

UNIVERSITY OF CALIFORNIA,  
IRVINE

Microscale-based Macroscale Rendering and Its Inverse Rendering

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

© 2021 Yu Guo

## **DEDICATION**

To Myself and My Family

# TABLE OF CONTENTS

	Page
<b>LIST OF FIGURES</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>vi</b>
<b>ACKNOWLEDGMENTS</b>	<b>vii</b>
<b>VITA</b>	<b>viii</b>
<b>ABSTRACT OF THE DISSERTATION</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>5</b>
<b>3 Microscale Based Surface Rendering</b>	<b>6</b>
3.1 Introduction . . . . .	7
3.2 Related Work . . . . .	10
3.2.1 Discretized layered BSDFs . . . . .	10
3.2.2 Analytic layered BSDFs . . . . .	11
3.2.3 Microfacet models for interfaces . . . . .	11
3.2.4 Capability comparison . . . . .	12
3.3 Background and Overview . . . . .	13
3.3.1 Assumptions . . . . .	13
3.3.2 Review of Veach’s path integral formulation . . . . .	14
3.3.3 Paper overview . . . . .	15
3.4 Position-Free Path Formulation . . . . .	16
3.4.1 Notation . . . . .	16
3.4.2 Position-free path integral . . . . .	17
3.4.3 Derivation . . . . .	20
3.4.4 Normal mapping . . . . .	22
3.4.5 Note about reciprocity . . . . .	23
3.4.6 Multiple slabs . . . . .	23
3.5 Our Estimators . . . . .	24
3.5.1 BSDF sampling . . . . .	24
3.5.2 BSDF evaluation . . . . .	25

3.5.3	Pdf estimation . . . . .	29
3.6	Applications and Results . . . . .	33
3.6.1	Validations . . . . .	33
3.6.2	Main Results . . . . .	34
3.6.3	Performance . . . . .	39
3.6.4	Limitations and future work . . . . .	39
3.7	Conclusion . . . . .	41
<b>4</b>	<b>Microscale Based Volumetric Rendering</b>	<b>42</b>
<b>5</b>	<b>Inverse Rendering for Macroscale Material Parameters</b>	<b>43</b>
<b>6</b>	<b>Inverse Rendering for Microscale Material Parameters</b>	<b>44</b>
<b>7</b>	<b>Conclusion and Future work</b>	<b>45</b>
<b>Bibliography</b>		<b>46</b>
<b>Appendix A Appendix for Chapter 3</b>		<b>49</b>
A.1	Detailed Derivations . . . . .	49
A.2	Efficient Weight Computation . . . . .	50
A.3	MIS with stochastic function and weight evaluation . . . . .	53

## LIST OF FIGURES

	Page
3.1 Teaser of LayeredBSDF . . . . .	6
3.2 Equal-time comparisons . . . . .	9
3.3 Comparison to previous work . . . . .	12
3.4 Small displacement assumption . . . . .	13
3.5 Outgoing lobes of a layered BSDF . . . . .	16
3.6 Example paths . . . . .	18
3.7 Our Monte Carlo estimators . . . . .	27
3.8 Validation of our pdf estimates . . . . .	32
3.9 Multiple importance sampling . . . . .	32
3.10 White furnace tests . . . . .	34
3.11 Top vs. bottom height variation . . . . .	35
3.12 Reflection and transmission A . . . . .	36
3.13 Reflection and transmission B . . . . .	36
3.14 Anisotropic media within layers . . . . .	37
3.15 Comparison to volumetric cloth . . . . .	38
3.16 Multi-layer BSDF . . . . .	39

## LIST OF TABLES

	Page
3.1 Notation used in the path formulation (§3.4) . . . . .	17
3.2 Render times of all our results . . . . .	40

## 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 an under reviewing paper (“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 L<sup>A</sup>T<sub>E</sub>X template for thesis and dissertation documents at UC Irvine [17].

# VITA

## Yu Guo

### EDUCATION

<b>Doctor of Philosophy in Computer Science</b> University of California, Irvine	<b>2016 – 2021</b> <i>Irvine, CA, US</i>
<b>Master of Science in Computational Sciences</b> University of Chinese Academy of Sciences	<b>2010 – 2013</b> <i>Beijing &amp; Shenzhen, China</i>
<b>Bachalar of Science in Applied Mathematics</b> Central South University	<b>2006 – 2010</b> <i>Changsha, China</i>

### RESEARCH EXPERIENCE

<b>Research Associate</b> Nanyang Technological University	<b>2013 – 2016</b> <i>Singapore</i>
---	--

## **REFEREED PUBLICATIONS**

<b>Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs</b> ACM Transactions on Graphics	<b>2018</b>
<b>MaterialGAN: Reflectance Capture using a Generative SVBRDF Model</b> ACM Transactions on Graphics	<b>2020</b>
<b>A Bayesian Inference Framework for Procedural Material Parameter Estimation</b> Computer Graphics Forum	<b>2020</b>
<b>Beyond Mie Theory: Systematic Computation of Bulk Scattering Parameters based on Microphysical Wave Optics</b> In submission ...	<b>2021</b>

# **ABSTRACT OF THE DISSERTATION**

Microscale-based Macroscale Rendering and Its Inverse Rendering

By

Yu Guo

Doctor of Philosophy in Computer Science

University of California, Irvine, 2021

Professor Shuang Zhao, Chair

The abstract of your contribution goes here.

# Chapter 1

## Introduction

In this dissertation, we first address a more general but efficient way to handle complex surface reflectance and volumetric scattering,

Next, we present an optimization based method for SVBRDF reconstruction and then extend it to bayesian inference.

To summarize, we develop a smart technique to render layered material, a framework to compute scatterings in participating media based on wave optics, and given a number of images, how to estimate the material properties. These techniques were presented at multiple conferences [7, 8, 6]. 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 ex-

pensive 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 and \*\*\*\*\* in Chapter 2. From Chapters 3 to 6, we present technical details of our \*\*\*\*, \*\*\*\*, \*\*\*\* and \*\*\*\*, respectively. Finally, we present our conclusion and discuss future research directions in Chapter 7.

# Chapter 2

## Background

# Chapter 3

## Microscale Based Surface Rendering

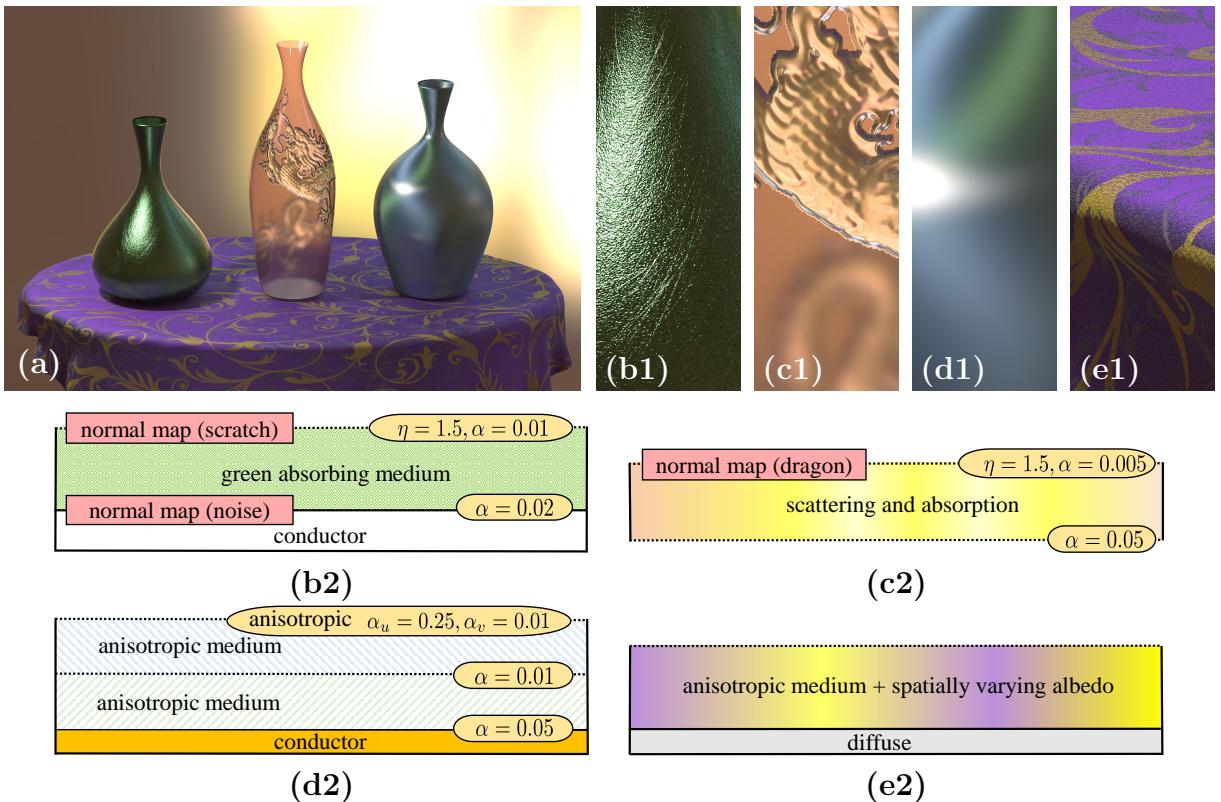


Figure 3.1: We introduce a new BSDF model leveraging an efficient Monte Carlo simulation algorithm applied locally to layered geometries. Our model enjoys the flexibility of using arbitrary layer interfaces and internal media and is capable of reproducing a wide variety of appearances. This example contains 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 [23] 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 [1] 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. [14] (with a recent follow-up [24]) 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 paper, 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.

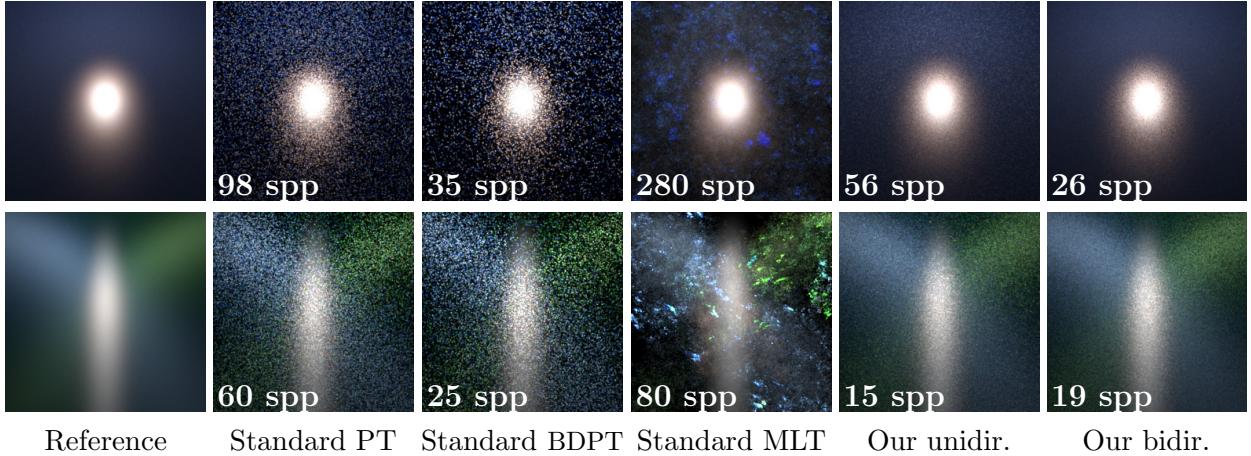


Figure 3.2: **Equal-time comparisons** of our unidirectional and bidirectional approach to standard transport algorithms, on a simple flat layered configuration lit by a small area light. For standard PT, BDPT and MLT, results are all generated using 3D tracing by applying these algorithms in a simple 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 [13]. 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 [9] 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 [19] 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. [14] 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 [24]. 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 [23] 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 [5] 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 [1] 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 [3], 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. [21] extends the microfacet model of Cook and Torrance [2] 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.[11] is capable of capturing interreflections between the facets and better conserves energy. Schüssler [18] 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

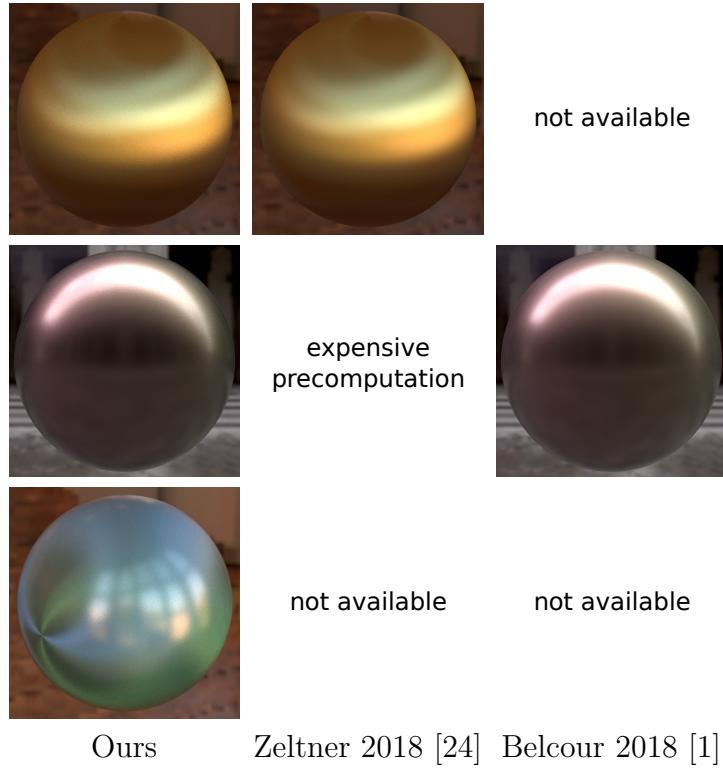


Figure 3.3: **Comparison to previous work.** The **top row** shows an example with anisotropic surface reflectance, where our solution closely matches Zeltner’s, but Belcour’s approach does not support anisotropy. The **middle row** shows an example with spatial variation in the parameters; here our 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 [24, 1]. 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 paper.

#### 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).

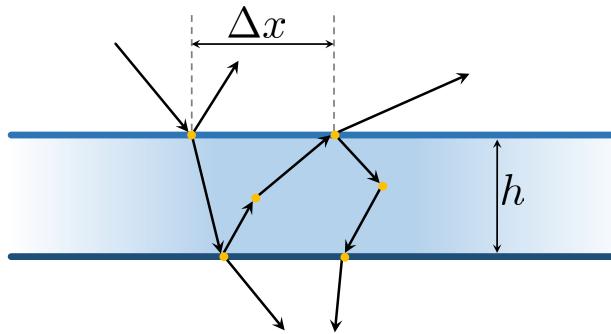


Figure 3.4: **Small displacement assumption:** when light hits a thin layer, it gets reflected and refracted by the interfaces and scattered and absorbed internally. Since the geometric thickness  $h$  of the layer is small, we assume the displacements (e.g.,  $\Delta x$ ) of light’s entrance and departure 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 [20], 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

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

where  $\bar{x} = (x_0, \dots, x_k)$  is a path with  $k$  segments and  $k + 1$  vertices on the surfaces or within the participating media of a scene.  $\Omega$  is the space of all paths and is defined as the union of  $\Omega_k$  for  $k \geq 0$ , where  $\Omega_k$  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 Paper overview

In Section ??, 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. This 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 ??, and summarize in Section 3.7.

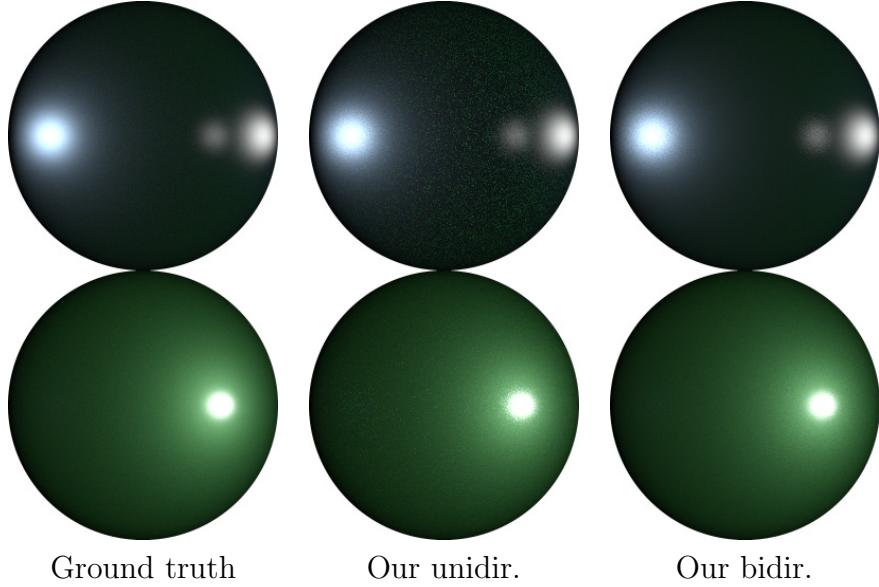


Figure 3.5: **Outgoing lobes of a layered BSDF** (reflection and transmission) visualized as projected hemispheres. **Left:** ground truth computed by sampling and binning the light paths. **Middle:** Our unidirectional estimator. **Right:** Our bidirectional estimator (same time).

## 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  $\omega_i$  and  $\omega_o$ , 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  $\cos \omega$  to denote the  $z$ -component of the unit vector  $\omega$ . 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  $\mathcal{S}^2$ . Please refer to Table 3.1 for the notation used in this section.

Table 3.1: Notation used in the path formulation (§3.4).

$\omega_i$	light direction
$\omega_o$	camera direction
$\cos \omega$	$z$ -component of the unit vector $\omega$
$\mathbb{I}(x)$	binary indicator function
$f_l(\omega_i, \omega_o)$	layered BSDF (our goal)
$f_s(z, \omega_i, \omega_o)$	interface BSDF at depth $z$
$f_\uparrow, f_\downarrow$	BSDFs $f_s$ at top and bottom interface
$f_p(\omega_i, \omega_o)$	phase function (normalized as a pdf)
$\sigma_s, \sigma_t$	scattering and absorption coefficient
$\hat{f}_p$	reduced phase function, $\hat{f}_p = \sigma_s f_p$
$z_i$	depth of $i$ -th path vertex
$\mathbf{d}_i$	direction of $i$ -th path segment
$\bar{x}$	light path $(\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_k, \mathbf{d}_k)$
$v_i$	$i$ -th vertex contribution
$s_i$	$i$ -th segment contribution
$\tau(z, z', \omega)$	transfer through segment
$\alpha_i$	$i$ -th segment cosine term exponent
$\mu(\bar{x})$	path space measure
$\sigma(\omega)$	solid angle measure on unit directions
$\lambda(z)$	line (Lebesgue) measure on real numbers
$p(\bar{x})$	pdf of path $\bar{x}$ in measure $\mu(\bar{x})$
$L_v(z, \omega_o)$	volume radiance
$L_s(z, \omega_o)$	outgoing surface radiance
$L_s^i(z, \omega_i)$	incoming surface radiance
$S(z, \omega)$	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  $\sigma_s$  and extinction coefficient  $\sigma_t$ ; we will use the notation  $\hat{f}_p = \sigma_s f_p$ .

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**  $\mathbf{d}_i$  is a unit vector on  $\mathcal{S}^2$  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:  $\bar{x} = (\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_k, \mathbf{d}_k)$ . The first and last directions are aligned with the input and output directions of the layered BSDF query, i.e.  $\mathbf{d}_0 = -\omega_i$  and  $\mathbf{d}_k = \omega_o$ . 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.

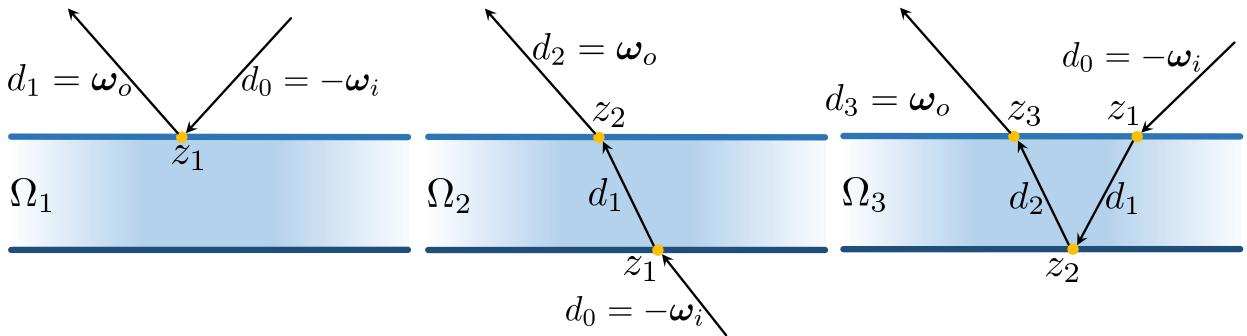


Figure 3.6: **Example paths** of lengths 1, 2 and 3. In our formulation, the exact positions of the vertices do not matter: the  $z_i$  only carry information about which interface the vertex occurs on. The first and last in the sequence of directions  $\mathbf{d}_i$  map to the incoming and outgoing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

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

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

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

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

$$\tau(z, z', \boldsymbol{\omega}) := \exp\left(\frac{-\sigma_t |z' - z|}{|\cos \boldsymbol{\omega}|}\right) \cdot \mathbb{I}\left(\frac{z' - z}{\cos \boldsymbol{\omega}} > 0\right). \quad (3.4)$$

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:

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

where

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

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**  $\Omega(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$  is the set of all paths of one or more vertices, such that the first direction of the path is equal to  $-\boldsymbol{\omega}_i$  and the last to  $\boldsymbol{\omega}_o$ . 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  $\sigma$  on the internal directions of the path, times the product of line measures  $\lambda$  on volumetric scattering vertices. That is, for a  $k$ -vertex path,

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

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

Finally, we can define the **layered BSDF value**  $f_l(\omega_i, \omega_o)$  as an integral over the set of paths  $\Omega(\omega_i, \omega_o)$ :

$$f_l(\omega_i, \omega_o) = \int_{\Omega(\omega_i, \omega_o)} f(\bar{x}) d\mu(\bar{x}). \quad (3.8)$$

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  $L_s(z, \omega)$  (for  $z \in \{0, 1\}$ ) and the volume radiance  $L_v(z, \omega)$  (for  $z \in [0, 1]$ ).

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

position-free setting:

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

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

$$S(z, \omega) = \tau(0, z, \omega) L_s(0, \omega) + \tau(1, z, \omega) L_s(1, \omega). \quad (3.10)$$

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  $1/|\cos \omega|$  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  $L_s(z, \omega)$  satisfies the standard rendering equation:

$$L_s(z, \omega) = \int_{\mathcal{S}^2} f_s(z, \omega, \omega') |\cos \omega'| L_s^i(z, \omega') d\sigma(\omega'), \quad (3.11)$$

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

$$L_s^i(z, \omega) = L_v(z, -\omega). \quad (3.12)$$

The BSDF value is defined as the radiance leaving the surface in direction  $\omega_o$ , under unit irradiance from a directional light in direction  $\omega_i$ . This is equivalent to evaluating  $L_s(1, \omega_o)$  under the boundary condition

$$L_s^i(z, \omega) = \frac{\delta(\omega - \omega_i)}{|\cos \omega_i|}. \quad (3.13)$$

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  $\omega$  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  $L_s(1, \omega_o)$  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  $\tau(z, z', \omega)$  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  $\alpha_i$  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

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

where  $\mathbf{n}(z)$  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 [20] (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  $f_s(\omega_i, \omega_o) = \eta_o^2 / \eta_i^2 \cdot f_s(\omega_o, \omega_i)$  [21], where  $\eta_i$  and  $\eta_o$  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  $\eta^2$  and  $1/\eta^2$  terms along the path for each medium with index  $\eta$ .

### 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  $\tau(z, z' \omega)$  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  $\omega_o$  given the incoming one  $\omega_i$  (or the reverse), while evaluation answers the BSDF query for given  $\omega_i$  and  $\omega_o$ . 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 [11].

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  $\omega_o$  given the incoming one  $\omega_i$  (or the reverse), with a pdf proportional, exactly or approximately, to the value  $f_l(\omega_i, \omega_o)$  (times the cosine term, if possible). This is straightforward: we draw  $\omega_o$  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  $-\omega_i$  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  $\omega_o$  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  $\omega_i$  and  $\omega_o$ , 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  $\omega_i$ , intuitively connecting it to an external

directional light source with direction  $\omega_i$ . 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  $\omega_o$ . At each scattering event, we need to find a direction  $\omega'_i$  so that  $\omega_i \rightarrow \omega'_i$  follows the BSDF at the interface. To this end, we draw  $\omega'_i$  by sampling the interface BSDF backwards, given  $\omega_i$ . 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 [22, 16] 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.

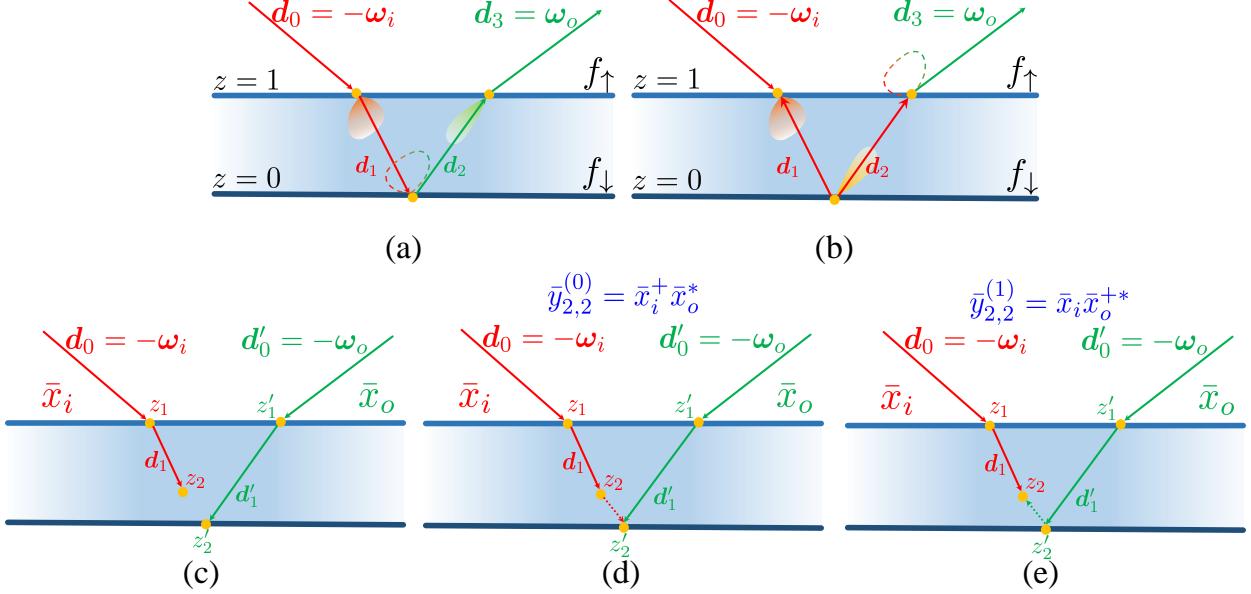


Figure 3.7: **Our Monte Carlo estimators for BSDF values.** (ab) Unidirectional estimator uses two path sampling strategies for “shading” a vertex on the bottom layer: (a) sampling the BSDF  $f_{\uparrow}$  of the top interface and connecting at the bottom (next event estimation); or (b) sampling  $d_2$  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  $\omega_i$  and  $\omega_o$ . (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  $\omega_i$  and  $\omega_o$ , consider two light transport paths, generated from the light and camera, respectively.

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

where  $\mathbf{d}_0 = -\boldsymbol{\omega}_i$  and  $\mathbf{d}'_0 = -\boldsymbol{\omega}_o$  (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):

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

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

$$\bar{x}_i^+ := (\mathbf{d}_0, z_1, \mathbf{d}_1, \dots, z_s, \mathbf{d}_s), \quad (3.17)$$

$$\bar{x}_o^+ := (\mathbf{d}'_0, z'_1, \mathbf{d}'_1, \dots, z'_t, \mathbf{d}'_t), \quad (3.18)$$

and let  $\bar{x}^*$  denote the adjoint (reversed) version of a light path  $\bar{x}$ , e.g.,

$$\bar{x}_o^{+*} = (-\mathbf{d}'_t, z'_t, -\mathbf{d}'_{t-1}, \dots, z'_1, -\mathbf{d}'_0).$$

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

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

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

It follows that the two Monte Carlo estimates will be:

$$\frac{f(\bar{y}_{s,t}^{(0)})}{p(\bar{y}_{s,t}^{(0)})} = \frac{f(\bar{x}_i^+)}{p(\bar{x}_i^+)} \frac{f(\bar{x}_o^*)}{p(\bar{x}_o^*)} s(z_s, z'_t, \mathbf{d}_s) v(z'_t, -\mathbf{d}_s, -\mathbf{d}'_{t-1}) \quad (3.21)$$

$$\frac{f(\bar{y}_{s,t}^{(1)})}{p(\bar{y}_{s,t}^{(1)})} = \frac{f(\bar{x}_i)}{p(\bar{x}_i)} \frac{f(\bar{x}_o^{+*})}{p(\bar{x}_o^+)} v(z_s, \mathbf{d}_{s-1}, \mathbf{d}'_t) s(z_s, z'_t, -\mathbf{d}'_t), \quad (3.22)$$

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 [20].

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):

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

For all  $s$  and  $t$  combinations, Eqs. (3.21) and (3.22) provide  $2n_i n_o$  estimators of  $f_l(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ . 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  $p(\boldsymbol{\omega}_o | \boldsymbol{\omega}_i)$ , the probability density of  $\boldsymbol{\omega}_o$  given  $\boldsymbol{\omega}_i$  (assuming that the sampling process draws  $\boldsymbol{\omega}_o$  and fixes  $\boldsymbol{\omega}_i$ ). 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  $p(\omega_o \mid \omega_i)$  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:

$$p(\omega_o \mid \omega_i) = \int_{\Omega(\omega_i, \omega_o)} \mathcal{P}(\bar{x}) d\mu(\bar{x}), \quad (3.24)$$

where

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

with  $k$  denoting the number of vertices in  $\bar{x}$ . Note that  $\mathbf{d}_k = \omega_o$ .

We introduce two nondeterministic methods, an unbiased and a fast approximate approach, to estimate  $p(\omega_o \mid \omega_i)$ . 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  $p(\omega_o \mid \omega_i)$  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:

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

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

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  $\omega_i \cdot \omega_o > 0$  (i.e.,  $f_l(\omega_i, \omega_o)$  captures reflection) and  $(L + 1)$  when  $\omega_i \cdot \omega_o < 0$  (i.e.,  $f_l(\omega_i, \omega_o)$  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.

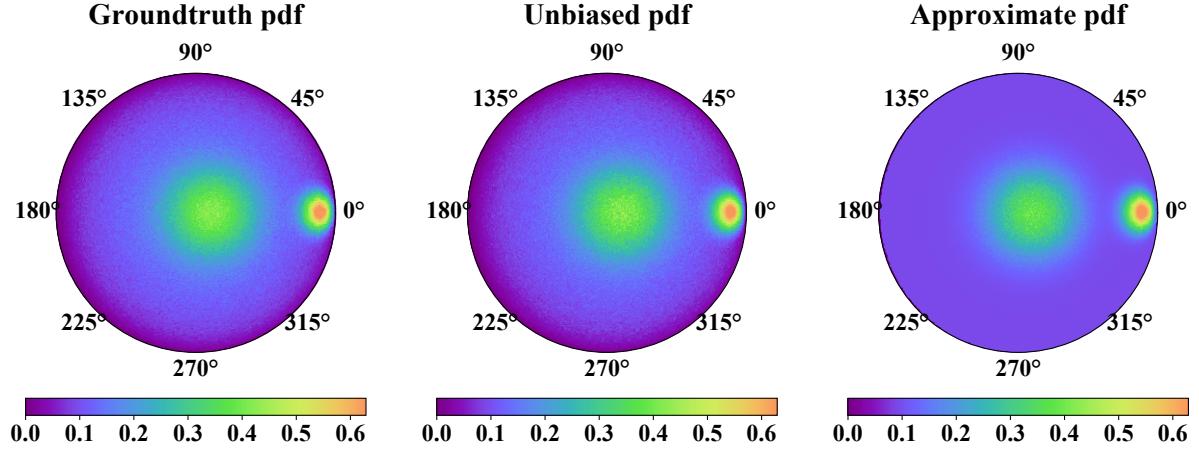


Figure 3.8: **Validation of our pdf estimates.** The visualization applies a  $\log(1 + x)$  map for better shape perception. **Left:** Ground truth by sampling and binning. **Middle:** Using the unbiased pdf from §3.5.3. **Right:** Using the approximate pdf from §3.5.3 matches the shape of the most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

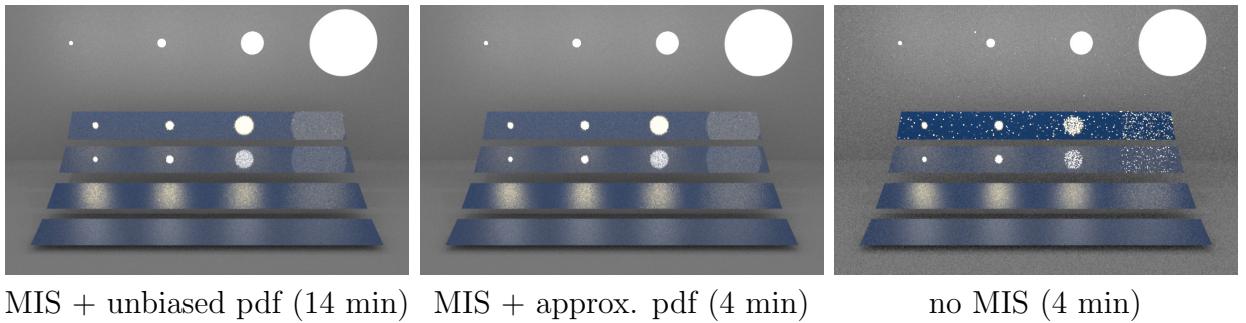


Figure 3.9: **Multiple importance sampling** using our BSDFs. The slabs in this figure use a layered material with rough dielectric on the top, rough gold conductor on the bottom, and blueish homogeneous scattering medium in between. **Left:** Using the unbiased pdf for MIS in a traditional global path 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 [12] 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 paper 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 ??-(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.

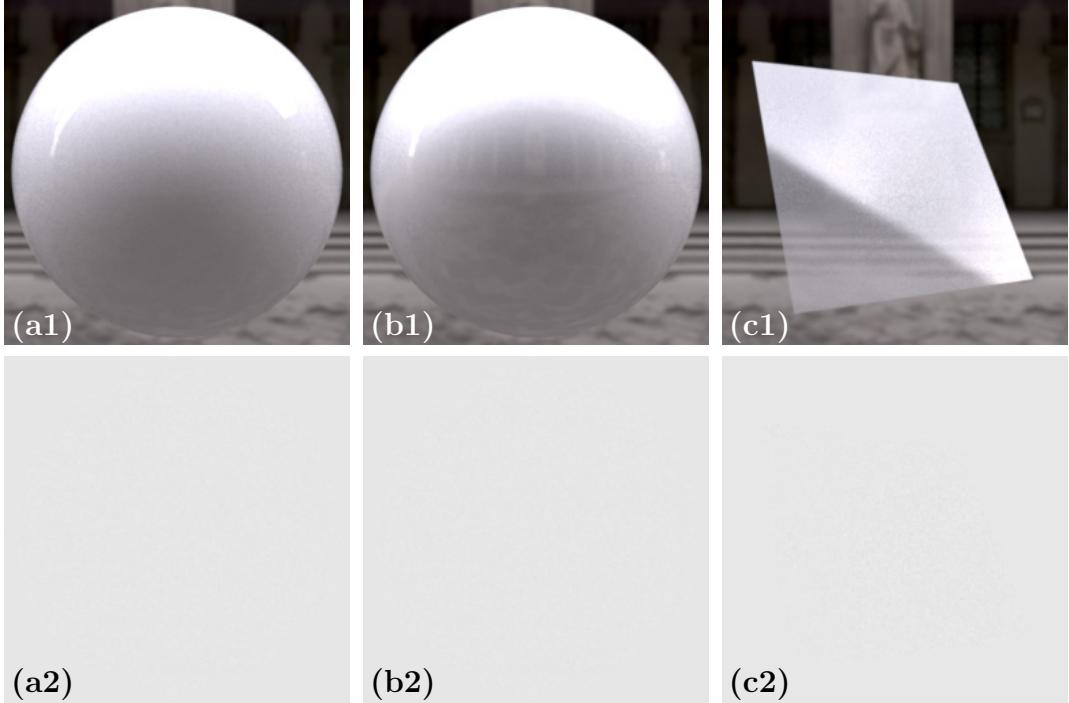


Figure 3.10: **White furnace tests.** We demonstrate that our BSDFs conserve energy properly via three layered BSDF examples respectively given by **(a)** a dielectric and a diffuse interface; **(b)** a dielectric and a conductor interface and participating medium; **(c)** two dielectric interfaces and 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

#### 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.

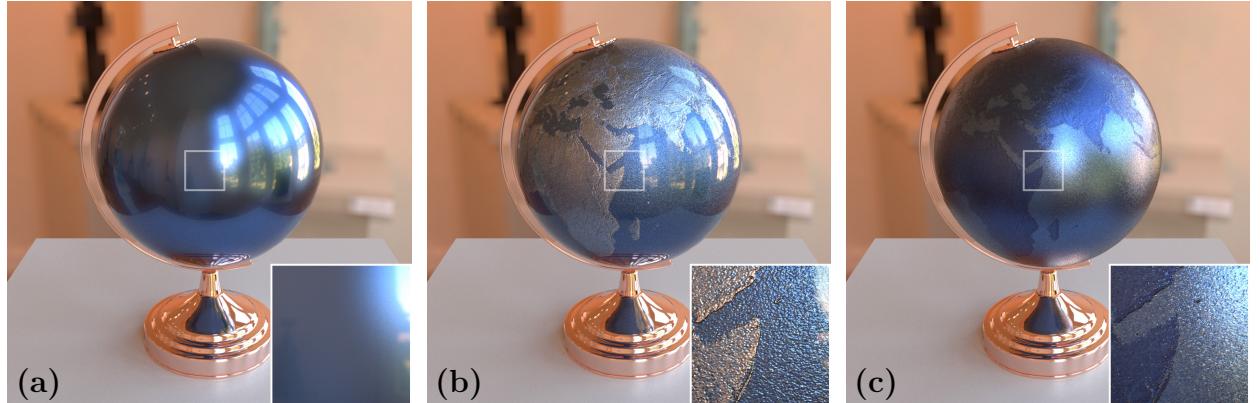
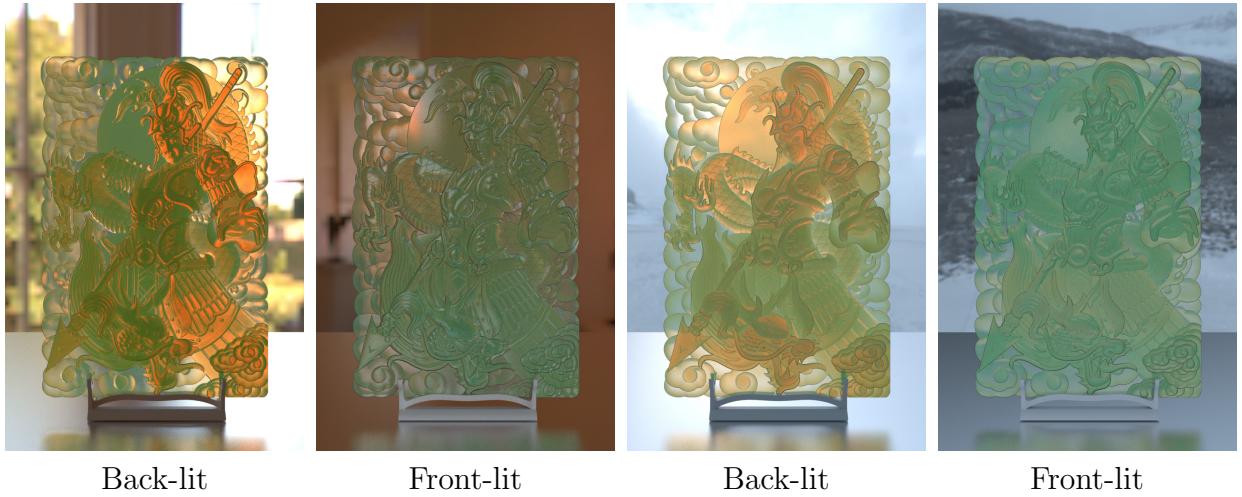


Figure 3.11: **Top vs. bottom height variation.** Thanks to the physically-based nature of our layered BSDF model, manipulating heights on its top and bottom interfaces has greatly varying effects on the final appearance. The height variation drives both normals and thickness differences (and thus medium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

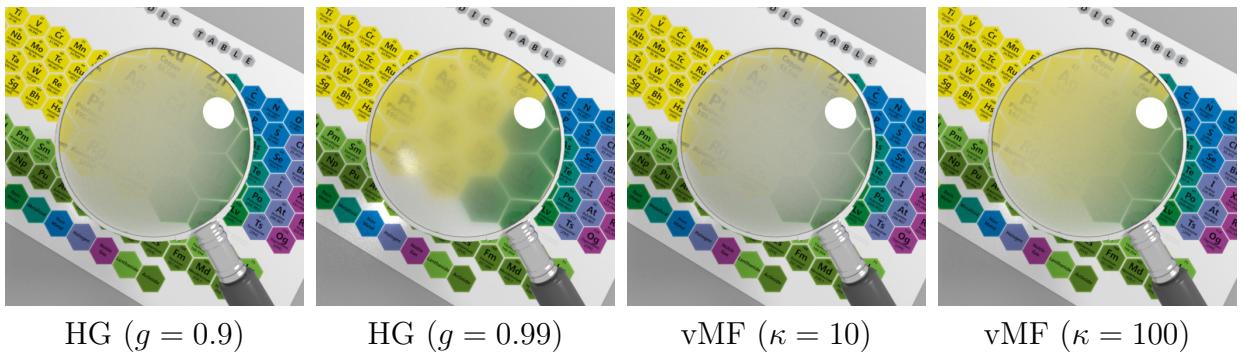
### Application: Complex thin sheet transmission

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.



**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.



**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 spa-

tially 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 [13, 25, 10] 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.

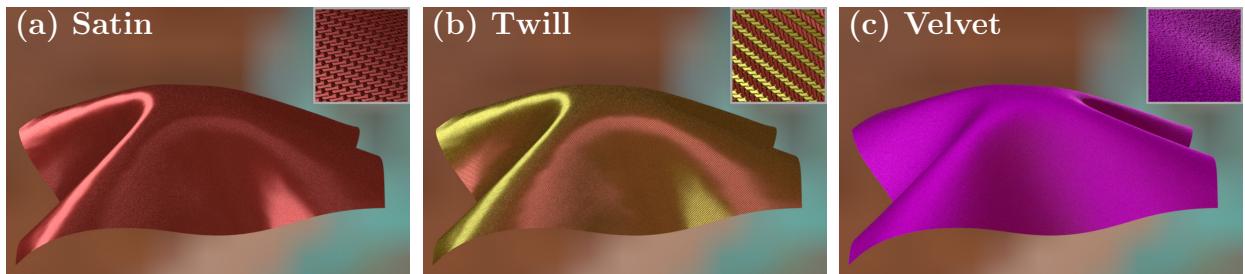


Figure 3.14: **Anisotropic media within layers.** Our layered BSDF offers the generality to use anisotropic layer media with microflake phase functions. This example shows three fabrics modeled with our BSDF model with anisotropic layer media: (a) satin; (b) twill; and (c) velvet.

Figure 3.15 shows a fabric rendered using fiber orientation data acquired by micro-CT imaging [25]. 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.

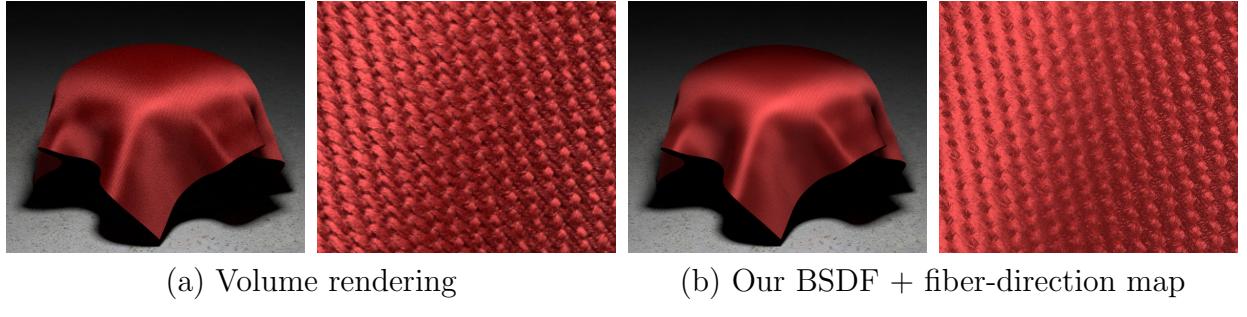


Figure 3.15: **Comparison to volumetric cloth.** (a) Images rendered from micro-CT volumetric data, using the microflake phase function. (b) Renderings using our approach using a single microflake volumetric layer, where we are using fiber direction maps extracted from the volumetric data. Our rendering is  $40\times$  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 that from (b). Our method offers the flexibility to conveniently 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.

### 3.6.4 Limitations and future work

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

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 paper figures. Please refer to the supplemental material for all the other renderings.

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

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 [15, 4] 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.

### 3.7 Conclusion

In this paper, 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 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**

## **Microscale Based Volumetric Rendering**

# **Chapter 5**

## **Inverse Rendering for Macroscale Material Parameters**

# **Chapter 6**

## **Inverse Rendering for Microscale Material Parameters**

# **Chapter 7**

## **Conclusion and Future work**

# Bibliography

- [1] Laurent Belcour. Efficient Rendering of Layered Materials Using an Atomic Decomposition with Statistical Operators. *ACM Transactions on Graphics*, 37(4):1–15, 2018.
- [2] Robert L Cook and Kenneth E. Torrance. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.
- [3] 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.
- [4] Jeppe Revall Frisvad, Toshiya Hachisuka, and Thomas Kim Kjeldsen. Directional Dipole Model for Subsurface Scattering. *ACM Transactions on Graphics*, 34(1):1–12, 2014.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Pat Hanrahan and Wolfgang Krueger. Reflection from Layered Surfaces Due to Subsurface Scattering. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’93, pages 165–174, 1993.
- [10] Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. The SGGX microflake distribution. *ACM Transactions on Graphics*, 34(4):1–11, 2015.
- [11] 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.

- [12] Wenzel Jakob. Mitsuba Renderer. <http://www.mitsuba-renderer.org>, 2010.
- [13] 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.
- [14] 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.
- [15] Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. A Practical Model for Subsurface Light Transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 511–518, 2001.
- [16] 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.
- [17] Lars Otten. LaTeX Template for Thesis and Dissertation Documents at UC Irvine. <https://github.com/lotten/uci-thesis-latex/>, 2012.
- [18] 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.
- [19] Jos Stam. An Illumination Model for a Skin Layer Bounded by Rough Surfaces. In *Eurographics Workshop on Rendering Techniques*, pages 39–52, 2001.
- [20] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.
- [21] 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.
- [22] 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.
- [23] Andrea Weidlich and Alexander Wilkie. Arbitrarily Layered Micro-facet Surfaces. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pages 171–178, 2007.
- [24] 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.

- [25] 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.

# 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  $t$ :

$$L_v(z, \omega) = S(z, \omega) + \int_0^{t'} \exp(-t\sigma_t) \int_{S^2} \hat{f}_p(\omega', \omega) L_v(z', \omega') d\omega' dt, \quad (\text{A.1})$$

where  $z' := z - t \cos \omega$  and  $t'$  denotes the distance between  $z$  and the closest layer boundary. Since  $t = (z - z') / \cos \omega$ , changing the integration variable from  $t$  to  $z'$  in Eq. (A.1) yields an additional factor of  $(\cos \omega)^{-1}$  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  $|\cos \mathbf{d}_i|$  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  $\mathbf{d}_i$  can yield an additional  $|\cos \mathbf{d}_i|^{-1}$  when  $z_{i+1}$  corresponds to a volumetric scattering. Thus, for each  $i$ , the path contribution involve a factor of  $|\cos \mathbf{d}_i|^{\alpha_i}$  with:

- $\alpha_i = 1$  if  $z_i$  and  $z_{i+1}$  are both on interfaces;
- $\alpha_i = 0$  (i.e., no  $\cos \mathbf{d}_i$  term) if (i)  $z_i$  is volumetric and  $z_{i+1}$  lies on an interface (so that no  $\cos \mathbf{d}_i$  terms appear during expansion), or (ii)  $z_i$  is interfacial and  $z_{i+1}$  is volumetric (so that both  $|\cos \mathbf{d}_i|$  and  $|\cos \mathbf{d}_i|^{-1}$  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  $2n_i n_o$  estimators of the form  $f(\bar{y}_{s,t}^{(u)})/p_{s,t}^{(u)}(\bar{y}_{s,t}^{(u)})$  with  $s \in \{1, 2, \dots, n_i\}$ ,  $t \in \{1, 2, \dots, n_o\}$ , and  $u \in \{0, 1\}$  via the multiple importance sampling (MIS) framework. This yields a combined estimator:

$$\sum_{s=1}^{n_i} \sum_{t=1}^{n_o} \sum_{u=0}^1 w_{s,t}^{(u)}(\bar{y}_{s,t}^{(u)}) \frac{f(\bar{y}_{s,t}^{(u)})}{p_{s,t}^{(u)}(\bar{y}_{s,t}^{(u)})}, \quad (\text{A.2})$$

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

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

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  $t$  naively 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 for the paper).

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

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

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

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

$$p_{s',t'}^{(u')}(\bar{y}_{s,t}) = p_{s'}^{(0)} p_{t'}^{(1)} q_{s'}^{(u')}, \quad (\text{A.6})$$

where

$$q_{s'}^{(u')} := \begin{cases} p(\mathbf{d}_{s'} \mid z_{s'}, \mathbf{d}_{s'-1}) & \text{if } u = 0, \\ p(-\mathbf{d}_{s'} \mid z_{s'+1}, -\mathbf{d}_{s'+1}) & \text{if } u = 1. \end{cases} \quad (\text{A.7})$$

It follows that

$$\frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{s'}^{(0)} p_{t'}^{(1)} q_{s'}^{(u')}}{p_s^{(0)} p_t^{(1)} q_s^{(u)}}. \quad (\text{A.8})$$

Note that, for any  $s' < s$ , we have

$$p_{s',t'}^{(u')}(\bar{y}_{s,t}) = p_{s'}^{(0)} \frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}} q_{s'}^{(u')} p_{t+1}^{(1)}. \quad (\text{A.9})$$

It follows that

$$\sum_{s'=1}^{s-1} \sum_{u'=0}^1 \frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{t+1}^{(1)}}{p_t^{(1)} q_s^{(u)}} \underbrace{\sum_{s'=1}^{s-1} \sum_{u'=0}^1 \frac{p_{s'}^{(0)} \frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}} q_{s'}^{(u')}}{p_s^{(0)}}}_{=: P_s^{(0)}}. \quad (\text{A.10})$$

Since

$$\frac{p_{t'}^{(1)}}{p_{t+1}^{(1)}} = \prod_{i=s'+1}^{s-1} p(-\mathbf{d}_i \mid z_{i+1}, -\mathbf{d}_{i+1}) p(z_i \mid z_{i+1}, -\mathbf{d}_i), \quad (\text{A.11})$$

it is easy to verify that  $P_s^{(0)}$  depends only on depths  $z_{s'}$  and directions  $\mathbf{d}_{s'}$  with  $s' \leq s$ , 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  $s = 1, 2, \dots, n_i$ . To this end,  $P_s^{(0)}(\bar{y})$  can be efficiently evaluated using the following relation:

$$P_s^{(0)} = \begin{cases} 0 & (s = 0), \\ \frac{p_{s-1}^{(0)}}{p_s^{(0)}} \left( P_{s-1}^{(0)} \frac{p_{t+2}^{(1)}}{p_{t+1}^{(1)}} + \sum_{u'} q_{s-1}^{(u')} \right) & (s > 1). \end{cases} \quad (\text{A.12})$$

Using Eq. (A.12), we can compute  $P_s^{(0)}(\bar{x}_i)$  for  $s = 1, 2, \dots, n_i$  in  $O(n_i)$  time.

Similarly, for all  $t' < t$ , we have

$$\sum_{t'=1}^{t-1} \sum_{u'=0}^1 \frac{p_{s',t'}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} = \frac{p_{s+1}^{(0)}}{p_s^{(0)} q_s^{(u)}} \underbrace{\sum_{t'=1}^{t-1} \sum_{u'=0}^1 \frac{\frac{p_{s'}^{(0)}}{p_{s+1}^{(0)}} p_{t'}^{(1)} q_{n-t'}^{(u')}}{p_t^{(1)}}}_{=: P_t^{(1)}}, \quad (\text{A.13})$$

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

$$w_{s,t}^{(u)}(\bar{y}_{s,t}) = \left( 1 + P_s^{(0)} + P_t^{(1)} + \sum_{u'=0}^1 \frac{p_{s-1,t+1}^{(u')}(\bar{y}_{s,t}) + p_{s+1,t-1}^{(u')}(\bar{y}_{s,t})}{p_{s,t}^{(u)}(\bar{y}_{s,t})} \right)^{-1}, \quad (\text{A.14})$$

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

$$I = \int_D f(x) dx \quad (\text{A.15})$$

can be approximated by the unbiased estimator

$$X_f = \frac{f(X)}{p(X)}. \quad (\text{A.16})$$

It is easy to see that  $X_f$  is an unbiased estimate of  $I$ :

$$E_X[X_f] = \int_D p(x) \frac{f(x)}{p(x)} dx = \int_D f(x) dx = I. \quad (\text{A.17})$$

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  $I_2$ :

$$I = I_1 + I_2 = \int_D w_1(x)f(x) dx + \int_D w_2(x)f(x) dx. \quad (\text{A.18})$$

The following are unbiased estimators for  $I_1$  and  $I_2$ :

$$X_f^1 = \frac{w_1(X)f(X)}{p_1(X)} \quad X_f^2 = \frac{w_2(X)f(X)}{p_2(X)}. \quad (\text{A.19})$$

This can be seen as follows:

$$E_X[X_f^1] = \int_D p_1(x) \frac{w_1(x)f(x)}{p_1(x)} dx = \int_D w_1(x)f(x) dx = I_1, \quad (\text{A.20})$$

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:

$$X_f = \frac{f(X, R)}{p(X)}. \quad (\text{A.21})$$

We can see that this estimator is still unbiased, by computing its expected value over  $X$  and  $R$ :

$$\begin{aligned} E_{X,R}[X_f] &= \int_D \int_0^1 p(x) \frac{f(x, r)}{p(x)} dr dx \\ &= \int_D p(x) \frac{\int_0^1 f(x, r) dr}{p(x)} dx \\ &= \int_D p(x) \frac{f(x)}{p(x)} dx = I \end{aligned} \quad (\text{A.22})$$

**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:

$$X_f^1 = \frac{w_1(X, R_1)f(X, R_2)}{p_1(X)} \quad (\text{A.23})$$

We can see that this estimator is unbiased, by computing its expected value over  $X$ ,  $R_1$  and  $R_2$ :

$$\begin{aligned} E_{X, R_1, R_2}[X_f^1] &= \int_D \int_0^1 \int_0^1 p_1(x) \frac{w_1(x, r_1)f(x, r_2)}{p_1(x)} dr_1 dr_2 dx \\ &= \int_D p_1(x) \frac{\int_0^1 w_1(x, r_1) dr_1 \cdot \int_0^1 f(x, r_2) dr_2}{p_1(x)} dx \\ &= \int_D p_1(x) \frac{w_1(x)f(x)}{p_1(x)} dx = I_1. \end{aligned} \quad (\text{A.24})$$

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  $\omega_o$  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  $\omega_o$ ; these evaluations

will consume vectors of uniform random numbers  $R_1$  and  $R_2$ , respectively.

**No approximation of pdfs in estimator denominators.** While in the main paper 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  $I$ .