A Probabilistic Programming Language for Scene Perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4390–4399, 2015.

[80] Philip Laven. MiePlot. `http://www.philiplaven.com/mieplot.htm`, 2011.

[81] Melvin Lax. Multiple Scattering of Waves. *Reviews of Modern Physics*, 23(4):287, 1951.

[82] Jonathan Leaf, Rundong Wu, Eston Schweickart, Doug L James, and Steve Marschner. Interactive Design of Periodic Yarn-level Cloth Patterns. *ACM Transactions on Graphics*, 37(6):1–15, 2018.

[83] Tzu-Mao Li, Jaakko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. Anisotropic Gaussian Mutations for Metropolis Light Transport through Hessian-Hamiltonian Dynamics. *ACM Transactions on Graphics*, 34(6):1–13, 2015.

[84] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling Surface Appearance from a Single Photograph using Self-augmented Convolutional Neural Networks. *ACM Transactions on Graphics*, 36(4):1–11, 2017.

[85] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Synthesizing 3d Shapes from Silhouette Image Collections using Multi-projection Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5535–5544, 2019.

[86] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *Proceedings of the European Conference on Computer Vision*, pages 72–87, 2018.

[87] Albert Julius Liu, Zhao Dong, Miloš Hašan, and Steve Marschner. Simulating the Structure and Texture of Solid Wood. *ACM Transactions on Graphics*, 35(6):1–11, 2016.

[88] Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Langevin Monte Carlo Rendering with Gradient-based Adaptation. *ACM Transactions on Graphics*, 39(4):1–14, 2020.

[89] Daniel W Mackowski and Michael I Mishchenko. Calculation of the T Matrix and the Scattering Matrix for Ensembles of Spheres. *JOSA A*, 13(11):2266–2278, 1996.

[90] Daniel W Mackowski and Michael I Mishchenko. A multiple sphere T-matrix Fortran code for use on parallel computer clusters. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 112(13):2182–2192, 2011.

[91] Stephen R Marschner, Stephen H Westin, Eric PF Lafortune, Kenneth E Torrance, and Donald P Greenberg. Image-based BRDF measurement including human skin. In *Eurographics Workshop on Rendering Techniques*, pages 131–144. Springer, 1999.

[92] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A Data-driven Reflectance Model. *ACM Transactions on Graphics*, 22(3):759–769, 2003.

[93] Johannes Meng, Marios Papas, Ralf Habel, Carsten Dachsbacher, Steve Marschner, Markus H Gross, and Wojciech Jarosz. Multi-scale Modeling and Rendering of Granular Materials. *ACM Transactions on Graphics*, 34(4):49–1, 2015.

[94] Michael I Mishchenko. Vector Radiative Transfer Equation for Arbitrarily Shaped and Arbitrarily Oriented Particles: A Microphysical Derivation from Statistical Electromagnetics. *Applied optics*, 41(33):7114–7134, 2002.

[95] Michael I Mishchenko. 125 Years of Radiative Transfer: Enduring Triumphs and Persisting Misconceptions. In *AIP Conference Proceedings*, volume 1531, pages 11–18, 2013.

[96] Michael I Mishchenko. *Electromagnetic Scattering by Particles and Particle Groups: An Introduction*. Cambridge University Press, 2014.

[97] Michael I Mishchenko, Larry D Travis, and Andrew A Lacis. *Multiple Scattering of Light by Particles: Radiative Transfer and Coherent Backscattering*. Cambridge University Press, 2006.

[98] Hans P Moravec. 3d Graphics and the Wave Theory. In *SIGGRAPH '81: Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques*, pages 289–296, 1981.

[99] Thomas Müller, Marios Papas, Markus Gross, Wojciech Jarosz, and Jan Novák. Efficient Rendering of Heterogeneous Polydisperse Granular Media. *ACM Transactions on Graphics*, 35(6):1–14, 2016.

[100] A Musbach, GW Meyer, F Reitich, and SH Oh. Full Wave Modelling of Light Propagation and Reflection. *Computer Graphics Forum*, 32(6):24–37, 2013.

[101] Radford M Neal et al. MCMC using Hamiltonian Dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

[102] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental Analysis of BRDF Models. In *Proceedings of the 16th Eurographics conference on Rendering Techniques*, pages 117–126, 2005.

[103] Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. Monte Carlo Methods for Volumetric Light Transport Simulation. *Computer Graphics Forum*, 37(2):551–576, 2018.

[104] Daniel O'Malley, John K Golden, and Velimir V Vesselinov. Learning to Regularize with a Variational Autoencoder for Hydrologic Inverse Analysis. *arXiv preprint arXiv:1906.02401*, 2019.

[105] Lars Otten. LaTeX Template for Thesis and Dissertation Documents at UC Irvine. `https://github.com/lotten/uci-thesis-latex/`, 2012.

[106] Bo Peterson and Staffan Ström. T Matrix for Electromagnetic Scattering from an Arbitrary Number of Scatterers and Representations of E(3). *Physical review D*, 8(10):3661, 1973.

[107] Matt Pharr, Wenzel Jakob, and Greg Humphreys. Physically Based Rendering: From Theory to Implementation. `https://github.com/mmp/pbrt-v4`, 2021.

[108] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*, 2015.

[109] Peiran Ren, Jiaping Wang, John Snyder, Xin Tong, and Baining Guo. Pocket Reflectometry. *ACM Transactions on Graphics*, 30(4):1–10, 2011.

[110] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and Controllable Neural Texture Synthesis and Style Transfer using Histogram Losses. *arXiv preprint arXiv:1701.08893*, 2017.

[111] Gareth O Roberts and Richard L Tweedie. Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, pages 341–363, 1996.

[112] Iman Sadeghi, Adolfo Munoz, Philip Laven, Wojciech Jarosz, Francisco Seron, Diego Gutierrez, and Henrik Wann Jensen. Physically-based Simulation of Rainbows. *ACM Transactions on Graphics*, 31(1):1–12, 2012.

[113] Vincent Schüssler, Eric Heitz, Johannes Hanika, and Carsten Dachsbacher. Microfacet-based Normal Mapping for Robust Monte Carlo Path Tracing. *ACM Transactions on Graphics*, 36(6):1–12, 2017.

[114] Raymond A Shaw, Alexander B Kostinski, and Daniel D Lanterman. Super-exponential Extinction of Radiation in a Negatively Correlated Random Medium. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 75(1):13–20, 2002.

[115] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[116] Brian E Smits and Gary W Meyer. Newton's Colors: Simulating Interference Phenomena in Realistic Image Synthesis. *Photorealism in Computer Graphics*, pages 185–194, 1992.

[117] Jos Stam. Diffraction Shaders. In *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 101–110, 1999.

[118] Jos Stam. An Illumination Model for a Skin Layer Bounded by Rough Surfaces. In *Eurographics Workshop on Rendering Techniques*, pages 39–52, 2001.

[119] Shlomi Steinberg. Analytic Spectral Integration of Birefringence-Induced Iridescence. *Computer Graphics Forum*, 38(4):97–110, 2019.

[120] Antoine Toisoul and Abhijeet Ghosh. Practical Acquisition and Rendering of Diffraction Effects in Surface Reflectance. *ACM Transactions on Graphics*, 36(5):1–16, 2017.

[121] Leung Tsang, Jin Au Kong, and Robert T Shin. *Theory of Microwave Remote Sensing*. John Wiley & Sons, 1985.

[122] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing Motion and Content for Video Generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[123] Hendrik Christoffel van der Hulst. *Light Scattering by Small Particles*. Courier Corporation, 1981.

[124] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.

[125] Eric Veach and Leonidas J Guibas. Metropolis light transport. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76, 1997.

[126] Eugene von Lommel. Die Photometrie der Diffusen Zurückwerfung. *Annalen der Physik*, 272(2):473–502, 1889.

[127] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet Models for Refraction Through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, pages 195–206, 2007.

[128] Bruce Walter, Shuang Zhao, Nicolas Holzschuch, and Kavita Bala. Single Scattering in Refractive Media with Triangle Mesh Boundaries. *ACM Transactions on Graphics*, 28(3):1–8, 2009.

[129] Chun-Po Wang, Noah Snavely, and Steve Marschner. Estimating Dual-scale Properties of Glossy Surfaces from Step-edge Lighting. *ACM Transactions on Graphics*, 30(6):1–12, 2011.

[130] PC Waterman. Matrix Formulation of Electromagnetic Scattering. *Proceedings of the IEEE*, 53(8):805–812, 1965.

[131] Andrea Weidlich and Alexander Wilkie. Arbitrarily Layered Micro-facet Surfaces. In *GRAPHITE '07: Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pages 171–178, 2007.

[132] Sebastian Werner, Zdravko Velinov, Wenzel Jakob, and Matthias B Hullin. Scratch Iridescence: Wave-optical Rendering of Diffractive Surface Structure. *ACM Transactions on Graphics*, 36(6):1–14, 2017.

[133] Tim Weyrich, Jason Lawrence, Hendrik PA Lensch, Szymon Rusinkiewicz, and Todd Zickler. *Principles of Appearance Acquisition and Representation*. Now Publishers Inc, 2009.

[134] Alexander Wilkie, Robert F Tobler, and Werner Purgathofer. Combined Rendering of Polarization and Fluorescence Effects. In *Eurographics Workshop on Rendering Techniques*, pages 197–204, 2001.

[135] Te-Kao Wu and L Tsai. Scattering by Arbitrarily Cross-sectioned Layered, Lossy Dielectric Cylinders. *IEEE Transactions on Antennas and Propagation*, 25(4):518–524, 1977.

[136] Mengqi Xia, Bruce Walter, Eric Michielssen, David Bindel, and Steve Marschner. A Wave Optics Based Fiber Scattering Model. *ACM Transactions on Graphics*, 39(6):1–16, 2020.

[137] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. Minimal BRDF Sampling for Two-shot Near-field Reflectance Acquisition. *ACM Transactions on Graphics*, 35(6):1–12, 2016.

[138] Ling-Qi Yan, Miloš Hašan, Bruce Walter, Steve Marschner, and Ravi Ramamoorthi. Rendering Specular Microgeometry with Wave Optics. *ACM Transactions on Graphics*, 37(4):1–10, 2018.

[139] Tizian Zeltner and Wenzel Jakob. The Layer Laboratory: A Calculus for Additive and Subtractive Composition of Anisotropic Surface Reflectance. *ACM Transactions on Graphics*, 37(4):1–14, 2018.

[140] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[141] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Building Volumetric Appearance Models of Fabric Using Micro CT Imaging. *ACM Transactions on Graphics*, 30(4):1–10, 2011.

[142] Shuang Zhao, Fujun Luan, and Kavita Bala. Fitting Procedural Yarn Models for Realistic Cloth Rendering. *ACM Transactions on Graphics*, 35(4):1–11, 2016.

[143] Shuang Zhao, Ravi Ramamoorthi, and Kavita Bala. High-order Similarity Relations in Radiative Transfer. *ACM Transactions on Graphics*, 33(4):1–12, 2014.

[144] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative Visual Manipulation on the Natural Image Manifold. In *Proceedings of European conference on computer vision*, pages 597–613, 2016.

Appendix A

Appendix for Chapter 3

A.1 Detailed Derivations

We now provide detailed derivations for the key equations in §3.4.

Position-free radiative transfer equation. Traditionally, the radiative transfer equation (RTE) involves an integral over free-flight distance t:

$$L_v(z, \boldsymbol{\omega}) = S(z, \boldsymbol{\omega}) + \int_0^{t'} \exp(-t\sigma_{\mathrm{t}}) \int_{\mathbb{S}^2} \hat{f}_p(\boldsymbol{\omega}', \boldsymbol{\omega})\, L_v(z', \boldsymbol{\omega}')\, \mathrm{d}\boldsymbol{\omega}'\, \mathrm{d}t, \tag{A.1}$$

where $z' := z - t\cos\boldsymbol{\omega}$ and t' denotes the distance between z and the closest layer boundary. Since $t = (z - z')/\cos\boldsymbol{\omega}$, changing the integration variable from t to z' in Eq. (A.1) yields an additional factor of $(\cos\boldsymbol{\omega})^{-1}$ which in turn gives our position-free RTE (3.9). Notice that the change-of-variable ratio only appears within the integration (and not in the source term S).

Cosines in path contribution. The contribution f of a light path $\bar{x}$ can be obtained by repeatedly expanding the rendering equation (3.11) and our position-free RTE (3.9).

Similar to the traditional path integral formulation, for each vertex z_i corresponding to an interface event (i.e., reflection or refraction), a cosine term $|\cos \boldsymbol{d}_i|$ is needed to ensure the measure of projected solid angle.

On the other hand, a segment of our light path connecting two depths z_i and z_{i+1} via direction $\boldsymbol{d}_i$ can yield an additional $|\cos \boldsymbol{d}_i|^{-1}$ when z_{i+1} corresponds to a volumetric scattering. Thus, for each i, the path contribution involve a factor of $|\cos \boldsymbol{d}_i|^{\alpha_i}$ with:

- $\alpha_i = 1$ if z_i and z_{i+1} are both on interfaces;
- $\alpha_i = 0$ (i.e., no $\cos \boldsymbol{d}_i$ term) if (i) z_i is volumetric and z_{i+1} lies on an interface (so that no $\cos \boldsymbol{d}_i$ terms appear during expansion), or (ii) z_i is interfacial and z_{i+1} is volumetric (so that both $|\cos \boldsymbol{d}_i|$ and $|\cos \boldsymbol{d}_i|^{-1}$ are present, canceling out each other);
- $\alpha_i = -1$ if z_i and z_{i+1} are both volumetric vertices.

Eq. (3.6) provides a compact way to encode these rules.

A.2 Efficient Weight Computation

Weights of Light Transport Paths. Given a light subpath $\bar{x}_i$ and a camera subpath $\bar{x}_o$ with n_i and n_o vertices respectively, our bidirectional estimator combines $2n_in_o$ estimators of the form $f(\bar{y}^{(u)}_{s,t})/p^{(u)}_{s,t}(\bar{y}^{(u)}_{s,t})$ with $s \in \{1, 2, \ldots, n_i\}$, $t \in \{1, 2, \ldots, n_o\}$, and $u \in \{0, 1\}$ via the multiple importance sampling (MIS) framework. This yields a combined estimator:

$$\sum_{s=1}^{n_i} \sum_{t=1}^{n_o} \sum_{u=0}^{1} w^{(u)}_{s,t}(\bar{y}^{(u)}_{s,t}) \frac{f(\bar{y}^{(u)}_{s,t})}{p^{(u)}_{s,t}(\bar{y}^{(u)}_{s,t})}, \tag{A.2}$$

where the weight $w_{s,t}^{(u)}$, when using the balanced heuristics [Veach 1997], is given by

$$w_{s,t}^{(u)}(\bar{y}_{s,t}) = \left(\sum_{s'=1}^{s+t-1} \sum_{u'=0}^{1} \frac{p_{s',s+t-s'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} \right)^{-1} \tag{A.3}$$

for any path $\bar{y}_{s,t}$ with $(s+t)$ vertices.

Notice that, compared to standard bidirectional path tracing that combines $n_i n_o$ estimators, our position-free formulation offers twice the number of estimators since the direction connecting two depths is not unique.

Efficient Weight Computation. Computing Eqs. (A.2) and (A.3) for all s and t naïvely has a time complexity of $O(n_i n_o(n_i+n_o))$ and is too slow to be practical. We now present our method that runs in $O(n_i n_o)$ time. Our approach is conceptually similar to Veach's method for standard BDPT but differs in the exact mathematical form due to our position-free path formulation (see §3.5).

Let $\bar{y}_{s,t} = (\boldsymbol{d}_0, z_1, \boldsymbol{d}_1, \ldots, z_n, \boldsymbol{d}_n)$ with $n = s+t$. For all $s', t' \in \{1, 2, \ldots, n\}$, define

$$p_{s'}^{(0)} := \prod_{i=1}^{s'-1} p(\boldsymbol{d}_i \mid z_i, \boldsymbol{d}_{i-1})\, p(z_{i+1} \mid z_i, \boldsymbol{d}_i), \tag{A.4}$$

$$p_{t'}^{(1)} := \prod_{i=n-t'+1}^{n-1} p(-\boldsymbol{d}_i \mid z_{i+1}, -\boldsymbol{d}_{i+1})\, p(z_i \mid z_{i+1}, -\boldsymbol{d}_i), \tag{A.5}$$

which denote the probability for constructing two subpaths containing the first s' and last t' vertices of $\bar{y}$, respectively. Then, for all u', s' and t', it holds that

$$p_{s',t'}^{(u')}(\bar{y}_{s,t}) = p_{s'}^{(0)}\, p_{t'}^{(1)}\, q_{s'}^{(u')}, \tag{A.6}$$

where

$$q_{s'}^{(u')} := \begin{cases} p(\boldsymbol{d}_{s'} \mid z_{s'}, \boldsymbol{d}_{s'-1}) & \text{if } u = 0, \\ p(-\boldsymbol{d}_{s'} \mid z_{s'+1}, -\boldsymbol{d}_{s'+1}) & \text{if } u = 1. \end{cases} \tag{A.7}$$

It follows that

$$\frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{s'}^{(0)}\, p_{t'}^{(1)}\, q_{s'}^{(u')}}{p_{s}^{(0)}\, p_{t}^{(1)}\, q_{s}^{(u)}}. \tag{A.8}$$

Note that, for any $s' < s$, we have

$$p_{s',t'}^{(u')}(\bar{y}_{s,t}) = p_{s'}^{(0)}\, \frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}}\, q_{s'}^{(u')}\, p_{t+1}^{(1)}. \tag{A.9}$$

It follows that

$$\sum_{s'=1}^{s-1} \sum_{u'=0}^{1} \frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{t+1}^{(1)}}{p_{t}^{(1)}\, q_{s}^{(u)}} \underbrace{\sum_{s'=1}^{s-1} \sum_{u'=0}^{1} \frac{p_{s'}^{(0)}\, \frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}} q_{s'}^{(u')}}{p_{s}^{(0)}}}_{=:\, P_s^{(0)}}. \tag{A.10}$$

Since

$$\frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}} = \prod_{i=s'+1}^{s-1} p(-\boldsymbol{d}_i \mid z_{i+1}, -\boldsymbol{d}_{i+1})\, p(z_i \mid z_{i+1}, -\boldsymbol{d}_i), \tag{A.11}$$

it is easy to verify that $P_s^{(0)}$ depends only on depths $z_{s'}$ and directions $\boldsymbol{d}_{s'}$ with $s' \le s$, which are all from the subpath $\bar{x}_i$. Further, $P_{s'}^{(0)}$ remains constant for all paths $\bar{y}_{s,t}$ with $s > s'$. This allows us to precompute $P_s^{(0)}$ using $\bar{x}_i$ for $s = 1, 2, \ldots, n_i$. To this end, $P_s^{(0)}(\bar{y})$ can be efficiently evaluated using the following relation:

$$P_s^{(0)} = \begin{cases} 0 & (s = 0), \\ \frac{p_{s-1}^{(0)}}{p_s^{(0)}} \left(P_{s-1}^{(0)} \frac{p_{t+2}^{(1)}}{p_{t+1}^{(1)}} + \sum_{u'} q_{s-1}^{(u')} \right) & (s > 1). \end{cases} \tag{A.12}$$

Using Eq. (A.12), we can compute $P_s^{(0)}(\bar{x}_i)$ for $s = 1, 2, \ldots, n_i$ in $O(n_i)$ time.

Similarly, for all $t' < t$, we have

$$\sum_{t'=1}^{t-1}\sum_{u'=0}^{1}\frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{s+1}^{(0)}}{p_s^{(0)}\, q_s^{(u)}}\underbrace{\sum_{t'=1}^{t-1}\sum_{u'=0}^{1}\frac{\frac{p_{s'}^{(0)}}{p_{s+1}^{(0)}}p_{t'}^{(1)}q_{n-t'}^{(u')}}{p_t^{(1)}}}_{=:\ P_t^{(1)}}, \tag{A.13}$$

where $P_t^{(1)}$ only depends on $\bar{x}_o$ can be computed in $O(n_o)$ time.

With both $P_s^{(0)}$ and $P_t^{(1)}$ precomputed, Eq. (A.3) becomes

$$w_{s,t}^{(u)}(\bar{y}_{s,t}) = \left(1 + P_s^{(0)} + P_t^{(1)} + \sum_{u'=0}^{1}\frac{p_{s-1,t+1}^{(u')}(\bar{y}_{s,t}) + p_{s+1,t-1}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})}\right)^{-1}, \tag{A.14}$$

which can be computed in constant time. This leads to a full bidirectional estimator with time complexity $O(n_i n_o)$.

A.3 MIS with stochastic function and weight evaluation

Introduction. While Monte Carlo integration and multiple importance sampling (MIS) are widely used in practice, we use extended versions of these techniques: our MIS weighting is based on approximate (not exact) pdfs, and our weight and function evaluation are both stochastic (i.e. they consume additional random numbers, and are equal to the true weight and function value only in expectation). For this reason, we review standard Monte Carlo and MIS estimators, and show that our extensions still lead to unbiased results.

Monte Carlo estimator. Let $f(x)$ be an integrable function on domain D, and let X be a random variable on domain D with probability distribution $p(x)$, such that $p(x) > 0$

whenever $f(x) \neq 0$. An integral

$$I = \int_D f(x)\ dx \tag{A.15}$$

can be approximated by the unbiased estimator

$$X_f = \frac{f(X)}{p(X)}. \tag{A.16}$$

It is easy to see that X_f is an unbiased estimate of I:

$$E_X[X_f] = \int_D p(x)\frac{f(x)}{p(x)}\ dx = \int_D f(x)\ dx = I. \tag{A.17}$$

Note, the cancellation of $p(x)$ is always possible due to the assumption that $p(x) > 0$ for all x where $f(x)$ is non-zero.

Combining estimators through MIS. Multiple importance sampling (MIS) combines two different sampling strategies (random variables) X_1 and X_2 on D, with pdfs $p_1(x)$ and $p_2(x)$, to compute the integral I more robustly. This is achieved by choosing weighting functions $w_1(x)$ and $w_2(x)$ such that $w_1(x) + w_2(x) = 1$ for all $x \in D$.

Furthermore, we shall require that if $p_1(x) = 0$ or $p_2(x) = 0$, the corresponding $f(x) = 0$. The integral I is thus split into I_1 and I_2:

$$I = I_1 + I_2 = \int_D w_1(x)f(x)\ dx + \int_D w_2(x)f(x)\ dx. \tag{A.18}$$

The following are unbiased estimators for I_1 and I_2:

$$X_f^1 = \frac{w_1(X)f(X)}{p_1(X)} \qquad X_f^2 = \frac{w_2(X)f(X)}{p_2(X)}. \tag{A.19}$$

This can be seen as follows:

$$E_X[X_f^1] = \int_D p_1(x) \frac{w_1(x) f(x)}{p_1(x)} \, dx = \int_D w_1(x) f(x) \, dx = I_1, \tag{A.20}$$

and the same argument works for I_2. Again, the reason the cancellation of $p_1(x)$ works is that either it is non-zero, or $f(x) = 0$, due to the assumption above.

Also note that we made no assumptions on the weights other than that they sum to 1. In particular, there is no requirement that the weights be derived from exact pdfs, and we are free to base them on approximate pdfs, among other choices.

Stochastic function evaluation. Now suppose that the function evaluation is itself stochastic, i.e. it is an unbiased estimator $f(x, R)$ of the true value of $f(x)$, that uses a uniform random number R on the interval $[0, 1)$ during its evaluation. The argument can be easily extended for the case of consuming multiple uniform random numbers. We use a single random number in the proof for brevity.

Because the function estimator is unbiased, we have $E_R[f(x, R)] = \int_0^1 f(x, r) \, dr = f(x)$ for all x. Therefore, our full estimator becomes:

$$X_f = \frac{f(X, R)}{p(X)}. \tag{A.21}$$

We can see that this estimator is still unbiased, by computing its expected value over X *and* R:

$$\begin{aligned} E_{X,R}[X_f] &= \int_D \int_0^1 p(x) \frac{f(x, r)}{p(x)} \, dr \, dx \\ &= \int_D p(x) \frac{\int_0^1 f(x, r) \, dr}{p(x)} \, dx \\ &= \int_D p(x) \frac{f(x)}{p(x)} \, dx = I \end{aligned} \tag{A.22}$$

Stochastic weight and function evaluation When both the weight evaluation and the function evaluation in an MIS estimator are stochastic, the resulting estimator is still unbiased, provided that the random numbers used by the weight and the function are independent (which enables us to rewrite the joint integral over both random choices into separate integrals). Specifically, consider an unbiased estimator $w_1(x, R_1)$ of the true value of $w_1(x)$, and an unbiased estimator $f(x, R_2)$ of the true value of $f(x)$, based on uniform random numbers R_1 and R_2 on the interval $[0, 1)$. (again, this can be easily extended for the case of consuming multiple uniform random numbers.) The estimator for integral I_1 will become:

$$X_f^1 = \frac{w_1(X, R_1) f(X, R_2)}{p_1(X)} \tag{A.23}$$

We can see that this estimator is unbiased, by computing its expected value over X, R_1 and R_2:

$$\begin{aligned} E_{X,R_1,R_2}[X_f^1] &= \int_D \int_0^1 \int_0^1 p_1(x) \frac{w_1(x, r_1) f(x, r_2)}{p_1(x)} \, dr_1 \, dr_2 \, dx \\ &= \int_D p_1(x) \frac{\int_0^1 w_1(x, r_1) \, dr_1 \cdot \int_0^1 f(x, r_2) \, dr_2}{p_1(x)} \, dx \\ &= \int_D p_1(x) \frac{w_1(x) f(x)}{p_1(x)} \, dx = I_1. \end{aligned} \tag{A.24}$$

The same argument can be used for X_f^2.

Discussion. Application to direct illumination integral. In our application, the integral of interest I is normally the direct illumination estimate at a shading point. The function $f(x)$ involves the product of the BSDF and illumination values; this is integrated over the unit sphere (or unit hemisphere for BRDFs with no transmission), which is the domain D. The random variables X_1 and X_2 are outgoing directions ω_o chosen by light sampling and BSDF sampling, respectively. For the case of light sampling, we need to stochastically evaluate the MIS weight and BSDF value for the chosen ω_o; these evaluations

will consume vectors of uniform random numbers R_1 and R_2, respectively.

No approximation of pdfs in estimator denominators. While we use approximate stochastic pdfs to define the weights, we never approximate the pdfs in the denominators of our estimators. In our case, the accurate values of these pdfs are already baked into the f/p estimates returned by the position-free Monte Carlo simulations.

Sum of stochastic weights. The sum of the stochastic approximations to weights w_1 and w_2 will generally not be exactly 1, but this is not required. We simply require that

1. the expected values of the weights sum to 1, so that the integral I separates correctly into I_1 and I_2,
2. X_f^1 and X_f^2 are unbiased estimators for I_1 and I_2, respectively.

The combination of these properties implies an unbiased estimate for I.