MODELING AND RENDERING APPEARANCE OF HAIR AND TEXTILE FIBERS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Pramook Khungurn

May 2017

MODELING AND RENDERING APPEARANCE OF HAIR AND TEXTILE FIBERS

Pramook Khungurn, Ph.D.

Cornell University 2017

Fibers are ubiquitous in our visual world. Hair is an important part of our appearance, and we wear and use clothes made from various types of fibers. Computer graphics models that can accurately simulate light scattering in these materials have applications in the production of media such as movies and video games. They can also significantly lower the cost of textile design by allowing designers to design fabrics entirely in silico, render realistic images for feedback, and then fabricate final products that look exactly as designed.

Recent research has shown that renderings of the highest quality—those showing realistic reflectance and complex geometric details—can be obtained by modeling individual fibers. However, this approach raises many open problems. For hair, the effect of fiber cross sections on light scattering behavior has never been carefully studied. For textiles, several competing approaches for fiber-level modeling exist, and it has been unclear which is the best. Furthermore, there has been no general procedure for matching textile models to real fabric appearance, and rendering such models requires considerable computing resources. In this dissertation, we present solutions to these open problems.

Our first contribution is a light scattering model for human hair fibers that more accurately takes into account how light interacts with their elliptical cross sections. The model has been validated by a novel measurement device that captures light scattered from a single hair fiber much more efficiently than previous methods.

Our second contribution is a general and powerful optimization framework for estimating parameters of a large class of appearance models from observations of real materials, which greatly simplifies development and testing of such models. We used the framework to systematically identify best practices in fabric modeling, including how to represent geometry and which light scattering model to use for textile fibers.

Our third contribution is a fast, precomputation-based, GPU-friendly algorithm for approximately rendering fiber-level textile models under environment illumination. Using only a single commodity GPU, our implementation can render high-resolution, supersampled images of micron-resolution fabrics with multiple scattering in tens of seconds, compared to tens of core-hours required by CPU-based algorithms. Our algorithm makes fiber-level models practical for applications that require quick feedback, such as interactive textile design.

We expect these contributions will make realistic physically-based virtual prototyping a reality.

BIOGRAPHICAL SKETCH

Pramook Khungurn was born on December 22nd, 1982 in Loei, Thailand. Upon graduating from Triam Udom Suksa School, Bangkok, in 2001, he received the King's Scholarship from the Royal Thai Government, allowing him to study abroad in the country of his choice: the USA. After repeating a high school year at Deerfield Academy, Massachusetts, he enrolled at the Massachusetts Institute of Technology and received Bachelor of Science degrees in Computer Science and Mathematics in 2006. That year, Pramook received another scholarship from the Royal Thai Government and continued at MIT to study for a Master of Engineering in Electrical Engineering and Computer Science, which he completed in 2007. He then worked as a lecturer in the Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok until 2011.

Pramook was first exposed to computer graphics as an academic discipline when he took a class in his senior year at MIT. He subsequently became interested and started doing graphics research on his own while teaching at Kasetsart University. Feeling that the need for guidance and more exposure, he decided to pursue a doctorate degree. He entered the computer science doctoral program at Cornell University in 2011 and received his Ph.D. in 2017.

ACKNOWLEDGEMENTS

I would like to first thank my advisors, Kavita Bala and Steve Marschner, for guiding and supporting me through the course of the Ph.D. program. I'm especially grateful for KB's responsiveness and care for my mental well-being. Her big-picture perspective and savviness have steered my research effort in productive directions, and her resourcefulness has led to new collaborations. I'm indebted to Steve's insightful research suggestions as well as the patience to go down many technical rabbit holes with me. Working with him, I learned new skills that I think I would never be able to obtain otherwise, including operating a scanning electron microscopes. I thank both advisors deeply for allowing me to pursue a project unrelated to my main research topic and even funding my trip to present its result. Lastly, by TAing their graphics courses, I was given opportunities to build rendering systems up from scratch, and the experiences were much more instructive than any courses or paper reading.

I'm grateful to the other members of my Special Committee—Steven Strogatz, Charles Van Loan, and Robert Kleinberg—for their time and suggestion.

Much of the research in this dissertation would not have been possible without the work of my coauthors, all of whom I had great pleasure to collaborate with: Daniel Schroeder, Shuang Zhao, Rundong Wu, and James Noeckel. I'm much indebted Bruce Walter for allowing me to join his research project midway and patiently teaching me about light transport simulation, appearance acquisition, and the spherical gantry. I also thank Fujun Luan for helping with my research project despite his busy schedule.

Several colleagues and friends contributed software, data, and materials that were essential to my research. I thank Wenzel Jakob for the Mitsuba renderer, Albert Liu for the Caliber geometric calibration system, and Cem Yuksel for the

3D hair model data he made freely available. Frederick von Stein and Mark Riccio scanned many of the cloth samples used in Chapter 5. Saba Weatherspoon, Siman Peng, Kavita Bala, and Uniliver R&D for provided the hair samples used in Chapter 4. A special thank goes to Hurf Sheldon for his ever helpful technical support.

The research in this dissertation was supported by the National Science Foundation (grants IIS-1513967, IIS-1161645, and IIS-1011919) and the Intel Science and Technology Center for Visual Computing.

For making my Ph.D. ride a joyful experience, I thank my lab mates and Cornell friends: Sean Bell, Nicholas Beaumont, Derek Chou, Daniel Hauagge, Balazs Kovacs, Kenta Labur, Timothy Langlois, Kevin Matzen, Steven McDowell, James Nguyen, Nicholas Savva, Chie Shu, Eston Schweickart, Paul Upchurch, Scott Wehrwein, Kyle Wilson, Chris Yu, Chun-Po Wang, and Zechen Zhang.

Last but not least, I would like to thank my parents for their continued support and for allowing me to freely pursue my interests. It's impossible to pay back all the blessings they have given me, but I sure hope I will be able to pay it forward in the future.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

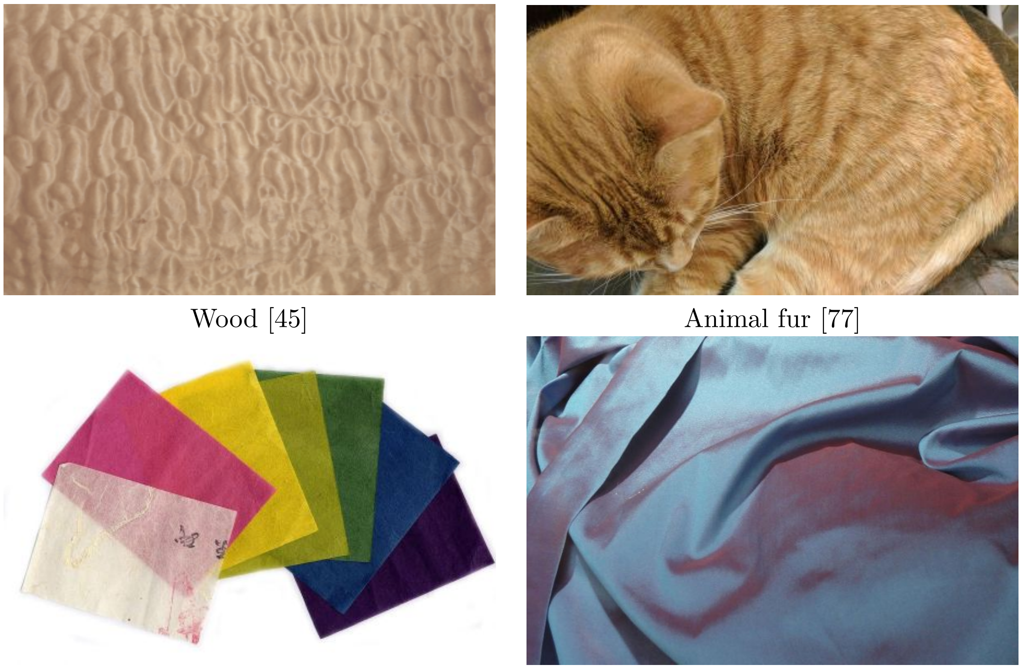
INTRODUCTION

> เธอสวย ทุกนาทีที่เคยสัมผัส
> รู้ทันทีว่าเธอคือคนพิเศษ
> ที่ฉันนั้นรอมานาน ที่ฟ้าให้มาเจอะกัน
> ให้ฉันมีเธอ
>
> DOUBLE YOU, *เธอสวย*

Physically-based rendering is the process of generating images by simulating the propagation of light inside virtual scenes made up of mathematical models of physical objects. By taking into account appropriate physical laws and using models with enough fidelity, the resulting images can be photorealistic. The afforded realism leads to applications in:

- **Media production.** Rendering realistic looking characters and environments are crucial to production of movies and video games.
- **Computer-assisted design.** Nowadays, various products are typically modeled entirely in silico before production. Highly accurate simulation enables *predictive rendering*, which can significantly lower the cost of the design process. Instead of a designer having to create a physical prototype to observe his/her design's appearance, he/she can render the model for feedback, iterate on the design, and then fabricate products that look exactly as simulated.

As mentioned earlier, physically-based rendering depends on mathematical models of physical objects. Each of these *appearance models* must prescribe an



Wood [45]

Animal fur [77]

Paper [86]

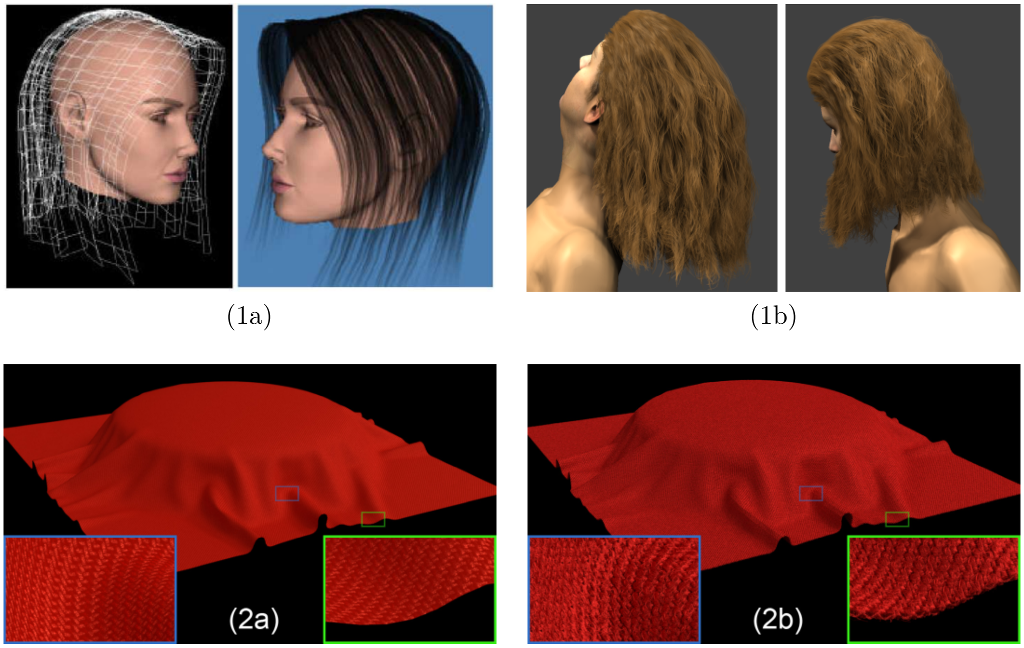
Fabric [89]

Figure 1.1: Examples of fibrous materials.

object's shape and light scattering behavior in such a way that, with it, light transport simulation can be carried out. Creating appearance models that can mimic real world objects is a rich research field in computer graphics because our world is full of objects with interesting shapes and light scattering behavior.

One particular class of such objects are those that are made of small—often microscopic—fibers. It includes natural objects such as wood, fur, and hair; and man-made objects such as paper and textiles. (See Figure 1.1.) These materials have complex appearance because the fibers themselves give rise to complex geometric details, and they scatter light collectively in complex ways.

This dissertation concerns appearance models for hair and textiles—two materials that are integral to the appearance of humans and their surroundings. Despite the fact that they are made of a large number of fibers, appearance models used


(1a)

(1b)

In (1a), the head of hair is approximated by hair strips [95]. With the help of a texture with transparent pixels, an illusion that individual strands are present can be achieve. However, the resulting orderly appearance does not look as realistic as when individual strands are explicitly modeled as in (1b) [10].
The cloth is abstracted to a flat sheet in (2a), but individual fibers are modeled in (2b). Consequently, thickness is present in the latter but not the former. Cloth texture also looks more realistic when fibers are explicitly modeled [109].

Figure 1.2: Modeling hair and textiles with macroscopic shapes (a) versus modeling their individual fibers (b).

in practice often simplify them to their overall, macroscopic shapes. A head of hair is often reduced to the shell that covers the fiber mass or to flat strips representing several nearby hair strands. Cloth is almost always modeled as a flat surface, ignoring the fact that its yarns and fibers give rise to non-flat textures. (Figure 1.2a) While these very simplified models are practical and have been used to great effect in media production, they cannot reproduce all the geometric and optical complexities inherent in fibrous materials. Fuzz and flyaway fibers cannot be represented, and, while some complex reflectance can be approximated, it does not look as realistic when the models are viewed close-up. The inability of surface models to reproduce these details makes them unsuitable for predictive rendering

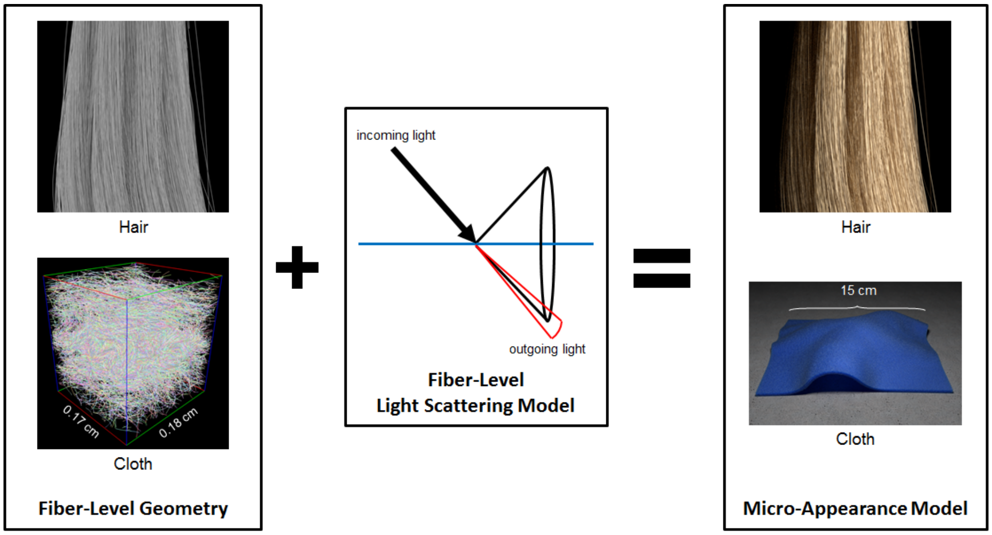


Figure 1.3: Components of a micro-appearance model.

where realism is of great interest.

In pursuit of a greater level of realism, computer graphics researchers have been developing *micro-appearance models*: those that model fibrous materials down to the level of individual fibers. More specifically, such a model must be made of (1) fine-scale geometry where the individual fibers are resolved and (2) models of how the fibers scatter light. (Figure 1.3) The researchers have successfully shown that high-fidelity renderings exhibiting both geometric and optical complexity can be obtained from these models. (Figure 1.2b) In creating and employing them, however, many important research problems must be solved, and many of them are still open.

The first problem is **geometric representation**: how exactly to represent the geometry of fibers. Fiber geometry may be represented by a collection of discrete surfaces [51, 76] or by volumetric grids of density values [35, 75, 109]. Hair strands are almost always represented by surfaces because the fibers are large and are often arranged in non-repeatable ways, making the volumetric representation much less efficient. However, textiles contain small fibers bundled together in repeatable units, so the shapes of individual fibers can afford to be less well-defined. As a result, it is not clear which geometric representation is better: volumes or large collections of micron-resolution surfaces.

The second problem is **optical representation**: how to model how fibers scatter light. Marschner et al. [49] formulated the *bidirectional curve scattering distribution function* (BCSDF), which has become a standard abstraction for a fiber's light scattering behavior. For hair, BCSDFs are often created based on the assumption that hair cross sections are circular [49, 16, 18], but are these models accurate enough given that hair cross sections are generally elliptical [71]? For fabrics, one may use the BCSDF if fibers are represented with surfaces [76], but Zhao et al. [109] also presented a volume-based light scattering model based on the microflake phase function [32]. Which model is better has remained an open problem.

The third problem, crucial to predictive rendering, is **appearance matching**: once the choice of the appearance model has been decided, how to tune its parameters so that the rendered images match the appearance of real materials. In 2009, Bonneel et al. [6] and Zinke et al. [114] presented two solutions for hair. In 2011, Zhao et al. presented one for fabrics [109]. However, these solutions are specific to the models used in their respective papers, and so are not useful as a general modeling tool. Can we design an algorithm that works on arbitrary appearance models?

The fourth problem is **rendering**: how to simulate light transport in micro-appearance models to generate images. While general rendering algorithms such as path tracing can be employed, they are often very time consuming when used with micro-appearance models because they have to trace rays through fine fiber arrangements and do so multiple times per image sample to simulate multiple scattering inside the material. Real-time rendering algorithms that aggressively approximate multiple scattering exist for hair [116, 70, 101]. For fabrics, Zhao et al. proposed *modular flux transfer* in 2013 [108]. However, while it is a magnitude faster than path tracing, it can still take minutes on a large compute cluster. Can we design a faster rendering algorithm for cloth?

1.1 Summary of Contributions

This dissertation presents solutions to the above open problems. Our contributions are as follows:

Our first contribution (Chapter 4) addresses the lack of light scattering models for hair that take into account elliptical cross sections. We present **a new model that takes into account the effects of cross-sectional ellipticity**. We also present **a new, highly efficient measurement device for light scattered from a single hair fiber**, which we used to validate our model. Experimental data showed features that ours predicts but others could not, but they also contained one that no models (include ours) had successfully accounted for.

Our second contribution (Chapter 5) addresses two problems on micro-appearance modeling of fabrics. We propose **an optimization-based framework for estimating parameters of a large class of models** from photographs of real materials. With its help, **we systematically compared competing models and identified the best ones**. For the study, we also invented **a more accurate, yet simple light scattering model for textile fibers** and **an algo-**

rithm for creating surface-based geometric representations of fibers. Our work, taken as a whole, comprises a complete and practical system for appearance modeling of fabrics.

Our third contribution (Chapter 6) addresses the prohibitive cost of rendering fabric micro-appearance models. We present **a precomputation-based, fast, GPU-friendly rendering algorithm** that, using a single commodity GPU, can render high-quality images in tens of seconds that Monte Carlo path tracing and other CPU-based algorithms can take minutes to complete on a large compute cluster. Our algorithm thus makes micro-appearance models practical for applications such as interactive textile design, where rapid turnaround is paramount.

1.2 Organization of the Dissertation

This dissertation is organized as follows.

Chapter 2 reviews the background material that is common to all the following chapters, including the theory of light transport simulation and an abstraction for light scattering behavior of fibers.

Chapter 3 surveys previous approaches to model and render appearance of hair and textile.

Chapter 4 describes our new light scattering model for human hair fibers with elliptical cross section.

Chapter 5 discusses our optimization-based framework for appearance matching and details the study of different competing approaches to micro-appearance modeling. We also describe new components of micro-appearance modeling that we invent for the study.

Chapter 6 describes our fast, GPU-friendly algorithm for approximately rendering fabric micro-appearance models.

Chapter 7 concludes the dissertation and presents potential future works.

CHAPTER 2

BACKGROUND

> ハイハイハイ　分かっています
> 大体の出番くらい
> 背景だし　背景だし
> 歌になってごめんなさい
>
> 背景コンビ,
> 背景放題やりほーだい?

We create appearance models so that we can simulate light transport and generate images of hair and fabrics. This chapter reviews relevant background material on light transport simulation and the light scattering behavior of fibers. We start by stating assumptions about light and light sources (Section 2.1), then discuss the governing equations that light transport simulators aim to solve (Section 2.2). Next, we discuss path tracing, a solution technique on which much of our work is based (Section 2.3). Lastly, we detail the mathematical formulation of the bidirectional curve scattering distribution function (BCSDF), an abstraction for light scattering behavior of fibers that is used throughout this dissertation (Section 2.4).

2.1 Modeling Light

Light is a complex phenomenon exhibiting such diverse behavior that simulating every aspect of it is infeasible. It is thus imperative to work with a simplified model of light that retains only the salient features. Following standard light transport simulation literature [20, 63], we assume *geometric optics*, which maintains the following assumptions:

1. Light travels in a straight line in any medium with a constant index of refraction.
2. Light may be absorbed, reflected, and refracted.
3. Light is unpolarized and incoherent.
4. Light energy does not transfer from one wavelength to another.
5. Light propagates in space and reaches equilibrium state instantaneously.

While the assumptions preclude many interesting effects such as diffraction, interference, polarization, fluorescence, phosphorescence, and relativistic phenomena, they are often adequate for the purpose of synthesizing photorealistic images. They also make simulating light transport much easier. For example, Assumptions 1 and 2 imply that light travels along straight line segments (given that the scene contains only media with constant refractive indices). Assumption 3 enables treating light energy as a scalar. Lastly, Assumption 4 implies that different wavelengths can be treated independently and, consequently, we can act as if there is only one wavelength.

Radiometry

With the model of light specified, we now discuss radiometric quantities involved in light transport simulation. Since the main purpose of this section is to introduce notations and conventions, we refer the reader to Dutre et al. [20] or Preisendorfer [66] for more complete treatments of the concepts.

The goal of light transport simulation is to determine the distribution of light energy in a scene. That is, we would like to answer the following question:

> Given a point $\mathbf{x} \in \mathbb{R}^3$ and a unit vector $\omega \in \mathbb{S}^2$, where $\mathbb{S}^2$ is the unit 2-sphere, what is the amount of light energy flowing along ω from $\mathbf{x}$ per unit time?

The quantity in question is the *outgoing radiance*, which we denote by $L_o(\mathbf{x}, \omega)$. It can be thought of as a function that maps the tuple $(\mathbf{x}, \omega) \in \mathbb{R}^3 \times \mathbb{S}^2$ to a scalar. The tuple $(\mathbf{x}, \omega)$ defines a *ray*, which is the set of points extending from $\mathbf{x}$ along ω; i.e. $\{\mathbf{x} + s\omega : s \in [0, \infty)\}$. As a result, the outgoing radiance can be thought of as a function of rays. The point $\mathbf{x}$ is called the ray's *origin*, and ω is called the ray's *direction*. By convention, ω *always points away from* $\mathbf{x}$. Moreover, we will use the word "direction" interchangeably with the term "unit vector."

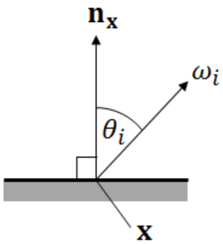
The related quantity *incoming radiance*, denoted by $L_i(\mathbf{x}, \omega)$, is defined to be the light energy that flows to $\mathbf{x}$ in direction $-\omega$. The reversal of direction in here is due to our desire to associate the radiance with the ray $(\mathbf{x}, \omega)$ instead of a ray with direction $-\omega$ whose origin is unspecified. It allows us to simply denote generic radiance associated with ray $(\mathbf{x}, \omega)$ by $L(\mathbf{x}, \omega)$.

Radiance is light power per unit area per unit solid angle, and so its unit is watt per meter square per steradian ($\frac{\mathrm{W}}{\mathrm{m}^2\ \mathrm{sr}}$). It can be shown that, under geometric optics, radiance is conserved along a ray. In other words, $L(\mathbf{x}, \omega) = L(\mathbf{x}', \omega)$ if $\mathbf{x}'$ lies on the ray $(\mathbf{x}, \omega)$, and there is no light interaction at any point along the line segment from $\mathbf{x}$ to $\mathbf{x}'$.

Another function of interest is the *irradiance* I. It maps a point $\mathbf{x}$, located on a surface, to the power of incoming light energy from all directions to $\mathbf{x}$. As it is power per unit area, its unit is watt per meter squared ($\frac{\mathrm{W}}{\mathrm{m}^2}$). It is related to radiance by the following *Lambert's cosine law*:

$$\mathrm{d}I(\mathbf{x},\omega_i) = L_i(\mathbf{x},\omega_i)\cos\theta_i\ \mathrm{d}\omega_i$$

where $\mathrm{d}I(\mathbf{x},\omega)$ denotes the infinitesimal irradiance due to the incoming radiance along ω_i, and θ_i denotes the angle ω_i makes with the normal vector $\mathbf{n_x}$ of the surface at point $\mathbf{x}$. Intuitively, the cosine factor arises to correct for the fact that the radiance $L_i(\mathbf{x},\omega_i)$ is defined with respect to the infinitesimal area that is perpendicular to the direction ω instead of the area on which $\mathbf{x}$ is located, which is perpendicular to $\mathbf{n_x}$.



Light Sources

To start light transport simulation, some scene elements must generate light energy. We assume two types of light sources:

1. **Area light source.** For any point $\mathbf{x}$ on a surface in the scene and any direction ω, we let $L_e(\mathbf{x},\omega)$ denote the radiance *emitted* by the surface from $\mathbf{x}$ in direction ω. If the surface does not emit light along the ray $(\mathbf{x},\omega)$, then $L_e(\mathbf{x},\omega) = 0$.
2. **Environment light source.** From any point $\mathbf{x}$ in the scene and for any direction ω, if the ray $(\mathbf{x},\omega)$ is not occluded by any scene geometry, then the point receives incoming radiance of $L_{env}(\omega)$.

The above do not include light sources with delta distributions such as point lights and directional lights. However, because they can be approximated by small area light sources or environment light sources with small support, our assumptions do not introduce significant limitations.

2.2 Light Transport Equations

Light transport simulation is formulated in terms of finding a solution to an equation that governs how light interacts with matter. The equation used depends on the presumed form of matter in the scene. When matter is abstracted with surface elements, the *rendering equation* [36] is used. When matter is abstracted with a volume of light scattering particles, the *radiative transfer equation* (RTE) [12] is used. In this section, we discuss the two equations in order.

2.2.1 Rendering Equation

The rendering equation applies in scenes populated by objects with well-defined surfaces. Space between and inside the objects are filled with vacuum or media that do not interact with light; for example, clear gas or liquid. In such a scene, light interaction only happens on object surfaces.

The rendering equation is given by:

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega_{\mathbf{x}}} f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) \cos\theta_i \ \mathrm{d}\omega_i. \tag{2.1}$$

According to the equation, the outgoing radiance $L(\mathbf{x}, \omega_o)$ from a surface point $\mathbf{x}$ in direction ω_o can be separated into two components: the *emitted* radiance $L_e(\mathbf{x}, \omega_o)$ and the *reflected* radiance, which we shall denote by $L_r(\mathbf{x}, \omega_o)$. The latter is given by an integral that gathers incoming radiance from all directions that are "above" the surface and convert it to outgoing radiance. More specifically, the integration domain $\Omega_{\mathbf{x}}$ is the hemisphere oriented along $\mathbf{n}_{\mathbf{x}}$; in other words, $\Omega_{\mathbf{x}} = \{\omega \in \mathbb{S}^2 : \omega \cdot \mathbf{n}_{\mathbf{x}} \geq 0\}$. The function f_r, called the *bidirectional reflectance distribution function* (BRDF), encapsulates the light reflecting behavior of the material at $\mathbf{x}$.

Mathematically, $f_r(\mathbf{x}, \omega_i, \omega_o)$ gives the infinitesimal outgoing radiance in direction ω_o produced by reflecting the infinitesimal irradiance coming from direction ω_i:

$$f_r(\mathbf{x}, \omega_i, \omega_o) = \frac{\mathrm{d}L_r(\mathbf{x}, \omega_o)}{\mathrm{d}I(\mathbf{x}, \omega_i)} = \frac{\mathrm{d}L_r(\mathbf{x}, \omega_o)}{L_i(\mathbf{x}, \omega_i) \cos\theta_i \ \mathrm{d}\omega_i}.$$

Intuitively, the integrand is the result of reflecting radiance from direction ω_i, and the integral gathers the result from all relevant directions.

In the way that the equation is stated in (2.1), there is an implicit assumption that the material is opaque, so there can be no light coming from below the surface. This assumption breaks down when we deal with refractive materials such as glass. In this case, we may generalize the BRDF to the *bidirectional scattering distribution function* (BSDF), which takes into account light coming from both sides. The rendering equation remains the same except for the fact that the $\cos\theta_i$ term becomes $|\cos\theta_i|$ to take into account the larger range of θ_i. However, to simplify subsequent discussions, we will not make this generalization in this dissertation.

The first step towards solving the rendering equation is to reduce the number of unknowns. Currently, two are present: L_o and L_i. We can write L_i in terms of L_o with the help of the *ray tracing function* $\mathfrak{R}(\mathbf{x}, \omega)$, which returns the first surface point $\mathbf{y}$ that the ray $(\mathbf{x}, \omega)$ intersects as we extend it from $\mathbf{x}$ or returns a special symbol $\varnothing$ if the ray does not intersect any surface:

$$L_i(\mathbf{x}, \omega_i) = \begin{cases} L_o(\mathbf{y}, -\omega_i), & \mathfrak{R}(\mathbf{x}, \omega_i) = \mathbf{y} \\ L_{\mathrm{env}}(\omega_i), & \mathfrak{R}(\mathbf{x}, \omega_i) = \varnothing \end{cases}. \tag{2.2}$$

In other words, if the ray hits a surface, the incoming radiance is the outgoing light from the hit point; otherwise, the radiance comes from environment illumination. We will discuss how (2.2) is used to solve the rendering equation in Section 2.3.2.

2.2.2 Radiative Transfer Equation

While the rendering equation governs light transport in scenes where light scattering happens only on surfaces of scene objects, the radiative transfer equation governs light transport in media inhabited by light-scattering particles. It can be used to render translucent solids (soap, skin, jade), colored liquids (milk, orange juice, wine), and gaseous phenomena (fog, smoke, clouds) [11, 23].

The RTE specifies the rate of change of radiance along a ray as an observer moves along it. We consider the ray $(\mathbf{x}_0, \omega_o)$ and parameterize it with the function $\mathbf{x}(s) = \mathbf{x}^* + s\omega_o$, where s denotes the distance from the origin $\mathbf{x}^*$. At a high level, the RTE is given by:

$$\frac{\mathrm{d}L_o(\mathbf{x}, \omega_o)}{\mathrm{d}s} = -(\text{absorption}) - (\text{out-scattering}) + (\text{in-scattering}) + (\text{emission}). \tag{2.3}$$

It says that radiance change is due to 4 types of light interactions with matter:

- **Absorption**. Light collides with matter and is converted to other forms of energy, such as heat.
- **Out-scattering**. Light along ω_o collides with matter and changes direction.
- **In-scattering**. Light along other directions collides with matter and changes direction to ω_o.
- **Emission**. Matter itself emits light into the environment.

The RTE thus calls for a *volumetric model* which characterizes the 4 interactions at all points in space. Typically, the interactions are functions of mainly 3D positions. However, when rendering hair and textile fibers, it is important that they also take the light's direction into account, so we will follow the treatment by Jakob et al. [32], which incorporates full directional dependency into the RTE. According to their treatment, the functions are:

- The *absorption coefficient* $\sigma_a(\mathbf{x}, \omega_o)$ specifies the fraction of radiance along ω_o that is absorbed by matter per unit length. In other words:

  $$\text{(absorption)} = \sigma_a(\mathbf{x}, \omega_o) L_o(\mathbf{x}, \omega_o).$$

- The *scattering coefficient* $\sigma_s(\mathbf{x}, \omega_o)$ specifies the fraction of radiance along ω_o that changes direction per unit length. In other words:

  $$\text{(out-scattering)} = \sigma_s(\mathbf{x}, \omega_o) L_o(\mathbf{x}, \omega_o).$$

- The *phase function* $f_p(\mathbf{x}, \omega_i, \omega_o)$ specifies the proportion of radiance from ω_i that scatters to direction ω_o relative to radiance from all directions that scatters to ω_o, given that the point $\mathbf{x}$ is illuminated by a field constant radiance from all directions. It is by definition a probability distribution over ω_i. Using the phase function, the in-scattering term is given by:

  $$\text{(in-scattering)} = \sigma_s(\mathbf{x}, \omega_o) \int_{\mathbb{S}^2} f_p(\mathbf{x}, \omega_i, \omega_o) L_o(\mathbf{x}, \omega_i)\ \mathrm{d}\omega_i.$$

  We note that the above formulation is counterintuitive. It is only valid when σ_s and f_p satisfy the following reciprocity relation:

  $$\sigma_s(\mathbf{x}, \omega_o) f_p(\mathbf{x}, \omega_i, \omega_o) = \sigma_s(\mathbf{x}, \omega_i) f_p(\mathbf{x}, \omega_o, \omega_i).$$

  We refer the reader to Jakob et al. [32] for details.

- The *emission coefficient* $Q(\mathbf{x}, \omega_o)$ specifies the rate of increase in radiance along the ray due to the matter at $\mathbf{x}$ emitting light. In other words,

  $$\text{(emission)} = Q(\mathbf{x}, \omega_o).$$

Expanding the terms in (2.3), the RTE is given by:

$$\begin{aligned}\frac{\mathrm{d}L(\mathbf{x},\omega_o)}{\mathrm{d}s} &= -\sigma_a(\mathbf{x},\omega_o)L(\mathbf{x},\omega_o) - \sigma_s(\mathbf{x},\omega_o)L(\mathbf{x},\omega_o) \\ &\qquad + \sigma_s(\mathbf{x},\omega_o)\int_{\mathbb{S}^2} f_p(\mathbf{x},\omega_i,\omega_o)L_o(\mathbf{x},\omega_i)\ \mathrm{d}\omega_i + Q(\mathbf{x},\omega_o).\end{aligned}$$

We further make a number of simplifications. First, because our light source models in Section 2.1 do not include volumetric light sources, we may set $Q(\mathbf{x},\omega_o)$ to zero and effectively drop that term from the RTE. Second, we define the *extinction coefficient* $\sigma_t(\mathbf{x},\omega_o) := \sigma_a(\mathbf{x},\omega_o) + \sigma_s(\mathbf{x},\omega_o)$ to group similar terms in the RTE together. Third, we assume that the ratio $\sigma_s(\mathbf{x},\omega)/\sigma_t(\mathbf{x},\omega)$ does not depend on ω, and define the *scattering albedo function* $\alpha(\mathbf{x}) := \sigma_s(\mathbf{x},\omega)/\sigma_t(\mathbf{x},\omega)$. Intuitively, the extinction coefficient specifies the density of matter, and the scattering albedo specifies its color. Requiring the scattering albedo to be invariant of direction means requiring the material to have the same color when looked at from all directions, which is sensible for most media. With the three simplifications, we may rewrite the RTE as:

$$\frac{\mathrm{d}L_o(\mathbf{x},\omega_o)}{\mathrm{d}s} = -\sigma_t(\mathbf{x},\omega_o)L(\mathbf{x},\omega_o) + \alpha(\mathbf{x})\sigma_t(\mathbf{x},\omega_o)\int_{\mathbb{S}^2} f_p(\mathbf{x},\omega_i,\omega_o)L_o(\mathbf{x},\omega_i)\ \mathrm{d}\omega_i.$$

The above is a first-order ordinary differential equation, which we can solve to yield the following *volume rendering equation*:

$$\begin{aligned}L_o(\mathbf{x},\omega_o) &= \int_0^s \tau(\mathbf{x}',\mathbf{x})\alpha(\mathbf{x}')\sigma_t(\mathbf{x}',\omega_o)\bigg(\int_{\mathbb{S}^2} f_p(\mathbf{x}',\omega_i,\omega_o)L_o(\mathbf{x}',\omega_i)\ \mathrm{d}\omega_i\bigg)\ \mathrm{d}s' \\ &\qquad + \tau(\mathbf{x}^*,\mathbf{x})L_o(\mathbf{x}^*,\omega_o) \qquad (2.4)\end{aligned}$$

where $\mathbf{x}' = \mathbf{x}^* + s'\omega_o$, and

$$\tau(\mathbf{a},\mathbf{b}) = \exp\Bigg(-\int_0^{\|\mathbf{b}-\mathbf{a}\|} \sigma_t\Big(\mathbf{a} + u\frac{\mathbf{b}-\mathbf{a}}{\|\mathbf{b}-\mathbf{a}\|}, \frac{\mathbf{b}-\mathbf{a}}{\|\mathbf{b}-\mathbf{a}\|}\Big)\ \mathrm{d}u\Bigg)$$

is the *transmittance* between $\mathbf{a}$ and $\mathbf{b}$, which is the fraction of radiance from $\mathbf{a}$ that survives absorption and out-scattering when it reaches $\mathbf{b}$.

2.3 Path Tracing

Path tracing is an algorithm for solving the two light transport equations that is often regarded as the "reference" solution to rendering problems in the computer graphics community. It works with the integral forms of the equations, namely (2.1) and (2.4), and employs *Monte Carlo integration* to compute a stochastic estimate of the integrals. As it estimates the integrals, it effectively traces paths a photon may follow *backward* from the camera to a light source.

In this section, we will review the concept of Monte Carlo integration first and then detail how path tracing is formulated to solve each of the two light transport equations.

2.3.1 Monte Carlo Integration

Monte Carlo integration is a technique for probabilistically estimating the value of a definite integral $\int_A f(x)\ \mathrm{d}x$ of some real function f over some domain A. The idea is to construct a random variable F according to the following random process:

1. Sample a value x according to a probability distribution p.
2. Set $F := f(x)/p(x)$.

A nice property of F is that it is an unbiased estimate of the integral:

$$E[F] = \int_A p(x) \frac{f(x)}{p(x)}\ \mathrm{d}x = \int_A f(x)\ \mathrm{d}x.$$

However, F might be noisy; in other words, it might have high variance. The first approach to reduce variance is to construct *independent, identically distributed* (i.i.d) random variables F_1, F_2, $\dots$, F_N, each having the same distribution as F, and compute their mean: $\bar{F}_N = (\sum_{i=1}^{N} F_i)/N$. The mean $\bar{F}_N$ will be much less noisy than F because the variance drops by a factor of $1/N$: $\mathrm{Var}(\bar{F}_N) = \mathrm{Var}(F)/N$. Another approach to reduce variance is *importance sampling*: carefully picking the distribution p so that it approximates f well. The two approaches are orthogonal and so can be applied in tandem.

When solving light transport equations, we would like to compute the incoming radiance $L_i(\mathbf{x}_o, \omega_0)$ where $\mathbf{x}_0$ is a point on the camera's sensor. We typically imagine a space $\mathbb{P}$ of all possible paths that a photon may follow from when it is emitted from a light source until it reaches $\mathbf{x}_0$. More formally, a path $\vec{\mathbf{x}}$ is a sequence of 3D positions alternated with directions $(\mathbf{x}_0, \omega_0, \mathbf{x}_1, \omega_1, \mathbf{x}_2, \dots, \mathbf{x}_{n-2}, \omega_{n-2}, \mathbf{x}_{n-1})$ where

- $\mathbf{x}_0$ is a point on the camera's sensor,
- $\mathbf{x}_{n-1}$ is a point on a light source,
- all other points in between are locations where the photon interacts with matter, and
- ω_i denotes the direction from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$.

(See Figure 2.1.) Each path has its associated *contribution* $f(\vec{\mathbf{x}})$ which indicates the amount of radiance the photon, traveling along the path, contributes to $L_i(\mathbf{x}, \omega_i)$. The incoming radiance is then given by:

$$L_i(\mathbf{x}, \omega_i) = \int_{\mathbb{P}} f(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}}. \tag{2.5}$$

Path tracing describes an algorithm for constructing an unbiased estimator $\widetilde{L}_i$ for the above integral. As a result, we can obtain a low-variance estimate of $L_i(\mathbf{x}, \omega_i)$ by computing the mean of the results obtained by running it several

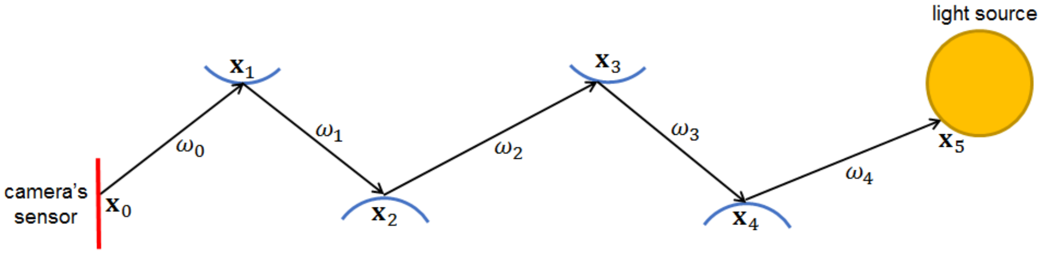


Figure 2.1: A path with 6 vertices with its components marked.

times and averaging the results: $L_i(\mathbf{x}_0, \omega_0) \approx (\sum_{j=1}^{N} \widetilde{L}_{i,j})/N$. A straightforward application of Monte Carlo integration suggests that sampling a path $\vec{\mathbf{x}}$ according to some probability distribution $p(\vec{\mathbf{x}})$ and computing $f(\vec{\mathbf{x}})/p(\vec{\mathbf{x}})$ is an unbiased estimate of $L_i(\mathbf{x}_0, \omega_0)$. However, the variant of path tracing used in this dissertation samples multiple related-paths instead of a single one.

As it is very hard to describe path tracing without referring to the equation it tries to solve, we will describe path tracing when applied to the rendering equation in Section 2.3.2. The description will cover most concepts that path tracing employs, and we will adapt these concepts to solving the radiative transfer equation in Section 2.3.3.

2.3.2 Surface Path Tracing

We now discuss how to derive an unbiased estimate of $L_i(\mathbf{x}_0, \omega_0)$ in surface-based scenes. Recall that, in this context, the incoming radiance $L_i(\mathbf{x}_0, \omega_0)$ can be rewritten in terms of the outgoing radiance and radiance from the environment light source as in (2.2). As a result, the first step in computing $L_i(\mathbf{x}_0, \omega_0)$ is to cast a ray from $\mathbf{x}_0$ in direction ω_0. If the ray does not hit any surface, we may return $L_{env}(\omega_0)$ and be done. If it hits a surface point $\mathbf{x}_1$, we have to compute $L_o(\mathbf{x}_1, -\omega_0)$,

```
Surface-Path-Tracing($\mathbf{x}_0, \omega_0$)
Cast a ray from $\mathbf{x}_0$ in direction $\omega_0$ and then determine the hit point $\mathbf{x}_1$.
if $\mathbf{x}_1 = \varnothing$
        $\widetilde{L}_i^{\text{dir}} = L_{env}(\omega_o)$
else
        $\widetilde{L}_i^{\text{dir}} = L_e(\mathbf{x}_1, -\omega_0)$
return $\widetilde{L}_i^{\text{dir}}$ + Surface-Reflected-Radiance($\mathbf{x}_1, -\omega_0$)
```

Figure 2.2: Pseudocode of the surface path tracing algorithm.

which, according to the rendering equation, is equal to:

$$L_o(\mathbf{x}_1, -\omega_0) = L_e(\mathbf{x}_1, -\omega_0) + \int_{\Omega_{\mathbf{x}_1}} f_r(\mathbf{x}_1, \omega_1, -\omega_o) L_i(\mathbf{x}_1, \omega_1) \cos\theta_1 \, \mathrm{d}\omega_1. \tag{2.6}$$

Since computing the emission term is easy (it is included in the scene description), the main task of path tracing is to compute the integral on the RHS.

Before we proceed with computing the integral, let us take stock and put what we have described in a more algorithmic from. We shall encapsulate the surface path tracing algorithm with a function called Surface-Path-Tracing that takes a point $\mathbf{x}_0$ and ω_0 as input and returns $L_i(\mathbf{x}_0, \omega_0)$. The core of the algorithm is the estimation of reflected radiance, which we shall abstract with the function Surface-Reflected-Radiance($\mathbf{x}_j, -\omega_j$). This function takes a point $\mathbf{x}_j$ on a surface, and a direction $-\omega_{j-1}$ as input, and it should return $L_r(\mathbf{x}_j, -\omega_{j-1})$. (We use the generic index j instead of 1 because this function will be called recursively.) The pseudocode for Surface-Path-Tracing is given in Figure 2.2.

Reflected Radiance

Now, we proceed to estimate the reflected radiance. In this dissertation, we use a variant of path tracing that employs *next event estimation*, a technique which significantly reduces variance when light sources are small. It splits the incoming radiance into two components: $L_i(\mathbf{x}_j, \omega_j) = L_i^{\mathrm{dir}}(\mathbf{x}_j, \omega_j) + L_i^{\mathrm{indir}}(\mathbf{x}_j, \omega_j)$. The *direct* incoming radiance $L_i^{\mathrm{dir}}(\mathbf{x}_j, \omega_j)$ comes directly from a light source without being scattered by matter; namely,

$$L_i^{\mathrm{dir}}(\mathbf{x}_j, \omega_j) = \begin{cases} L_{env}(\omega_j), & \mathfrak{R}(\mathbf{x}_j, \omega_j) = \varnothing \\ L_e(\mathbf{x}_{j+1}, -\omega_j), & \mathfrak{R}(\mathbf{x}_j, \omega_j) = \mathbf{x}_{j+1} \end{cases}.$$

The *indirect* incoming radiance $L_i^{\mathrm{indir}}(\mathbf{x}, \omega_i)$ is incoming radiance that is scattered at least once by matter. In other words,

$$L_i^{\mathrm{indir}}(\mathbf{x}_j, \omega_j) = \begin{cases} 0, & \mathfrak{R}(\mathbf{x}_j, \omega_j) = \varnothing \\ L_r(\mathbf{x}_{j+1}, -\omega_j), & \mathfrak{R}(\mathbf{x}_j, \omega_j) = \mathbf{x}_{j+1} \end{cases}$$

With the splitting of the incoming radiance, we may rewrite the reflected radiance as follows:

$$\begin{aligned} L_r(\mathbf{x}_j, -\omega_{j-1}) &= \int_{\Omega_{\mathbf{x}_j}} f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1})(L_i^{\mathrm{dir}}(\mathbf{x}_j, \omega_j) + L_i^{\mathrm{indir}}(\mathbf{x}_j, \omega_j)) \cos\theta_j \ \mathrm{d}\omega_j \\ &= \int_{\Omega_{\mathbf{x}_j}} f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) L_i^{\mathrm{dir}}(\mathbf{x}_j, \omega_j) \cos\theta_j \ \mathrm{d}\omega_j && (2.7) \\ &\quad + \int_{\Omega_{\mathbf{x}_j}} f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) L_i^{\mathrm{indir}}(\mathbf{x}_j, \omega_j) \cos\theta_j \ \mathrm{d}\omega_j && (2.8) \end{aligned}$$

We approximate (2.7) by sampling a direction ω_j' towards a light source in the scene according to some probability distribution $p_j'(\omega_j')$. (We refer the reader to Pharr et al.'s book [63] for how to sample directions to light sources.) With ω_j', an unbiased estimate of the integral is given by:

$$\begin{aligned} &\int_{\Omega_{\mathbf{x}_j}} f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) L_i^{\mathrm{dir}}(\mathbf{x}, \omega_j) \cos\theta_j \ \mathrm{d}\omega_j \\ &\qquad \approx \frac{f_r(\mathbf{x}_j, \omega_j', -\omega_{j-1}) L_i^{\mathrm{dir}}(\mathbf{x}_j, \omega_j') \cos\theta_j'}{p_j'(\omega_j')}. \end{aligned}$$

We note that the variance of the estimate can be reduced through *multiple importance sampling*: instead of estimating (2.7) with a single directional sample, multiple samples are taken according to different probability distributions, and the estimates according to these samples are combined with the help of carefully chosen weights [92]. However, we do not include the technique in our treatment to keep it simple.

For the integral (2.8), we sample a direction ω_j according to another probability distribution $p_j(\omega_j)$, which is typically chosen to be proportional to $f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) \cos\theta_j$. A Monte Carlo estimate of the integral is given by:

$$\begin{aligned}\int_{\Omega_{\mathbf{x}_j}} & f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) L_i^{\mathrm{indir}}(\mathbf{x}_j, \omega_j) \cos\theta_j \; \mathrm{d}\omega_j \\ & \approx \frac{f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) L_i^{\mathrm{indir}}(\mathbf{x}_j, \omega_j) \cos\theta_j}{p_j(\omega_j)}.\end{aligned}$$

To compute L_i^{indir}, we cast the ray $(\mathbf{x}_j, \omega_j)$ and find its hit point $\mathbf{x}_{j+1}$. If $\mathbf{x}_{j+1} = \varnothing$, then $L_i^{indir}(\mathbf{x}_j, \omega_j)$ is equal to zero. Otherwise, $L_i^{indir}(\mathbf{x}_j, \omega_j)$ is given by $L_r(\mathbf{x}_{j+1}, -\omega_j)$, which we can approximate by recursively invoking $\textsc{Surface-Reflected-Radiance}(\mathbf{x}_{j+1}, -\omega_j)$; in other words,

$$L_i^{\mathrm{indir}}(\mathbf{x}_j, \omega_j) \approx \begin{cases} 0, & \mathfrak{R}(\mathbf{x}_j, \omega_j) = \varnothing \\ \textsc{Surface-Reflected-Radiance}(\mathbf{x}_{j+1}, -\omega_j), & \mathfrak{R}(\mathbf{x}_j, \omega_j) = \mathbf{x}_{j+1} \end{cases}.$$

While the estimate of (2.7) may be computed with multiple samples, it is important that the estimate of (2.8) be computed with only one sample. Otherwise, the recursion tree's size will grow exponentially with depth, making computing the estimate too computationally expensive.

Terminating the Recursion

The above recursion, however, leads to an infinite loop if the camera is located inside a closed surface. To make sure it terminates, we employ the *Russian roulette estimator*. Given a random variable X, its Russian roulette estimator with *termination probability* $0 < p_t < 1$ is the random variable

$$Y = \begin{cases} 0, & \text{with probability } p_t \\ X/(1-p_t), & \text{with probability } 1-p_t \end{cases}.$$

Because

$$E[Y] = p_t \cdot 0 + (1-p_t)E[X]/(1-p_t) = E[X],$$

Y is an unbiased estimate of X. So, to make sure that the recursion terminates with probability 1, we compute the Russian roulette estimate of (2.8) instead of the canonical Monte Carlo estimate we discussed before. More specifically,

$$\int_{\Omega_{\mathbf{x}_j}} f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) L_i^{\text{indir}}(\mathbf{x}_j, \omega_j) \cos\theta_j \; \mathrm{d}\omega_j$$
$$\approx \begin{cases} 0, & \text{with probability } p_t \\ \frac{f_r(\mathbf{x}_j,\omega_j,-\omega_{j-1}) L_i^{\text{indir}}(\mathbf{x}_j,\omega_j)\cos\theta_j}{(1-p_t)p_j(\omega_j)}, & \text{with probability } 1-p_t \end{cases}.$$

Here, the probability that the recursion does not terminate after k recursive calls is $(1-p_t)^k$, which tends to 0 as k tends to ∞. The pseudocode for Surface-Reflected-Radiance with Russian roulette is given in Figure 2.3.

Sampled Paths

It is instructive to identify the paths sampled by the path tracing algorithm. Consider the pseudocode in Figure 2.2. The variable $\widetilde{L}_i^{\text{dir}}$ contains an estimate of the

```
SURFACE-REFLECTED-RADIANCE($\mathbf{x}_j, -\omega_{j-1}$)
    ▷ Reflected radiance due to direct incoming radiance
    Sample direction $\omega'_j$ with probability $p'_j(\omega'_j)$.
    Cast ray $(\mathbf{x}_j, \omega'_j)$ and determine the hit point $\mathbf{x}'_{j+1}$.
    if $\mathbf{x}'_{j+1} = \varnothing$
        $\widetilde{L}_i^{\text{dir}} = L_{env}(\omega'_j)$
    else
        $\widetilde{L}_i^{\text{dir}} = L_e(\mathbf{x}'_{j+1}, \omega'_j)$
    $\widetilde{L}_r^{\text{dir}} = \widetilde{L}_i^{\text{dir}} f_r(\mathbf{x}_j, \omega'_j, -\omega_{j-1}) \cos\theta'_j / p'_j(\omega'_j)$
    ▷ Reflected radiance due to indirect incoming radiance
    Toss a coin that shows head with probability $p_t$.
    if the coin shows head
        return $\widetilde{L}_r^{\text{dir}}$
    else
        Sample direction $\omega_j$ with probability $p_j(\omega_j)$
        Cast ray $(\mathbf{x}_j, \omega_j)$ and determine the hit point $\mathbf{x}_{j+1}$.
        if $\mathbf{x}_{j+1} = \varnothing$
            $\widetilde{L}_i^{\text{indir}} = 0$
        else
            $\widetilde{L}_i^{\text{indir}} =$ SURFACE-REFLECTED-RADIANCE($\mathbf{x}_{j+1}, -\omega_j$)
        return $\widetilde{L}_r^{\text{dir}} + \widetilde{L}_i^{\text{indir}} f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) \cos\theta_j / ((1 - p_t) p_j(\omega_j))$
```

Figure 2.3: Pseudocode of the algorithm to compute reflected radiance in surface-based scenes.

contribution of paths of the form $(\mathbf{x}_0, \omega_0, \mathbf{x}_1)$. However, there is only one such path, and its contribution is given by $L_e(\mathbf{x_1}, -\omega_0)$. (Here, we abuse the notation and let $L_e(\mathbf{x_1}, -\omega_0) = L_{env}(\omega_0)$ if $\mathbf{x_1} = \varnothing$.) On the other hand, in Figure 2.3, the variable $\widetilde{L}_i^{\text{dir}}$ contains an estimate of the contribution of paths of the form $(\mathbf{x}_0, \omega_0, \mathbf{x}_1, \ldots, \mathbf{x}_j, \omega'_j, \mathbf{x}'_{j+1})$. The sampled path's contribution and probability are given by:

$$\begin{aligned} f(\mathbf{x}_0, \omega_0, \mathbf{x}_1, \ldots, \mathbf{x}_j, \omega'_j, \mathbf{x}'_{j+1}) &= \left(\prod_{k=1}^{j-1} f_r(\mathbf{x}_k, \omega_k, -\omega_{k-1}) \cos\theta_k \right) \\ &\qquad f_r(\mathbf{x}_j, \omega'_j, -\omega_{j-1})(\cos\theta'_j) L_e(\mathbf{x}'_{j+1}, -\omega'_j) \\ p(\mathbf{x}_0, \omega_0, \mathbf{x}_1, \ldots, \mathbf{x}_j, \omega'_j, \mathbf{x}'_{j+1}) &= (1-p_t)^{j-1} \left(\prod_{k=1}^{j-1} p_k(\omega_k) \right) p'_j(\omega'_j) \end{aligned}$$

As a result of the above observations, we can say that path tracing decomposes the space into subspaces of paths according to their lengths: $\mathbb{P} = \mathbb{P}_1 \cup \mathbb{P}_2 \cup \mathbb{P}_3 \cup \cdots$, where $\mathbb{P}_k$ denotes the set of all paths from the form $(\mathbf{x}_0, \omega_0, \mathbf{x}_1, \omega_1, \ldots, \mathbf{x}_{k-1}, \omega_{k-1}, \mathbf{x}_k)$ having exactly $k+1$ vertices. Accordingly, the radiance $L_i(\mathbf{x}_0, \omega_0)$ may be written as:

$$\begin{aligned} L_i(\mathbf{x}_0, \omega_0) &= \int_{\mathbb{P}} f(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}} \\ &= \int_{\mathbb{P}_1} f(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}} + \int_{\mathbb{P}_2} f(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}} + \int_{\mathbb{P}_3} f(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}} + \cdots. \end{aligned}$$

In the call to Surface-Path-Tracing, the integral involving $\mathbb{P}_1$ is evaluated exactly. Then, Surface-Reflected-Radiance is called recursively $J+1$ times where J is the number of times the coins in Line 9 of Figure 2.3 shows up tail before showing head. (In other words, J is a geometric random variable with parameter p_t.) The call to Surface-Reflected-Radiance$(\mathbf{x}_j, -\omega_j)$ estimates the integral involving P_{j+1} with $f(\mathbf{x}_0, \ldots, \mathbf{x}'_{j+1}) / p(\mathbf{x}_0, \ldots, \mathbf{x}'_{j+1})$. Hence, the estimate that path tracing computes is:

$$\begin{aligned} L_i(\mathbf{x}_0, \omega_0) &\approx L_e(\mathbf{x}_1, -\omega_0) \\ &\qquad + \sum_{j=1}^{J+1} \left(\frac{\prod_{k=1}^{j-1} f_r(\mathbf{x}_k, \omega_k, -\omega_{k-1}) \cos\theta_k}{(1-p_t)^{j-1} \prod_{k=1}^{j-1} p_j(\omega_j)} \ \frac{f_r(\mathbf{x}_j, \omega'_j, -\omega_{j-1})(\cos\theta'_j)}{p'_j(\omega'_j)} \right). \end{aligned}$$

Non-Recursive Path Tracing

With the knowledge of the forms, contributions, and probabilities of paths that path tracing samples, we can rewrite the algorithm to be non-recursive. The algorithm takes the form of a loop in which it first estimates the direct incoming radiance (Line 1 to 7 of Figure 2.3) and then samples the direction ω_j to extend the path (Line 12 to 13 of Figure 2.3). The loop terminates when the Russian roulette fires. Throughout its lifetime, we maintain the *throughput* variable:

$$T_j := \frac{\prod_{k=1}^{j-1} f_r(\mathbf{x}_k, \omega_k, -\omega_{k-1}) \cos\theta_k}{(1-p_t)^{j-1} \prod_{k=1}^{j-1} p_j(\omega_j)},$$

which is used to scale the BRDF's response of the incoming direct radiance before adding it to the running estimate of $L_i(\mathbf{x}, \omega_0)$. The pseudocode of the non-recursive version is given in Figure 2.4.

2.3.3 Volume Path Tracing

We now turn our attention to how to apply path tracing to the radiative transfer equation. To simplify the derivation, we assume that the scene contains only volume-based models of matter (Section 2.2.2) and no surface-based ones (Section 2.2.1). As a consequence, the only light source available is the environment light source. We refer the reader to Raab et al. for how to perform path tracing in a scene with both types of models [67].

Again, given a point $\mathbf{x}_0$ and direction ω_0 pointing out of it, we would like to find $L_i(\mathbf{x}_0, \omega_0)$, which is the same as $L_o(\mathbf{x}_0, -\omega_0)$. Fortunately, we shall see that this quantity can be computed recursively, and so let us abstract the volume path tracing processing with the function Volume-Path-Tracing$(\mathbf{x}_j, -\omega_j)$, which

```
Non-Recursive-Surface-Path-Tracing($\mathbf{x}_0, \omega_0$)
    ▷ Compute direct radiance to the camera
    Cast ray from $(\mathbf{x}_0, \omega_0)$ to determine the hit point $\mathbf{x}_1$.
    if $\mathbf{x}_1 = \varnothing$
        return $L_{env}(\omega_o)$
    $\widetilde{L}_i = L_e(\mathbf{x}_1, -\omega_0)$
    $j = 1$ and $T_j = 1$
    while true
        ▷ Reflected radiance due to direct incoming radiance
        Sample direction $\omega'_j$ with probability $p'_j(\omega'_j)$.
        Cast ray $(\mathbf{x}_j, \omega'_j)$ and determine the hit point $\mathbf{x}'_{j+1}$.
        if $\mathbf{x}'_{j+1} = \varnothing$
            $\widetilde{L}_i^{\mathrm{dir}} = L_{env}(\omega'_j)$
        else
            $\widetilde{L}_i^{\mathrm{dir}} = L_e(\mathbf{x}'_{j+1}, \omega'_j)$
        $\widetilde{L}_i = \widetilde{L}_i + T_j \widetilde{L}_i^{\mathrm{dir}} f_r(\mathbf{x}_j, \omega'_j, -\omega_{j-1}) \cos\theta'_j / p'_j(\omega'_j)$
        ▷ Russian roulette
        Toss a coin that shows head with probability $p_t$.
        if the coin shows head
            break
        ▷ Sample a new direction to extend the path
        Sample direction $\omega_j$ with probability $p_j(\omega_j)$
        Cast ray $(\mathbf{x}_j, \omega_j)$ and determine the hit point $\mathbf{x}_{j+1}$.
        if $\mathbf{x}_{j+1} = \varnothing$
            break
        else
            $T_{j+1} = T_j f_r(\mathbf{x}_j, \omega_j, -\omega_{j-1}) \cos\theta_j / ((1 - p_t) p_j(\omega_j))$
        $j = j + 1$
    return $\widetilde{L}_i$
```

Figure 2.4: Pseudocode of the non-recursive version of surface path tracing.

outputs $L_o(\mathbf{x}_j, -\omega_j)$.

We would like to apply the volume rendering equation (2.4) to compute $L_o(\mathbf{x}_j, -\omega_j)$. Before doing so, however, we must determine what the ray origin $\mathbf{x}^*$ is. Since the scene does not have any surface that can occlude the environment light source, the ray origin $\mathbf{x}^*$ should be the point at infinity: $\mathbf{x}_j^* = \lim_{s\to\infty}(\mathbf{x}_j + s\omega_j)$. As a result, we have that $L_o(\mathbf{x}_j^*, -\omega_j) = L_{env}(\omega_j)$. We also need to rewrite the outer integral in (2.4) so that the variable of integration measures the distance from $\mathbf{x}_j$ instead of the one from $\mathbf{x}^*$, which is not well-defined. Making the assumption that $\sigma_t(\mathbf{x}, \omega) = \sigma_t(\mathbf{x}, -\omega)$ for all $\mathbf{x}$ and ω, the volume rendering equation becomes:

$$\begin{aligned} L_o(\mathbf{x}_j, -\omega_j) = \tau(\mathbf{x}_j, \mathbf{x}_j^*)L_{env}(\omega_j) + \int_0^\infty & \tau(\mathbf{x}_j, \mathbf{x}_{j+1})\sigma_t(\mathbf{x}_{j+1}, \omega_j)\alpha(\mathbf{x}_{j+1}) \\ & \left(\int_{\mathbb{S}^2} f_p(\mathbf{x}_{j+1}, -\omega_{j+1}, -\omega_j) L_o(\mathbf{x}_{j+1}, -\omega_{j+1})\ \mathrm{d}\omega_{j+1} \right) \mathrm{d}s_j \end{aligned}$$

where $\mathbf{x}_{j+1} = \mathbf{x}_j + s_j\omega_j$.

We will use Monte Carlo integration to estimate both the outer and the inner integral, and we will discuss the outer one first. To do so, let us abstract the estimate of the inner integral with the random variable $\widetilde{L}_s$ so that we may simplify the volume rendering equation to:

$$L_o(\mathbf{x}_j, -\omega_j) = \tau(\mathbf{x}_j, \mathbf{x}_j^*)L_{env}(\omega_j) + \int_0^\infty \tau(\mathbf{x}_j, \mathbf{x}_{j+1})\sigma_t(\mathbf{x}_{j+1}, \omega_j)\alpha(\mathbf{x}_{j+1})\widetilde{L}_s\ \mathrm{d}s_j$$

It turns out that the whole RHS, not just the integral, can be approximated stochastically. We sample a value of s_j with probability $p(s_j)$ while allowing for the possibility of the event that $s_j = \infty$, which can happen if the scene is finite, and there is no matter outside its boundary. Then, the following is an unbiased estimate of $L_o(\mathbf{x}_j, -\omega_j)$:

$$L_o(\mathbf{x}_j, -\omega_j) \approx \begin{cases} \tau(\mathbf{x}_j, \mathbf{x}_j^*)L_{env}(\omega_j) / \Pr(s_j = \infty), & s_j = \infty \\ \tau(\mathbf{x}_j, \mathbf{x}_{j+1})\sigma_t(\mathbf{x}_{j+1}, \omega_j)\alpha(\mathbf{x}_{j+1})\widetilde{L}_s / p(s_j), & s_j \neq \infty \end{cases}. \tag{2.9}$$

```
Woodcock-Tracking($\mathbf{x}_j, \omega_j$)
if the ray $(\mathbf{x}_j, \omega_j)$ does not intersect the scene's boundary.
    return $\infty$.
else
    Let $\sigma_{\max}$ be at least $\sup\{\sigma_t(\mathbf{x}_j + s\omega_j, \omega_j) : 0 \le s < \infty\}$.
    Let $\xi_0$ and $\xi_1$ be two independent, uniform random numbers from $[0, 1)$.
    $s_j = -\ln(1 - \xi_0)/\sigma_{max}$
    if $\xi_1 < \sigma_t(\mathbf{x}_j + s_j\omega_j, \omega_j)/\sigma_{\max}$
        return $s_j$
    else
        return $s_j$ + Woodcock-Tracking($\mathbf{x}_j + s_j\omega_j, \omega_j$)
```

Figure 2.5: Pseudocode of Woodcock tracking.

A simple and popular technique for sampling s_j is *Woodcock tracking* [99], which was first introduced in the neutron transport community as a means to sample the distance a neutron travels in a medium before it is absorbed or scattered. Its pseudocode is given in Figure 2.5. Its only requirement is that we must be able to find an upper bound on the extinction coefficient along a ray, which is straightforward if the scene is finite and the matter density in the scene is also finite. Woodcock tracking yields $p(s_j) = \tau(\mathbf{x}_j, \mathbf{x}_{j+1})\sigma(\mathbf{x}_j, \omega_j)$ for all finite values of s_j, and $\Pr(s_j = \infty) = \tau(\mathbf{x}_j, \mathbf{x}_j^*)$. So, employing it greatly simplifies (2.9) to:

$$L_o(\mathbf{x}_j, -\omega_j) \approx \begin{cases} L_{env}(\omega_j), & s_j = \infty \\ \alpha(\mathbf{x}_{j+1})\widetilde{L}_s, & s_j \neq \infty \end{cases}.$$

Having estimated the outer integral, we now turn to the estimate $\tilde{L}_s$ of the inner integral. We note that the quantity $L_o(\mathbf{x}_{j+1}, -\omega_{j+1})$ may be estimated by recursively calling Volume-Path-Tracing($\mathbf{x}_{j+1}, -\omega_{j+1}$). Like the surface path tracing case, we compute the Russian roulette estimate of the inner integral instead of employing only the standard Monte Carlo integration in order to make sure that

```
Volume-Path-Tracing($\mathbf{x}_j, -\omega_j$)
$s_j$ = Woodcock-Tracking($\mathbf{x}_j, \omega_j$)
if $s_j = \infty$
    return $L_{env}(\omega_j)$
$\mathbf{x}_{j+1} = \mathbf{x}_j + s_j\omega_j$
Toss a coin that shows head with probability $p_t$
if the coin shows head
    return 0
Sample $\omega_{j+1}$ with probability $p(\omega_{j+1})$.
return $\frac{\alpha(\mathbf{x}_{j+1}) f_p(\mathbf{x}_{j+1}, -\omega_{j+1}, -\omega_j)}{(1-p_t)p(\omega_{j+1})}$ × Volume-Path-Tracing($\mathbf{x}_{j+1}, -\omega_{j+1}$)
```

Figure 2.6: Pseudocode of volume path tracing.

the recursion terminates. More specifically, $\tilde{L}_s$ is given by:

$$\tilde{L}_s \approx \begin{cases} 0, & \text{with probability } p_t \\ \frac{f_p(\mathbf{x}_j, -\omega_{j+1}, -\omega_j)\textsc{Volume-Path-Tracing}(\mathbf{x}_{j+1}, -\omega_{j+1})}{(1-p_t)p(\omega_{j+1})}, & \text{with probability } 1 - p_t \end{cases}$$

where the direction ω_{j+1} is sampled with probability $p(\omega_{j+1})$, which is often made to be equal to $f_p(\mathbf{x}_j, -\omega_{j+1}, -\omega_j)$.

The pseudocode of volume path tracing is given in Figure 2.6. Like what we have done with surface path tracing, we can rewrite it so that it becomes non-recursive, and the pseudocode of the non-recursive version is given in Figure 2.7.

2.4 Modeling Light Scattering from Fiber Surfaces

In Section 2.2, we have seen that matter can be modeled either by surfaces, which are inherently two-dimensional, or by volumes, which are inherently three-dimensional. We shall see later that both representations can model the geometry of hair and textile fibers well. Nevertheless, when modeling fibers with surfaces,

```
Non-Recursive-Volume-Path-Tracing($\mathbf{x}_0, \omega_0$)
$\widetilde{L}_i = 0$
$j = 0$ and $T_j = 0$
while true
    $s_j =$ Woodcock-Tracking($\mathbf{x}_j, \omega_j$)
    if $s_j = \infty$
        $\widetilde{L}_i = \widetilde{L}_i + T_j L_{env}(\omega_j)$
        break
    $\mathbf{x}_{j+1} = \mathbf{x}_j + s_j \omega_j$
    Toss a coin that shows head with probability $p_t$
    if the coin shows head
        break
    Sample $\omega_{j+1}$ with probability $p(\omega_{j+1})$.
    $T_{j+1} = T_j \frac{\alpha(\mathbf{x}_{j+1}) f_p(\mathbf{x}_{j+1}, -\omega_{j+1}, -\omega_j)}{(1-p_t) p(\omega_{j+1})}$
    $j = j + 1$
return $\widetilde{L}_i$
```

Figure 2.7: Pseudocode of the non-recursive version of volume path tracing.

we seldom use the BRDF or the BSDF to model their light scattering behavior. Instead, we opt for the *bidirectional curve scattering distribution function* (BCSDF), which was first proposed by Marschner et al. [49] and then named and generalized by Zinke and Weber [115]. The benefit of the BCSDF is that it greatly simplifies light transport simulation because it collapses all interactions occurring inside the fiber—i.e. from light striking the surface to eventually leaving the surface at another point—into a single scattering event.

In this section, we review the definition and theory of the BCSDF. Our treatment is similar to that of Zinke [113] but simpler. We now start by defining concepts necessary to define the model.

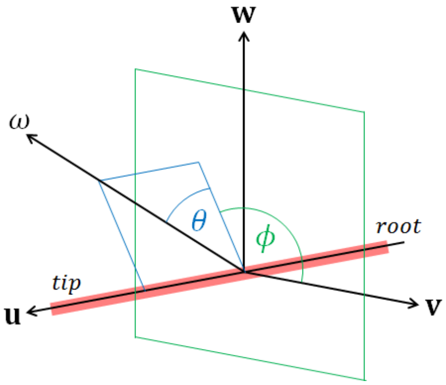


Figure 2.8: Fiber coordinate system as described in Marschner et al. [49].

Angles

We follow the notation of Marschner et al. [49]. All calculations are done in a coordinate system with right-handed orthonormal basis vectors **u**, **v**, and **w**, and the axis of symmetry of the hair fiber runs along the **u** axis. The direction $-\mathbf{u}$ points towards the root, and **u** towards the tip. A 3D direction vector ω can be described by two spherical angles: the *longitudinal angle* $\theta \in [-\pi/2, \pi/2]$ and the *azimuthal angle* $\phi \in [0, 2\pi]$:

$$\omega = \begin{bmatrix} \sin\theta \\ \cos\theta\cos\phi \\ \cos\theta\sin\phi \end{bmatrix} .$$

(See Figure 2.8.) We let ω_i and ω_o denote the incoming and outgoing direction, respectively. The spherical angles of the two directions are denoted by θ_i, ϕ_i, θ_o, and ϕ_o.

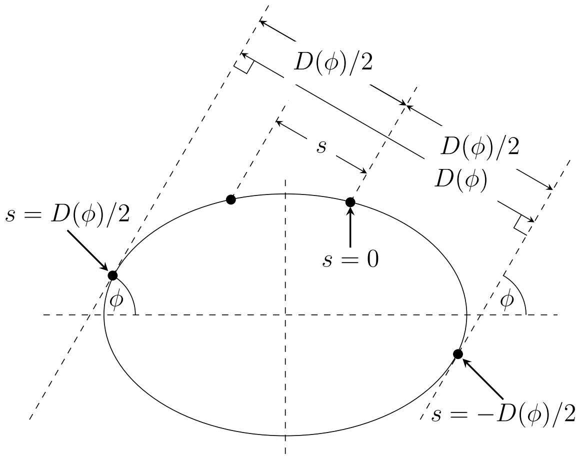


Figure 2.9: Projected diameter $D(\phi)$ and displacement parameter s of a fiber's cross section.

Projected diameter

For the purpose of modeling light scattering, we assume that the fiber is a generalized cylinder whose cross section (which might not be circular) is constant along its length.[1] The *projected diameter along azimuthal angle* ϕ is the width of the fiber when the observer views the fiber from azimuthal direction ϕ. More precisely, it is the furthest distance between two lines in the normal plane that are tangent to the cross section and make angle ϕ with the v-axis. (See Figure 2.9.) We denote the projected diameter by the symbol $D(\phi)$. For example, for a circular cross section of radius 1, $D(\phi_i) = 2$ for all ϕ_i. For an elliptic cross section with minor radius 1 and major radius a, $D(0°) = D(180°) = 2$ and $D(90°) = D(270°) = 2a$.

[1] The actual geometry might be different, and we will discuss how to handle the discrepancy later.

Ray spaces

Define the *fiber ray space* (FRS) as the set of all rays exiting the surface of the fiber (that is, the dot product of the ray's direction with the surface normal at that point is non-negative). Each ray in the FRS can be parameterized by four values: the longitudinal angle, θ, the azimuthal angle, ϕ, the displacement along the fiber's axis, t, and the displacement along the projected diameter, s, which ranges from $-D(\phi)/2$ to $D(\phi)/2$. We will indicate functions defined on the FRS by a superscript F.

A related space of rays, useful when we wish to ignore the width of the fiber, is the set of rays originating from a specific point along the fiber's axis. We call this space the *conventional ray space* (CRS). A ray in CRS is parameterized by one direction $\omega = (\theta, \phi)$ and the displacement along the fiber's axis, t. We will indicate functions defined on the CRS by a superscript C.

Radiometry

Radiance incident to or exitant from the fiber's surface is a function of the rays in FRS. Hence, we may write it as $L^F(\theta, \phi, s, t)$ or $L^F(\omega, s, t)$.

Marschner et al. defined two new radiometric quantities: the *curve radiance* $\overline{L}$ and the *curve irradiance* $\overline{E}$. The curve radiance is power per unit solid angle per unit projected length of the fiber. The curve radiance is a function of ω and t, so we may write $\overline{L}(\omega, t)$. It is related to the radiance in FRS by:

$$\overline{L}(\omega, t) = \int_{-D(\phi)/2}^{D(\phi)/2} L^F(\omega, s, t)\ \mathrm{d}s.$$

The curve irradiance is incoming power per unit length of the fiber, which is a function of t and thus denoted $\overline{E}(t)$. It is equal to the integral of the incoming curve radiance scaled by the cosine of the longitudinal angle:

$$\overline{E}(t) = \int_{\mathbb{S}^2} \overline{L}(\omega, t) \cos\theta \ \mathrm{d}\omega = \int_{\mathbb{S}^2} \int_{-D(\phi)/2}^{D(\phi)/2} L^F(\omega, s, t) \cos\theta \ \mathrm{d}s\mathrm{d}\omega.$$

The cosine factor is present as a consequence of Lambert's cosine law. Note that the quantity $\overline{L}(\omega, t) \cos\theta \ \mathrm{d}\omega$ can be interpreted as the infinitesimal irradiance created by rays whose directions are within $\mathrm{d}\omega$ of ω.

Curve radiance can be related to ordinary radiance by the following *thin fiber assumption:*

$$L^F(\omega, s, t) = L(\omega, 0, t) = L(\omega, t)$$

for all s. That is, we assume the fiber is so thin that the radiance distribution is constant across its width. This means all rays have the same radiance as the ray in the same direction that passes through the fiber axis, so $L(\omega, t)$, as well as $\overline{L}(\omega, t)$ is a function of rays in the CRS. The assumption implies that:

$$\overline{L}(\omega, t) = D(\phi) L(\omega, t).$$

In all the discussion that follows, we will make this assumption. For brevity, we will drop the t parameter from all the radiometric functions, leaving the dependence on t implicit.

2.4.1 Bidirectional Curve Scattering Distribution Function

Assume that the fiber is struck by incoming curve radiance $\overline{L}_i(\omega_i) = D(\phi_i) L_i(\omega_i)$ from a differential solid angle $\mathrm{d}\omega_i$ around the direction ω_i, which produces differential curve irradiance $\overline{L}_i(\omega_i) \cos\theta_i \ \mathrm{d}\omega_i$. The fiber scatters this incoming irradiance into a distribution of differential curve radiance over the outgoing directions, and let us denote its value at direction ω_o by $\mathrm{d}\overline{L}_o(\omega_i \to \omega_o)$. The outgoing curve radiance along a specific direction ω_o, as a result of the whole incoming curve radiance distribution, is given by:

$$\overline{L}_o(\omega_o) = \int_{\omega_i \in \mathbb{S}^2} \mathrm{d}\overline{L}_o(\omega_i \to \omega_o)$$

The *bidirectional curve scattering distribution function* (BCSDF), denoted by $S(\omega_i, \omega_r)$ or $S(\theta_i, \phi_i, \theta_r, \phi_r)$, is the ratio of the outgoing differential curve radiance in direction ω_o to the incoming differential curve irradiance from direction ω_i:

$$S(\omega_i, \omega_o) = \frac{\mathrm{d}\overline{L}_o(\omega_i \to \omega_o)}{\overline{L}_i(\omega_i)\cos\theta_i \ \mathrm{d}\omega_i}$$

This function and the thin fiber assumption imply the following relationship between radiance scattered from hair and the incoming radiance:

$$\begin{aligned} L_o(\omega_o) = \frac{\overline{L}_o(\omega_o)}{D(\phi_o)} &= \frac{1}{D(\phi_o)} \int_{\omega_i \in \mathbb{S}^2} \mathrm{d}\overline{L}_o(\omega_i \to \omega_o) \\ &= \frac{1}{D(\phi_o)} \int_{\mathbb{S}^2} S(\omega_i, \omega_o) \overline{L}_i(\omega_i) \cos\theta_i \ \mathrm{d}\omega_i \\ &= \frac{1}{D(\phi_o)} \int_{\mathbb{S}^2} S(\omega_i, \omega_o) D(\phi_i) L_i(\omega_i) \cos\theta_i \ \mathrm{d}\omega_i, \end{aligned} \tag{2.10}$$

or, equivalently,

$$L_o(\omega_o) = \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} S(\omega_i, \omega_o) \frac{D(\phi_i)}{D(\phi_o)} L_i(\omega_i) \cos^2\theta_i \ \mathrm{d}\phi_i \mathrm{d}\theta_i. \tag{2.11}$$

because $\mathrm{d}\omega_i = \cos\theta_i \ \mathrm{d}\phi_i \mathrm{d}\theta_i$.

A BCSDF is said to be *energy conserving* if

$$\int_{\mathbb{S}^2} S(\omega_i, \omega_o) \cos\theta_o \ \mathrm{d}\omega_o \leq 1$$

for all ω_i. The inequality above implies that the outgoing curve irradiance produced by the BCSDF does not exceed the curve irradiance coming in.

Multimodal, Factored BCSDFs

Hair and textile fibers are often made up of dielectric material. This means that light can reflect externally off the surface or transmit into the fiber. Once inside, it may reflect off the surface many times before finally transmit out. Marschner et al. classify all possible sequences of interactions into *modes*:

1. the *R mode* means reflecting out at first contact,
2. the *TT mode* means transmitting into the cross section and then transmitting out when light hits the surface the second time,
3. the *TRT mode* means transmitting, reflecting internally, and then transmitting out.
4. The *TRRT mode* means transmitting, reflecting internally twice, and then transmitting out,

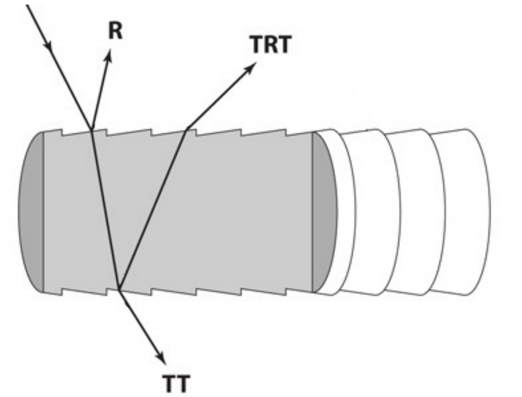
and so on. (See Figure 2.10.) For brevity, we denote each mode by a non-negative integer p, where $p = 0$ denotes the R mode, $p = 1$ denotes the TT mode, $p = 2$ denotes the TRT mode, and so on.

To abstract away all the modes into a single interaction, the BCSDF is decomposed into a number of *per-mode scattering functions*:

$$S(\omega_i, \omega_o) = \sum_{p=0}^{\infty} S_p(\omega_i, \omega_o).$$

Each per-mode scattering function is factored into a product between *the longitudinal scattering function* (LSF) M_p and the *azimuthal scattering function* (ASF) N_p:

$$S_p(\omega_i, \omega_o) = M_p(\theta_i, \theta_o) N_p(\theta_d, \phi_i, \phi_o) \tag{2.12}$$



This image originally appeared in Marschner et al.'s paper [49]. We modified it to show only the light interactions and not other extraneous details.

Figure 2.10: The first three light scattering modes out of a model human hair fiber.

where, depending on the scattering model, θ_d can be equal to the incoming angle, θ_i, or the half angle, $|\theta_i - \theta_o|/2$. The Marschner model [49] and the d'Eon model [17] use the half angle, but the model that we will present later uses the incoming angle.

We say that the whole collection of ASFs is *energy conserving* if, for all θ_d and ϕ_i,

$$\int_0^{2\pi} \sum_{p=0}^{\infty} N_p(\theta_d, \phi_i, \phi_o) \, \mathrm{d}\phi_o \leq 1.$$

That is, the incoming energy from a single direction should be split among the modes, and no extra energy should be introduced. We say that an ASF $N_p(\theta_d, \phi_i, \phi_o)$ is *reciprocal* if

$$\frac{N_p(\theta_d, \phi_i, \phi_o)}{D(\phi_o)} = \frac{N_p(\theta_d, \phi_o, \phi_i)}{D(\phi_i)} \tag{2.13}$$

for all θ_d, ϕ_i, and ϕ_o. We choose this definition because it simplifies the outgoing radiance integral in (2.11):

$$\begin{aligned}\int_{-\pi/2}^{\pi/2}\int_0^{2\pi}\sum_{p=0}^{\infty} M_p(\theta_i,\theta_o)N_p(\theta_d,\phi_i,\phi_o)\frac{D(\phi_i)}{D(\phi_o)}L_i(\omega_i)\cos^2\theta_i\ \mathrm{d}\phi_i\mathrm{d}\theta_i\\ = \int_{-\pi/2}^{\pi/2}\int_0^{2\pi}\sum_{p=0}^{\infty} M_p(\theta_i,\theta_o)N_p(\theta_d,\phi_o,\phi_i)L_i(\omega_i)\cos^2\theta_i\ \mathrm{d}\phi_i\mathrm{d}\theta_i.\end{aligned}$$

We say that an LSF is *energy conserving* if

$$\int_{-\pi/2}^{\pi/2} M_p(\theta_i,\theta_o)\cos^2\theta_o\ \mathrm{d}\theta_o \leq 1 \tag{2.14}$$

for all θ_i. This implies that the LSF by itself does not introduce any extra energy. For example, if the ASF is diffuse and energy preserving (that is, if there is only one mode, and $N_p(\theta_d,\phi_i,\phi_o) = 1/(2\pi)$ for all values of θ_d, ϕ_i, and ϕ_o), then the per-mode BCSDF is energy conserving. Moreover, provided that θ_d is equal to θ_i instead of the half angle, we can *prove* that the BCSDF is energy conserving if both the ASFs and the LSFs are energy conserving.

Lemma 2.1. *If both the LSFs and the ASFs are energy conserving, so is the BCSDF.*

Proof.

$$\begin{aligned}\int_{\mathbb{S}^2} S(\omega_i,\omega_o)\cos\theta_o\ \mathrm{d}\omega_o &= \int_{-\pi/2}^{\pi/2}\int_0^{2\pi}\sum_{p=0}^{\infty} M_p(\theta_i,\theta_o)N_p(\theta_i,\phi_i,\phi_o)\cos^2\theta_o\ \mathrm{d}\phi_o\mathrm{d}\theta_o\\ &= \int_0^{2\pi}\sum_{p=0}^{\infty}\left(\int_{-\pi/2}^{\pi/2} M_p(\theta_i,\theta_o)\cos^2\theta_o\ \mathrm{d}\theta_o\right)N_p(\theta_i,\phi_i,\phi_o)\ \mathrm{d}\phi_o\\ &\leq \int_0^{2\pi}\sum_{p=0}^{\infty} N_p(\theta_i,\phi_i,\phi_o)\ \mathrm{d}\phi_o \leq 1.\end{aligned}$$

as required. □

The full scattering models we shall construct in later chapters satisfy all the definitions above and therefore are provably energy conserving. The model proposed by Marschner et al. [49], however, is not, and d'Eon et al. later devised a new model to address this shortcoming [17].

2.4.2 Simulating Light Transport with BCSDF

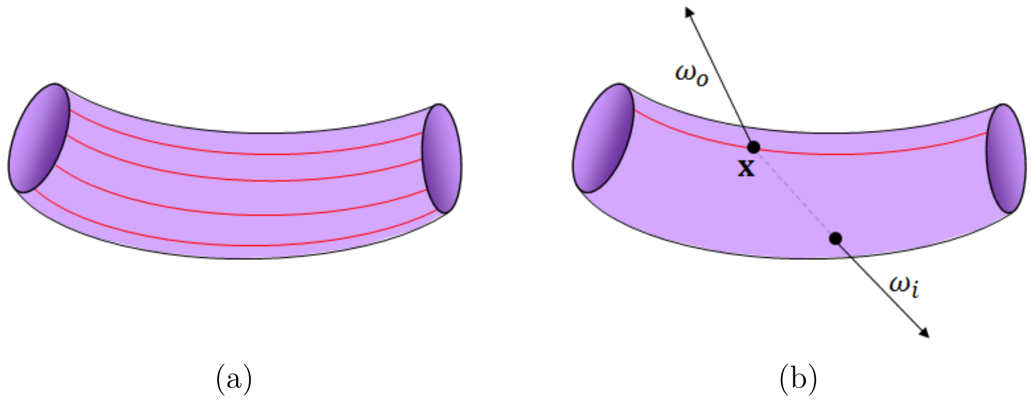
The end goal of formulating the BCSDF is to use it in light transport simulation. However, the light transport equations and the simulation processes discussed so far were not formulated to directly use it. In this section, we discuss how to modify components of the simulation processes to incorporate the BCSDF.

Simulating with the BCSDF in Surface-Based Scenes

Recall that light scattering behavior in surface-based scenes is modeled by the BRDF. While the BCSDF is a bidirectional function like the BRDF, it is not a drop-in replacement.

The first problem is that their semantics are different. The BCSDF treats matter as a one-dimensional entity: a curve with no surface area. On the other hand, the BRDF treats matter as a two-dimensional entity with a well-defined surface area. To reconcile the difference, we conceptually think of the fiber surface as a generalized cylinder obtained by sweeping a fixed cross section along a center curve. In this way, we may think of the surface as if it were made of a continuum of one-dimensional curves, each of which is obtained by tracking a specific point on cross section as it moves along the center curve. (Figure 2.11a) Moreover, this interpretation allows us to simply substitute the BCSDF in place of the BRDF in the rendering equation.

The second problem is that, in a path tracer, when estimating scattered light according to (2.11), the sampled incoming direction ω_i might point below the fiber surface. (Figure 2.11b) As a result, if we extend the ray $(\mathbf{x}, \omega_i)$, it would hit the inside of the fiber surface, and so the incoming radiance along the direction is



(a) When using the BCSDF with a surface-based model of fibers, we think of the surface as being made of a continuum of curves. (b) When estimating scattered light, we always skip the first hit along the ray if it hits the fiber from the inside. This allows us to simulate subsurface scattering behavior that the BCSDF intends to model.

Figure 2.11: How we integrate the BCSDF with a surface-based model of fiber.

always occluded. To allow for the subsurface scattering behavior that the BCSDF intends to model, we always skip the first hit along the ray if it hits the inside of the fiber.

Simulating with the BCSDF in Volume-Based Scenes

In order to simulate light transport in volumetric scenes, we must specify the extinction coefficient σ_t, the albedo α, and the phase function f_p. Because the albedo and the phase function govern light scattering behavior (as opposed to light absorbing behavior), we must convert the BCSDF to them. We can turn the BCSDF $S(\omega_i, \omega_o)$ to a phase function by normalizing it so that it is a probability distribution with respect to ω_i:

$$f_p(\omega_i, \omega_o) := \frac{S(\omega_i, \omega_o)\cos\theta_i}{\int_{S^2} S(\omega_i, \omega_o)\cos\theta_i \,\mathrm{d}\omega_i}. \tag{2.15}$$

However, because we normalize the BCSDF, we lost information related to the fiber's color. As a result, we must encode such information in the albedo. A good albedo should make its product with the phase function as close as possible to the BCSDF. Hence, we may choose the albedo by solving the optimization problem below:

$$\alpha := \underset{\alpha'}{\operatorname{argmin}} \int_{\mathbb{S}^2} \int_{\mathbb{S}^2} (\alpha' f_p(\omega_i, \omega_o) - S(\omega_i, \omega_o) \cos\theta_i)^2 \, \mathrm{d}\omega_i \mathrm{d}\omega_o \tag{2.16}$$

It is important to note that Equations (2.15) and 2.16 are used when a fixed BCSDF is given, and we would like to convert it to volume rendering parameters. In Chapter 5, we discuss a related problem of fitting volume rendering parameters based on an *unknown* BCSDF to photographs of real fabrics. In that setting, we optimize for the BCSDF parameters and convert it to a phase function with (2.15). However, instead of using (2.16), we think of the albedo as a separate parameter to be fitted independently from the BCSDF.

CHAPTER 3

PREVIOUS WORK

君の前前前世から僕は
君を探しはじめたよ
そのぶきっちょな
笑い方をめがけて
やってきたんだよ

RADWIMPS, 前前前世

The previous chapter discusses how to simulate light transport given models of matter without discussing their specifics. This chapter surveys the models that researchers have used to represent hair and fabrics. We discuss those for hair in Section 3.1 and those for fabrics in Section 3.2.

Throughout the discussion, the reader will notice several recurring themes. The first theme is that, for each of the two materials, we will discuss models for its geometry and models for its light scattering behavior. This is because, as discussed in Chapter 1, an appearance model has to specify both.

The second theme is that, similar to the classification in Chapter 2, appearance models can be classified as surface-based ones or volume-based ones. Often, a surface-based light scattering model can only be used with a surface-based geometric model, and the same is true for volume-based models.

The third theme is that, when discussing geometry of hair and fabrics, we distinguish between two scales. *Microgeometry* is the shapes of individual fibers, which can generally be thought of as long and thin rods in 3D space. *Macrogeometry* is the overall shape of the fiber assembly. In hair, it is the hairstyle. In fabrics, it is the shape of the fabric piece. While macrogeometry alone is enough for geometric modeling, microgeometry adds fine and subtle details such as fuzz and flyaway fibers.

3.1 Hair Models

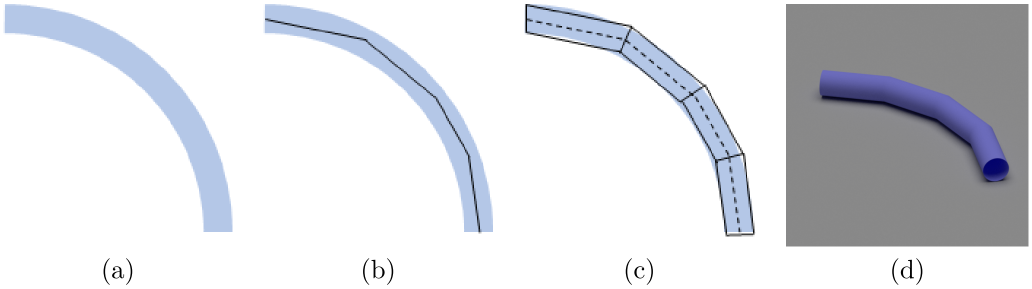
3.1.1 Hair Geometry Models

Computer graphics practitioners (e.g. media creators) often model hair with surfaces that represent its macrogeometry. On the other hand, computer graphics researchers pursue a greater level of realism and so almost always model hair explicitly at the fiber level.

A strand of hair can be thought of as a filled cylinder along a curve in 3D space. A surface-based model represents such a fiber as a collection of discrete surfaces that approximates its outer shell. Such a model generally has to make two choices: how to represent the curve, and how to represent the surface that surrounds it.

A popular choice for representing the fiber curve is a *polyline*, which is a sequence of 3D line segments connecting consecutive items in a sequence of 3D points. In many real-time rendering systems [106, 116, 70, 101, 100], it is used as the sole representation of microgeometry because it can be readily translated to GPU-friendly primitives such as lines and camera-facing quads. A fiber curve can also be modeled as a non-straight curve such as a B-spline [61, 10].

A representation for fiber surface is needed when rendering with a ray tracer. If the fiber curve is a smooth, the surface can be modeled as a generalized cylinder:



(a) (b) (c) (d)

(a) A fiber can be thought of as a curved cylinder. (b) The center curve of the fiber is approximated with a polyline. (c) Then, we cover the polyline's segments with straight cylinders of equal radii. Consecutive cylinders meet at miter joints. (d) We do not put circular caps at the ends of the fiber, so one can see its inside from the ends. We do so because it makes it easy to integrate this geometric representation with the BCSDF. (See Section 2.4.2.)

Figure 3.1: The surface-based geometric representation of fibers used in this dissertation.

the shape obtained by sweeping a 2D shape, typically a circle, along the curve. Nevertheless, ray intersection becomes a much more complicated problem [8], and we have not seen any work that renders hair or textile fibers in this way. Instead, we may tessellate generalized cylinders into polygonal meshes [53]. The most popular alternative is to represent the fiber curve with a polyline and cover its segments with simple 3D surfaces. Watanabe and Suenaga uses trigonal prisms [97]. On the other hand, following Moon et al. [51, 52], we use cylinders having the same radii and require that consecutive cylinders meet at miter joints in all of our works. However, we do not put circular caps at fiber ends like they do. While the surface is not smooth at every point, it does not lead to noticeable artifacts unless the fibers are viewed at the zoom level of a microscope, which is outside the scopes of our works.

Because a human head contains above 100,000 hair strands, it is impractical to generate the fibers manually. Hair curves can be generated procedurally by distributing hair roots on the scalp and growing the curves according to a hairstyle specification, the approaches to which are surveyed in Ward et al.'s article [95]. When hair geometry is obtained through physical simulation, a small number of *guide hairs* are generated at the beginning and are used in simulation. They are then interpolated to generate other strands which form most of the hair volume at render time [13]. A level-of-detail representation that generates curves to fill larger hair structures on the fly has also been proposed [96]. In contrast to the aforementioned procedural approaches, Jakob et al. capture individual strands in real hair samples through the use of multiple photographs taken with shallow depths of field [33]. Data obtained from thermal imaging [27], photographs taken from single view point under different lighting conditions [58], multi-view photographs [98], and a light stage [59] have been used to guide hair curve generation.

Fiber microgeometry can also be modeled with volumetric data; in other words, a 3D array of density values. However, the representation is not as popular as the surface-based one. Kajiya and Kay [35] pioneered the approach in the context of fur rendering. Following Chang et al. [13], Pretrovic et al. represents hair during simulation with a few guide hairs, but use them to generate volumetric geometry at render time [61].

3.1.2 Hair Scattering Models

In addition to a volumetric model for hair geometry, Kajiya and Kay also presented the first analytical light scattering model for hair fibers: a phase function based on the Phong surface shading model [35]. By explicitly modeling fibers as cylinders, it captures the linear highlights characteristic of hair and fur, but since the model assumes hair fibers are opaque, it cannot model effects arising from refraction. Extensions to incorporate forward scattering into the Kajiya–Kay model have been proposed [24, 41].

When hair geometry is modeled by surfaces, light scattering behavior is often modeled by a BCSDF. Marschner et al. presented one based on treating a hair fiber as a transparent dielectric cylinder [49]. As discussed in the previous chapter, their model decomposes the scattering behavior of fibers into scattering modes according to the number of reflections and refractions that a light ray undergoes as it interacts with the hair fiber. They factored the scattering function for each mode into a product of a longitudinal term and an azimuthal term. The former enabled them to model the longitudinal shifts of highlights, and the latter the glints due to the fiber's cross section.

Marschner et al.'s model has been widely used and has generated much subsequent research. Zinke and Weber generalized the BCSDF to the *bidirectional fiber scattering distribution function* (BFSDF) to make it capable of capturing near-field appearance [115]. Zinke et al. added a diffuse component to the Marschner model and provided a procedure for acquiring model parameters from a single photograph of a hair sample [114]. Sadeghi et al. refined the model to be more intuitive for artists [73]. Xu et al. gave a fast approximation to the model for interactive rendering and appearance editing [101]. Some research has also gone toward analytic importance sampling for variants of this model [28, 56].

The Marschner model, however, has two important shortcomings. First, as mentioned in the last chapter, it is not energy conserving. Second, it deals with the caustics in one scattering mode in an *ad hoc* way, which introduces extra non-physical parameters to the model. d'Eon et al. addressed both problems by introducing a new longitudinal component designed to conserve energy and an azimuthal component based on numerical integration [17].

Researchers have also worked on BCSDFs for animal fur. Yan et al. model a fur strand as a circular cylinder having two layers with different absorbing materials, reflecting more accurately animal fur's anatomy [104]. While all other works assume circular cross sections, Ogaki et al. presented a model for fur having arbitrary user-specified cross-sectional shapes that can even vary along the length of the fiber. To deal with such geometry, they trace photons through the fiber and tabulate the resulting density estimates [55].

3.2 Fabric Models

3.2.1 Fabric Geometry Models

In computer graphics research, three approaches have been proposed to represent fabric geometry. The first approach is to model only macrogeometry by abstracting the fabric to a two-dimensional surface, which is in turn represented by a mesh or curved surface [74, 1, 29]. Surfaces can be quite successful in distant views, but close-up views, especially at edges and silhouettes, look incorrect because the three-dimensional structure of yarns and fibers is missing.

The second approach is to use volumetric data to model individual fibers. Xu et al. used the approach to represent yarns in knitwear [103]. Schröder et al. generate textile fibers procedurally and convert the fibers to a volumetric representation [75]. Zhao et al. obtained theirs volumetric data from computed microtomography (micro CT) scans of real cloth samples and were able to reproduce detailed and irregular appearance of cloth at an unprecedented level [109].

The third approach is to model fabric as a collection of fibers, each represented with surfaces covering a center curve. These curves in woven fabrics can be generated from procedural models that describe their arrangements in a yarn. Two such models have been used in the computer graphics community: Sriprateep and Bohetz's [84] and Schröder et al.'s [76]. They have been incorporated in level-of-detail represention of fabric by Zhang et al. [107] and Wu and Yuksel [100], respectively. The curves can also be generated from measured data, and we will describe an algorithm to obtain them from micro CT scans of cloth samples in Chapter 5. Zhao et al. use the fiber curves generated by our algorithm to fit Schröder et al.'s model to real yarns [111].

3.2.2 Fabric Scattering Models

To model light scattering based on a macrogeometry-only model, a popular option is to employ bidirectional texture functions (BTFs) [74]. This approach captures view-dependent appearance by exhaustive sampling, but often suffers from undersampling and shows limited quality at edges and silhouettes and under grazing illumination.

The second approach, also coupled with the macrogeometry-only approach to geometry representation, is to use a surface-based BRDF model coupled with textures. Adabala et al. [1] generated textures from weave pattern data and modeled scattering behavior with a microfacet BRDF. Irawan and Marschner [29] proposed a procedural texture model based on weave patterns and a reflectance model based on the analysis of specular reflection from spun fibers. Sadhegi et al. [72] constructed cloth BRDFs from weave patterns and a yarn scattering model. Real-time cloth rendering systems such as those of Zhang et al.'s and Wu's and Yuksel's [100]

Scattering model	Geometry model		
	Fabric mesh	Volume	Fiber mesh
BTF	Sattler [74]		
BRDF	Adabala [1] Irawan [29] Sadeghi [72]		Zhang [107] Wu [100]
Volumetric		Zhao [109]	
Fiber-based		Schröder [75] Chapter 5	Schröder [75]* Schröder [76] Chapter 5

*Schröder et al. used fiber geometry coupled with fiber-based appearance models to produce ground truth for comparison with their volumetric geometry representation.

Table 3.1: Approaches to fabric appearance modeling.

use simple BRDFs in their implementations.

The third approach is to specify volumetric appearance as dictated by the radiative transfer equation. Zhao et al. [109] modeled appearance of cloth with an anisotropic microflake phase function, the foundation of which was laid by Jakob et al. [32].

The last approach employs the fiber-based scattering functions such as the BCSDF or the BFSDF. Schröder et al. [75] used the Zinke and Weber's BFSDF [115] with fiber-based geometry and a derived volume-based approximation. However, because the shift to volume rendering caused changes in appearance, they had to include special corrections in their volumetric rendering algorithm.

Table 3.2.2 summarizes the range of approaches to fabric modeling and situates our work (Chapter 5) in this space.

CHAPTER 4

MODELING APPEARANCE OF ELLIPTICAL HAIR FIBERS

เพียงเจอ ไอ้หนุ่มผมยาว
น้องสาวก็ลืมพี่หมดสิ้น
รักเรา เจ้าไม่ถวิล
ไปหลงลมลิ้นเจ้าศิลปินตัวดี
มันเจ็บใจ เจ็บใจเหมือนไม่พอ
เจ้าชมมันรูปหล่อเสียงดี
มันบาดใจ บาดใจเหลือที่
เจอะอย่างนี้หุ่นอย่างพี่ต้องขอลา
โฮ โฮ้โห่โฮ้โอโอโอ่ โฮ้โห่โอ๊ะโอโอโอ่
เจ้าศิลปินผมยาว

ดนู ฮันตระกูล, *ไอ้หนุ่มผมยาว*

In the last chapter, we see that most previous scattering models for hair fibers are based on the assumption that hair cross sections are circles. However, human hair fibers generally have elliptical cross sections. In this chapter, we present a new light scattering model for elliptical fibers and a study of the effects of elliptical cross sections to hair appearance. They constitute a step forward in achieving realism in hair rendering.

This work originally appeared in the paper “Azimuthal Scattering from Elliptical Hair Fibers,” which appeared in *ACM Transactions on Graphics* in 2017 [38]. It is joint with Steve Marschner.

4.1 Introduction

The appearance of hair arises from the interaction of light with the geometry of fibers. The overall cylindrical shape of fibers causes the characteristic linear highlights on a head of hair, and the scaly surface of the cuticle covering the hair separates them into distinct colored highlights. The fibers' particular cross-sectional shape controls how scattered light depends on both the locations of the viewer and light source and the orientation of the fiber, determining whether highlights appear smooth or glittery. Scattering models that are used to render hair account for all three phenomena, but research has mainly focused on the first two. This chapter is about how to improve our models for the effects of cross section on the distribution of scattered light.

It's well documented that hair cross sections are generally non-circular, and often roughly elliptical [71]. However, prior scattering models for hair are all based on an analysis of light rays interacting with circular cylinders, including the model of Marschner et al. [49] that includes an approximate correction for eccentricity.

Creating efficient scattering models that properly account for elliptical cross-sections is a difficult undertaking, and we seek to answer two questions:

- Is cross section *important*? Are the changes in appearance caused by non-circular cross-sections significant enough to warrant the effort?
- Are ellipses a *correct* model? Does an elliptical cross-section predict what happens in real hair fibers?

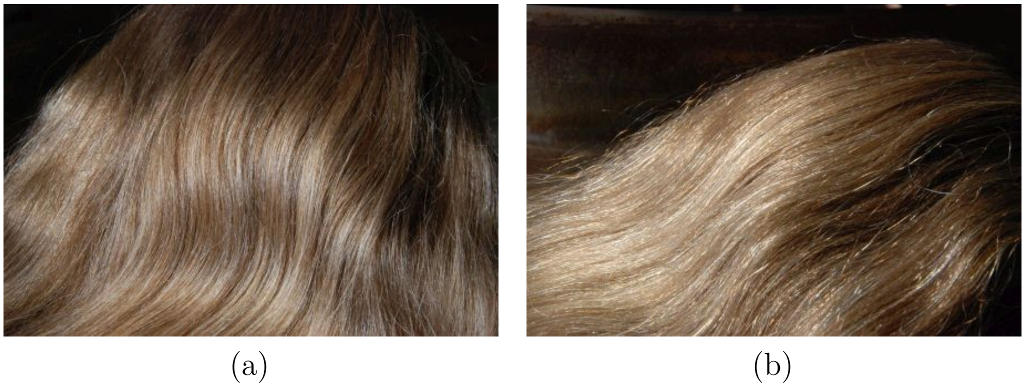
To answer the question of importance, we have extended previous hair models to account for elliptical cross-sections. By contrast to Marschner et al.'s previous model, our model accurately describes the geometric optics of elliptical cylinders. It predicts dramatic changes in scattering behavior, particularly for transmitted and internally reflected light, as we move from circular to elliptical fibers. These changes are significant even for mild eccentricity, and the behavior is very different from the eccentricity correction proposed by Marschner et al.

These predictions of our elliptical theory need to be substantiated: after all, hairs are not exactly elliptical, and many details of their structure and of the physics of light are omitted by the model. The elliptical model might predict features that do not occur because of imperfections in real fibers, or its inherent assumptions might cause it to miss visually important phenomena.

The experiments of Bustard and Smith [9] and later Marschner et al. provide promising support for an elliptical model, but their measurements were limited, being based on small numbers of data points gathered on just a few fibers. No one has examined the complete scattering function in enough detail to see whether we have modeled the whole behavior.

Therefore, to answer the question of correctness, we have made new measurements of a range of different types of hair fibers, using a new device designed to give a very complete picture of fiber scattering functions. Our goal is to capture the complete function with enough resolution to observe all the important features, but accuracy in absolute magnitudes is a secondary concern. With this goal in mind, the measurement system is built around image-based measurements, capturing tens of thousands of data points with each exposure.

The conclusions of our study are clear, answering both questions in the affirmative and introducing a new phenomenon:



Here, we show two photographs of Caucasian hair taken from two different angles under the same lighting, (a) with the sun behind the camera and (b) with the sun roughly 90 degrees to the camera's left. Bright glints can be observed in (b) but not in (a).

Figure 4.1: Orientation dependent glints in human hair.

- The structured internal-reflection glints predicted by the elliptical model dramatically contradict previous models. They also are clearly observable in real hair, as long as it is light enough in color for internal reflection to be significant.
- There is a forward-scattering feature present in all hair scattering functions, previously unremarked upon in graphics papers: an ideal-specular lobe, which we call the E mode, that becomes particularly bright at grazing angles.

For rendering, the key implication of non-circular fibers is that scattering depends jointly on the azimuthal positions of the light and viewer, not only on the azimuthal *difference* as in the circular case. The azimuthal component of a scattering model must be a function of two variables, not one, and the brightness of fibers depends on their orientation, even when the light and viewer remain fixed. This leads to a strong glinty appearance in hair under conditions when this orientation dependence is strong (Figure 4.1).

It should be emphasized that we do not propose a production-ready, practical algorithm to efficiently compute scattering from elliptical fibers in a shader. The renderings in this chapter are made using tabulated scattering functions, which limits the ability to model continuous variation in fiber properties. Our study has established what should be modeled to provide a truly high-fidelity hair scattering model. Models that are more efficient and include the E mode should be developed in the future.

In the remainder of this chapter, we extend previous models to accurately account for cross-section (Sections 4.3 and 4.4); examine the predictions of the theory (Section 4.5); describe our measurement device and its results (Sections 4.6 and 4.7); and examine the agreement between theory and measurements (Sections 4.8 and 4.9).

4.2 Previous Work

Researchers in the optics community have characterized some aspects of elliptical fibers' light scattering behavior. Marcuse determined the range of azimuthal angles of backward scattered light [47]. Adler et al. studied the Fourier transform of the angle of backward scattered caustics as a function of incoming angle [2]. In comparison, our work is more comprehensive because we characterize both forward and backward scattering lobes.

We are only aware of three other works in the graphics community that measure light scattered from fiber-like structures. The first is Marschner et al. [49], and the second is Sadeghi et al. [72], and the third is Yan et al. [104]. All three works used a gonioreflectometer for the task, and so only 1D or coarse 2D slices of the 4D scattering function were feasible to acquire. Our device, on the other hand, can acquire the whole hemisphere of scattered light in one exposure, thus yielding more comprehensive data at greater efficiency than the previous method. However, the measurements are less accurate due to factors we shall describe in Section 4.6. As such, it is more suitable for observing qualitative behavior of the scattering function than for directly acquiring the function itself.

4.3 Background

We model light scattering behavior of hair fibers with the bidirectional curve scattering distribution function (BCSDF), which we discussed in detail in Section 2.4. We have also discussed previous BCSDFs for hair in Section 3.1.2. In this section, we shall review common parameters to these models and how to visualize them.

4.3.1 Common Model Parameters

Previous hair scattering models [49, 17] and the model we will present share the following parameters:

- The index of refraction of the material η.
- The absorption coefficient of the material σ.
- For each mode p, the longitudinal shift α_p.
- For each mode p, the width of the longitudinal lobe β_p.
- The aspect ratio of the cross section a. If the cross section is an ellipse, the aspect ratio is the ratio of the length of the major axis to the length of the minor axis. Hence, the circle has $a = 1$.

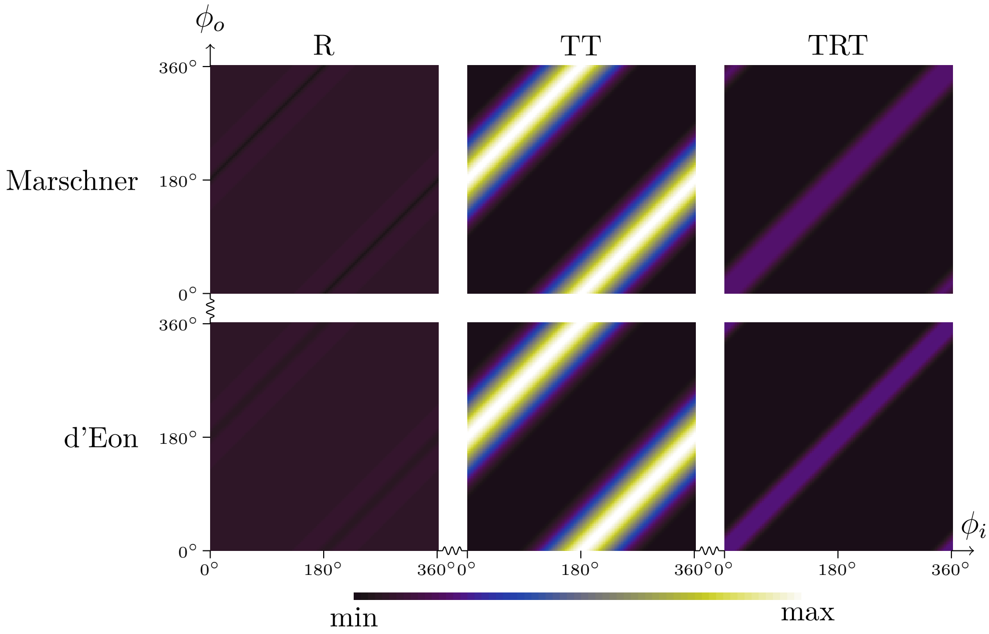
Note that the Marschner model only covers scattering modes up to $p = 2$. It also contains extra parameters such as the glint scale factor k_G and the azimuthal caustic width w_c in order to support its ad hoc modeling of the TRT mode. The d'Eon model requires that β_p is the same for all p and calls the single β parameter the "surface roughness" of the hair fiber. The d'Eon model also only models circular hair fibers, so a is always 1.

4.3.2 Visualization

We introduce two types of visualization for parts of the scattering models and measured data.

The first, the (ϕ_i, ϕ_o)-plot, shows $N_p(\theta_d, \phi_i, \phi_o)$ for a fixed θ_d as a scalar field over the (ϕ_i, ϕ_o)-plane. Each point's color indicates the intensity of light scattered to ϕ_o when it arrives from ϕ_i. The (ϕ_i, ϕ_o)-plot allows us to see the complete picture of how the ASF evolves with ϕ_i, and it exposes the inherent symmetry due to reciprocity. It also provides a convenient way of observing how the ASF evolves with changing θ_i. Examples of (ϕ_i, ϕ_o)-plots are given in Figure 4.2.

The second, the (θ_o, ϕ_d)-plot, shows a slice of the whole scattering function $S(\omega_i, \omega_o)$ for a fixed incoming direction ω_i. The slice is shown as a color map over the (θ_o, ϕ_d)-plane where $\phi_d = \phi_o - \phi_i$. Note that, by using the difference angle ϕ_d instead of the outgoing angle ϕ_o, the direction of retroreflection is fixed at $\phi_d = 0°$ and the direction of forward scattering is fixed at $\phi_d = 180°$. In this way, the (θ_o, ϕ_d)-plot does not shift vertically as ϕ_i changes, making it easier to notice changes in the scattering function as ϕ_i changes. Also, as we will see in the next section, it represents the data that is captured by one snapshot of our measurement



The (ϕ_i, ϕ_o)-plots were generated with the following parameters: $\eta = 1.55, \sigma = 0, \beta_0 = \beta_1 = \beta_2 = 5°$, and $a = 1$ (circle). For the Marschner model, we set $k_G = 1, w_c = 10°, \Delta\eta' = 0.2$, and $\Delta h_M = 0.5$. For all plots, $\theta_d = 0$. The colors between different modes of the same models are in the same scale, but the plots are not in the same scale between the models.

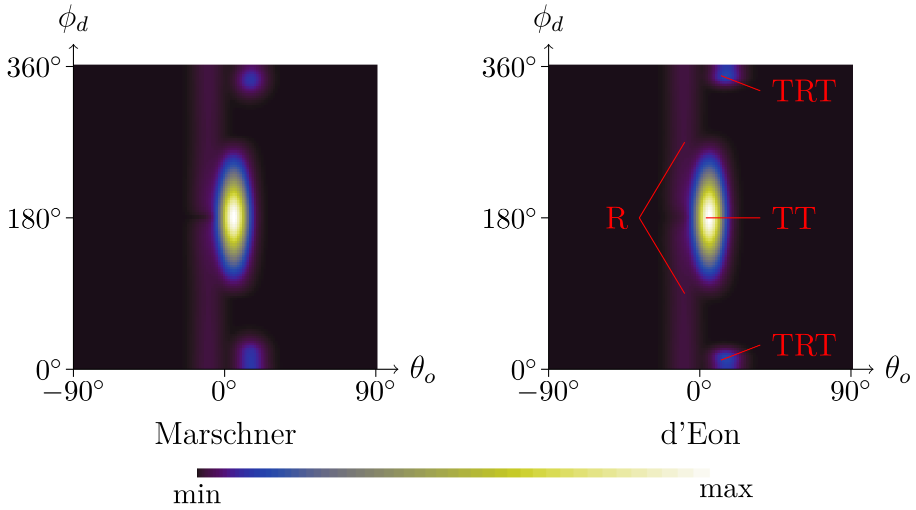
Notice that the plots are *shift invariant*, meaning that each vertical slice of the plot is a shifted version of other slices. This property arises from the rotational symmetry of the circular cross section. All observable features in shift invariant plots are slanted bars along the southwest–northeast direction. One can also see that the d'Eon model is very similar to the Marschner model for the circular fiber.

Figure 4.2: The (ϕ_i, ϕ_o)-plots of the first three modes of the Marschner and d'Eon scattering models.

system. As examples, (θ_o, ϕ_d)-plots of the Marschner and d'Eon models are given in Figure 4.3.

Inspection of the (ϕ_i, ϕ_o)-plots in Figure 4.2 provides us with heuristics on how to recognize the first three scattering modes in the (θ_o, ϕ_d)-plots:

- The R mode has a large support in the (ϕ_i, ϕ_o)-plots, covering all but a small area around the lines $\phi_o = \phi_i \pm 180°$. Hence, in (θ_o, ϕ_d)-plots, it should appear as two vertical strips covering the whole range of ϕ_d except a small



The (θ_o, ϕ_d)-plots were generated with the following longitudinal shifts: $\alpha_0 = -10°, \alpha_1 = 5°$, and $\alpha_2 = 15°$. The other model parameters are the same as those in Figure 4.2. For both plots, $\theta_i = 0°$ and $\phi_i = 0°$. The plots are not to scale with each other.

The R mode manifests as two vertical strips, the TT mode as the brightest lobe at the vertical center of the plot, and the TRT mode as blobs near the top and bottom ends.

Figure 4.3: The (θ_o, ϕ_d)-plots of the Marschner and d'Eon scattering models.

portion around $\phi_d = 180°$.

- The TT mode has narrower support around $\phi_o = \phi_i \pm 180°$ and is the brightest among the three modes. Hence, it should appear as the brightest lobe, centered vertically around $\phi_d = 180°$.
- The TRT mode has the narrowest support around the line $\phi_o = \phi_i$. Hence, it should appear as blobs located vertically near $\phi_d = 0°$ and $\phi_d = 360°$.

According to Marschner et al. [49], the modes are also separated in θ due to the tilting of the hair scales. Note that, at the beginning of Section 2.4, we said that the direction $-\mathbf{u}$ points towards the root and $\mathbf{u}$ points towards the tip. This means that the left side of the (θ_o, ϕ_d)-plot $(-90°)$ should correspond to scattering toward the root, and the right side $(90°)$ should correspond to scattering towards the tip. Marschner et al. observe that the R mode is shifted towards the root (i.e.

to the left), but other modes are shifted towards the tip (i.e. to the right). As such, the R mode should appear to the left of the ideal specular line ($\theta_o = -\theta_i$) while other modes appear to the right as can be seen in Figure 4.3.

4.4 Modeling

Having reviewed the previous models and established our notations, in this section we construct a scattering model for hair fiber with elliptical cross sections. Our model is a factored model like that of Marschner et al.:

$$S(\omega_i, \omega_o) = \sum_{p=0}^{\infty} M_p(\theta_i, \theta_o) N_p(\theta_i, \phi_i, \phi_o) \tag{4.1}$$

However, we make the ASF depend on θ_i rather than the half angle $|\theta_i - \theta_o|/2$ for the reason we discussed in Section 2.4.1.

4.4.1 Longitudinal Scattering Function

The LSF for Mode p is defined by two parameters: the *width of the lobe* β_p and the *longitudinal shift* α_p. In the original model of Marschner et al., the LSF was simply a Gaussian distribution in θ. As observed by d'Eon et al. [17] and other researchers, this LSF is not normalized on the sphere, and loses energy away from normal incidence. Although d'Eon et al. proposed an elegant and improved alternative, we have noticed some problems which we discuss in Appendix A.1. In this paper, we use a simpler LSF that is a normalized version of the original Gaussian:

$$M_p(\theta_i, \theta_o) = \frac{g(\theta_o; \mu(\theta_i), \beta_p)}{G(\mu(\theta_i), \beta_p)}.$$

where $g(x;\mu,\sigma) = \exp(-(x-\mu)^2/2\sigma^2)/(\sqrt{2\pi}\sigma)$ is the Gaussian distribution, and the normalization factor G is calculated to ensure normalization over the sphere. In order to achieve a lobe shifted by α_p without the lobe becoming excessively concentrated near grazing, we center the Gaussian at

$$\mu(\theta_i) = \text{clamp}(-\theta_i + \alpha_p, [-\pi/2, \pi/2]).$$

The normalization factor G is defined to ensure cosine-weighted normalization over the sphere:

$$G(\mu,\sigma) = \int_{-\pi/2}^{\pi/2} g(\theta;\mu,\sigma)Q(\theta)\ \mathrm{d}\theta \approx \int_{-\pi/2}^{\pi/2} g(\theta;\mu,\sigma)\cos^2(\theta)\ \mathrm{d}\theta$$

where Q is a polynomial that approximates $\cos^2\theta$ from above[1]:

$$Q(\theta) = 0.002439\theta^8 - 0.04301\theta^6 + 0.3322\theta^4 - 0.999745\theta^2 + 1.0001 \geq \cos^2\theta.$$

It can be easily shown that this normalization factor implies the LSF is energy conserving in the sense of (2.14). The indefinite integral of the product of a Gaussian and a polynomial has a closed form, so the integral can be evaluated analytically as follows.

Lemma 4.1. *Let $P(x) = p_0 + p_1 x + \cdots + p_k x^k$. The indefinite integral of $P(x)g(x;\mu,\sigma)$ is given by*

$$\frac{A}{2}\text{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right) - \sigma^2 B(x) g(x;\mu,\sigma) + C \tag{4.2}$$

where $B(x) = b_0 + b_1 x + \cdots + b_{k-1}x^{k-1}$,

$$b_j = \begin{cases} p_k, & j = k-1, \\ p_{k-1} + \mu b_{k-1}, & j = k-2, \\ p_{j+1} + \mu b_{j+1} + (j+2)\sigma^2 b_{j+2}, & 0 \leq j < k-2, \end{cases}$$

[1] Alternatively, one can also define G more directly in terms of the antiderivative of $g(\theta;\mu,\sigma)\cos^2\theta$. However, we found that its closed form involves the imaginary error function, a special function which is not a part of many libraries. We approximated $\cos^2\theta$ with a polynomial because the antiderivative depends only the real error function, whose implementation is more readily available.

and $A = p_0 + \mu b_0 + \sigma^2 b_1$.

Proof. The formula can be verified by simply taking the derivative. □

We note that Zinke [113] also defines an LSF that is a scaled Gaussian lobe. The difference between his function and ours is that his normalization factor $G(\mu, \sigma)$ does not contain the polynomial Q in the integrand and so is much larger than ours, implying that his function loses more energy.

4.4.2 Azimuthal Scattering Function

We derive the ASF by tracing rays through the elliptical cross section. Due to Bravais's Law [87, 49], the paths of reflected and refracted rays in a cylinder of any cross section behave the same as rays in 2D interacting with the cross section, though with a different refractive index that depends on θ_i. In this 2D world, the fiber ray space is parameterized by an angle ϕ and a displacement s, which still ranges from $-D(\phi)/2$ to $D(\phi)/2$. So, radiance is now written as $L(\phi, s)$. The curve radiance, now parameterized only by ϕ, is defined in a similar way as that of the 3D version:

$$\overline{L}(\phi) = \int_{-D(\phi)/2}^{D(\phi)/2} L(\phi, s)\ \mathrm{d}s. \tag{4.3}$$

Characterizing a Cross Section

We assume that each interaction of light with the surface results in perfect specular reflection or refraction; we will account for surface roughness later. For our purpose, a cross section is characterized by:

- the projected diameter $D(\phi)$, defined in Section 2.4;
- for each mode p, the *exit direction* $\phi_p^{\mathrm{e}}(\phi, s)$;
- for each mode p, the *exit displacement* $s_p^{\mathrm{e}}(\phi, s)$; and
- for each mode p, the *attenuation* $A_p(\phi, s)$;

where ϕ_p^{e} and s_p^{e} describe the direction and displacement of a ray entering at (ϕ, s) after it has undergone $p+1$ interactions with the cross section's surface. $A_p(\phi, s)$ is the attenuation of the radiance along the ray as it interacts with the surface and the material inside the fiber. Reversibility of paths implies that, if $\phi_o = \phi_p^{\mathrm{e}}(\phi_i, s_i)$ and $s_o = s_p^{\mathrm{e}}(\phi_i, s_i)$, then

$$\phi_i = \phi_p^{\mathrm{e}}(\phi_o, s_o),\ s_i = s_p^{\mathrm{e}}(\phi_o, s_o), \text{ and } A_p(\phi_i, s_i) = A_p(\phi_o, s_o)$$

The relationship between exiting and incident radiance in mode p is then

$$L_{o,p}(\phi, s) = A_p(\phi, s) L_{i,p}(\phi_p^{\mathrm{e}}(\phi, s), s_p^{\mathrm{e}}(\phi, s)). \tag{4.4}$$

Here, $L_{i,p}$ and $L_{o,p}$ are the incoming and outgoing radiance, respectively, restricted to mode p. (Unless otherwise stated, the subscript p denotes restriction to mode p from now on.)

Defining the ASF

Having defined the relationship between $L_{i,p}$ and $L_{o,p}$, we want to derive a suitable relationship between $\overline{L}_{i,p}$ and $\overline{L}_{o,p}$, which is the ASF. Substituting (4.4) into (4.3) and assuming $L_{i,p}(\phi, s) \equiv \overline{L}_{i,p}(\phi)/D(\phi)$ gives

$$\overline{L}_{o,p}(\phi_o) = \int_{-D(\phi_o)/2}^{D(\phi_o)/2} A_p(\phi_o, s_o) \frac{\overline{L}_{i,p}(\phi_p^{\mathrm{e}}(\phi_o, s_o))}{D(\phi_p^{\mathrm{e}}(\phi_o, s_o))}\, \mathrm{d}s_o.$$

To define a BCSDF requires a change of variables from s_o to ϕ_i. As observed in prior work, this approach has the problem that the change-of-variables factor $|\mathrm{d}s_o/\mathrm{d}\phi_i|$ is not always finite. In the model of Marschner et al., this was worked around by explicitly removing the singularities, but for non-circular cross sections it is not simple to compute where the singularities occur. On the other hand, surface roughness prevents these singularities from appearing in practice, so we follow the approach pioneered by Zinke and Weber [115] and d'Eon et al. [17] and apply a blur to account for roughness before changing variable.

We will make use of an *angular kernel function* $K^{\gamma_p}(\phi)$ defined with the per-mode width parameter γ_p on the interval $[0, 2\pi)$. The angular kernel function obeys the normalization $\int_0^{2\pi} K^{\gamma_p}(\phi)\ \mathrm{d}\phi = 1$. To maintain symmetry, we convolve $\overline{L}_{i,p}(\phi_i)$ and $\overline{L}_{o,p}(\phi_o)$ each with K^{γ_p}:

$$\overline{L}_{o,p}(\phi_o) = \left[K^{\gamma_p} * \int_{-D(\cdot)/2}^{D(\cdot)/2} A_p(\cdot, s_o) \left[K^{\gamma_p} * \frac{\overline{L}_{i,p}}{D}\right] (\phi_p^{\mathrm{e}}(\cdot, s_o))\ \mathrm{d}s_o\right](\phi_o)$$

This equation can be expanded and written as an integral over ϕ_i:

$$\begin{aligned}
\overline{L}_{o,p}(\phi_o) &= \int_0^{2\pi} K^{\gamma_p}(\phi_o - \phi_o')\bigg[\int_{-D(\phi_o')/2}^{D(\phi_o')/2} A_p(\phi_o', s_o) \\
&\qquad \left(\int_0^{2\pi} K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi_o', s_o) - \phi_i)\frac{\overline{L}_{i,p}(\phi_i)}{D(\phi_i)}\ \mathrm{d}\phi_i\right)\mathrm{d}s_o\bigg]\ \mathrm{d}\phi_o' \\
&= \int_0^{2\pi} \frac{\overline{L}_{i,p}(\phi_i)}{D(\phi_i)}\bigg[\int_0^{2\pi} K^{\gamma_p}(\phi_o - \phi_o') \\
&\qquad \left(\int_{-D(\phi_o')/2}^{D(\phi_o')/2} K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi_o', s_o) - \phi_i)A_p(\phi_o', s_o)\ \mathrm{d}s_o\right)\ \mathrm{d}\phi_o'\bigg]\ \mathrm{d}\phi_i \\
&= \int_0^{2\pi} \frac{R_p(\phi_i, \phi_o)}{D(\phi_i)}\overline{L}_{i,p}(\phi_i)\ \mathrm{d}\phi_i
\end{aligned}$$

where

$$R_p(\phi_i, \phi_o) = \int_0^{2\pi} K^{\gamma_p}(\phi_o - \phi_o')\left(\int_{-D(\phi_o')/2}^{D(\phi_o')/2} K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi_o', s_o) - \phi_i)A_p(\phi_o', s_o)\ \mathrm{d}s_o\right)\ \mathrm{d}\phi_o',$$

and we call this function the *blurred response function.* In Appendix A.2, we show that it is reciprocal: $R_p(\phi_i, \phi_o) = R_p(\phi_o, \phi_i)$. So the integral on the right hand side can be written in terms of ϕ'_i and s_i instead of ϕ'_o and s_o.

With the blurred response function, we could define the ASF as $R_p(\phi_i, \phi_o)/D(\phi_i)$, which obeys reciprocity in the sense of (2.13). However, in this exact form it is difficult to ensure energy conservation, so we define the ASF for the cross section in a slightly different form:

$$N_p(\phi_i, \phi_o) = \frac{R_p(\phi_i, \phi_o)}{D^{\gamma_p}(\phi_i)}, \tag{4.5}$$

where the *blurred diameter function* $D^{\gamma_p}(\phi)$ is the convolution of the projected diameter function with the angular kernel:

$$D^{\gamma_p}(\phi) = \int_0^{2\pi} D(\phi')K^{\gamma_p}(\phi - \phi') \, \mathrm{d}\phi'.$$

The ASF defined above is energy conserving, but it is *not* reciprocal in the sense of Equation (2.13). However, it is close to being reciprocal because it satisfies a similar equation:

$$\frac{N_p(\phi_i, \phi_o)}{D^{\gamma_p}(\phi_o)} = \frac{N_p(\phi_o, \phi_i)}{D^{\gamma_p}(\phi_i)}.$$

That is, instead of the relevant values being equal relative to diameter, they are equal *relative to blurred diameter.* We call this property *approximate reciprocity.* A proof of energy conservation and approximate reciprocity is available in Appendix A.2. The property might be exploited in the design of sampling algorithms of ASFs. However, we did not use it in our implementation.

Discussion

Our ASF is a generalization of d'Eon's ASF to non-circular cross section. One difference between their ASF and ours is that their ASF only blurs over the outgoing direction ϕ_o, but ours blurs over both the incoming and outgoing directions. d'Eon's ASF is reciprocal because it is defined only for circular cross section: it is a function of $\phi_i - \phi_o$, so blurring over either ϕ produces the same result. However, with a non-circular cross section, blurring over ϕ_i is different from blurring over ϕ_o, so blurring over just one angle would result in significant non-reciprocity.

Note also that, in this section, we only have defined the ASF $N_p(\theta_i, \phi_i, \phi_o)$ in terms of the functions $D(\phi)$, $\phi_p^{\mathrm{e}}(\phi, s)$, $s_p^{\mathrm{e}}(\phi, s)$, and $A_p(\phi, s)$, which depend on the incoming azimuthal angle θ_i as well as on the specific shape of the cross section.

4.4.3 Complete Description of the Model

The model is defined by the following parameters:

- The aspect ratio of the elliptical cross section.
- The index of refraction η.
- The absorption coefficient σ of the material inside the fiber.
- For each mode p, the longitudinal shift α_p.
- For each mode p, the longitudinal lobe's width β_p.
- For each mode p, the angular kernel' width used in azimuthal scattering γ_p.

The projected diameter function $D(\phi)$ is defined by the shape of the cross section and is unchanged with respect to θ_i. The three other functions that characterize the cross section depend on Bravais's effective index of refraction:

$$\eta' = \sqrt{\eta^2 - \sin^2\theta_i}/\cos\theta_i.$$

In particular, $\phi_p^{\mathrm{e}}(\phi, s)$ and $s_p^{\mathrm{e}}(\phi, s)$ are defined by tracing the ray (ϕ, s) through the cross section, substituting η' for η, and allowing the ray to intersect the cross section $p+1$ times before going out. For the attenuation,

$$A_p(\phi, s) = \exp(-\sigma \ell_p(\phi, s)/\cos\theta_i) F_p(\phi, s)$$

where $\ell_p(\phi, s)$ is the length of the portion of the path that lies inside the cross section, and $F_p(\phi, s)$ is the product of the Fresnel factors associated with all the intersections with the cross section's surface. If the ray does not emerge from the cross section due to total internal reflection, we say that $A_p(\phi, s) = 0$.

4.4.4 Implementation

We have implemented the full scattering model as a part of a path tracer written in Java.

Rendering

As previously discussed in Section 3.1.1, we model each hair fiber with a collection of circular cylinders. Because the actual projected diameter of fibers is a constant, the outgoing radiance integral (2.11) reduces to:

$$\int_{-\pi/2}^{\pi/2} \int_0^{2\pi} S(\omega_i, \omega_o) L_i(\omega_i) \cos^2\theta_i \; \mathrm{d}\phi_i \mathrm{d}\theta_i$$

We note that, while the actual cross sections are circular, the BCSDF S above can be that of an elliptical cross section. The discrepancy between the actual cross section and the BCSDF's cross section does not cause an accuracy problem: when viewed from a distance, the circular fibers will look as bright as elliptical fibers. To see why this assertion is true, recall that (1) the BCSDF outputs curve radiance, which is the total outgoing energy across the projected diameter, and (2) outgoing radiance is curve radiance per unit projected diameter. Hence, the outgoing radiance is the curve radiance of an elliptical fiber divided by the projected diameter of a circular fiber. Thus, if we accumulate radiance across the circular fiber's projected diameter (which happens when the viewer is far away from the fiber), we get the elliptical fiber's curve radiance, i.e., its total brightness.

Nevertheless, renderings made with circular cylinders can still be different from those made with elliptical cylinders because different geometry results in different paths that light propagate through the scene. (For example, elliptical fibers can cast wider shadows.) Still, we chose to use circular cylinders because it is hard to join two elliptical cylinders in a way that results in one continuous surface.

Cross Sections

We always use an ellipse whose area is π and whose major axis is the v-axis. In other words, the equation of the ellipse in the vw-plane is $v^2/a + aw^2 = 1$. We tabulate the projected diameter function $D(\phi)$ and then evaluate it by linear interpolation.

Unlike with circular cross sections, closed-form formulae for ϕ_p^{e}, s_p^{e}, and A_p are unwieldy for elliptical cross sections, so we compute these functions using a simple 2D ray tracing procedure.

Computing the ASF

We only evaluate the first four ASFs (that is, up to $p = 3$). For each mode, we tabulate the ASF $N_p(\theta_i, \phi_i, \phi_o)$. Because the ASF is an even function in θ_i regardless of the cross sectional shape, we only consider θ_i ranging from 0 to $\pi/2$. On the other hand, we did not consider the symmetry of the cross section and let the azimuthal angles ϕ_i and ϕ_o assume their full range of values: $[0, 2\pi)$. Our grid resolution is $128 \times 256 \times 256$. Each entry of the table contains a single-precision floating point number for each of the three color channels. The total space of the tables is about 400MB.

To precompute the ASFs, we evaluate $N_p(\theta_i, \phi_i, \phi_o)$ in slices of fixed θ_i, slice by slice, by kernel density estimation. For a fixed value of θ_i, we randomly generate $N = 1,000,000$ incoming rays $(\phi_{i,1}, s_{i,1})$, $\ldots$, $(\phi_{i,N}, s_{i,N})$ where ray $(\phi_{i,j}, s_{i,j})$ has power $2\pi D(\phi_{i,j})/N$. For each ray, we compute $\phi_{o,j} = \phi_p^{\mathrm{e}}(\phi_{i,j}, s_{i,j})$, $s_{o,j} = s_p^{\mathrm{e}}(\phi_{i,j}, s_{i,j})$, and $A_j = A_p(\phi_{i,j}, s_{i,j})$. Then, $N_p(\theta_i, \phi_i, \phi_o)$ is approximated as:

$$N_p(\theta_i, \phi_i, \phi_o) \approx \frac{2\pi}{N D^{\gamma_p}(\phi_i)} \sum_{j=1}^{N} A_j D(\phi_{i,j}) K^{\gamma_p}(\phi_i - \phi_{i,j}) K^{\gamma_p}(\phi_o - \phi_{o,j})$$

We use the 1D Gaussian $g(\phi; 0, \gamma_p)$ as our kernel function K^{γ_p} (thereby making $K^{\gamma_p}(\phi_i - \phi_{i,j})K^{\gamma_p}(\phi_o - \phi_{o,j})$ a 2D Gaussian). When evaluating the above sum, we exclude points that are of 2D Euclidean distance at least $3\gamma_p$ from (ϕ_i, ϕ_o). Because we also take into account that the angles ϕ_i and ϕ_o wrap around, our kernel function approximates the Gaussian detector function introduced in [17].

Sampling

Efficient evaluation of the outgoing radiance in a path tracer requires importance sampling of $(\theta_i.\phi_i)$ given (θ_o, ϕ_o). To do so, we drop the incoming radiance term and rewrite the integral as:

$$\sum_{p=0}^{3} \int_{-\pi/2}^{\pi/2} M_p(\theta_i, \theta_o) \cos^2 \theta_i \left(\int_0^{2\pi} N_p(\theta_i, \phi_i, \phi_o) \, \mathrm{d}\phi_i \right) \mathrm{d}\theta_i.$$

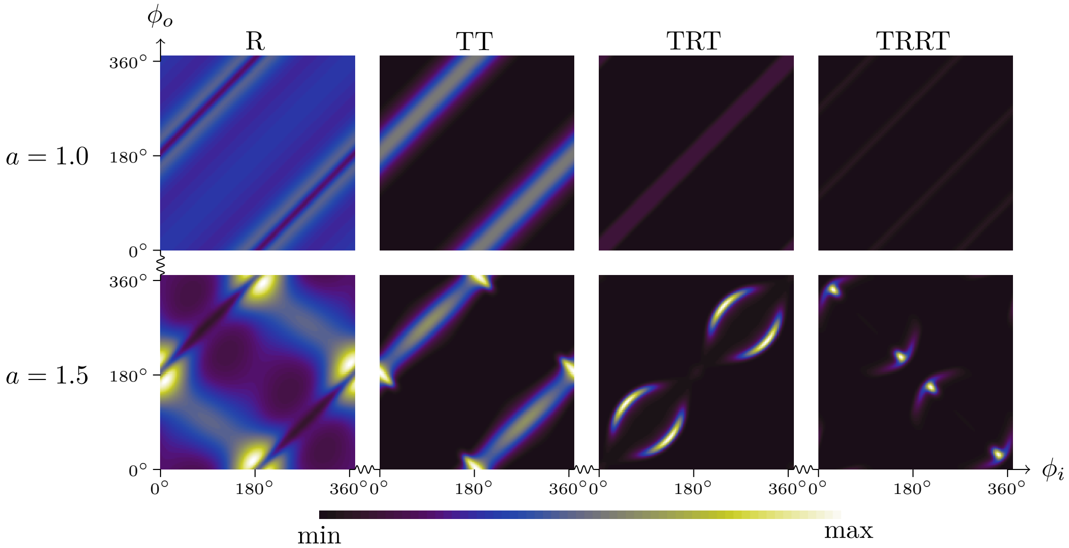
The rewritten integral suggests we divide the sampling of θ_i into three steps: (1) sample the mode p; (2) given p, sample θ_i; and then, (3) given p and θ_i, sample ϕ_i.

To carry out the first two steps, for each mode p, we precompute a 3D table T_p indexed by θ_o, ϕ_o, and θ_i with

$$T_p[\theta_o, \phi_o, \theta_i] = M_p(\theta_i, \theta_o) \cos^2 \theta_i \left(\int_0^{2\pi} N_p(\theta_i, \phi_i, \phi_o) \, \mathrm{d}\phi_i \right).$$

In our implementation, T_p has dimension of $128 \times 128 \times 256$. With the table, the power of each mode can be computed by summing the $T_p[\theta_o, \phi_o, \cdot]$ entries. We can then use the power of the modes to sample a mode. After sampling the mode, we can use the entries in the $T_p[\theta_o, \phi_o, \cdot]$ row to sample a θ_i. After sampling p and θ_i, we can use the table for the ASF to sample ϕ_i. We use binary search to locate the sampled table cell. Efficiency, however, may be improved by more sophisticated probability distribution function inversion techniques [14].

Our implementation entails two space requirements. First, to sample the ASF efficiently, a table of partial sums of $N_p(\theta_i, \phi_i, \phi_o)$ as a function of ϕ_i is required, which incurs 133MB of space in addition to the storage required for N_p itself; thus, the storage pertaining to N_p is 533MB. Second, we must build T_p, which is approximately 67MB in size.



All the plots were calculated with $\eta = 1.55$, and $\sigma = 0$. Plots in the same columns are to scale with one another, but not across the columns. The darkest color (black) corresponds to the minimum value of plots in the same column, and the brightest color (white) corresponds to the maximum value.

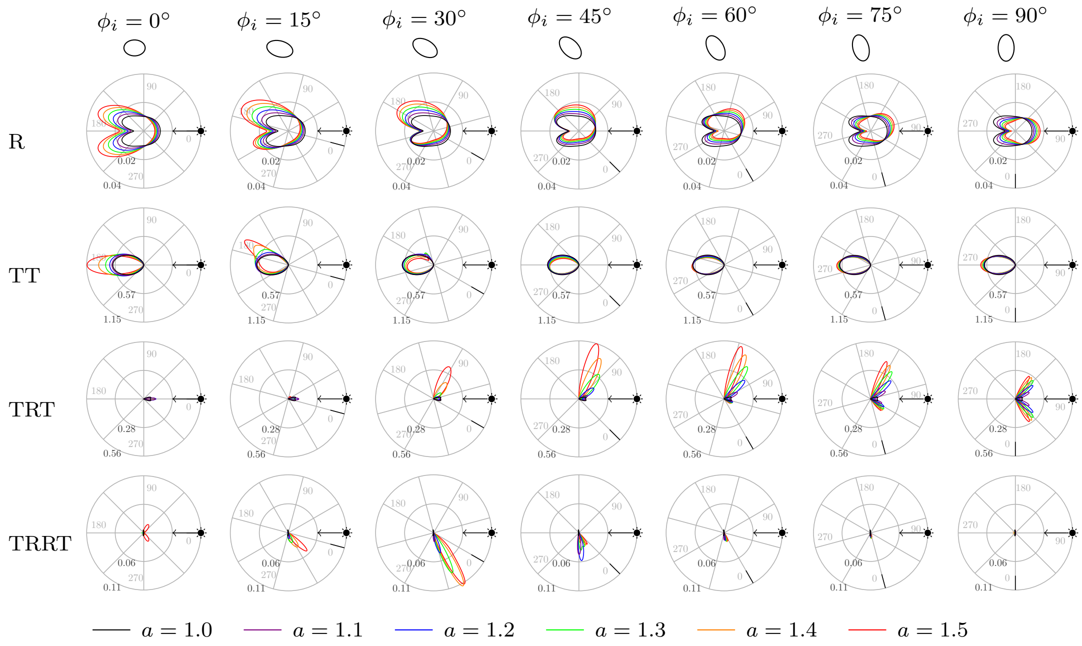
According to the model, elliptical fibers give rise to many features not present in circular fibers such as the bright bars in the R mode (⟍), the extra lobes in the TT mode (∴), the eyes in the TRT mode (ₒᵒ), and the winged blobs in the TRRT mode.

Figure 4.4: (ϕ_i, ϕ_o)-plots of the first four modes of the ASF at $\theta_i = 0°$ for an elliptical fiber with $a = 1.0$ and $a = 1.5$.

4.5 Prediction

In this section, we compute the ASFs of elliptical fibers and report the predictions of our model for the first four scattering modes: R, TT, TRT, and TRRT. We only consider fibers with $a \leq 1.6$ and $\sigma = 0$. While this is only a small subset of the possible range of parameters, it is sufficient to illustrate the differences between elliptical ASFs and circular ones and to show the qualitative trends that the ASF follows as a and θ_i change.

We provide two types of visualizations of the ASF. The first type (Figure 4.4 and Figure 4.6) is the (ϕ_i, ϕ_o)-plots discussed at the end of Section 4.3. The second type (Figure 4.5) is a polar plot of the ASF with ϕ_o as the free variable. Displaying

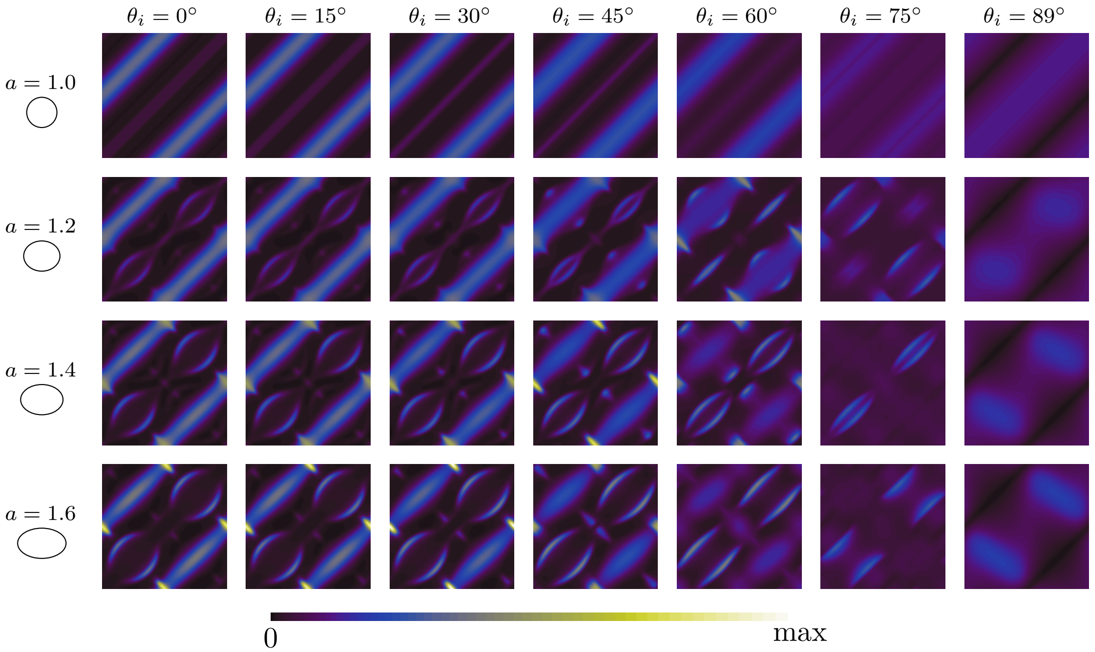


The plots are generated from the equation $r = N_p(0°, \phi_i, \phi_o)$ with ϕ_o being the free variable. They are rotated so that the light is coming from the right. All ASF values were calculated with $\eta = 1.55$, $\theta_i = 0°$, $\sigma = 0$, and $\gamma_p = 5°$. Plots in the same row are to scale with one another, but not across the rows. Due to the symmetry of ellipses, plots of the ASF with ϕ_i being a multiple of 15° from 90° to 360° can be deduced from the plots we show here. In general, one can observe that higher aspect ratios are associated with brighter lobes in all modes.

Figure 4.5: Polar plots of the ASF of elliptic cross sections with six different aspect ratios.

the ASFs of different shapes on the same plot makes it easy to compare them. The polar plots in Figure 4.5 correspond to vertical slices through these images. The (ϕ_i, ϕ_o)-plots allow us to see the complete picture of how the ASF evolves with ϕ_i, and they expose the inherent symmetry due to reciprocity. It also provides a convenient way of observing how the ASF evolves with changing θ_i.

In Figure 4.4, the four modes of a circular fiber and an elliptical fiber are plotted separately to show the pronounced effect of ellipticity and the distinct features of each mode. In Figure 4.6, we plot the sums of the four modes for a number of combinations of cross sectional shapes and longitudinal angles to show how the modes evolve with these parameters.



All functions were calculated with $\eta = 1.55$, $\sigma = 0$, and $\gamma_p = 0°$. All the plots are to scale with one another. The darkest color (black) corresponds to the value 0, and the brightest color corresponds to the highest value in all the plots, which is roughly 1.8. While the R mode gets brighter as θ_i approaches 90°, all other modes becomes dimmer and eventually disappear at grazing angles.

Figure 4.6: (ϕ_i, ϕ_o)-plots of the sum of the R, TT, TRT, TRRT modes of the ASFs of elliptical fibers with four aspect ratios under seven values of θ_i.

We now make observations of the modes, one by one.

R mode. From the first row of Figure 4.5, we see that the R mode of elliptical fibers (colored curves) is quite similar to circular ones (black curves). All the plots contain two forward lobes. More eccentric ellipses scatter more forward and less backward than less eccentric ones when $\phi_i = 0°$, and the reverse holds at $\phi_i = 90°$.

The (ϕ_i, ϕ_o) plot in Figure 4.4 shows that the R mode of an elliptical fiber contains two bands of brighter reflection (⟍). More specifically, this brighter reflection occurs when when $\phi_i + \phi_o \approx 180°$, which is to say, the half vector is normal to the flat side of the ellipse.

However, as can be seen from Figure 4.5, the R mode remains quite dim even for elliptic fibers, so its visual significance is low when the other components are present.

TT mode. In Figure 4.5, the plots of the TT mode with $\phi_i = 15°$ and $\phi_i = 30°$ indicate that more elliptical fibers have two lobes in the TT mode. As in the circular case, there is a lobe that peaks at the forward scattering direction $\phi_o \approx \phi_i + 180°$, but there is also an extra lobe, present only when ϕ_i is near 0° or 180° (when the incident light is seeing a narrower cross section) that peaks somewhat to the side. In the (ϕ_i, ϕ_o)-plot, the forward scattering lobe manifests as the two long parallel strips (⫽) as their centers are 180° "above" the line $\phi_i = \phi_o$. The extra lobe manifests as the small elongated blobs perpendicular to the bars (∷).

Figure 4.5 and Figure 4.6 suggests that the extra TT lobe is the brightest feature of the ASF (at least for highly transparent fibers). Moreover, as the cross section becomes more elliptical, the extra lobe gets brighter. The behavior of the lobes as θ_i increases is more complicated. The extra lobe gets brighter as θ_i increases and then quickly disappears, but the forward scattering lobes seem to get broader and dimmer slowly as θ_i increases.

In Appendix A.3, we argue that the extra lobe arises from a group of incoming rays whose directions become very similar as they exit the cross section. As such, its origin is similar to that of the glints as described by Marschner et al. [49].

TRT mode. The TRT mode of a circular fiber is dim compared to elliptical fibers. At $\theta_i = 0$, it contains two caustic lobes, which are dim and small in extent. As θ_i increases, the lobes merge into one brighter lobe, which then becomes broader and dimmer as θ_i continues to increase.

The TRT modes of elliptical fibers have much more complex behaviors. In the (ϕ_i, ϕ_o)-plot, the mode appears to have two components: the two "eyes" ( ) and the center blob ( ). The center blob is very dim and appears as a single lobe only when ϕ_i is near 0° or 180°; that is, when light strikes the narrow sides of the ellipses. The eyes contain two asymmetric lobes, many times brighter than in a circular fiber, that appear in the range of ϕ_i complementary to the center blob. Our observation is consistent with Marschner et al. [49], who observed that as ϕ_i changes, the two TRT lobes seem to merge into a single lobe as the cross section rotates so that the narrower side faces the light.

Figure 4.6 suggests that the eyes will become brighter as the fiber becomes more eccentric. The distance between the two eyelids and the curvature also seem to increase as a increases, so we can say that more elliptical fibers have "wider" eyes, meaning that the peaks occur farther from the incident direction. On the other hand, the center blob becomes broader and dimmer.

In fibers with small eccentricity, as θ_i increases, the eyelids get closer to each other, eventually collapse into a line, and then quickly become broader and dimmer. In highly eccentric cross sections ($a = 1.6$ in particular), the eyelids do not collapse but seem to maintain their distance while inverting the direction of curvature.

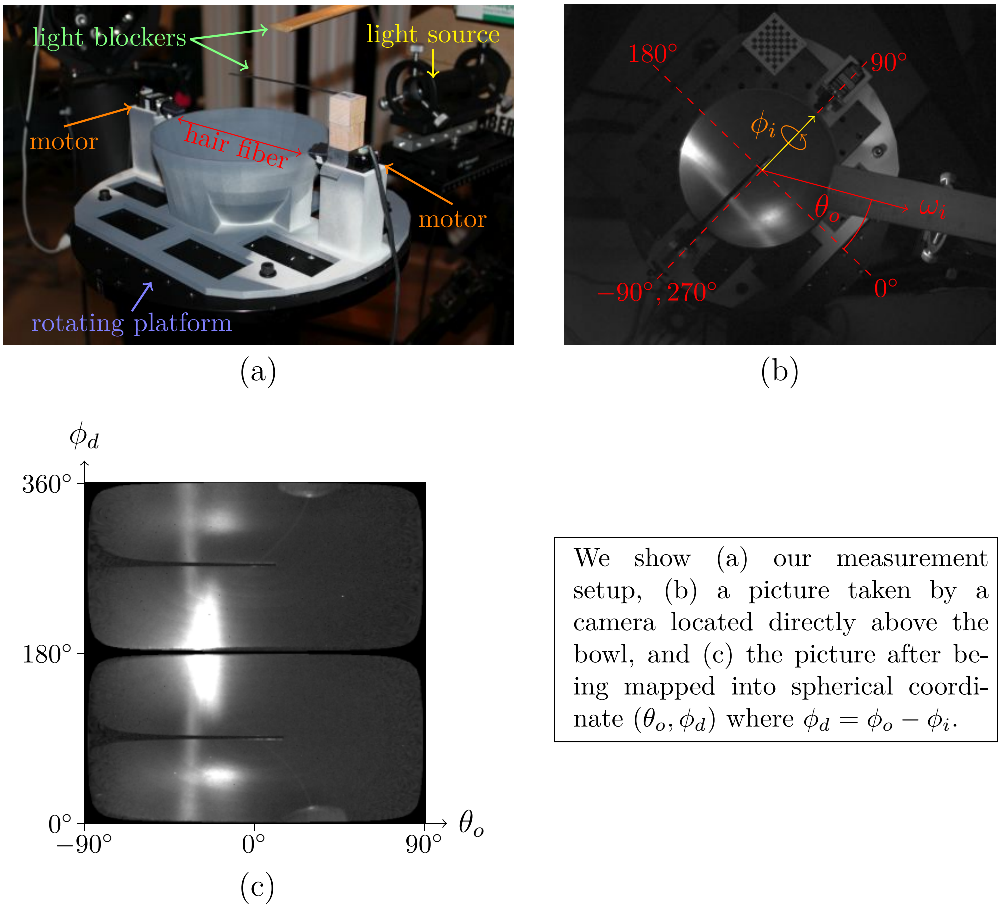
TRRT mode. In the (ϕ_i, ϕ_o)-plot, the TRRT mode of an elliptical fiber manifests as four "winged blobs" that occur near the line $\phi_i + \phi_o = 0$, when the half vector is aligned with the long axis of the ellipse. The TRRT mode is insignificant in circular fibers, but Figure 4.6 suggests that the blobs should be observable, particularly since they occur at angles far from where other brighter components dominate. When θ_i is small, the blobs are dimmer than the TT and TRT modes, but they can remain bright and become brighter as θ_i increases and the TT and TRT modes start to disappear. As a result, we include the TRRT mode in our implementation while other works only include up to the TRT mode [49, 73].

In general, the blobs of the TRRT mode get larger and brighter as θ_i increases, and, after a point, they seem to quickly disappear. The angle θ_i where the blobs disappear seems to decrease as a increases.

To summarize, elliptical cross sections give rise to *much* more structure in the scattering function than is seen in circular fibers, with bright lobes in the TT and TRRT modes, which can appear and disappear quickly as ϕ_i changes. The TRT lobes, while similarly bright (and much brighter than in circular fibers), are more stable, but their locations and brightness vary based on eccentricity and orientation of the fiber. As a result, the model predicts that hair fibers with different aspect ratios will have glints at different angles.

4.6 Measurement Device

To investigate how the behavior of real fibers compares to the predictions discussed in the last section, we designed and built a new image-based system for measuring scattering from individual fibers. This new system provides much higher angular resolution than previous methods based on 4-axis gonioreflectometers [49], and thereby produces a more complete picture of the scattering function than has been seen before. The device, however, was designed for observing qualitative features of the scattered light and not for accurately measuring the BCSDF.



(a)

(b)

(c)

We show (a) our measurement setup, (b) a picture taken by a camera located directly above the bowl, and (c) the picture after being mapped into spherical coordinate (θ_o, ϕ_d) where $\phi_d = \phi_o - \phi_i$.

Figure 4.7: Device for measuring light scattering from a fiber.

4.6.1 Description

Figure 4.7a shows our measurement system. Its main component is a bowl-shaped receiving surface which approximates a hemisphere of radius 7.5cm. The top part of the hemisphere is left out so that the height of the bowl is 7.3cm. The bowl was created by 3D printing and painted a diffuse gray.

To perform a measurement, we suspend a hair fiber above the bowl, running along the bowl's diameter so that the fiber passes through the center of the hemisphere.[2] Light from a white LED is channeled through optics to produce a narrow

[2] Short fibers must be extended—for example, by attaching other fibers to both ends—to be

beam that illuminates a short length of the fiber at the center. The illuminated section is between 3mm and 6mm long at normal incidence, and the solid angle of the beam is adjustable between 7×10^{-5} and 2×10^{-4} steradian The two ends of the fiber are attached to two stepper motors, which rotate in tandem around the hair's axis to change ϕ_i, and the bowl rests on a rotating platform so that θ_i can be controlled by rotating the whole apparatus relative to the source.

The bowl receives the light scattered by the hair fiber into the lower hemisphere, and reflects it approximately diffusely. We measure this reflected light by taking a photograph using a camera mounted on one arm of the gonioreflectometer. The arm was positioned so that the camera views the bowl from directly overhead. The relative position between the camera and the bowl was computed by the Caliber camera localization and calibration system [44]. Because one photograph only captures a hemisphere of scattered light, we need two photographs to capture a complete sphere. In particular, if we take a picture where the direction of incoming light is (θ_i, ϕ_i), then we have to take another photograph with light direction $(-\theta_i, \phi_i + 180°)$ to capture the complementary hemisphere. The motors and the rotating platform provide two degrees of freedom and the surface of the bowl provides two more, enabling measurement of the complete 4D scattering function.

A picture taken by the camera is shown in Figure 4.7b, annotated to illustrate the angles that correspond to ones in the theoretical setting. The intensity of each pixel of the photograph that corresponds to the bowl's surface is a point sample of the hemisphere of scattered light. These point samples are the data produced by the system. We typically parameterize them with the spherical coordinates (θ_o, ϕ_d) where θ_o ranges from $-90°$ to $90°$, and $\phi_d = \phi_o - \phi_i$ ranges from $0°$ to $360°$. The

at least 20cm long before mounting. This way, we have used the device to measure a short wool fiber. (See Section 5.4.)

coordinates can be computed easily from the unit vector from the bowl's center to the point on the bowl surface corresponding to the pixel.

In Figure 4.7c, we visualize the sphere of scattered light by plotting the point samples of the complete sphere on the plane where the horizontal axis is the azimuthal angle θ_o, and the vertical axis is the angle $\phi_d = \phi_o - \phi_i$. This is equivalent to the (θ_o, ϕ_d)-plot discussed in Section 4.3.

4.6.2 Practical Design Issues

The device is designed to allow the illuminating beam to clear the edge of the bowl all the way around, so it has a blind spot about 5° wide around the equator of the sphere—all directions for which $\phi_i \approx \phi_o$ or $\phi_i \approx \phi_o + 180°$.

In addition to the desired light, which travels from the source to the hair to the bowl to the camera, other paths contribute to the image as well. The direct path from the source to the hair to the camera produces a bright line at the center of the image, and the path from the source to the bowl to the camera[3] produces a bright area on the rim of the bowl right next to the source. To combat lens flare these paths are blocked by two occluders; this is important to enable the camera to record the very dim illumination on the inside of the bowl, but they create two more narrow blind spots in the data.

Moreover, diffuse interreflection paths from other points on the bowl's surface also contribute; these are mitigated by the relatively low reflectance of the bowl and contribute an approximately constant component that is near the noise floor of

[3] The source is designed to miss the bowl, but when its aperture is fully open the beam spills slightly onto the rim, and even when it is closed down further, stray light still produces significant illumination on the rim.

our measurements. We will quantify this diffuse interreflection in the next section.

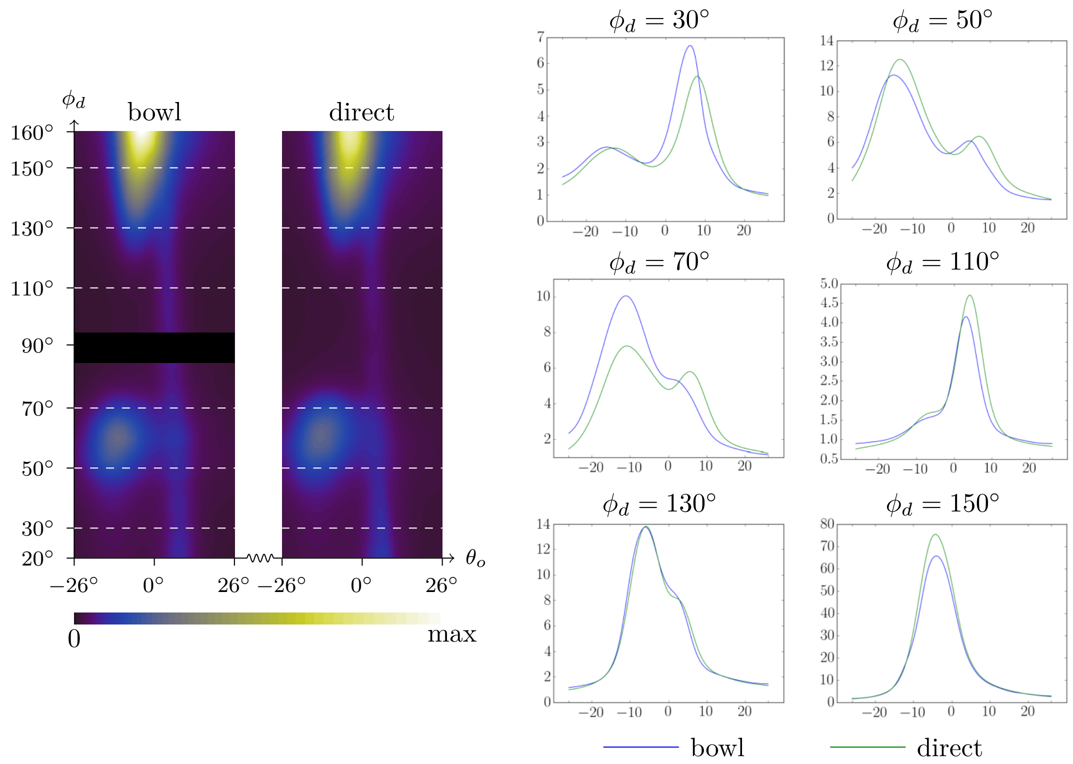
We wrote earlier that we take a "photograph" to capture an outgoing hemisphere of scattered light. In practice, this is a high dynamic range image acquired through three exposures typically at 5, 15, and 30 seconds with the QImaging Retiga scientific camera. The exposure times were chosen so that the brightest parts of the bowl surface were not overexposed. The image formation model we used to merge these photographs is detailed in Appendix A.4.

4.6.3 Quantitative Characterization

To characterize the accuracy of our device, we compared our results to direct measurements of light scattered from a hair fiber using the following procedure. We mount a Caucasian hair fiber[4] on the bowl, rotate the platform so that $\theta_i = 90°$, and take a photograph of the bowl surface. We then rotate the fiber by 180° in θ and in ϕ, so that the incident direction is the same, but the scattered light just measured using the bowl are visible from above. Subsequently, we acquire 4,048 images of the hair, corresponding to outgoing directions with $-26° \leq \theta_o \leq 26°$ and $20° \leq \phi_d \leq 160°$. From each photograph, we identify the illuminated region, sum up the pixel values inside, divide the sum by $\cos \theta_o$, and interpret the resulting value as the amount of light scattered from the hair fiber in the corresponding direction. This process yields 4,048 unstructured samples of the scattered light.

The data from the above measurement process cannot be directly compared to the bowl measurement for several reasons:

[4] We used a fiber from Sample C, the details of which are given in Section 4.7.



A segment around $\phi_d = 90°$ of the bowl measurement is blacked out because it contains the blocker, and point samples in the area were removed before the reconstruction of the smooth function. Our optimization yielded the optimal translation with $\Delta\theta_o = 0.62°$ and $\Delta\phi_d = -0.85°$, $k_d = 2.58 \times 10^{-2}$, and $I_a = 24.23$. Here, I_a is in the unit of pixel values at the time we performed the direct measurement. The camera settings between the bowl and the direct measurement were identical, except that, in the direct measurement, we turned the aperture of the camera down so the camera took in 16 times less light, i.e. 32 f-stops.

The comparison shows that the device can locate peaks in the BCSDF, making it suitable for qualitatively observing light scattering behavior of hair fibers. However, it is not accurate enough to give definite numerical values of the BCSDF.

Figure 4.8: Comparison between the direct measurement and the measurement made by our device.

1. The numbers of samples are not the same between measurements, and the samples are not at the same (θ_o, ϕ_d) coordinates.
2. The bowl measurement is much more noisy because of the roughness of the bowl surface.
3. The bowl data represents light reflected off the bowl surface, not the light directly scattered by the fiber.

4. Due to uncertainty in the automatic calibration of camera positions when performing the direct measurement, the data of the two measurements are not aligned perfectly. (For the bowl measurement, we manually calibrated the camera so that the vertices of 3D model of the bowl aligned with those in the photograph.)

To remedy the first two issues, we reconstructed a function from (θ_o, ϕ_d)-plane to the light intensity I by Gaussian kernel smoothing: if the point samples are $(\theta_{o,1}, \phi_{d,1}, I_1), \ldots, (\theta_{o,N}, \phi_{d,N}, I_N)$, then the intensity value at (θ_o, ϕ_d) is given by:

$$I(\theta_o, \phi_d) = \frac{\sum_{j=1}^{N} K(\theta_o, \phi_d; \theta_{o,j}, \phi_{d,j}) I_j}{\sum_{j=1}^{N} K(\theta_o, \phi_d; \theta_{o,j}, \phi_{d,j})}$$

where K is a 2D circular Gaussian function with standard deviation of 2°.

To remedy the third issue, we recognize that the intensity in the direct measurement I_{direct} is related to the intensity in the bowl measurement I_{bowl} by:

$$I_{\text{bowl}} = k_d(I_{\text{direct}} + I_a) \tag{4.6}$$

where k_d is the diffuse reflectance of the bowl surface, and I_a is the intensity of ambient light, which comes from multiple reflection off the bowl surface. We optimized for k_d and I_a by minimizing the square error between the RHS and LHS of (4.6) at evenly spaced discrete sample points in the rectangle $-26° \leq \theta_o \leq 26°$ and $20° \leq \phi_d \leq 160°$.

To remedy the last issue, we optimized for a translation in the (θ_o, ϕ_d)-plane that, when applied to the direct sample points, minimizes the square error mentioned in the last paragraph. The shift in ϕ_d accounts for error in the rotation of the turntable, and the shift in θ_o partially accounts for error in the vertical position of the hair fiber.

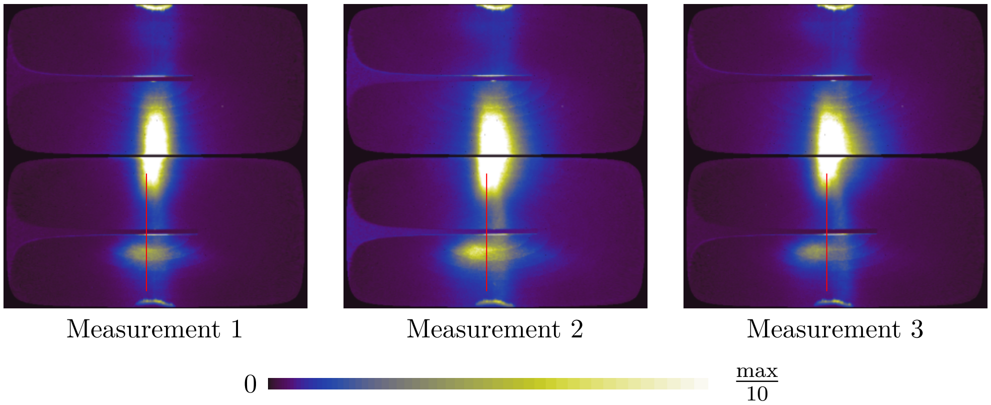
The plots of the direct and the bowl measurement after all the above adjustments are given in Figure 4.8. We can see that both measurements contain peaks that appear at approximately the same locations. That I_a is positive is consistent with the fact that interreflection can only add to the brightness of the reflected light from the bowl. Now, consider the plot with $\phi_d = 110°$, which contains a peak of the R mode whose value is approximately 4. Because $k_d I_a \approx 0.63$, we infer that interreflection can be as bright as 15% of the R mode, implying that the R mode and other brighter modes can be clearly distinguished from background. However, while the peaks can be located and observed easily, their amplitudes in the two measurements do not match. Taking the direct measurement as ground truth, relative error can be as large as 101%. (This might be due to the fact that, even after applying the translations, the measurements did not align well enough.) In conclusion, we demonstrated that our device is adequate for observing and locating peaks in the BCSDF but not suitable for quantitatively measuring the function.

Nevertheless, we would like to point out the expediency of our device: the whole hemisphere of outgoing light can be acquired in less than a minute. On the other hand, the direct measurement took about 6 hours to perform but yielded only a small portion of the hemisphere.

4.6.4 Repeatability

We next test the repeatability of our measurements. To do so, we used the device to capture the scattered light from the same portion[5] of a human hair fiber three

[5] We marked a point 5cm from the root end of a fiber with black ink. When mounting the fiber on the bowl, we made sure that the marked part just cleared the end of the 3D printed part we attached to one of the motors' shaft. The ink mark was about 2mm wide, so the illuminated part in each measurement could be different, but they will be no further than 2mm apart.


Measurement 1

Measurement 2

Measurement 3

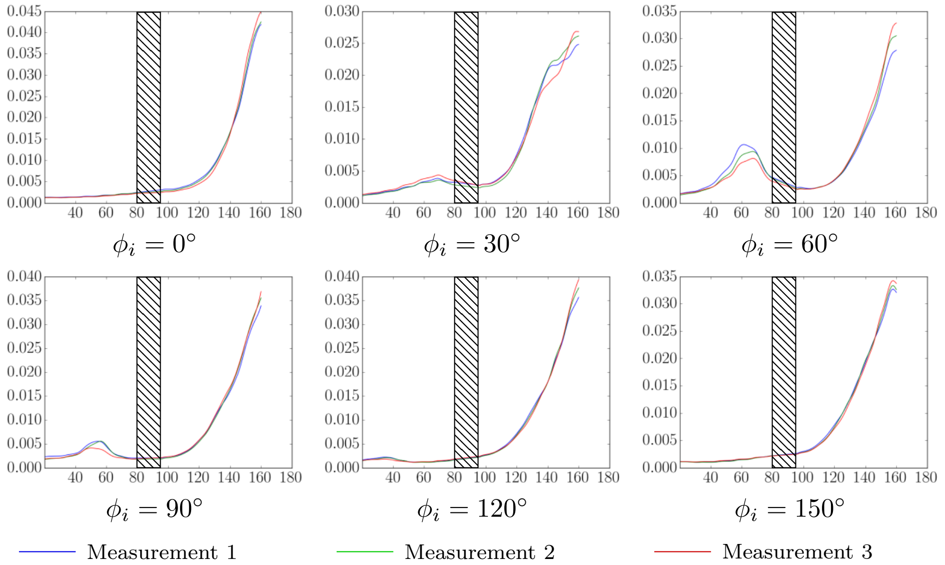
Here, we show the (ϕ_d, θ_o)-plots of the same sphere of scattered light, acquired 3 times. While the images appear similar, there are differences in brightness. In creating the visualization, we rendered all pixels whose brightness is greater than one tenth of the maximum value to be white. This was done to make the brightness difference in the TRT blobs near the bottom easy to see. The red lines in the pictures are the 1D slices graphed in Figure 4.10.

Figure 4.9: Visualization of one of the 180 spheres acquired in the three measurements of the same hair fiber.

times. For each measurement, we set $\theta_i = 90°$, rotated the step motors so ϕ_i takes 180 equally spaced values in the 360° range, and acquired the full sphere of scattering light for each of the ϕ_i values. This process results in 180 photos which constitute a 3D slice of the BCSDF with θ_i hold fixed.

We visualize one of 180 spheres from the three measurements in Figure 4.9. It can be seen that, while all three measurements contain the same features at roughly the same locations, the brightness and shapes of the features do not match exactly: those of Measurement 2 have the highest magnitude, followed by those of Measurement 1 and then Measurement 3. We hypothesize that the differences in brightness is due to differences in positioning the fiber in the beam, which is not uniform across its vertical extent.

Nevertheless, the measurements generally agree up to a constant factor. To see this, we also show graphs of a 1D slice of the measurements for several angles



$\phi_i = 0°$ $\phi_i = 30°$ $\phi_i = 60°$

$\phi_i = 90°$ $\phi_i = 120°$ $\phi_i = 150°$

Here, we show plots of 1D slices of three measurements made by our device. The slices' locations are indicated by the red lines in Figure 4.9. The x-axis represents ϕ_d, in degrees. We throw away data from $\phi_d = 80°$ to $\phi_d = 95°$ because they are contaminated by the blocker. To account for the difference in scale between the measurements, we made the y-axis represent the ratio of brightness at each point to the sum of brightness across all points in the same measurement. (That is, we divide the brightness values by their sum.) The fact that all graphs (after correcting for scale differences) roughly coincide implies that the device is reliable enough to observe qualitative features in hair BCSDFs.

Figure 4.10: Graphs of 1D slices of three repeated measurements for 6 different values of ϕ_i.

in the Figure 4.10. Here, the brightness values are divided by the sum of the values in the same measurements to account for the difference in scale between the measurements. We can see that the graphs have generally the same shapes.

From the measurements, we can conclude that our device is not reliable at recording absolute brightness of the scattered light. This is the direct consequence of our not controlling many factors that can affect the recorded brightness such as the vertical position of the fiber, the uniformity of the light beam, and the reflectance properties of the bowl itself. However, because the same features consistently appeared in all measurements, the device is suitable for observing large-scale features of the BCSDF. We believe that the shortcomings of the device can

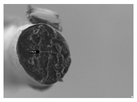
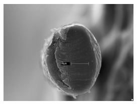
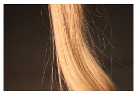
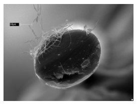
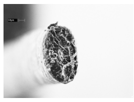
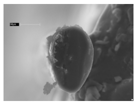
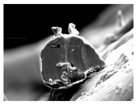
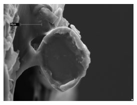
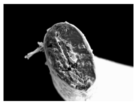
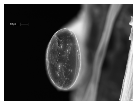
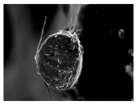
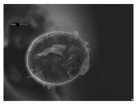
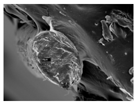
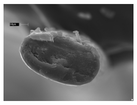
	Source	Photo	Fiber 1		Fiber 2	
			SEM image	Aspect ratios	SEM image	Aspect ratios
A	Caucasian			1.136 1.213 1.469 ($\bar{x} = 1.273$)		1.256 1.268 1.411 1.429 ($\bar{x} = 1.341$)
B	Caucasian			1.446 1.501 ($\bar{x} = 1.473$)		1.077 1.195 1.246 1.353 1.436 1.552 ($\bar{x} = 1.310$)
C	Caucasian			1.603 1.717 ($\bar{x} = 1.660$)		1.548 1.553 1.746 1.762 ($\bar{x} = 1.605$)
D	artificial wig			1.127 1.385 ($\bar{x} = 1.256$)		1.147 1.159 1.233 1.310 ($\bar{x} = 1.212$)
E	African			1.834 1.923 1.933 ($\bar{x} = 1.897$)		1.561 1.647 1.789 ($\bar{x} = 1.666$)
F	Chinese			1.308 1.361 ($\bar{x} = 1.334$)		1.035 1.123 1.142 1.183 ($\bar{x} = 1.121$)
G	Indian			1.310 1.482 ($\bar{x} = 1.396$)		1.503 1.524 1.609 1.616 1.780 1.839 ($\bar{x} = 1.645$)

Table 4.1: Hair samples.

be overcome by better engineering, but, as it stands, the device is adequate for our purpose.

4.7 Measurements

In this section, we present measurements of a broad range of hair fibers and discuss features that we observed.

4.7.1 Hair Samples and Measurement Setups

We measured hair fibers taken from 7 hair samples, which are identified by letters from A to G. We took 2 hair fibers from each hair sample, for a total of 14 fibers. The details of each fiber are given in Table 4.1.

The setups for all the measurements are almost the same, but with differences in the setting of the aperture in the light source optics. We measured all fibers that are numbered "1" first with the aperture fully open. As noted previously, this setup results in increased stray light. To reduce contamination of the data, we measured the fibers that are numbered "2" with the aperture closed to roughly one-third of the maximum area. This resulted in less stray light and sharper scattering patterns, but required longer exposures to capture the dimmer signals.

For each fiber, we performed three types of measurements.

Aspect Ratio Estimation

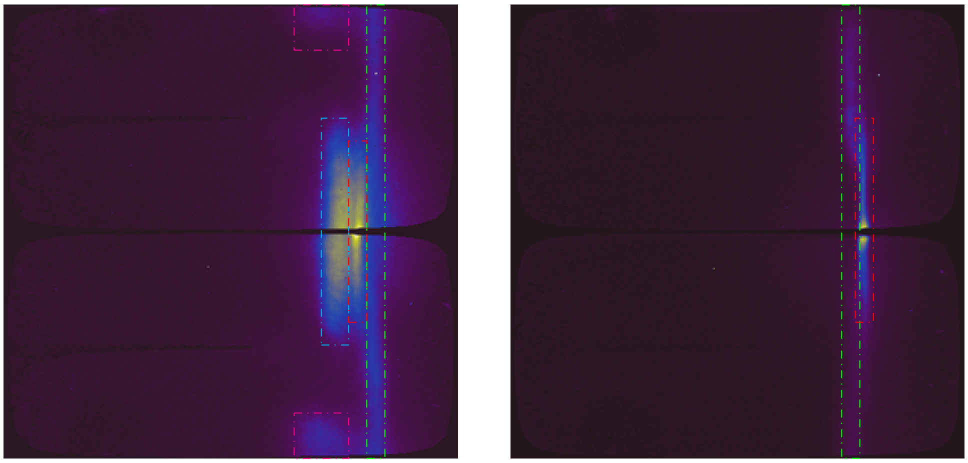
As this chapter seeks to study the effect of ellipticity on the scattering behavior of hair fibers, we also made measurements to directly determine the fibers' ellipticity. After the scattering measurements for each fiber, we cut the fiber at 2 to 6 locations near the illuminated spot and used a scanning electron microscope to image the cut ends. This procedure images the fiber's cross sections at locations up to 3cm from the illuminated spot. For each of the micrographs, we identified 20–60 points on the boundary of the cross section and fit an ellipse[6], taking the ratio between the major and minor axes of this ellipse to be the aspect ratio of the cross section. All the resulting aspect ratios and their averages can be found in Table 4.1. The full set of micrographs and fitting data is given in the supplementary material of [38].

This process produces several samples of the aspect ratio of each fiber, but the data show that the aspect ratio can vary considerably along the length of a single fiber. Therefore they give only a rough indication of the aspect ratio at the scattering point. Several other factors also contribute to error in the measurements, including: (1) the cut might not be exactly perpendicular to the fiber's axis; (2) the cutting might have distorted the cross section's shape; and, (3) in some images, the cross sections were viewed at an angle 10° to 20° from the fiber axis, which had to be estimated manually. Despite these uncertainties, the data still provide useful information about how elliptic each fiber is.

4.7.2 (θ_o, ϕ_d)-Measurements

We start with (θ_o, ϕ_d)-measurements as they are easily derived from a single photograph of the bowl. The measurements allow us to identify the observable features in the scattered light. To study how these features evolve as the incoming scattering angle θ_i changes, we varied the incoming longitudinal angle θ_i from 80° to -80° with the resolution of 2°. For each fiber sample, we chose the azimuthal angle ϕ_i so that the features characteristic of the TRT mode can be observed. For each

[6] We used the fitting algorithm described in [22] and the code from [91].



(a) Fiber B2

(b) Fiber E2

The two (θ_o, ϕ_o)-plots were generated from measurements at $\theta_i = -30°$.

Observed features marked by colored rectangles: R mode is green, TT mode is cyan, and TRT mode is magenta. Fiber E2 has dark color, so the TT and TRT mode are not present. The red rectangle surrounds the E mode, a phenomenon not predicted by any hair scattering models including ours.

Figure 4.11: Observable features in the (θ_o, ϕ_o)-plot of Fiber B2 and Fiber E2.

θ_i, we captured a (θ_o, ϕ_d)-plot, resulting in 81 plots. From these plots, we created videos (available in the supplementary material of [38]) that show the evolution of the scattering function as a function of θ_i. We also provide a sampling of these plots in the supplementary material of [38].

Observable features

We found four different features in the generated plots, not all present in all fibers, which we attribute to different modes of scattering from the fibers. The features are identified for an example plot of Fiber B2 in Figure 4.11a. The features are:

- A bright, horizontally narrow, forward scattering line that always occurs at $\theta_o = -\theta_i$ (within the accuracy of our image registration). The feature is surrounded by the red rectangle in Figure 4.11.

- Two vertical strips; one originating from the top edge of the plot and the other from the bottom edge. Both strips curve towards the previous forward scattering line. (Green.)
- One or more forward scattering lobes that are horizontally wider than the previous features. (Cyan.)
- Two small blobs, one appearing near the bottom edge, and the other near the top edge. (Magenta.)

According to the heuristics discussed at the end of Section 4.3, we believe the green and magenta features are the R and TRT modes, respectively.

The four features are not present in all the plots. As can be seen in Figure 4.11b and the supplementary material of [38], the cyan and the magenta features cannot be observed in plots of fibers from Samples E, F, and G, all of which are strongly pigmented fibers that appear black. This fact suggests that these two features are caused by light being transmitted through the fibers while the other two are not. As such, we believe that the cyan feature is the TT mode.

There are some disagreements between the data and the predictions made by previous models. The first is that previous models predict that the TT mode should appear more or less as one contiguous lobe. This prediction matches the wider forward scattering blobs of all light colored fibers except those from Sample C where there seem to be two separated blobs: one in the top hemisphere and the other in the bottom. On the other hand, the elliptical model allows the TT mode to manifest as two lobes: one for the slanted bar and one for the extra lobe discussed in the last section.

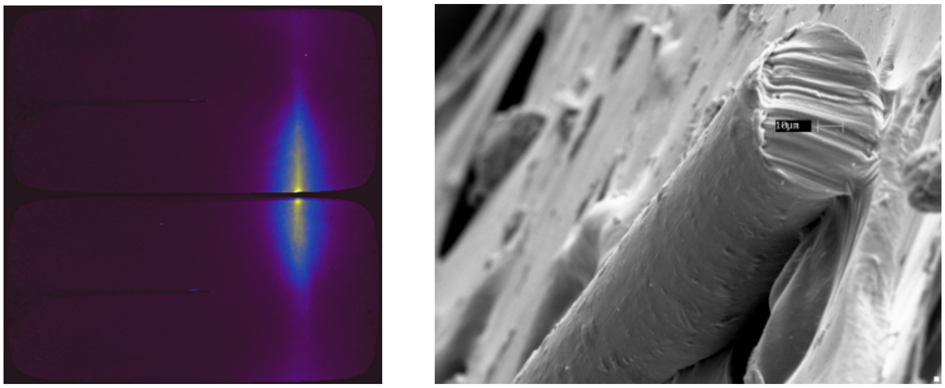
The second is that previous models predict that TRT blobs should appear only for values of ϕ_d that are within 20° of 0°—that is, quite close to the top or bottom of the plots. However, in the measurements such blobs appear as far as almost 90° from the plane of incidence. We showed through modeling of ASFs of elliptical fibers in Section 4.4 that with an elliptical cross section these blobs can still be explained by the TRT mode.

However, all models, ours included, do not predict the red feature.

The E Mode

The feature marked in red appears even for black hair, and its intensity increases at grazing angles. As such, it seems to be caused by reflection from the surface; therefore we consider it part of the R mode. However, it is distinct enough to need its own name, so we refer to it as the "E" mode. Here, E stands for "equi-angle peak" (EAP), a feature of light scattered from human hair reported by Stamm et al. [85] which we believe is the same as our red feature.

In particular, Stamm et al. observed a faint peak at the ideal specular angle which grows brighter as θ_i increases in magnitude. Nevertheless, they only measured light scattered backward ($\phi_d = 0°$) while we measured the whole range of ϕ_d. This enables us to see that the E mode is the brightest around forward directions and is very sharp in the θ_o direction. It is clearly separated from the rest of the R mode, and it grows stronger as θ_i approaches grazing angles, just like Stamm's EAP. Hence, Stamm' EAP was the part of the E mode that crosses the $\phi_d = 0°$ line. Note that Marschner et al. also measured a full hemisphere of scattered light, but not with high enough resolution to distinguish the E mode from other modes.



(a) (b)

Here, we show (a) a (θ_o, ϕ_d)-plot of Fiber D2, which is artificial, and (b) an SEM micrograph showing the surface of the artificial hair, which has no scales. Notice that all the modes' centers are aligned horizontally.

Figure 4.12: The absence of longitudinal shifts in a fiber without scales.

The cause of the E mode is unclear. Stamm et al. provided an explanation in terms of reflection from subsurface interfaces, but it does not fit our full 3D observations. We conjecture that the E mode may be a wave optics effect related to the phenomenon of emerging specularity on flat surfaces [26], but a proper explanation will have to await further investigation.

Longitudinal Shifts

With the exception of fibers from the artificial wig, we found that all fibers exhibit longitudinal shifts of the R, TT, and TRT modes, consistent with what has been observed in previous works. That is, the R mode shifts to one side of the $\theta_o = -\theta_i$ line while other modes shift to the other side. On the other hand, as can be seen in Figure 4.12a, all the modes of fibers from the artificial wig are centered horizontally at $\theta_o = -\theta_i$. This observation can be explained by the fact that fibers from the artificial wig have no scales (see Figure 4.12b) and behave like smooth cylinders. Consequently, all modes of scattered light are contained in the cone defined by

$\theta_o = -\theta_i$.

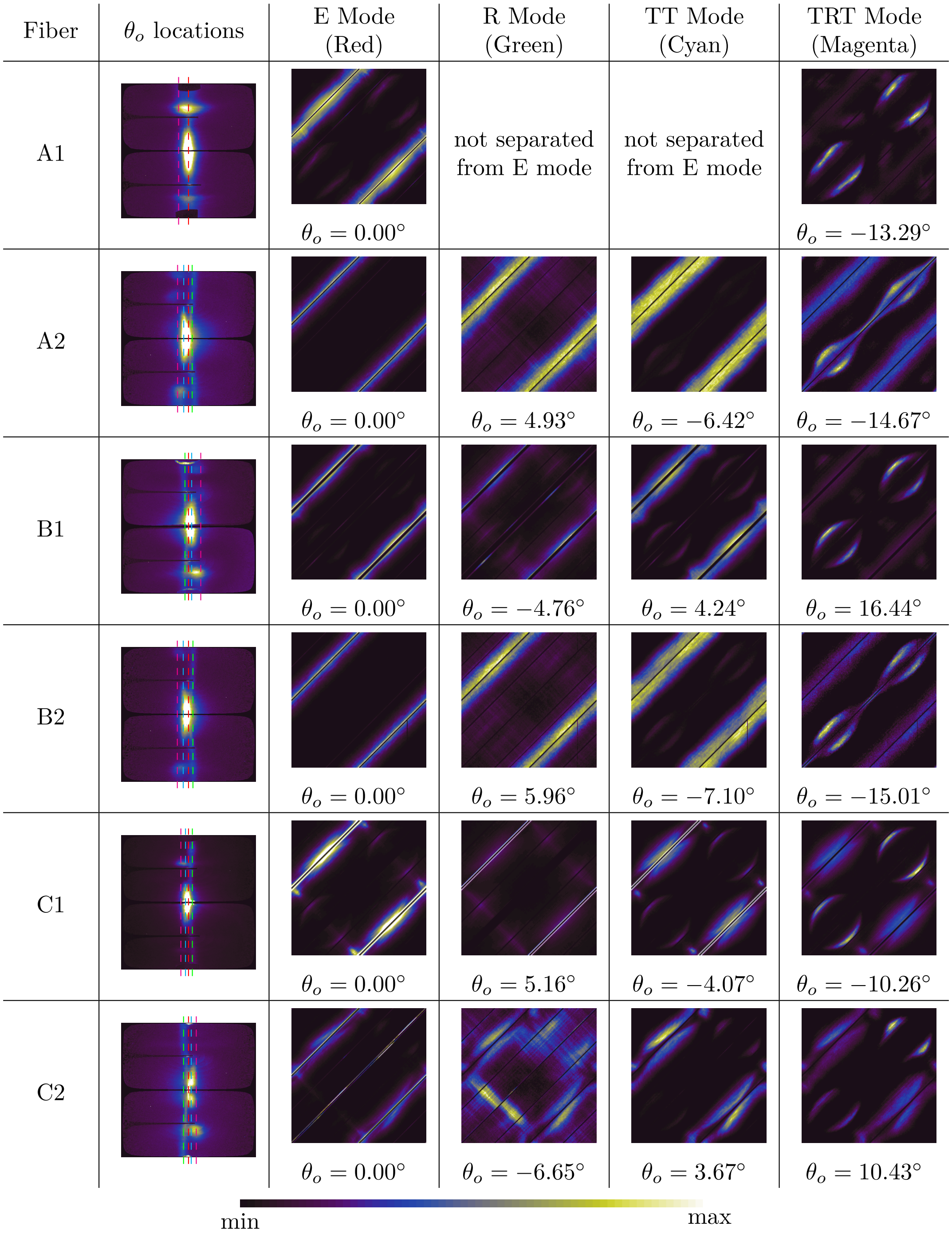
Evolution with respect to θ_i

With more detailed measurements than previously available, we can more completely characterize how the features change with respect to θ_i. For brevity, we discuss the evolution in detail in the supplementary material of [38]. The general trend is that the R mode and the E mode are bright near grazing angles but dim where θ_i is low in absolute value. In contrast, other modes behave in the exact opposite way.

4.7.3 (ϕ_i, ϕ_o)-Measurements

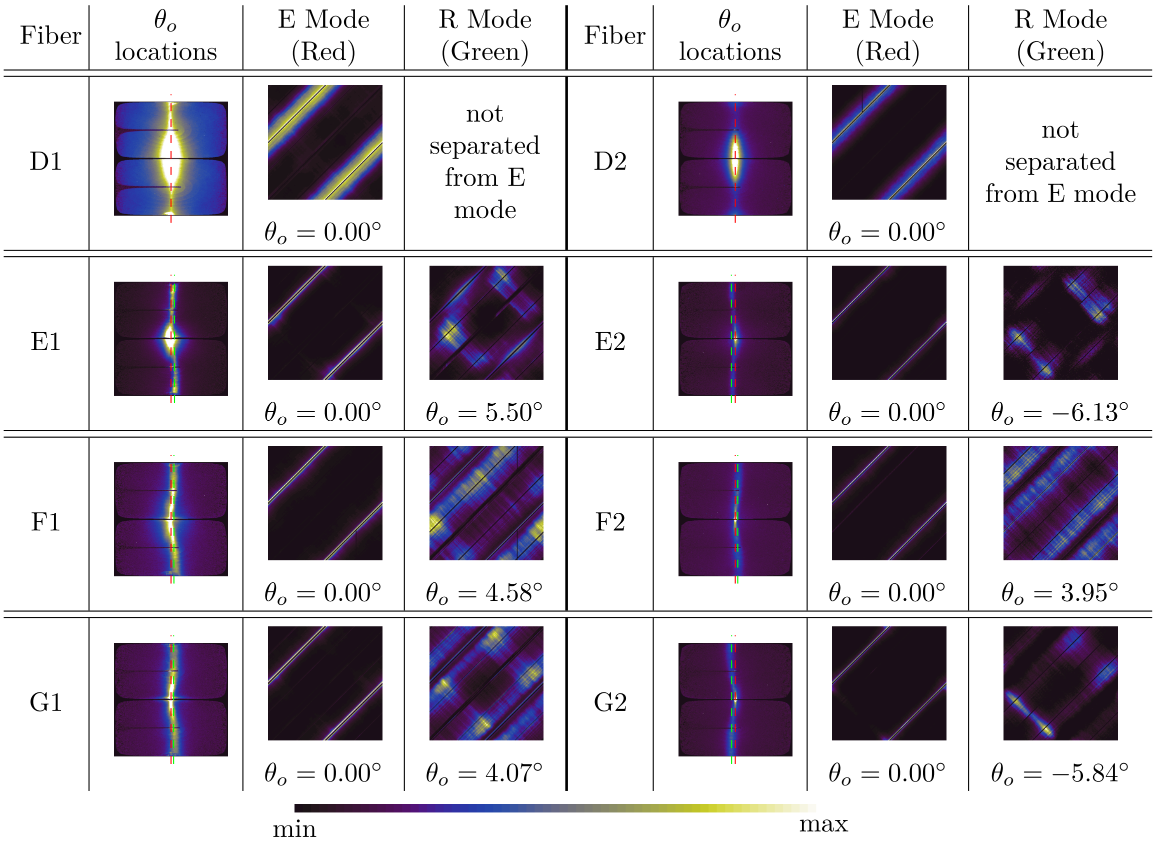
To get a more complete picture of the azimuthal scattering behavior, we created (ϕ_i, ϕ_o)-plots like the ones in Section 4.3.2. To do so, we set the rotating platform to a fixed position to hold θ_i constant and then rotate the fiber to achieve 180 equally spaced ϕ_i values from 0° to 360°. For each ϕ_i value, we captured the two photographs to cover the complete sphere and warped the photos into spherical coordinates as explained previously. To counteract the contribution of interreflection, we subtracted from each pixel 0.5% of the maximum intensity of all images taken with the same θ_i. From each of these 180 processed images, we selected a fixed column, corresponding to a particular value of θ_o, and arranged the columns to form a (ϕ_i, ϕ_o)-plot.

When conducting the measurements, we did not have means to determine the absolute ϕ_i angle. As such, the generated (ϕ_i, ϕ_o)-plot is a shifted version of the plots in Section 4.3. To make comparison with scattering models easier, we selected



None of the plots are to-scale with one another. For the first three modes of Fiber C1, due to the large spill on the far rim of the bowl, we clamp the maximum pixel value to about 50% of the real maximum pixel value to make it possible to observe the features more clearly. For the E mode of Fiber C2, we excluded the bright narrow band around the backward scattering direction when computing the maximum pixel value because it was also caused by spilled light. The complete set of plots with $\theta_i = -30°$ and $-60°$ is given in the supplementary material of [38].

Figure 4.13: (ϕ_i, ϕ_o)-plots of light-colored human hair fibers (Sample A to C) generated at $\theta_i = 0°$.



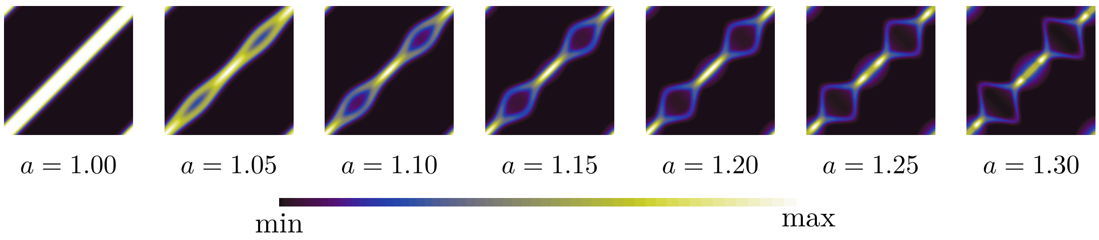
Fiber	θ_o locations	E Mode (Red)	R Mode (Green)	Fiber	θ_o locations	E Mode (Red)	R Mode (Green)
D1		$\theta_o = 0.00°$	not separated from E mode	D2		$\theta_o = 0.00°$	not separated from E mode
E1		$\theta_o = 0.00°$	$\theta_o = 5.50°$	E2		$\theta_o = 0.00°$	$\theta_o = -6.13°$
F1		$\theta_o = 0.00°$	$\theta_o = 4.58°$	F2		$\theta_o = 0.00°$	$\theta_o = 3.95°$
G1		$\theta_o = 0.00°$	$\theta_o = 4.07°$	G2		$\theta_o = 0.00°$	$\theta_o = -5.84°$

All (ϕ_i, ϕ_o)-plots were generated with $\theta_i = 0°$. None of the plots are to-scale with one another. Only the E and R modes are shown because other modes are either (1) not separated from the two in case of the wig fiber or (2) not visible at all in case of dark hair fibers. The complete set of plots with $\theta_i = -30°$ and $-60°$ is given in the supplementary material of [38].

Figure 4.14: (ϕ_i, ϕ_o)-plots of fibers from the artificial wig (Sample D) and dark-colored human hair (Sample E to G).

this unknown ϕ_i offset to align the observed features across the measurements and with the model in Section 4.4. Then, for convenience, we consider the left side of the generated plots to correspond to $\phi_i = 0°$.

For each fiber, we created three sets of (ϕ_i, ϕ_o)-plots corresponding to $\theta_i = 0°$, $-30°$, and $-60°$ for each fiber to see the effect of longitudinal angle on the ASF. We chose negative angles instead of positive ones because the area with most of the scattered energy (around $\theta_o = 30°$ and $60°$, respectively) does not contain one



The (ϕ_i, ϕ_o)-plots were generated using the Marschner model [49] with parameters $\eta = 1.55$, $\theta_d = 0$, $k_G = 1$, $w_c = 10^\circ$, $\Delta\eta' = 0.2$, and $\Delta h_m = 0.5$. The plots are not to scale with one another. The original paper only recommends using the model for only $0.85 \leq a \leq 1/0.85 = 1.176$, but we show plots with a outside the range as well.

The Marschner model with low aspect ratio approximates the TRT mode of Fiber A2 and B2 well though the variations in brightness are incorrect. At higher aspect ratios, it cannot produce the bright, separated eyelids seen in real fibers.

Figure 4.15: The (ϕ_i, ϕ_o)-plots of the TRT mode of the Marschner model for a number of aspect ratios.

of the light blockers.

We choose four different values of θ_o so that they align with the features we believe to be the E, R, TT, and TRT modes. We identified the R, TT, and TRT modes using the heuristics discussed at the end of Section 4.3. For the E mode, we use $\theta_o = -\theta_i$. As we did not generate a plot when the corresponding mode is not present or when it is not well separated from other brighter modes, this process resulted in at most 12 plots being generated for each fiber. The plots are available in Figure 4.14, Figure 4.13, and the supplementary material of [38].

The above measurements, which are intended to explore the ASF, actually do not measure the ASF directly. According to the separable model discussed in Section 4.4, these plots should show $M(\theta_i, \theta_o)N_p(\theta_d, \cdot, \cdot)$, which differs from $N_p(\theta_d, \cdot, \cdot)$ only by a constant, with additive contamination from other modes. However, in reality, the shifts and widths of the lobes modeled by M also change as a function of ϕ_i: the non-constant shifts cause some parts of the ASF to appear dimmer or even not to be observable in the slices we took, and the changing width produces a change in intensity. These effects are especially significant in the R mode, which, as we previously observed, curves quite noticeably in the (θ_i, ϕ_d)-plots.

It is important to keep these considerations in mind when comparing the measurements to the theoretical N function: qualitative agreement in location and shape of features can be expected, but intensities are not expected to match, and some features can be missing from the measured ASFs.

We now make observations of the scattering modes. While we list the E mode first in the figures, we will describe it last because it is usually contaminated by other modes.

R mode. All the fibers except those from Samples D and F exhibit two bands of brighter reflection that are predicted by the elliptical model but not previous circular models.

TT mode. Fibers A1, B1, C1, and C2 exhibit both the two slanted bars (⫽) and the small elongated blobs (⁘). Note that previous circular models predict the slanted bar but not the small blobs, so the elliptical model is more accurate. Fiber A2 and B2 exhibit only the strips, so their cross sections might be close to circular. As with the R mode, contamination from the TRT mode is sometimes visible in these plots.

TRT mode. The two eyes (ₒ°) predicted by the elliptical model can be observed. In fibers where the eyes open narrowly (A2 and B2), there exists a band along the $\phi_i = \phi_o$ line joining the two eyes. In other fibers, the eyes are completely separated.

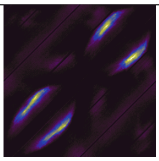
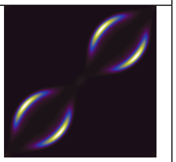
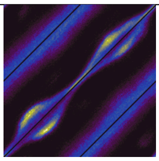
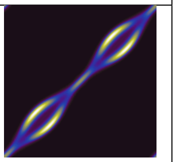
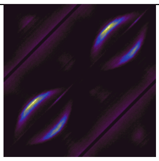
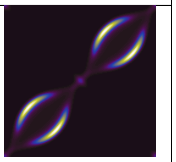
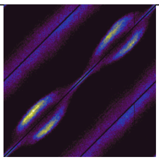
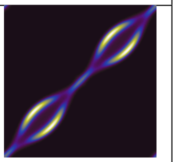
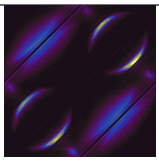
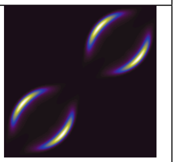
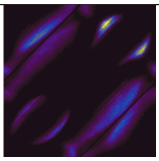
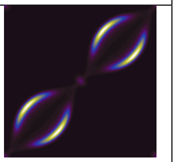
On the other hand, circular scattering models predict that the TRT mode appears as a narrow band of uniform width around the $\phi_i = \phi_o$ line. The Marschner model, with its approximation of elliptical fibers, can predict the TRT modes of Fiber A2 and B2 reasonably well (though it puts too much energy at retroreflection and not enough in the "eyelids"), but not those of other fibers. (See Figure 4.15.)

E mode. The E mode consists of narrow and bright bands around the forward scattering lines, which get broader with increasing θ_i. The E mode plots also show features resembling the TT mode, including slanted bars (particularly fibers C1 and C2) and perpendicular blobs (especially for $\theta_i = -60°$ in more elliptical fibers; see supplementary material of [38]), which we interpret as contamination from the TT mode.

Conclusions. From these observations, it is clear that real hair fibers exhibit features that are better explained by elliptical cross sections. These features include (1) the brighter bands in the R mode, (2) the small blobs in the TT mode, and (3) the eyes in the TRT mode.

4.8 Model Evaluation

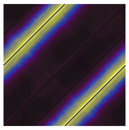
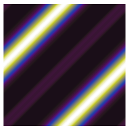
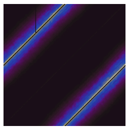
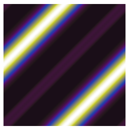
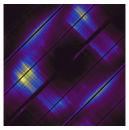
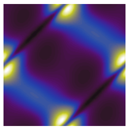
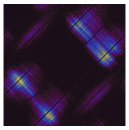
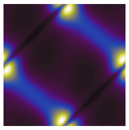
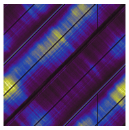
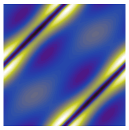
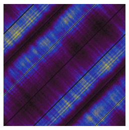
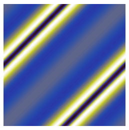
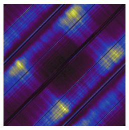
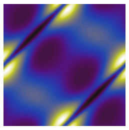
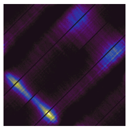
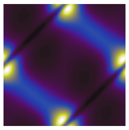
In this section, we further evaluate the elliptical scattering model by comparing the model (ϕ_i, ϕ_o)-plots against the measured data. We also present renderings produced by the model so that its visual impact can be assessed.

	a		(ϕ_i, ϕ_o)-plots of TRT mode	
Fiber	Measured	Fitted	Measured	Fitted
A1	1.136 to 1.496	1.51		
A2	1.256 to 1.429	1.13		
B1	1.446 to 1.501	1.35		
B2	1.077 to 1.552	1.17		
C1	1.603 to 1.717	1.65		
C2	1.548 to 1.762	1.42		

Here, we only show the (ϕ_i, ϕ_o)-plots of the TRT mode at $\theta_i = 0°$. Plots of other modes at other θ_i values are shown in the supplementary material of [38]. The plots' colors are not to scale with one another.

The generated plots match well to the measured plots, and the fitted aspect ratios are close to the measured ones.

Table 4.2: Results of fitting fiber aspect ratios for Fiber A to C to match the captured (ϕ_i, ϕ_o)-plots of the TRT mode.

	a		(ϕ_i, ϕ_o)-plots	
Fiber	Measured	Fitted	Measured	Fitted
D1	1.127 to 1.385	1.00	R+TT+TRT mode 	
D2	1.147 to 1.310	1.00	R+TT+TRT mode 	
E1	1.834 to 1.933	1.56	R mode 	
E2	1.561 to 1.789	1.99	R mode 	
F1	1.308 to 1.361	1.07	R mode 	
F2	1.035 to 1.183	1.00	R mode 	
G1	1.310 to 1.482	1.26	R mode 	
G2	1.503 to 1.839	1.81	R mode 	

For fibers from Sample D, we show the sum of the first three modes because all the modes are present in all the measured (ϕ_i, ϕ_o)-plots. For other dark-colored fibers, we only show the R mode as it is the only visible mode that the model covers. All plots are for $\theta_i = 0°$. Plots of other modes at other θ_i angles are given in the supplementary material of [38]. The plots' colors are not to-scale with one another.

Table 4.3: Results of fitting fiber aspect ratios for Fiber D to G (those that do not show clear TRT modes) to match the captured (ϕ_i, ϕ_o)-plots.

4.8.1 Comparison with Measured Data

To validate our model's ability to predict scattering behaviors, we fit the model to the measured (ϕ_i, ϕ_o)-plots and then compared the plots generated from the fitted parameters with the original data. We:

- used the θ_i values from the measurements,
- fixed σ to be 0 (perfectly transparent fibers),
- fixed η to be 1.55, and
- fixed γ_p to be 5° for all p.

This leaves the aspect ratio as the only parameter to be determined.

For each fiber, we pick a mode that is visible and varies the most as a changes:[7]

- the TRT mode for light-colored human hair fibers (Sample A, B, and C),
- the TT mode for artificial fibers (Sample D), and
- the R mode for dark-colored fibers (Sample E, F, and G).

Selecting a mode limits us to work with three measured plots: one for each of three values of θ_i. We then find a for which the three corresponding (ϕ_i, ϕ_o)-plots are the most similar to the three measured plots. To do so, we iterate through values of a ranging from 1.00 to 1.99 by increments of 0.01. For each a, we generate three plots with θ_i being 0°, $-30°$, and $-60°$. Next, all plots, including the measured plots,

[7] It is possible to fit the aspect ratio using a (ϕ_i, ϕ_o)-plot that contains more than one mode. However, the problem to solve becomes more complicated because the contribution of each mode is scaled by an unknown factor, which has to be optimized for. We chose to involve only one mode because it yields a much simpler optimization problem.

are normalized so that the pixel values add up to 1. (That is, they are regarded as probability distributions over the (ϕ_i, ϕ_o)-plane.) The score associated with this a value is the sum of the L_1-distances between the three pairs of generated and measured plots. The a value with the lowest score is regarded as the best fit for the fiber. The plots of the mode being fitted at $\theta_i = 0°$, generated from the best aspect ratios, are shown in Table 4.2 and Table 4.3. The supplementary material of [38] contains plots of other modes at other values of θ_i.

For light-colored human hair fibers, the model can be fitted well to the TRT mode. We invite the reader to consult the supplementary material of [38] to see that the fitted aspect ratios yield reasonable plots for the TT mode too. Moreover, the fitted aspect ratios either fall within or close to the range of measured aspect ratios. The successes of matching to the TRT mode tells us that, while fibers are not perfectly elliptical, their TRT modes usually behave like those of elliptical fibers.

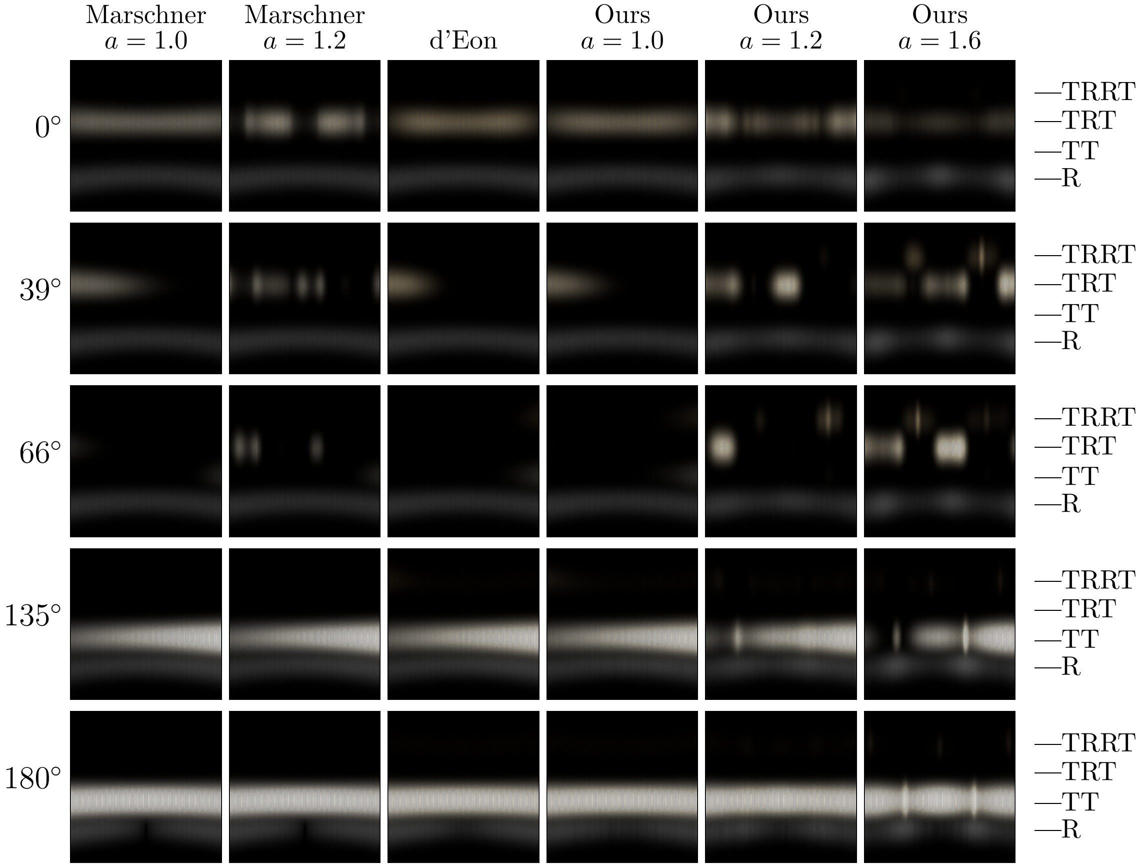
As fibers from Sample D do not separate the scattering modes in the θ_o direction, we could only match the parameter against the TT mode, which is the brightest and completely obscures other modes. Because the TT modes of fibers from Sample D behave much like a perfectly circular fiber, the matching process results in $a = 1.00$ being the best fit aspect ratio. However, SEM micrographs indicated that the fibers are not perfectly circular, and the (ϕ_i, ϕ_o)-plots generated using the average measured aspect ratio would display perpendicular blobs, which are not present in the measured plots. This discrepancy might be caused by the artificial fiber's index of refraction being lower than that of human hair fibers.

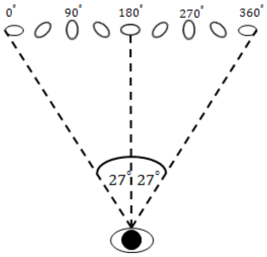
For dark-colored fibers, we could only match against the R mode. While our model can capture the two bright bands and can somewhat match the sizes of the bands, the best-fit aspect ratios can be far from the measured values, especially for Fibers E1, E2, and F2. Moreover, the generated plots generally do not look like the measured data. The two bands are not symmetric to each other as predicted, indicating that the R mode is sensitive to the asymmetry in the shape of the cross sections. We also observe dark bands around the forward scattering lines in the measured plots, but this is caused by the R mode curving away from the fixed longitudinal angle we used to create the (ϕ_i, ϕ_o)-plot, a phenomenon not accounted for by our model, and so doesn't indicate a disagreement in the ASF.

In conclusion, our elliptical ASF model can predict the azimuthal behavior in the TT and TRT modes well and can capture some important features of the R mode, indicating that changing from a circular to an elliptical model for the cross section can produce models much more faithful to real fibers. However, some important effects seen in the measurements, including curving of the R mode in θ_o direction and the existence of the E mode, require additional improvements in scattering models beyond upgrading the ASF to account for elliptical cross-sections.

4.8.2 Rendering Results

To compare azimuthal scattering behavior of our models to older models, we show single scattering behavior of a planar array of vertical hair strands whose cross sections are rotated by 0° to 360° from left to right in Figure 4.16. We also provide, in Figure 4.17, renderings with full multiple scattering of hair geometries illuminated by a constant environment light source and an area light source positioned at various horizontal angles around them. The renderings were done at roughly half a megapixel resolution with 512 samples per pixel. An image typically took 15





We show renderings of an array of vertical hair fibers whose cross sections are rotated by 0° to 360° from left to right. We compare (1) our model with aspect ratio 1.0, 1.2, and 1.6, (2) the circular model from [17], and (3) the classic model from Marschner et al. [49] with aspect ratio 1.0 and 1.2. (We do not show the Marschner model with aspect ratio parameter of 1.6 because the model was claimed to work with aspect ratio up to about $1/0.85 \approx 1.18$.)

The scenes all have one directional light source with a horizontal direction. The angle that this direction makes with the line from the eye to the center of the image plane is indicated in the leftmost column of the table. In this way, 0° indicates that the light's direction is the same as the viewing direction, and 180° indicates that the light's direction is anti-parallel to the viewing direction. We chose the tilt angles of the 4 modes so that they separate into distinct horizontal bands whose altitudes are indicated by the labels to the right to the table. The renderings only take into account single scattering.

Figure 4.16: Comparison between renderings of different hair scattering models in a simple scene with parallel hair strands.

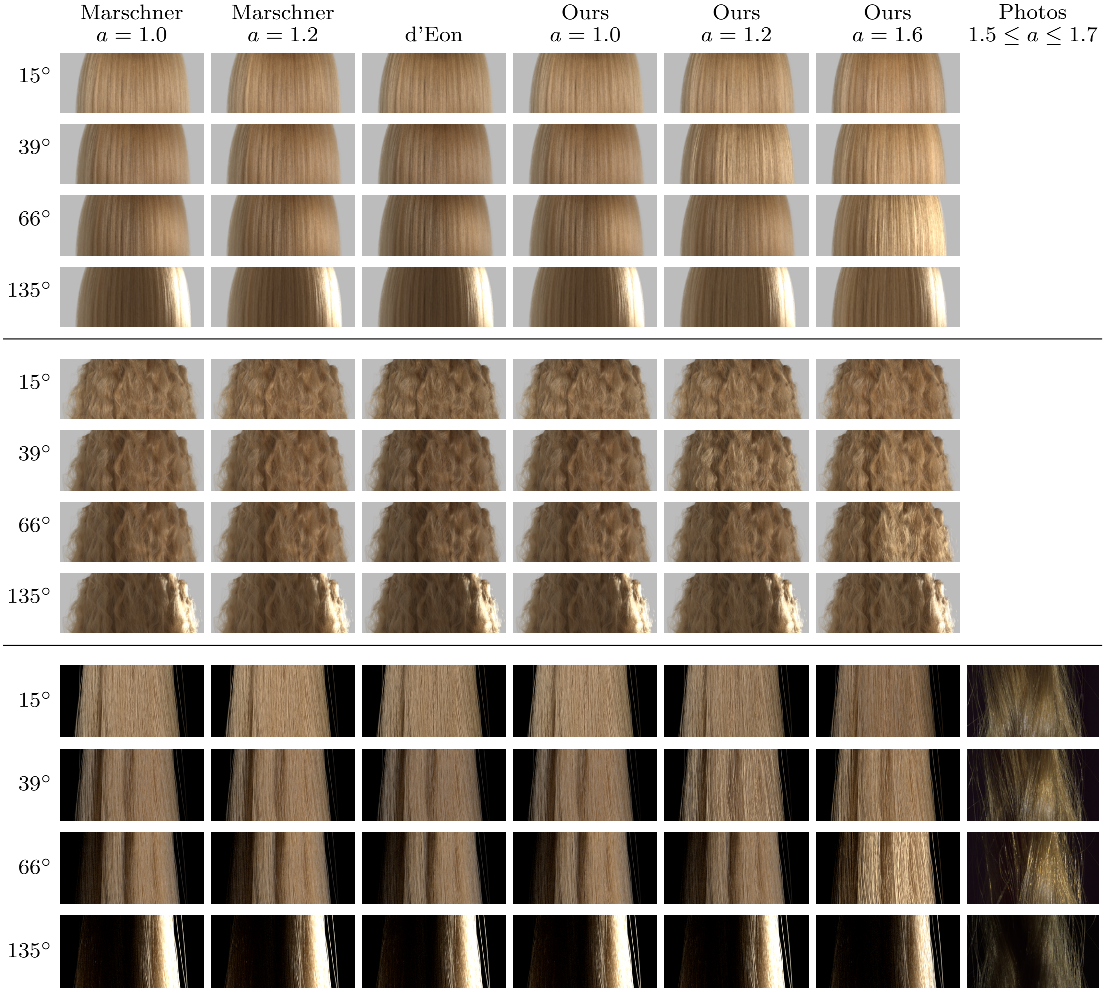
to 30 minutes on a 192-core compute cluster, depending on hair geometry. Videos comparing different models being lit by revolving an area source are available in the supplementary material of [38].[8]

From the single-fiber measurements and model predictions, we know that the most important effects of eccentricity on the azimuthal distribution are (1) that the TRT highlight occurs at a larger difference in ϕ, (2) that the TRT highlight occurs over a wider range of angles, and (3) that the TRT and TT highlights vary substantially in brightness with the orientation of the fiber.

The most obvious result of these effects is that the TRT component, which is a relatively subtle effect in circular hair, produces a bright, glittery highlight that occurs at a large scattering angle. In Figure 4.16, this effect manifests as bright blobs produced by our elliptical models when the light angles are 39° and 66° away from the camera. Moreover, in Figure 4.17, elliptic fibers with aspect ratio 1.2 become much brighter at 39°, and those with aspect ratio 1.6 at 66°. These bright, colorful glints create a strong texture in the image, with an almost metallic appearance, that is not seen in circular hair. Previous models for hair scattering have never reproduced glints with any serious attempt at accuracy. (While the Marschner model does produce some brightness variation in the TT mode, the glints it produces are generally too dim to notice in real scenes.) The ability to render them correctly will enable more natural looking and realistic results.

This glint angle for aspect ratio 1.6 is confirmed by a simple experiment of illuminating the swatch that was the source of two of our hair samples (Sample 3) from the same angles (bottom row in Figure 4.17). The photographs illustrate the occurrence of glints at angles between 60 and 80 degrees. The glints in photographs

[8] The video frames are rendered with 128 samples per pixel instead of 512.



We show renderings with full multiple scattering of three pieces of hair geometry lit by an area source rotating around them. The leftmost column contains the angle, in the horizontal plane, that the light makes with the camera's view direction. Observe that our model with $a = 1.2$ becomes brighter than other models at 39°, and so does our model with $a = 1.6$ at 66°. The final column shows photographs of Sample C for reference; note glints appearing at 66°. The supplementary material of [38] contains the parameters for the models and the uncropped renderings. (While we attempted to find absorption parameters that would match the overall color of Sample C, the real hair is considerably more absorbing.)

Figure 4.17: Comparison between full globally illuminated renderings of different light scattering models in scenes containing full heads of hair.

do not increase the overall brightness as much as in renderings, and they occur over a wider range of angles. They show up in short segments in the photographs while whole hair strands seem to light up in the renderings. We surmise that this effect is caused by real hair having natural variation that is lacking in our models: both eccentricity and orientation likely vary from fiber to fiber and along fibers. We randomized the orientation of each hair strand in our models at the root, but eccentricity is perfectly constant and the fibers do not twist.

A second difference can be seen in the TT component. One can observe that, in Figure 4.16, some fibers with our elliptical models become much brighter than surrounding fibers when the light source is at 135° from the camera. Such an effect is also present at 180° in the model with aspect ratio 1.6. It is the result of the TT component's focusing light strongly for a narrow range of orientations (that is, when one of the perpendicular blobs appears), leading to another, subtler glint effect. These TT glints have not been reported in previous measurements or modeled by any previous scattering models. However, they are hard to observe in less contrived rendering situations as we see minimal differences between forward scattering highlights between the models in Figure 4.17.

When the hair appearance is dominated by the R component and by multiple scattering, the effects of eccentricity are not as dramatic as the glints. They amount to moderate change in the overall hair color because eccentricity changes the distribution of path lengths through the fibers, resulting in different colors of modes other than the R mode. This suggests that approximate methods for multiple scattering that have been developed for the circular case [116, 80] can likely be used with elliptical fibers.

4.9 Conclusions

We have presented presented a study of the scattering behavior of hair fibers, focused on azimuthal effects. We designed a new measurement device that enabled us to observe a more complete picture of the light scattered by a fiber, which showed that real hair fibers behave differently from what was predicted by previous scattering models. Features not previously predicted include the two bands in the R mode, the perpendicular blobs in the TT mode, the eyes in the TRT mode, and the E mode itself. Using geometric optics to model light scattered from ideal rough elliptical fibers, we learned that the features in the R, TT, and TRT modes are products of the geometry of the cross section. However, the geometric optics analysis does not predict the E mode, and we conjecture that this mode is a wave optics phenomenon. Rendered images made using a model built on the new cross-section analysis better reproduce the glittery appearance that can be observed in real hair fibers.

4.9.1 Limitations and Future Work

While we did not model the E mode, light scattering behavior of small circular cylinders has been studied by physicists [90], and their work might inform us on how to model the E mode of elliptical fibers.

We are primarily concerned with azimuthal effects, but the measured data also motivate improvements to the longitudinal component of the model. In particular, the curving of the R mode needs to be addressed, and many components become more longitudinally blurred than the simple constant-β model predicts. The recent work of d'Eon et al. [18] on non-separable BCSDF can potentially handle these phenomena, but it only deals with circular cross sections.

Our implementation currently requires big tables to be built for each combination of hair parameters. This makes it inconvenient to render a head of hair where fibers have different parameters or where parameters vary along each fiber. A scheme to compactly compress the ASFs or efficient algorithms to analytically evaluate and sample them would enable rendering of more complex and realistic hair.

Lastly, our measurement techniques can be used with any type of fiber. As a result, they can be used to study fibers such as cloth fibers and animal fur.

CHAPTER 5

MODELING AND MATCHING APPEARANCE OF FABRICS

> ใส่ตามสบาย ม่อฮ่อมใส่แล้วสมชายชาญ
> ใส่ไปทำงานหรือเตรียมใส่ไปแอ่วสาว
> ใส่ไปไร่นาป่าเขา ใส่ไปกินเหล้ากินข้าว
> ไพร่ผู้ดี เศรษฐี ขี้ข้า หรือเจ้า
> ใส่ม่อฮ่อมสีความเป็นเงา
> วัฒนธรรมหมู่เฮาจาวเหนือ
>
> จรัล มโนเพ็ชร, *ม่อฮ่อม*

In the last chapter, we saw how we can design a light scattering model for hair fibers from first principles so that it captures some characteristic behavior of their cross sections. However, the light scattering model's description alone is never enough to produce renderings. We must also specify geometry and model parameters. This chapter presents a framework for obtaining these components when creating micro-appearance models for cloth.

The material in this chapter originally appears in the paper "Matching Real Fabrics with Micro-Appearance Models," which has been published in *ACM Transactions on Graphics* in 2015 [39]. The work is joint with Daniel Schroeder, Shuang Zhao, Steve Marschner, and Kavita Bala.

5.1 Introduction

Appearance models that can reproduce the rich appearance of fabric are important in a wide range of applications including textile design, product visualization, retail, and entertainment. Yet, photorealistic rendering of fabrics remains very challenging. Recent research that models fabrics at the scale of fibers [109, 76] has produced the most realistic renderings to date. By directly modeling the geometric arrangement of fibers, these methods can reproduce distinctive specular highlights caused by woven structures, subtle diffuse effects of multiple scattering, and details like fuzz and flyaway fibers.

These appearance models are instances of an approach we call *micro-appearance models*, which combines an explicit model of a material's microgeometry with a simple light scattering model. Previous work has shown that these components together can reproduce the subtle and complex light-scattering behaviors seen at larger scales. The approach is also general enough to encompass all types of textiles, including traditional weaves and knits as well as non-woven fabrics.

Using micro-appearance models entails answering a number of questions, which previous work still left open:

- **How to represent microgeometry?** Microgeometry can either be represented by volumes (e.g., [109]), or collections of individual fibers (e.g., [76]). It is unclear which approach is better.
- **How to model light scattering?** Zhao et al. [109] proposed a simple microflake phase function, but it does not capture scattering behavior in grazing configurations correctly. Schröder et al. [76] used a scattering function derived from that of hair fibers, which may handle grazing behaviors better. However, its effectiveness in this regard has never been assessed against measurements.
- **How to compute model parameters?** Zhao et al. [109] employs a simple binary search to fit their model, but it cannot be generalized to more com-

plicated ones. Most other previous work specifies parameters manually with the exception of Schröder et al. [76] which automatically derives the diffuse color from cloth photographs. To our knowledge, there has been no general framework for fitting model parameters.

This chapter attempts to answer the above three questions. To do so, we introduce a number of innovations. To identify the best microgeometry model, we develop an algorithm that converts a micro CT scan of cloth fabric to an explicit mesh of the fibers that compose the fabric so that we can compare the effectiveness of the two approaches. To identify the best light scattering model, we use the methodology for developing hair scattering function [49] to develop an improved scattering model for textile fibers that takes into account reflection from and refraction through fiber surfaces. To compute the parameters of the model, we develop an appearance matching framework which takes into account multiple fabric observations under different lighting conditions. It uses stochastic gradient descent to optimize the parameters, so it is general enough to fit any parameter with respect to which the partial derivative of a single path tracing sample can be computed.

These innovations enabled us to systematically evaluate micro-appearance modeling approaches against one another and against measurements. In general, we gather ground-truth data in the form of gonioreflectometric measurements of real cloth samples. The appearance matching framework uses some of the data to optimize for parameter values. Performance can then be assessed by comparing the rest of the ground-truth data to renderings yielded by the models and the fitted parameters. We believe that this evaluation procedure is essential to developing and effectively testing new models. Using the procedure, we provide an extensive evaluation of the effectiveness of our appearance matching pipeline and the relative performance of different microgeometry and light scattering models using six fabrics with very different characteristics.

The conclusions we draw from our investigation are:

1. A scattering model based on previous models for hair scattering works much better than the microflake model, especially at reproducing bright grazing highlights.
2. In choosing between fiber- and volume-based models for cloth microgeometry, there is no clear winner: both are capable of matching measurements when used with the right scattering model. However, we did find that smooth orientation fields are important to achieving good results with volume models.
3. Our system worked well for a number of fabrics, but we also experienced a number of difficult cases, which point out venues for improving both the system itself and the light scattering model.

Our work as a whole comprises a complete and practical appearance modeling system which we believe is an important step forward in achieving predictivity and photographic realism for textiles. It can generate fabric models that capture both far-field reflectance properties and near-field fine textures of different types of textiles.

The system implemented in this chapter, however, does not cover all aspects of fabric appearance. The experiments we shall present focus on light *reflected* from fabric and do not examine light *transmitted* through the fabric. Moreover, our system currently only handles fabrics with a single yarn type and color. While it can be readily extended to address these aspects, the extension is not in the scope of this dissertation and is left for future work.

5.2 Previous Work

5.2.1 Parameter Estimation

For the volume-based cloth model of Zhao et al. [109], the authors selected rendering parameters for the scanned cloth volumes by binary searching the values of single parameters in sequence to match statistics of a reference image of the cloth sample. By contrast, our method recovers all parameters of the fiber scattering model simultaneously, and compares the rendered cloth to several photographs under different lighting and viewing conditions. As in our approach, Gkioulekas et al. [23] recover the unknown parameters of their scattering model through stochastic gradient descent. Their method combines gradient descent and Monte Carlo rendering to recover linear combinations of predefined materials from a material dictionary for a wide range of translucent media. Recently, Schröder et al. [76] introduced an image-based technique to reverse engineer physical fabric samples. Their approach, however, focuses on recovering weave patterns and diffuse yarn colors and requires many other parameters needed for rendering to be specified manually. Sadeghi et al. [72] estimated parameters in two stages. They first determined their yarn scattering model's parameters to match to a dense sample of yarn BSDF measurements, and then they determined parameters of the yarn curves. Both stages, however, were carried out manually.

We note that there are two approaches to the number of training data used for parameter fitting. On one hand, works such as Zhao et al.'s and ours use a *sparse* sampling of appearance. Indeed, Zhao et al. uses one photograph whose pixel values are then averaged into 3 numbers. We use 16 (as will be discussed in Section 5.5.2) and the images are averaged in the same way. On the other hand, Sadeghi et al.'s and Gkioulekas et al.'s use a *dense* sampling.[1] One criticism to the sparse sampling approach is that the samples might not be representative enough to capture all the details of the fabric's scattering behaviors. For example, Sadeghi et al. documented that fabrics exhibit multiple highlights due to weave patterns, and not including such highlights may lead to incorrect parameter values, especially lobe widths.

It is instructive to compare our approach to that of Sadeghi et al. [72]. Appearance is produced by a combination of microgeometry and optical properties, and both methods seek to determine geometry and scattering properties separately. Sadeghi et al. consider the yarn as the basic unit, so they begin by measuring the optical properties of a yarn very accurately and then finish by adjusting the geometry to match the overall appearance. We consider the fiber as the basic unit and proceed in the opposite order. We establish geometry first—both the arrangement of fibers within yarns and also the geometry of yarns in the cloth—using micro CT scanning and then adjust the scattering properties to match appearance. In a sense, we rely on detailed microgeometry to give rise to complex structures in fabric's scattering behaviors such as multiple highlights and to compensate for approximated lobe widths values. Encouraged by the success of Zhao et al. and constrained by the need to render images during the fitting process, we have taken the approach of fitting the optical properties to the far-field BRDF. As discussed in Section 5.7, this approach produced excellent results for many fabrics, but there

[1] Gkioulekas et al. takes only 18 photographs, but each pixel of each photograph is considered a measurement, so the sampling is in fact very dense.

are others where additional training configurations may be helpful.

5.2.2 Derivative Estimation

Our appearance matching framework extends a path tracer to estimate derivatives of images with respect to parameters. We use the same mathematical formulation for derivatives of a path-traced pixel as that described in the work of Pfeiffer and Marroquim [62], which we were unfortunately not aware of when our paper [39] was published. There is a slight difference between their work and ours in the way the derivatives are computed, but the main differences are in the optimization algorithm used to compute the optimal parameters: Pfeiffer and Marroquim use the Gauss–Newton–Krylov method, but we use the simpler stochastic gradient descent. Since stochastic gradient descent is designed to tolerate noisy gradients, it can be said that our algorithm is more robust to noise in theory.

Hašan and Ramamoorthi [25] estimate derivatives with respect to the albedos of volumes. Pfeiffer and Maroquim's formulation reduces to the same form when applied to estimate derivatives with respect to albedo, but supports other parameters as well. The operator-theoretic method of Gkioulekas et al. [23] considers the full set of rendering parameters of an isotropic volume, but only obtains derivatives with respect to the coefficients of a convex linear combination of predefined materials. By comparison, the approach in Section 5.5 can individually estimate derivatives with respect to any rendering parameter that has a differentiable effect on the samples used to render the image.

5.2.3 Fiber Generation

We have reviewed a number of techniques for generating hair and textile fibers in Chapter 3. Since, the work in this chapter proposes an algorithm for generating fibers from volumetric data, we review previous works on this problem.

Constructing fiber geometry from volumetric data has been well studied in the neuroscience community for the purpose of understanding the brain's networking. Approaches include tracing particles through a direction field [4], growing a level set [60], and simulating water diffusion in the volume [37].

While Shinohara et al. used micro CT images to extract yarn positions in fabric [79], we are not aware of any previous works, especially in the graphics community, that generate cloth fibers from micro CT images.

5.3 Overview

Our appearance modeling pipeline is outlined in Figure 5.1. Our appearance matching process takes as input a set of photographs of a fabric under different lighting and viewing configurations, together with the corresponding scene geometry, and finds fits for the parameters of our new light scattering model. The light scattering model is described in Section 5.4. The appearance matching process is introduced in Section 5.5. The appearance matching process takes input as photographs and fiber microgeometry, the latter of which is constructed as described in Section 5.6.

We now describe each part of the pipeline in more detail. First, in Section 5.4,

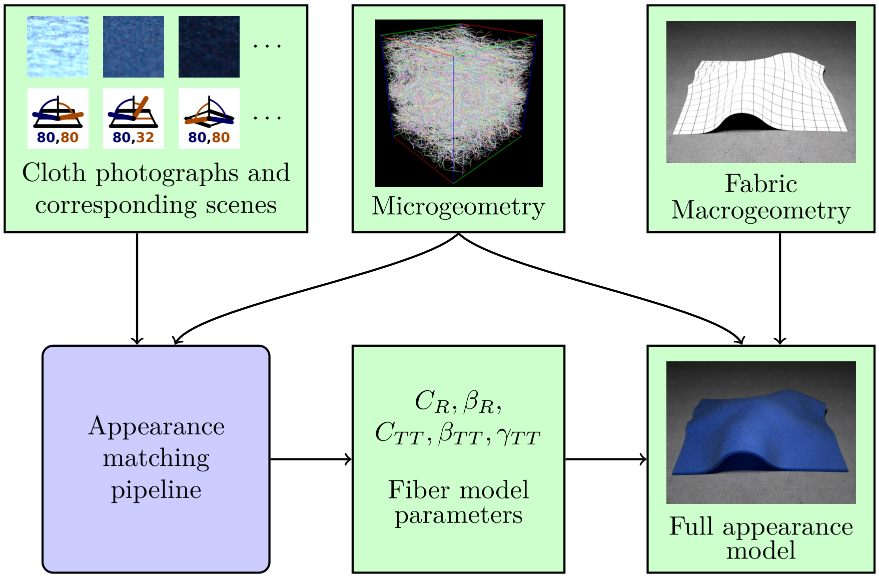


Figure 5.1: Appearance modeling pipeline.

we introduce a new light scattering model for textile fibers. The scattering model has two terms. The first term models light reflected directly off fiber surfaces, and contains a Fresnel term that makes the reflection brighter at grazing angles to address the inaccuracy observed in Zhao et al.'s model. The second term models light transmitted forward through the fibers, accounting for the fact that textile fibers are generally translucent. While the scattering model is conceptually simple, it cannot be fitted using the simple iterative binary search method described in Zhao et al. [109]. Therefore, we introduce a new appearance matching process in Section 5.5.

The appearance matching is done using gradient descent optimization to find values for the parameters of the scattering model that achieve the best match between the photographs and physically-based renderings of the cloth microgeometry model. Multiple scattering contributes significantly to the renderings (the majority of the reflected light is due to multiple scattering in most cases). Thus, the optimization has to account for it. To do so, we extend our renderer, a Monte

Carlo path tracer, to compute derivatives of the output image with respect to the parameters of the scattering model. The derivatives are computed as unbiased estimates of the true derivatives and are used in a *stochastic gradient descent* optimization method, which converges to a minimum despite the uncertainty in the individual estimates of pixel values and their derivatives (under the condition that the objective function is convex). Our process is agnostic both to the microgeometry model and to the scattering model, allowing us to directly compare the abilities of different models to recover cloth appearance.

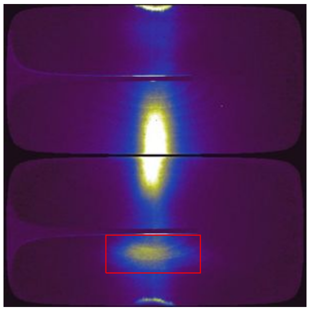
A goal of this chapter is to study the effectiveness of two representations of fabric microgeometry. While previous work has been able to create micron-scale volumetric representation from micro CT images of real fabrics, no work has addressed the creation of fiber mesh representations from such data. To enable direct comparison of the two approaches, we develop an algorithm to construct fiber meshes from micro CT images in Section 5.6. The method is based on identifying fiber centers in slices of micro CT volumes and connecting them.

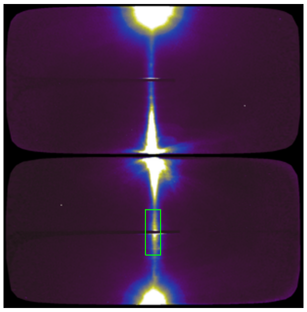
Finally, in Section 5.7 we present the results of our investigation into fabric appearance models, carried out using the tools developed in the earlier sections, and draw conclusions about which methods should be used.

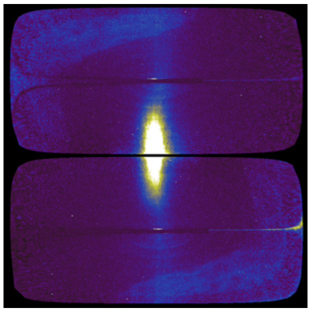
5.4 Fiber Scattering Model

In this section, we describe a fiber scattering model, developed with the goal of addressing the shortcomings of the microflake scattering function used by Zhao et al. [109]. Our model builds upon the considerable research in scattering models for rendering hair [49, 115, 17]. Like hair, textile fibers are long cylindrical structures made of dielectric material, so they can be expected to exhibit similar specular reflection geometry. However, because textile fibers are smaller and less visible individually, as well as more irregular in cross section, we use a simpler model than the full hair model.

Like many other light scattering model for fibers, our model is a multi-modal, factored BCSDF whose mathematical foundation was laid out in Section 2.4. In agreement with previous BSDF models for cloth [29, 72], informal measurements of individual fibers suggest an azimuthally uniform R mode and a TT mode with a single forward scattering lobe. Moreover, the more detailed TRT mode that appears in hair fibers does not appear to be important for textile fibers. Below, we provide pictures taken with the bowl device in Chapter 4, warped into the (θ_o, ϕ_o)-plane, showing the scattered light from a Caucasian hair (used in the previous chapter), a fiber of the Silk sample used in this chapter, and a fiber from a wool fabric.


Hair


Silk


Wool

We emphasize the TRT mode, present in scattered light from the hair, with a red rectangle. We can see that such a structure is not visible in the scattered light from the silk fiber and the wool fiber. We might say that the blob surrounded by the green rectangle is the TRT mode of the silk, but observe that it is very small when compared to the TRT blot of the hair. (The two big blobs on the top and

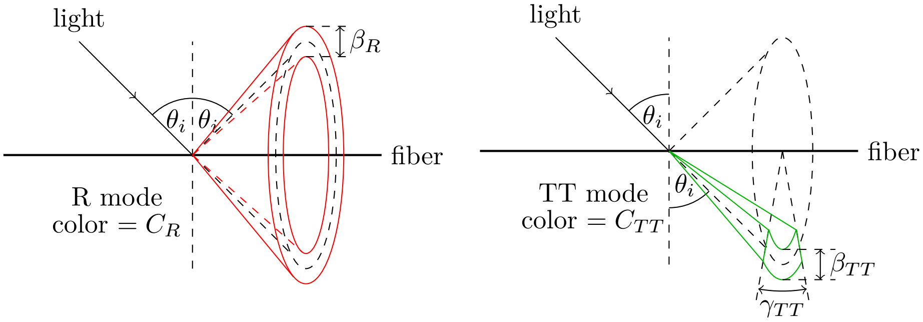


Figure 5.2: The two modes of our scattering function.

the bottom edges of the silk fiber's image are light spilt directly on the device from the light source. They are not part of the scattered light.)

The above casual observation motivated us to include only the first two modes in our model:

$$S(\omega_i, \omega_o) = \frac{M_R(\theta_i, \theta_o)}{2\pi} + M_{TT}(\theta_i, \theta_o) N_{TT}(\theta_i, \phi_i, \phi_o). \tag{5.1}$$

The two modes are depicted in Figure 5.2.

Our model has five parameters that determine the intensities and widths of the two modes:

- C_R: the color of the R mode
- C_{TT}: the color of the TT mode
- β_R: the longitudinal width of the R mode
- β_{TT}: the longitudinal width of the TT mode
- γ_{TT}: the azimuthal width of the TT mode

We make use of two differently normalized Gaussian-like functions. One is $\bar{g}$, a renormalized Gaussian in θ:

$$\bar{g}(\theta; \mu, \sigma) = \frac{g(\theta; \mu, \sigma)}{G(\mu, \sigma)} \tag{5.2}$$

where $g(\theta; \mu, \sigma)$ denotes the Gaussian distribution with mean μ and standard deviation σ, and $G(\mu, \sigma)$ is a normalization factor defined in Section 4.4.1.

The other is the von Mises distribution f (also used in [29]), which is the analog of the Gaussian distribution on the circle:

$$f(\phi; \mu, \sigma) = \frac{\exp(\sigma^{-2} \cos(\phi - \mu))}{2\pi I_0(\sigma^{-2})} \tag{5.3}$$

where $I_0(x)$ is the modified Bessel function of order 0.

5.4.1 The R Mode

The R mode accounts for light that reflects specularly from the surface of the fiber. The total amount scattered into this mode depends on θ due to Fresnel reflection; we model this dependence using a heuristic formula, since the surface is not planar and the actual fraction transmitted depends on the cross section and surface properties, neither of which we wish to model. We introduce a parameter C_R that specifies the reflectance at $\theta_i = 0$, then use Schlick's approximation to let reflectance increase to 1 as the incident direction becomes parallel to the fiber:

$$\mathcal{F}_R(\theta_i) = C_R + (1 - C_R)(1 - \cos\theta_i)^5. \tag{5.4}$$

In the absence of a particular cross section, we assume that the reflected light is distributed uniformly in ϕ, so that

$$N_R(\theta_i, \phi_i, \phi_o) = \frac{1}{2\pi}.$$

Moreover, it is scattered to a small range of θ that increases with surface roughness, which we model using the normalized lobe $\bar{g}$:

$$M_R(\theta_i, \theta_o) = \mathcal{F}_R(\theta_i)\bar{g}(\theta_o; -\theta_i, \beta_R). \tag{5.5}$$

Because the values of C_R may be different among the red, green, and blue channels, our model allows reflection from fiber surfaces to be colored. Although Fresnel's equations do not predict this, we deliberately make the reflection colored as we found empirically that this led to the model being better at reproducing the color of some fabrics.

5.4.2 The TT Mode

The TT mode represents light that transmits into the fiber and then out. It is responsible for the remaining fraction $1 - \mathcal{F}_R(\theta_i)$ of incoming light. It is colored (via the parameter C_{TT}) to account for light absorbed by colorants in the interior of the fiber.

As with the R mode, we model the longitudinal spread using a normalized Gaussian, but since transmitted light is generally focused forward, we model the dependence on ϕ using the von Mises distribution centered at $\phi_i + \pi$:

$$M_{TT}(\theta_i, \theta_o) = C_{TT}(1 - \mathcal{F}_R(\theta_i))\bar{g}(\theta_o; -\theta_i, \beta_{TT}) \tag{5.6}$$

$$N_{TT}(\theta_i, \phi_i, \phi_o) = f(\phi_o; \phi_i + \pi, \gamma_{TT}) \tag{5.7}$$

where γ_{TT} controls the azimuthal width of the forward scattering peak.

The two components of the model together define a simple but expressive model for scattering from fibers. It can model rougher fibers like cotton or wool compared to smoother fibers like nylon or silk by adjusting β_R and β_{TT}; it models the color of fibers primarily using C_{TT}, and the effects of different cross sections that produce more or less strongly forward-directed scattering are modeled by adjusting γ_{TT}.

5.4.3 Volumetric Appearance Model

The scattering model can be adapted into a volumetric appearance model compatible with the anisotropic RTE [32]. To do so, we need to specify a normalized phase function, an albedo, and a directionally varying coefficient of attenuation.

To specify a phase function, consider a single color channel. For each value of ω_o, we can construct a probability distribution over the sphere of incoming directions that is proportional to $S(\omega_i, \omega_o)$ according to (2.15) and use this distribution as the (direction-dependent) phase function. Our implementation uses the probability distribution, computed by tabulation of both ω_i and ω_o, for importance sampling of the fiber scattering model as the phase function. However, because C_R and C_{TT} can have different values in different channels, we construct a separate phase function for each channel. When rendering, we render three monochrome images for each channel before combining them to a single colored image.

We constrain the albedo $\alpha = \sigma_s/\sigma_t$ to be a constant throughout the fabric volume, and leave it as a parameter to be fit by our fitting process.

For the extinction coefficient, we choose a function such that σ_t is maximal when the light's direction is perpendicular to the local fiber direction and decreases smoothly to 0 when the light's direction is parallel to the local fiber direction. In particular, we choose:

$$\sigma_t(\omega) = \sigma_{t,\max}\sqrt{1 - (\omega \cdot \mathbf{d})^2} \tag{5.8}$$

where $\mathbf{d}$ is the local fiber direction, and $\sigma_{t,\max}$ is the maximum coefficient of extinction, which is a parameter to the model. In theory, $\sigma_{t,max}$ is a parameter our fitting process can fit. However, we set it to constant as will be discussed in the next section.

5.5 Appearance Matching

In this section we explain our appearance matching method. As outlined in Section 5.3, the method receives as input photographs of a material and corresponding scenes with the same geometric configuration of camera, cloth sample, and light source. Its goal is to find model parameters that result in renderings that match the photographs. The method quantifies the differences between photographs and renderings with an objective function, and minimizes its value through stochastic gradient descent on the model parameters.

To implement this method, we must obtain derivatives of the objective function with respect to the model parameters, which in turn requires differentiating the rendered images with respect to the parameters. We begin with an introduction to a derivative estimation method from stochastic simulation and its application to path tracing. We then discuss the choice of objective function and the gradient descent optimization.

5.5.1 Derivative Estimation

In the path integral formulation of physically based rendering (Section 2.3.1), the pixel intensity is proportional to the incoming radiance to a point on the camera's sensor. Consider a scattering model parameter ϑ. The incoming radiance depends on ϑ as follows:

$$L_i(\mathbf{x}_0, \omega_0; \vartheta) = \int_{\mathbb{P}} f_\vartheta(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}}.$$

where $\mathbf{x}_0$ is a point on the camera's sensor, ω_0 is a direction, $\mathbb{P}$ is the space of all paths starting with $(\mathbf{x}_0, \omega_0)$, and $f_\vartheta(\vec{\mathbf{x}})$ is the contribution of the path $\vec{\mathbf{x}}$, with the dependence on ϑ being made explicit.

Assuming that f_ϑ is *Lipschitz continuous*[2], it is possible to compute the derivative of $I(\vartheta)$ by passing the derivative operator through the integral:

$$\frac{\mathrm{d}L_i(\mathbf{x}_0, \omega_0; \vartheta)}{\mathrm{d}\vartheta} = \frac{\mathrm{d}}{\mathrm{d}\vartheta} \int_{\mathbb{P}} f_\vartheta(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}} = \int_{\mathbb{P}} \frac{\mathrm{d}f_\vartheta(\vec{\mathbf{x}})}{\mathrm{d}\vartheta}\ \mathrm{d}\vec{\mathbf{x}}.$$

This integral, then, can be estimated by Monte Carlo integration:

$$\frac{\mathrm{d}L_i(\mathbf{x}_0, \omega_0; \vartheta)}{\mathrm{d}\vartheta} = \int_{\mathbb{P}} \frac{(\mathrm{d}f_\vartheta/\mathrm{d}\vartheta)(\vec{\mathbf{x}})}{p(\vec{\mathbf{x}})} p(\vec{\mathbf{x}})\ \mathrm{d}\vec{\mathbf{x}} = E_{\vec{\mathbf{X}}}\left[\frac{(\mathrm{d}f_\vartheta/\mathrm{d}\vartheta)(\vec{\mathbf{X}})}{p(\vec{\mathbf{X}})}\right]$$

where p is *any fixed* probability distribution, and $\vec{\mathbf{X}}$ is a path random variable with distribution p.

Now, we want to evaluate the derivative at $\vartheta = \vartheta_0$. Since p is an arbitrary fixed probability distribution, we can use importance sampling based on the value ϑ_0 to pick an efficient probability distribution p_{ϑ_0}. Then, an unbiased estimator of $I'(\vartheta_0)$ is given by:

$$\frac{(\mathrm{d}f_\vartheta/\mathrm{d}\vartheta)|_{\vartheta=\vartheta_0}(\vec{\mathbf{x}})}{p_{\vartheta_0}(\vec{\mathbf{x}})} \tag{5.9}$$

[2] A function $f : X \to Y$ is Lipschitz continuous if, for all $x_1, x_2 \in X$, we have that $d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$ for some constant $K \geq$, where d_X and d_Y are metrics on X and Y, respectively.

where $\vec{\mathbf{x}}$ is a path sampled according to p_{ϑ_0}.

From Section 2.3, f consists of a product of terms such as BSDF evaluations, volume transmittances, and radiance from emitters. Abstracting the origins of the individual terms, this may be written as:

$$f_\vartheta(\vec{\mathbf{x}}) = \prod_{k=1}^{K} f_{k,\vartheta}(\vec{\mathbf{x}}).$$

As a result,

$$\frac{\mathrm{d}f_\vartheta(\vec{\mathbf{x}})}{\mathrm{d}\vartheta} = f_\vartheta(\vec{\mathbf{x}}) \sum_{k=1}^{K} \frac{(\mathrm{d}f_{k,\vartheta}/\mathrm{d}\vartheta)(\vec{\mathbf{x}})}{f_{k,\vartheta}(\vec{\mathbf{x}})}.$$

Substituting the above into (5.9), the expression evaluated by the estimator for $\mathrm{d}L_i/\mathrm{d}\vartheta$ is:

$$\frac{f_{\vartheta_0}(\vec{\mathbf{x}})}{p_{\vartheta_0}(\vec{\mathbf{x}})} \left[\sum_{k=1}^{N} \overbrace{\frac{(\mathrm{d}f_{k,\vartheta}/\mathrm{d}\vartheta)|_{\vartheta=\vartheta_0}(\vec{\mathbf{x}})}{f_{k,\vartheta_0}(\vec{\mathbf{x}})}}^{h_{k,\vartheta_0}(\vec{\mathbf{x}})} \right].$$

Implementation

Instrumenting a path tracer to estimate $\mathrm{d}L_i/\mathrm{d}\vartheta$ is relatively straightforward. First, we must extend the implementations of the BCSDF and the phase function so that they can output the (partial) derivatives with respect to the model parameters (i.e., C_R, β_R, and so on). How we compute derivatives depends on how different parts of the BCSDF are evaluated. When a part is evaluated directly (for example, the term for the R mode), we symbolically differentiate the expression for that part and write another piece of code to carry out the derived calculation. Other parts, such as the probability distribution in the derived phase function, are tabulated. For these parts, we also tabulate the derivatives with respect to relevant parameters as we tabulate the parts.

Second, we must modify the path tracing algorithm so that it computes both the estimate for the incoming radiance L_i and the estimate for its derivative $\mathrm{d}L_i/\mathrm{d}\vartheta$ at ϑ_0. This modification is the simplest when we work with the non-recursive versions of the path tracing (Figure 2.4 and Figure 2.7). The idea is to maintain the running sum

$$H_j := \sum_{k=1}^{j} h_{j,\vartheta_0}(\vec{\mathbf{x}}) = \sum_{k=1}^{j} \frac{(\mathrm{d}f_{k,\vartheta}/\mathrm{d}\vartheta)|_{\vartheta=\vartheta_0}(\vec{\mathbf{x}})}{f_{k,\vartheta_0}(\vec{\mathbf{x}})}$$

in addition to the throughput variable T_j. The derivative can be obtained by multiplying T_j, R_j, and some other appropriate terms together. The pseudocode for the derivative-computing surface path tracer is given in Figure 5.3 and one for the derivative-computing volume path tracer is given in Figure 5.4.

What we have described so far is how to estimate the derivative with respect to one parameter. However, to optimize for the best parameters, we need the gradient, which is the vector of partial derivatives with respect to all parameters. Computing partial derivatives is equivalent to finding the derivative of each parameter independently, so we can modify the path tracing algorithm to compute all partial derivatives in parallel by simply maintaining the derivative estimate $\widetilde{\mathrm{d}L_i}$ and the running sum H_j separately for each parameter.

As examples of outputs that the modified path tracer produce, Figure 5.5 shows visualizations of the (partial) derivatives of images of the Fleece fabric model under two lighting/viewing configurations.

5.5.2 Measurements

To derive a model that matches a real piece of fabric, we need measurements of that fabric to match against. For each fabric, we took 16 photographs of a flat

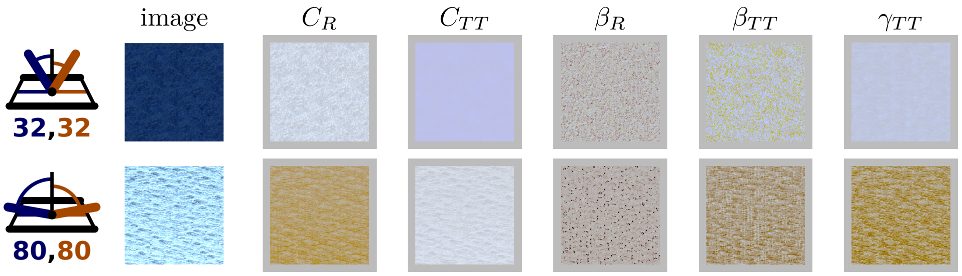
```
Derivative-Computing-Surface-Path-Tracing($\mathbf{x}_0, \omega_0, \vartheta_0$)
    ▷ Instead of the BRDF $f_r$, we use the BCSDF $S$ as the light scattering model.
      We also assume that all fibers has the same BCSDF.
    ▷ This function returns two values: the estimate of $L_i(\mathbf{x}_0, \omega_0)$
      and its derivative $(\mathrm{d}L_i/\mathrm{d}\vartheta)|_{\vartheta=\vartheta_0}$ where $\vartheta$ is a parameter of $S$.
    Cast ray from $(\mathbf{x}_0, \omega_0)$ to determine the hit point $\mathbf{x}_1$.
    if $\mathbf{x}_1 = \varnothing$
        return $(L_{env}(\omega_o), 0)$
    $\widetilde{L}_i = L_e(\mathbf{x}_1, -\omega_0)$ and $\widetilde{\mathrm{d}L}_i = 0$
    $j = 1$ and $T_j = 1$ and $H_j = 0$
    while true
        Sample direction $\omega'_j$ with probability $p'_j(\omega'_j)$.
        Cast ray $(\mathbf{x}_j, \omega'_j)$ and determine the hit point $\mathbf{x}'_{j+1}$.
        if $\mathbf{x}'_{j+1} = \varnothing$
            $\widetilde{L}_i^{\mathrm{dir}} = L_{env}(\omega'_j)$
        else
            $\widetilde{L}_i^{\mathrm{dir}} = L_e(\mathbf{x}'_{j+1}, \omega'_j)$
        $\widetilde{L}_i = \widetilde{L}_i + T_j \widetilde{L}_i^{\mathrm{dir}} S(\omega'_j, -\omega_{j-1}) \cos\theta'_j / p'_j(\omega'_j)$
        $\widetilde{\mathrm{d}L}_i = \widetilde{\mathrm{d}L}_i + T_j \widetilde{L}_i^{\mathrm{dir}} S(\omega'_j, -\omega_{j-1}) \cos\theta'_j / p'_j(\omega'_j) \left( H_j + \frac{\mathrm{d}S(\omega'_j, -\omega_{j-1})/\mathrm{d}\vartheta|_{\vartheta=\vartheta_0}}{S(\omega'_j, -\omega_{j-1})} \right)$
        Toss a coin that shows head with probability $p_t$.
        if the coin shows head
            break
        Sample direction $\omega_j$ with probability $p_j(\omega_j)$
        Cast ray $(\mathbf{x}_j, \omega_j)$ and determine the hit point $\mathbf{x}_{j+1}$.
        if $\mathbf{x}_{j+1} = \varnothing$
            break
        else
            $T_{j+1} = T_j S(\omega_j, -\omega_{j-1}) \cos\theta_j / ((1 - p_t) p_j(\omega_j))$
            $H_{j+1} = H_j + \frac{(\mathrm{d}S(\omega_j, -\omega_{j-1})/\mathrm{d}\vartheta)|_{\vartheta=\vartheta_0}}{S(\omega_j, -\omega_{j-1})}$
        $j = j + 1$
    return $(\widetilde{L}_i, \widetilde{\mathrm{d}L}_i)$
```

Figure 5.3: Pseudocode of the derivative-computing surface path tracing algorithm.

```
Derivative-Computing-Volume-Path-Tracing($\mathbf{x}_0, \omega_0, \vartheta_0$)
    ▷ We assume all points in the volume has and the same albedo $\alpha$ and
      the same phase function $f_p$, which is derived from the BCSDF.
    ▷ This function returns two values: the estimate of $L_i(\mathbf{x}_0, \omega_0)$
      and its derivative $(\mathrm{d}L_i/\mathrm{d}\vartheta)|_{\vartheta=\vartheta_0}$ with respect to parameter $\vartheta$.
      Here, $\vartheta$ is a parameter of $f_p$ or one of the components of the albedo $\alpha$.
      We do not differentiate with respect to parameters related to $\sigma_t$.
    $\widetilde{L}_i = 0$ and $\widetilde{\mathrm{d}L_i} = 0$
    $j = 0$ and $T_j = 1$ and $R_j = 0$
    while true
        $s_j =$ Woodcock-Tracking($\mathbf{x}_j, \omega_j$)
        if $s_j = \infty$
            $\widetilde{L}_i = \widetilde{L}_i + T_j L_{env}(\omega_j)$
            $\widetilde{\mathrm{d}L_i} = \widetilde{L}_i + T_j H_j L_{env}(\omega_j)$
            break
        $\mathbf{x}_{j+1} = \mathbf{x}_j + s_j \omega_j$
        Toss a coin that shows head with probability $p_t$
        if the coin shows head
            break
        Sample $\omega_{j+1}$ with probability $p(\omega_{j+1})$.
        $T_{j+1} = T_j \frac{\alpha f_p(\mathbf{x}_{j+1}, -\omega_{j+1}, -\omega_j)}{(1-p_t)p(\omega_{j+1})}$
        $H_{j+1} = H_j + \frac{\mathrm{d}(\alpha f_p(\mathbf{x}_{j+1}, -\omega_{j+1}, -\omega_j)/\mathrm{d}\vartheta|_{\vartheta=\vartheta_0}}{\alpha f_p(\mathbf{x}_{j+1}, -\omega_{j+1}, -\omega_j)}$
        $j = j + 1$
    return $\widetilde{L}_i$
```

Figure 5.4: Pseudocode of the derivative-computing volume path tracing algorithm.

sample of the cloth illuminated by a 10cm × 10cm square light source located about 61cm from the sample. Between measurements, we move the camera and the light source around hemispheres centered at a point on the fabric. Each photograph is cropped to a square covering roughly 1cm × 1cm area of the material around the center point of the camera's orbit. Figure 5.6 visualizes the 16 measurement configurations we use for appearance matching, with the corresponding cropped photographs of Fleece.

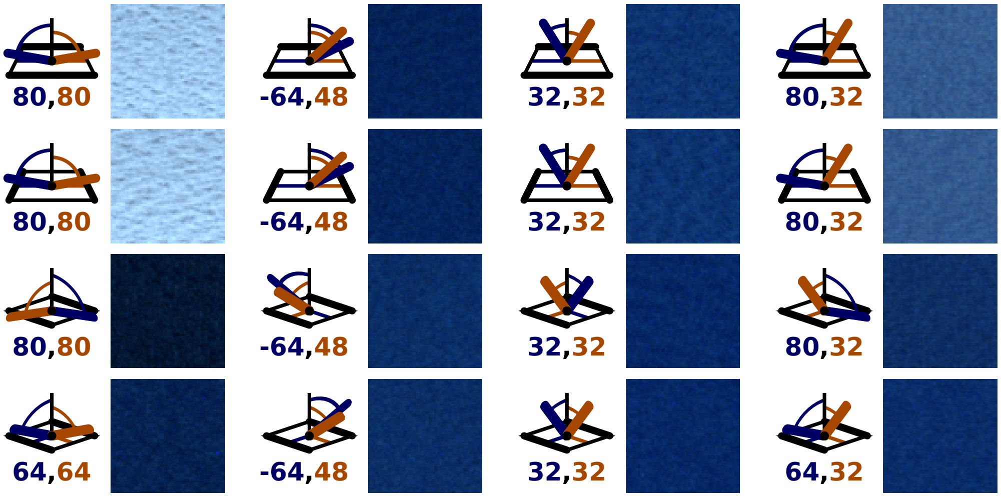


For two view configurations, we show the rendered Fleece and visualizations of its derivatives with respect to each parameter of the scattering model of Section 5.4. In the derivative images, gray (shown in the borders) indicates a value of zero, while lighter and darker values in each channel indicate positive and negative values, respectively. For scalar parameters, the image shows how each channel changes with respect to the parameter. For the color parameters, each channel of the parameter affects a single channel of the rendering, so the image visualizes the derivative of each channel of the rendering with respect to the color parameter in that channel. Derivative magnitudes are not to scale across images.

Figure 5.5: Renderings and derivatives with respect to fitted parameters of the Fleece model.

A number of factors influenced our choice of using 16 measurements. First, because there are 12 parameters to fit[3], there must be at least 12 observations to have a well-posed problem. Second, we generally would like to use as few measurements for fitting as possible because rendering detailed geometry with full light transport simulation is time consuming. Third, however, we would like enough diversity in lighting and viewing configurations. We settled on 16 measurements, which allows us to include 4 types of light/camera elevations (the columns of Figure 5.6), 2 types of fabric rotations (the odd rows versus the even rows), and whether the light source and the camera are in the same plane or not (Row 1 and 2 versus Row 3 and 4).

[3] The number of parameters comes from the combination of using the volumetric microgeometry model and the phase function derived from the fiber scattering function in Section 5.4. The parameters are C_R (3), β_R (1), C_{TT} (3), β_{TT} (1), γ_{TT} (1), and α (3), so in total there are $3 + 1 + 3 + 1 + 1 + 3 = 12$ parameters.



The icons indicate the orientation of the material (bolded edges), the light source (brown), and the camera (blue); the normal angles to the camera and light are given beneath. Due to limitations of the measurement apparatus, normal angles of 80 degrees in the final row are constrained to 64 degrees.

Figure 5.6: Measurements used in fitting process.

5.5.3 Objective Function

To optimize for parameter values, we need an objective function, which is a scalar valued function that summarizes the difference between the photographs and the corresponding renderings. We denote the objective function by $f(\vec{R}_{\vec{\vartheta}}, \vec{M})$ where $\vec{\vartheta}$ is the vector of values of all the rendering parameters, $\vec{R}_{\vec{\vartheta}}$ is the vector of pixel intensities of the rendered images when the parameters are set to $\vec{\vartheta}$, and $\vec{M}$ is the same vector of the photographs.

The photographs and the rendered images show views of the material under matching conditions, but do not show the same piece of fabric, so the objective function must compare the images without depending on the details being the same. The simplest way to do this is to average the whole image so that differences in the spatial details do not matter. Therefore, we form the measurement vector $\vec{M}$ by concatenating the average intensities of the 16 measured images in each color channel. In effect, we rely on fiber-level microgeometry to ensure a texture that is at least plausible. We optimize only against the average intensities of our measurements, guaranteeing that the base color and highlight of the material, which are visible at near and far scales, are captured accurately.

We now discuss our choice of the objective function f. Recall that the rendered measurements $\vec{R}_{\vec{\vartheta}}$ and their derivatives are only available as unbiased estimates from the renderer, so we will not be able to calculate f or its gradient exactly. Moreover, if one desires the theoretical convergence properties of stochastic gradient descent, we must calculate an unbiased estimate of f and its gradient. Thus, f must interchange with expectations as follows:

$$\mathrm{E}[f(\vec{R}_{\vec{\vartheta}}, \vec{M})] = f(\mathrm{E}[\vec{R}_{\vec{\vartheta}}], \vec{M}). \tag{5.10}$$

We observe that all multivariate polynomials in the components of $\vec{R}_{\vec{\vartheta}}$ satisfy the above property, as long as any rendered values $\vec{R}_{\vec{\vartheta}}(i)$ that occur in the same product are uncorrelated, e.g., by being calculated in separate renderings.

Our objective function takes the form of a weighted sum of terms f_{image} calculated per image:

$$f(\vec{R}, \vec{M}) = \sum_{i=1}^{N} w_i f_{\text{image}}(\vec{R}(i), \vec{M}(i)) \tag{5.11}$$

where $\vec{R}(i)$ and $\vec{M}(i)$ contain the average intensities in each channel of the render and photograph of configuration i, respectively. The per-image terms are:

$$\begin{aligned} w_i &= \frac{1}{\max(a_i, \tau)^2} \\ f_{\text{image}}(\vec{R}(i), \vec{M}(i)) &= \sum_{c \in \{r,g,b\}} (\vec{R}(i)_c - \vec{M}(i)_c)^2. \end{aligned} \tag{5.12}$$

The weight w_i is chosen such that each term of the sum in (5.11) approximates the square of the relative error between f_{image} and $\vec{M}(i)$. For simplicity, we use the average intensity $a_i = (\vec{M}(i)_r + \vec{M}(i)_g + \vec{M}(i)_b)/3$ of photograph i instead of assigning different weights to different channels. We calculate the *relative* error of each measurement to avoid overfitting the intensities of particularly bright images such as when the camera views the specular highlight produced by the light source at grazing angles.

Additionally, we threshold the weights of dark photographs to not be lower than $1/\tau$ where τ is set to 0.02 for all the fittings. Introducing the threshold prevents the optimization from overfitting the intensities of very dark images, where even differences due to the dark current noise produced by the camera may result in a large relative error in intensity.

As mentioned earlier, we want to evaluate an unbiased estimate of f, given the unbiased estimates of average intensities obtained from the renderer. Because f_{image} contains squares of the per-channel average intensities $R_{i,c}$ of the rendered images, each image must be rendered twice independently to make them uncorrelated. In this way, we may compute the squared difference without introducing bias.

5.5.4 Stochastic Gradient Descent Optimization

We now wish to explore the space of rendering parameter configurations $\vec{\vartheta}$ to minimize the value of f. For convenience, consider f as a direct function of the parameters $\vec{\vartheta}$. In stochastic gradient descent, we start with an initial parameter value $\vartheta^{(0)}$. We then iteratively modify the parameter values in the opposite direction of the (estimated) gradient:

$$\vec{\vartheta}^{(i+1)} = \vec{\vartheta}^{(i)} - \alpha_i \, \nabla f\big(\vec{\vartheta}^{(i)}\big), \tag{5.13}$$

where α_i is a scaling factor, often called the *learning rate*, that changes as the iteration proceeds. Despite the noise in the gradient estimate, it can be shown that the iteration converges to the global optimum given that f is convex over the search space and that the learning rates decrease at the appropriate speed: namely, if $\sum_{i=1}^{\infty} \alpha_i^2 < \infty$ and $\sum_{i=1}^{\infty} \alpha_i = \infty$ [7].

In our implementation, we choose the harmonic series $\alpha_i = a/i$ as the learning rates, where a is a constant. While the choice of a has no effect on the convergence guarantee, choosing a value that makes the optimization converge quickly is critical in practice. To make finding one possible, we non-linearly scale the parameter space using the process described in Section 5.5.5.

We perform the gradient descent as follows. Initializing $\vec{\vartheta}$ to a starting value $\vec{\vartheta}$ (defined in the next section), we run six different stochastic gradient descent iterations for exponentially bracketed choices of the learning rate multiplier a, running each for 60 iterations. To slow the $1/i$ decay of the step size while preserving the convergence guarantees of SGD, we initialize the iteration number i to 50. We select the optimized parameters with the lowest reported objective function value $f(\vec{R}_{\vec{\vartheta}}, \vec{M})$ as our result. In some instances, the gradient descent yielding the least residuals continued to oscillate intensely around a local minimum of the objective function after 60 iterations; in such cases, we performed a final 10 iterations at one tenth the last learning rate to descend to the local minimum.

5.5.5 Parameter Rescaling and Ranges

A suitable learning rate is hard to find when regions in parameter space have large differences in their gradient magnitudes. Such a situation complicates the selection

Parameter	Lower Bound	Upper Bound	ϑ
C_R	0.001	0.999	0.1
C_{TT}	0.001	0.999	0.85
β_R	1.0°	10.0°	5.0°
β_{TT}	10.0°	45.0°	10.0°
γ_{TT}	1.0°	45.0°	10.0°
α	0.001	0.999	0.85
γ	0.005	1.5	N/A
d	N/A	N/A	4000.0

The final column lists the default parameter values comprising the $\vec{\vartheta}$ of the rescaling process described in Section 5.5.5. α and d denote the albedo and density multiplier when rendering volume geometry (the latter is held fixed). Parameter fitting for the microflake model by the method of [109] use the stated ranges for α and γ. For the black Velvet, a lower bound of 0.04 was used for C_R.

Table 5.1: Fitting domains of rendering parameters.

of the learning rate because:

- The process may enter a neighborhood where gradients have small magnitude. If the learning rate is too small, it will spend a long time moving in short increments through this region.
- The region surrounding a local optimum may have gradients with large magnitude. If the learning rate is too large, the optimization process will repeatedly overshoot the local minimum many times before converging.

For example, a rendering of cloth will change much more dramatically with respect to the TT mode color, C_{TT}, when the parameter value is large than when the value is small. As such, a large learning rate is needed when C_{TT} is small, and a small learning rate is needed when C_{TT} is large. A situation might arise where no single learning rate works well on all regions.

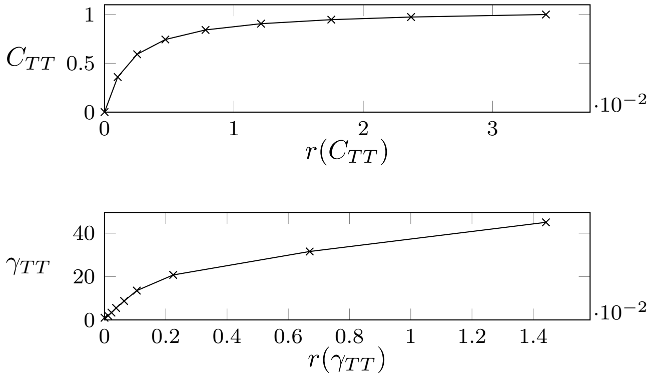
We mitigate the effect of disparity in gradient magnitudes by automatically defining a mapping r from the rendering parameters $\vec{\vartheta}$ to a space of search parameters $r(\vec{\vartheta})$. The stochastic gradient descent is performed in this rescaled space instead of the space of rendering parameters. Our goal is to find a mapping such that a unit change in the remapped space $r(\vec{\vartheta})$ corresponds to a constant change in the objective function value. To approximate this, we choose a starting configuration $\vec{\vartheta}$ of the rendering parameters and rescale each parameter individually based on its effect on the objective function near this configuration.

For each parameter $\vec{\vartheta}_p$, we define a mapping r_p for the parameter as follows. We fix all other parameters to the values specified by $\vec{\vartheta}$, and calculate the average intensities $\vec{R}$ obtained by setting parameter $\vec{\vartheta}_p$ to an ascending sequence of values $c_1, \ldots, c_k$ spanning the range we will permit the optimization to explore for this parameter. We then define r_p at the values c_j as shown below, and extend it to a piecewise linear function on the domain $[c_1, c_k]$:

$$\begin{aligned} r_p(c_1) &= 0 \\ r_p(c_{j+1}) - r_p(c_j) &= \sum_{i=0}^{N} w_i \sqrt{f_{\text{image}}(\vec{R}_{\vec{\vartheta},\vec{\vartheta}_i=c_j}(i), \vec{R}_{\vec{\vartheta},\vec{\vartheta}_i=c_{j+1}}(i))}. \end{aligned} \tag{5.14}$$

As $\vec{\vartheta}_p$ increases from c_j to c_{j+1}, the increase in the function $r_p(\vec{\vartheta}_p)$ is equal to the change in the averages $\vec{R}$, though we actually compute the square root to counter the nonlinearity of f_{image}. Consequently, regions of the domain of permitted values of $\vec{\vartheta}_p$ that correspond to a large change in $\vec{R}$ are mapped to larger regions in the space of search parameter values $r_p(\vec{\vartheta}_p)$. Figure 5.7 visualizes the calculated rescaling curves r_p for parameters of the scattering model of Section 5.4 that have large disparity between derivative magnitudes in different regions.

The rescaling function r for all the parameters, a vector function, is formed by assembling the per-parameter functions r_p. However, we treat scalar parameters (β_R, γ_R, and γ_{TT}) differently from color parameters (C_R and C_{TT}). For each scalar parameter p, we compute r_p as detailed above and use the same function in the



The curves relate the parameters $r(\vartheta)$ optimized by the gradient descent to the rendering parameters ϑ. The starting parameter configuration $\vec{\vartheta}$ is given in Table 5.1.

Figure 5.7: Calculated rescaling curves for the C_{TT} and γ_{TT} parameters of the scattering model of Section 5.4 for Fleece.

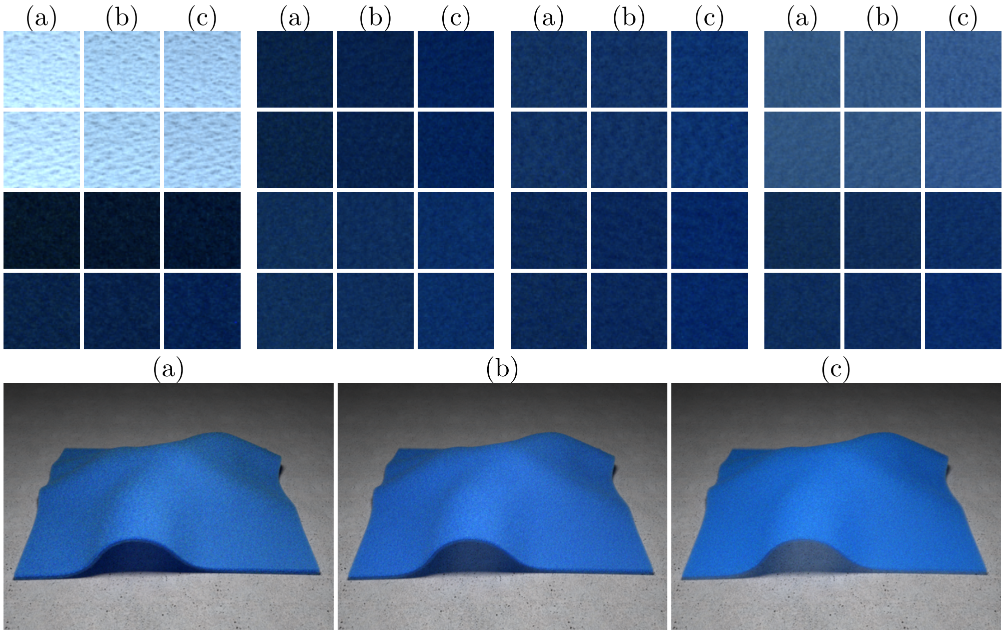
assembly of r. On the other hand, we treat a color parameter p as three separate scalar parameters that always have the same value when computing the rescaling curve r_p. Consequently, changing the (scalar) value of p is equivalent to changing the value of three scalar parameters at the same time. Thus, when assembling r, we set the per-channel function r_p^{red}, r_p^{green}, and r_p^{blue} to $r_p/3$ instead of r_p.

Compared to not rescaling the parameters, our approach handles the images' different sensitivity to unit change of different parameters better. For example, we found that the rendered images are much more sensitive to changes in C_{TT} than in γ_{TT}, so at least a component-wise linear rescaling was essential. However, adding non-linearity was also very important. Before employing the rescaling method we proposed, we found it difficult to fit to a material that was relatively bright in at least one channel in grazing configurations. As a specific example, the rendered images were so sensitive to C_{TT} in the grazing configurations that, unless the initial condition was very near to the correct value, the gradients were large enough to push the C_{TT} value to both extremes of the domain unless the step size was very small.

Table 5.1 lists the ranges of values we permit each parameter to take during the optimization. The varying ranges permitted for the lobe widths are due to practical considerations. We observed that values of β_R greater than 10 degrees led to an implausibly diffuse fiber-level appearance without significantly affecting the average intensities fitted against by the optimization. Moreover, values of β_{TT} less than 10 degrees led to instability in the fitting process, as they greatly increased the sensitivity of the rendered images to the C_{TT} parameter in grazing views with intense highlights. For the black Velvet, we constrained C_R to be at least 0.04.

The rightmost column of Table 5.1 lists the default parameter values used when rendering images to define the rescaling functions r_p. We chose values we believed would be typical of an "average" material, so that the rescaling is defined relative to a region of the parameter space we expect to be most heavily explored by the optimization. For each parameter, we choose a nine-value sequence $c_1, \ldots, c_9$ for use in defining the function r, assigning values c_j more finely towards the upper bound of C_{TT} and towards the lower bounds of C_R, β_R, β_{TT}, and γ_{TT}, capturing the greater sensitivity of the rendered images to values of the parameters in those regions. More details on these values can be found in Appendix B.1.

For volume geometry, we include the single-scattering albedo α in the fitting process but leave the density multiplier d fixed. As shown in Figure 5.8, we observed that when fitting against the photographs, all of which place both the camera and light source above the plane of the material, the value of d did not significantly affect the ability of the optimization to match the photographed appearance. Under draped configurations, though, fitted parameters with a low d resulted in an unacceptably "thin" appearance.



When the camera and light are both above the plane of the material, reducing the density from (a) 4000 to (b) 2000 to (c) 1000 has minimal effect on the ability of the optimization to recover the same appearance. In draped configurations, though, a scale of 1000 leads to an unrealistically thin appearance.

Figure 5.8: Fitted Fleece results for volumes of different density scales d.

5.6 Fabric Geometry Construction

In this section, we show how to construct fabric microgeometry. We construct two representations: a surface-based one and a volumetric one. The surface-based representation is a collection of discrete surfaces modeling the surfaces of individual fibers, while the volumetric representation stores density and fiber direction in a high resolution voxel grid.

Both representations are created from micro CT scans of cloth samples according to the pipeline in Figure 5.9. We use Zhao et al.'s [109] image processing pipeline to compute a preliminary volumetric representation, a voxel array containing (1) the density of the material in each voxel, and, (2) the local direction of the fiber at that voxel. From this volume, we infer locations of textile fibers and then

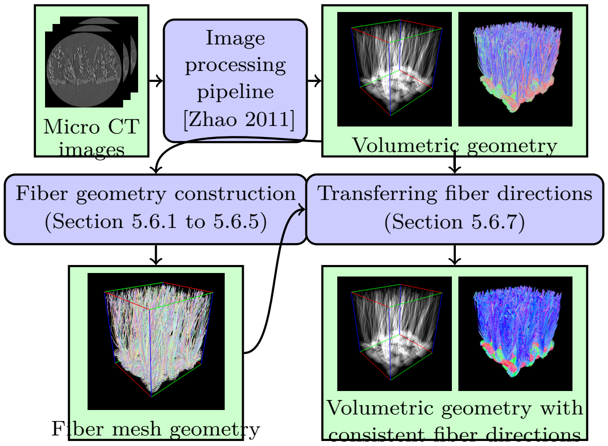

Micro CT images
Image processing pipeline [Zhao 2011]
Volumetric geometry
Fiber geometry construction (Section 5.6.1 to 5.6.5)
Transferring fiber directions (Section 5.6.7)
Fiber mesh geometry
Volumetric geometry with consistent fiber directions

Figure 5.9: Fabric geometry creation pipeline.

construct cylindrical surfaces to cover them. This gives us the surface-based fiber representation. Lastly, we use this fiber model to improve the consistency of the preliminary fiber direction volume, resulting in the final volumetric representation.

We first focus on the process that converts the volumetric representation to fibers. The input to this process is a voxel array with density and direction, and the output is fiber geometry in the form of a collection of 3D polylines. Each segment of polyline acts as the axis of symmetry of a cylinder with circular cross-section, and consecutive cylinders sharing a vertex are joined with a miter joint, producing a continuous 3D surface for rendering.

We assume that the input volume is roughly axis-aligned; i.e., the weft fibers are roughly along the x-axis, the warp fibers along the y-axis, and in case of fabrics such as velvet, vertical fibers along the z-axis.

To produce fiber geometry, the following steps are taken:

1. *Volume decomposition.* The input density volume is decomposed into three volumes corresponding to the x-, y-, and z-axis. In this way, the warp, weft,

and vertical fibers can be processed separately along their length.

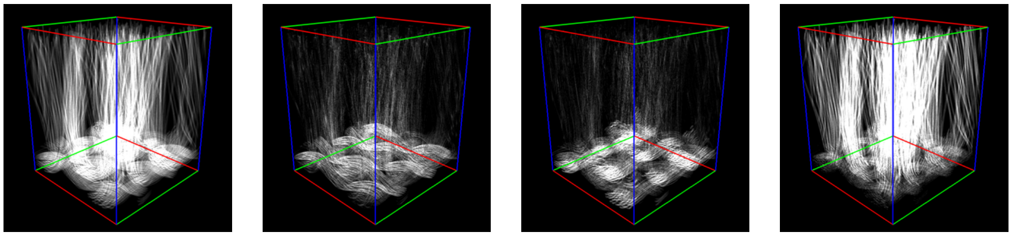
2. *Fiber center detection.* The 2D slices of each volume along the corresponding axis are processed independently to detect cross sectional centers for fibers going through the slices.
3. *Polyline creation and smoothing.* Nearby fiber centers in adjacent slices are linked together to form chained polylines in 3D. These polylines act as skeletons of the reconstructed fibers. Short polylines are removed, and the union of all polylines from the three volumes become the polylines for the entire volume. These polylines are then smoothed.
4. *Radius determination.* The single radius of all the cylinders constituting the fiber surfaces is then determined.

We now discuss each of these steps in more detail.

5.6.1 Volume Decomposition

The first step is to separate the model into three subvolumes containing only the fibers oriented (approximately) along the warp, the weft, and the vertical directions. The three subvolumes are created simply by associating each voxel with the coordinate axis closest to its direction vector (determined by selecting the component of the direction vector that is largest in absolute value). The three subvolumes are later processed independently to recover fibers that run primarily in each volume's associated direction.

Figure 5.10 depicts the result of decomposing a micro CT scan volume of Velvet, which has *pile* fibers sticking up perpendicular to the overall plane of the fabric.



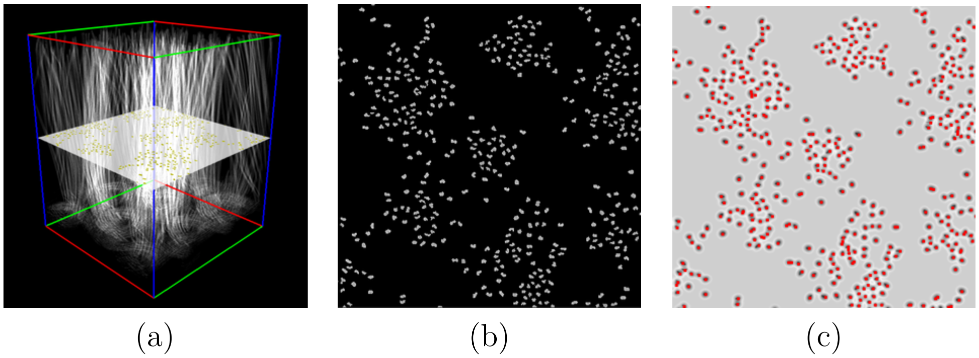
original x-dominant y-dominant z-dominant

Figure 5.10: The density volume obtained from a micro CT scan of Velvet and its three decomposed volumes.

We see that the x- and y-dominant volumes primarily contain voxels in the woven part (one for the warp, the other for the weft), and the z-dominant volume mostly contains the pile fibers. Noise voxels can be observed in all the volumes and can give rise to short extraneous fibers, which are removed later in the pipeline.

5.6.2 Fiber Center Detection

Each subvolume is processed in 2D slices perpendicular to its dominant direction, with the goal of locating where each fiber crosses each slice. Since fibers are roughly perpendicular to the slices, they appear as compact blobs of higher density. To determine the centers of these blobs, we apply the standard blob detector which convolves the slice image with Laplacian-of-Gaussian filters at several scales, and then finds the local minima of the response in both scale and spatial domain [43]. Figure 5.11 shows fiber centers detected in a slice of the z-dominant axis of the Velvet volume in Figure 5.10. The result is a collection of fiber locations $\{(i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m)\}$ for each slice, which must be matched up across slices to produce 3D fibers.



(a) A slice of the z-dominant subvolume of Velvet in Figure 5.10 in 3D view, (b) the slice viewed as a 2D image, and (c) the result of the convolution of the Laplacian of Gaussian filter to the slice image with the detected blob positions depicted as small red squares.

Figure 5.11: Processing a slice of the Velvet volume.

5.6.3 Fiber Building

In this step, we connect the detected fiber centers to create polylines representing the individual fibers. Contrary to previous approaches to fiber detection that grow fibers one after another [33, 46], we view this as a matching task: we first decide which detected centers in all pairs of neighboring slices belong to the same fiber, then extract maximal paths in the resulting graph to determine the polylines to generate.

We connect fibers across slices by solving a series of bipartite graph matching problems, each matching the fiber centers detected in slice k to the centers in slice $k+1$. These fiber centers become vertices in a weighted bipartite graph. Edges are constructed between vertices on different slices with weights inversely proportional to the in-plane distance between the end points. In particular, between point a in slice k and point b in slice $k+1$, there is an edge with weight:

$$w(a, b) = \exp(-d(a, b)^2/(2\sigma^2))$$
$$d(a, b) = \sqrt{(i_a - i'_b)^2 + (j_a - j'_b)^2}.$$

or zero if $d(a,b) > \sigma$. We used $\sigma = 5$ voxels for all the fabric volumes we processed in this chapter.

The maximum weighted bipartite matching can be solved to an approximation ratio of 1/2 using an $O(n^2 \log n)$ greedy algorithm [19], which is much faster at our problem size than the optimal but $O(n^3)$ Hungarian algorithm. We found that the greedy algorithm worked well with our data.

Due to noise, this process generates many short polylines in addition to the long polylines corresponding to well-tracked fibers. We retain only polylines with at least 10 vertices when we collect the results from the x, y, and z volumes together. Figure 5.12 shows the result of this step on the Velvet volume.

5.6.4 Polyline Smoothing

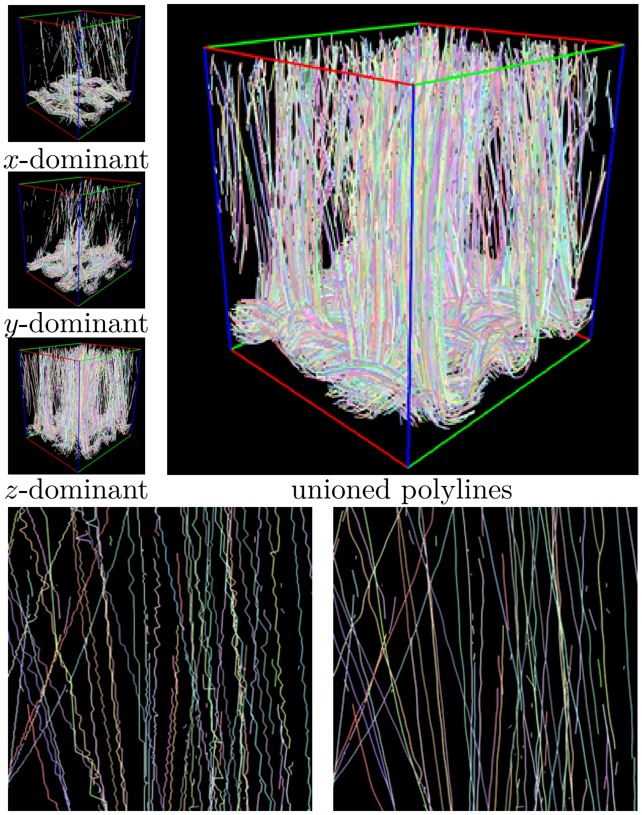
Because fiber centers are located independently per slice, the raw polylines are noisy (see Figure 5.12). To reduce noise we smooth them as described in [46]. New vertex positions $p_1, p_2, \ldots, p_n$ are computed to minimize the energy:

$$\mathcal{E} = \sum_i \alpha \|p_i - p_i^{(0)}\|^2 + \|p_{i-1} - 2p_i + p_{i+1}\|^2$$

where $p_i^{(0)}$ is the original 3D position of the ith vertex, and $\alpha = 0.1$.

5.6.5 Radius Determination

Given the collection of polylines determined above, we now need to compute a value for the fiber radius to fully define the set of 3D cylinders that represent the



x-dominant

y-dominant

z-dominant

unioned polylines

before smoothing

after smoothing

Figure 5.12: Polylines generated by the fiber growing process and the effect of smoothing on some generated polylines.

fibers. We choose the radius to match the volume covered by the fibers to the voxels that is occupied by fibers in the original volume.

More concretely, we first upsample the original volume by a factor of 4. Then, given a candidate value r of the radius, we rasterize cylinders of radius r around all polylines into a volume of the same resolution. The score for the value r is given by:

$$\mathrm{score}(r) = |C(r) \cap V| - |C(r) \otimes V|$$

where,

- $C(r)$ is the set of voxels contained in one of the cylinders when the radius

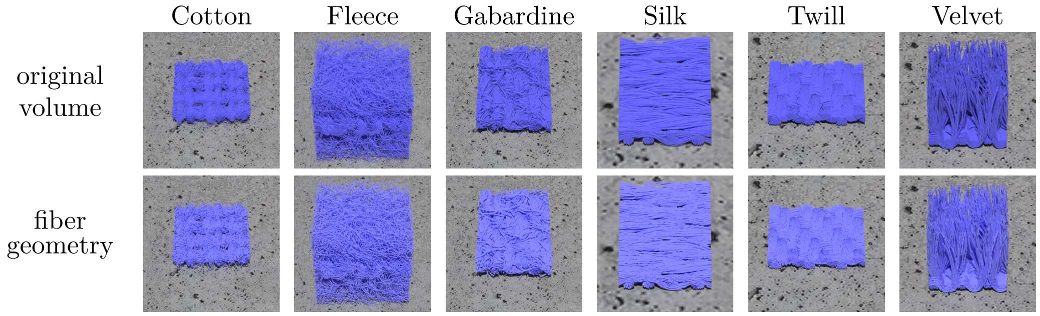


Figure 5.13: Visualizations of original micro CT scan volumes of six pieces of fabric and their reconstructed fibers.

is set to value r,

- V is the set of non-empty voxels in the original volume, and
- $\otimes$ is the symmetric difference operator:

$$A \otimes B = (A \cup B) - (A \cap B).$$

The radius values are found by trying out 20 evenly spaced values between w to $4w$, where w is the width of a voxel in the original volume.

5.6.6 Results and Discussions

Figure 5.13 compares renderings of the original micro CT scan volumes of six fabrics with the constructed fiber geometry. In general, the fibers agree with the geometry of the volume but look a bit thinner because (1) the pipeline filtered out many short fibers, and (2) the radii were determined in part by trying to minimize the number of voxels covered by the cylinders but not the original volumes.

We note that our pipeline can break highly curved fibers into disconnected pieces when different parts of the fiber belong to different decomposed volumes.

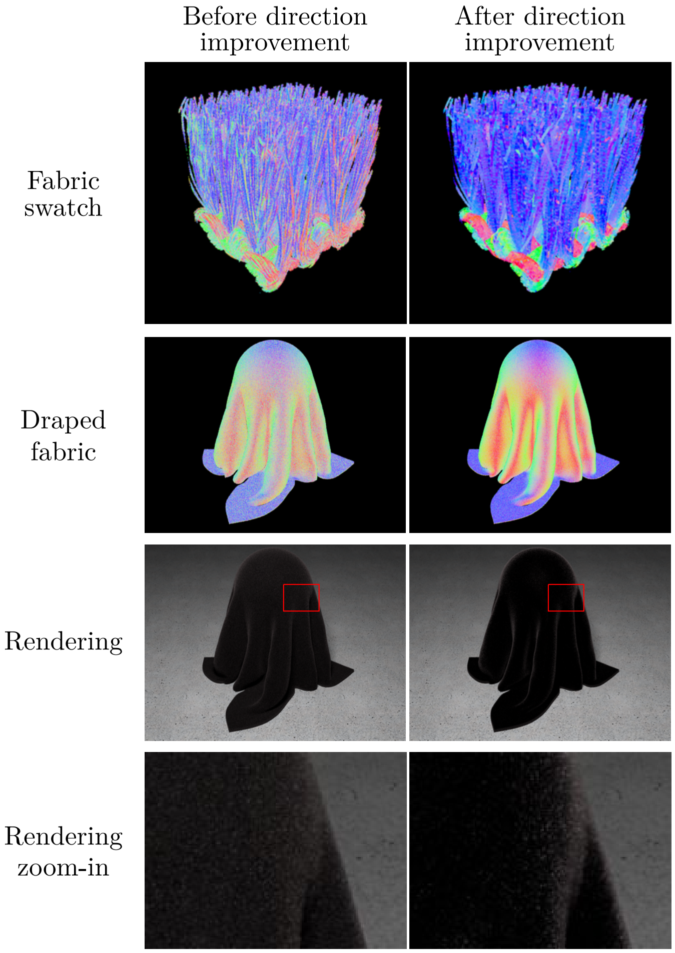
We do not attempt to reconnect the fibers, since, as we shall see in the final rendering of the fabric model, some broken fibers are acceptable at typical viewing distances where fibers are not clearly resolved.

Our pipeline requires a scan resolution that resolves individual fibers well. It also requires that the fiber cross sections appear circular or elliptical at the scanned resolution, which might not be true for synthetic fibers such as Rayon, Nylon, and acetate.

5.6.7 Improving volume direction fields

In real cloth, fine-scale irregularities in the fibers have a major effect on luster: smooth, well aligned fibers (as in many silks and synthetics) reflect coherently and produce bright highlights, whereas kinky and irregularly arranged fibers (cotton, wool) reflect to a range of directions, producing a less shiny appearance. In fiber or volume models, noise in the fiber curves or the direction volumes can introduce a similar change in appearance—noise makes it impossible to match the appearance of smooth fibers as seen in Silk or Velvet.

Smoothing of polylines is very effective in removing this noise without disturbing fabric structure or over-smoothing less organized fabrics. But when smoothing directions in a volume, naïve approaches can easily smooth out important features by mixing the directions belonging to different fibers. Rather than pursuing more complex noise-reduction methods for the volume direction fields and to keep the comparison between fibers and volumes on an equal footing, we smoothed the volume direction fields by simply transferring the directions from the fiber models onto the volume model. This is implemented by setting the direction of any voxel



The first two rows depict the local fiber direction at the first interaction between the eye ray and the volume, visualizing the absolute values of the x, y, and z components of direction as an RGB color. The volume in the second row is obtained by tiling the swatch from the first row and then warping the tiled volume with a shell map, as done in [109].

Figure 5.14: The effect of fiber direction consistency on appearance.

contained in a fiber cylinder to a unit vector parallel to the fiber. To increase coverage, we use cylinders of radius $2r$ instead of r, where r is the radius determined in Section 5.6.5.

The result of the above process can be observed in the second column of Figure 5.14. We can see that the direction field becomes much less noisy, and the highlight in the rendering becomes much more prominent relative to the base color of the material.

In summary, we have shown how to compute a surface-based fiber representation from the micro CT data, and how to transfer the smoothed directions from this representation to create a volumetric representation. We will now compare these two representations to evaluate their ability to represent real fabrics accurately.

5.7 Results

In this section, we present the results of our appearance matching pipeline and compare various model representations for a range of fabrics. We first detail our use of the geometry processing pipeline. Then, we present the matching results and evaluate our appearance-matched models against the photographed materials in different configurations to validate our approach.

5.7.1 Data Acquisition and Processing

We processed 6 cloth samples whose details are given in Table 5.2. All samples, except for Gabardine, were scanned with the XRadia VERSA XRM-500 scanner at the Cornell Imaging Multiscale CT Facility. Gabardine's scan was made at the High-Resolution X-ray Computed Tomography Facility at The University of Texas at Austin. Each volumetric scan, except Fleece, was rotated so that the fibers are aligned with the x-, y-, and z- axis according to the requirement in Section 5.4. The volumes were then cropped so that they can be easily tiled.

Name	Material	Weave	Color	Voxel size	Data size	s	t	h	ϵ_d	ϵ_J	Fiber radius
Fleece	N/A	unwoven*	blue	6.60 μm	500 × 540 × 586	2.00	3.00	11	0.23	-14	8.68 μm
Gabardine	wool	twill	red	5.00 μm	671 × 457 × 233	1.00	2.00	16	0.45	-10	16.05 μm
Silk	silk	satin	red	1.40 μm	780 × 530 × 160	1.00	2.00	11	0.55	-30	4.72 μm
Velvet	N/A	unwoven	black	3.03 μm	430 × 478 × 524	0.25	0.75	15	0.58	-155	9.25 μm
Twill	N/A	twill	green	2.51 μm	570 × 715 × 165	1.00	2.00	11	0.46	-16	4.88/7.27 μm
Cotton	cotton	gauze	white	5.50 μm	461 × 440 × 160	0.25	0.75	13	0.41	-5	8.97μm

Note: The data sizes were computed after the rotation and cropping were performed. The s, t, h, ϵ_d and ϵ_J are the parameters for the image processing pipeline described in [109]. Twill fabric has two fiber radii because the warp and the weft fibers are of different sizes. *Fleece is a knit fabric processed so that the visible surface is similar to felt.

Table 5.2: Cloth samples, volumetric model parameters, and fiber radii.

Some volumes and their associated fibers received the following special processing:

- The Silk volume was sheared to align the warp and weft yarns with the x- and y- axes, respectively.
- The warp and weft yarns of Twill are composed of different types of fibers with different radii. To capture this difference, we independently ran the radius determination procedure for fibers generated from the x-dominant and y-dominant subvolumes.
- To reduce seams when tiling fabrics with evident regular structure — i.e., Silk, Gabardine, and Twill — we created Wang tiles [15] of the volumes based on their top-down views and used the Wang tiles as source volumes to generate fibers instead of the original volumes.
- Some x- and y-slices of the Cotton volume were removed to ensure that the spacing between the warp and weft yarns roughly match the photographs taken for parameter fitting. This is done so that the area of the rendered image that is covered by a fiber is roughly the same as that of the photographs. Since the weave pattern of the material is quite loose, the inter-yarn spacing may change locally depending on the handling of the material.

The photographs of cloth samples were taken using the Cornell spherical gantry equipped with a Canon EOS 50D camera and a 10cm × 10cm LED area light source. The fabrics were mounted on a turntable whose top is a black metal plate. As a result, the photographs did not capture the appearance of the fabrics alone but the appearance of the fabrics with the metal plate underneath them. We simulate this condition by inserting a black mesh underneath all the fabrics being rendered.

For each of the fabric samples, we created three complete fabric models:

- The first uses the fiber mesh geometry we reconstruct (Section 5.6) and our fiber scattering model (Section 5.4). This model allows us to explore the capability of the fiber geometry/fiber appearance approach. We refer to this combination as “Fiber/BCSDF”.
- The second uses the volumetric geometry representation with improved direction field (Section 5.6.7) and the phase function derived from the fiber scattering function (Section 5.4.3). This model allows us to assess the volumetric geometry/fiber appearance approach. We refer to this combination as “Volume/BCSDF”.
- The third uses the same volumetric geometry representation as the second approach with the microflake-based phase function as specified in [109]. This model allows us to compare the above two new approaches against Zhao et al.’s work. We refer to this combination as “Volume/microflake”.

We compute the scattering parameters for the Volume/microflake model by adapting the fitting procedure from Zhao et al.’s work to the measurements used for the other models we evaluate. Zhao et al. photograph a curved sample of the material, isolate a region of the image with varied appearance, and assign microflake model parameters to match the mean and variance of the pixels in the region. To approximate this, we concatenate the images used by our appearance matching method and treat the concatenated image as the region to optimize. As shown in Figure 5.15, the microflake model is incapable of reproducing the extreme highlights seen in two of the fitting images, so we omit these when fitting the model.

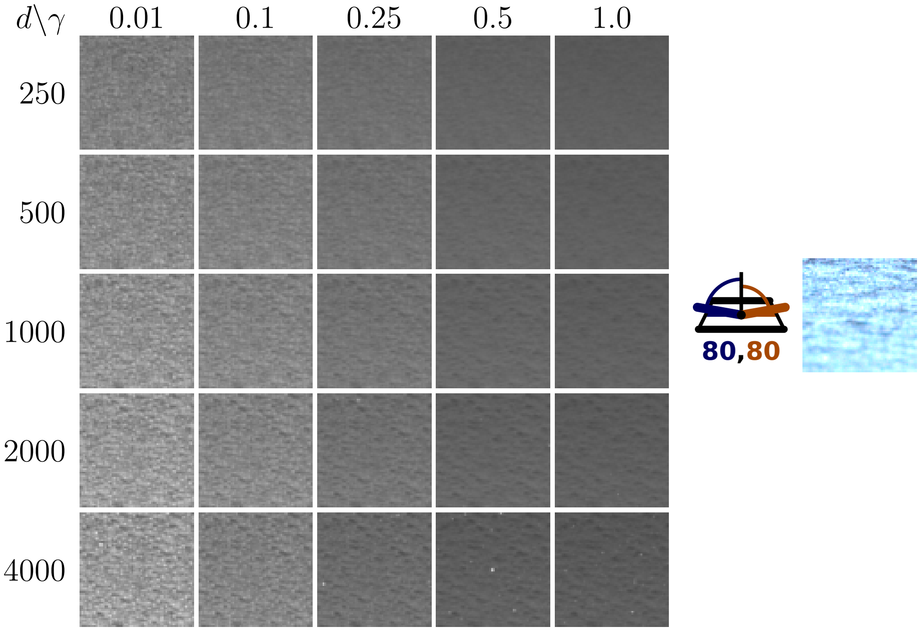
We performed all appearance matching using PCs with 2.27 GHz Intel XEON CPUs and at least 64 GB of RAM. A scene configuration is rendered into a 64×64 image at 64 samples per pixel. An iteration with 16 configurations, each rendered two times, took about 14 minutes to complete on a single core. We performed 60 iterations, and the whole process took about 14 core-hours for a fabric sample modeled with either the Fiber/BCSDF and Volume/BCSDF models. The Volume/microflake fitting took 2 core-hours because the microflake model was faster to compute and 27 binary search iterations were performed.

5.7.2 Validation

Validation in planar configurations Figure 5.19 shows the photographs used by the appearance matching process for each material alongside renderings produced by the three models. Figures 5.20 and 5.21 show the scatter plots of the average pixel values of the photographs versus the renderings. The supplementary material of [39] contains 492 more validation configurations that were not used for fitting. The corresponding fitted parameter values are given in Table 5.3.

Validation in non-planar configurations To evaluate the qualitative appearance in more natural configurations, Figure 5.22 shows the fabrics wrapped around a cylinder of radius 1.5cm and the renderings of the three models in the same configuration. Figures 5.23 and 5.24 plot the average pixel values of each column of the images in Figure 5.22, allowing us to quantitatively compare the methods.[4] Figure 5.25, and Figure 5.26 shows the renderings of the fabrics in a simple draped

[4] The photographs and the corresponding renderings are not aligned horizontally. As a result, we shift the photograph in the x-axis so that the peak in the $\phi_i - \phi_o = 162°$ configuration aligns with the peaks of the renderings.



The camera and light source are at a normal angle of 80 degrees and in a plane with the imaged point on the Fleece volume. Bracketing the density scale multiplier d and microflake standard deviation γ, no configuration is able to match the intensity of the photograph. The albedo is left fixed at 0.999 as the image brightens monotonically with albedo.

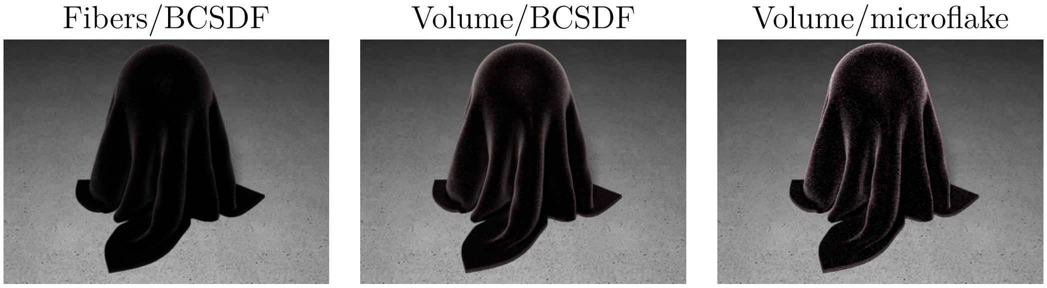
Figure 5.15: Insufficient brightness of microflake phase function at grazing angles.

configuration. In Figure 5.16, we also show Velvet in a more elaborate draped configuration which better reveals the fabric's characteristic highlights.

5.7.3 Discussion

We now discuss the results of each fabric in turn before drawing conclusions.

Fleece. The appearance of Fleece is matched well by the Fiber/BCSDF and Volume/BCSDF models. The Volume/microflake model also performs well in non-grazing configurations, but it is unable to reproduce the brightness in grazing



Fibers/BCSDF Volume/BCSDF Volume/microflake

Our appearance matching process yielded parameters that resulted in different highlight brightness between the images. However, the highlight consistently emerges as the fabric turns away from the camera in all models.

Figure 5.16: Renderings of Velvet in a draped configuration.

configurations (i.e., the first two rows of the first column in Figure 5.19 and the $\phi_i - \phi_o = 126°$ and $162°$ in Figures 5.22 and 5.23). The second row of Figure 5.25 shows that this discrepancy has a large impact on appearance particularly when the light source and the camera are on the opposite side of the fabric. Because the microflake model has no transmission component, it produces an opaque appearance, so the microflake Fleece looks less soft than our models.

Gabardine. In Figure 5.19 and Figure 5.22, our two models are able to match the appearance in all configurations well while the Fiber/BCSDF performs somewhat better in the $(80°, 80°)$ grazing configurations in Figure 5.19. On the other hand, the Volume/microflake still cannot produce bright highlights in these grazing configurations. Figure 5.22 and 5.23 also exhibit the same trends. All models are close to the photographs when the fabric is in retroreflection configurations ($\theta_i - \theta_o$ low). The Volume/microflake model, however, becomes too dark in grazing configurations ($\theta_i - \theta_o$ high), but the other two models become somewhat too bright. This indicates that the new models are improvements over the previous approach for Gabardine.

Silk. Our model still performs better than the Volume/microflake at grazing configurations. However, in Figure 5.19, the grazing highlights rendered with our models have a slight cyan tint. Additionally, in Figure 5.23, the BCSDF models performs less well in the green and blue channel in configurations with low $\theta_i - \theta_o$ in contrast to their good performance in the red channel.

These behaviors are because the red channel of C_R in the two BCSDF models is much higher than the blue and green channels. The parameters cause light reflected off fibers to be very bright in the red channel and dim in the others. They also cause the light transmitted through fibers to have a cyan tint. We surmise that the fitting process arrived at these values because it needs to set the red channel of C_R high to match the chromaticity of the dim training configurations, which are weighted higher by the objective function: we can see in Figure 5.20 that the two BCSDF models match the red channel of the dim configurations better than brighter ones. As such, the behaviors are likely to be caused by the models' additional expressiveness and our choice of objective function.

However, in Figures 5.22 and 5.25, the three models look very similar in retroreflection configurations, and the cyan tint of the highlight is very hard to notice. Thus, the suboptimal behaviors of the BCSDF models do not have a significant negative impact, but the inability of the Volume/microflake model to become bright in forward scattering configurations takes away a large part of the overall appearance.

Velvet. Velvet was a challenging material to fit because its fitting residuals were higher than for other materials. As a result, the optimization with the settings used for the other materials did not consistently recover the highlights and speckled appearance that characterize it. These effects are more likely to arise from the R mode, given the dark base color of the material, so we constrained C_R to be at least 0.04 for Velvet, to force the high-residual fit towards a region more likely to produce acceptable results.

In Figure 5.19, the Fiber/BCSDF model matches the training data better than the two volumetric models, which are too bright in many configurations. Nevertheless, Figure 5.21 suggests that all models fit poorly to validation examples. This indicates that more training data might lead to parameters that generalize better to the observed data.

In all draped configurations (Figures 5.16, 5.22, and 5.26), the largest difference between the models is the brightness of the highlights. Note though that all the highlights appear at the same locations: where the fabric turns away from the camera. Moreover, the 10°, 126° and 162° columns of Figure 5.24 are evidence that all models can predict the locations of bright highlights under cylindrical configurations despite magnitudes being off. This suggests that all 3 models, together with a consistent direction field, can model Velvet's appearance, but the appearance matching process needs to be improved to obtain better parameter values.

Still, we note that Velvet is a tricky material because its pile fibers move when touched, and we did not control their directions both when the micro CT scanning was performed and when the photographs were taken. While using more training examples can improve results post hoc, controlling the microgeometry might be required to get truly good results.

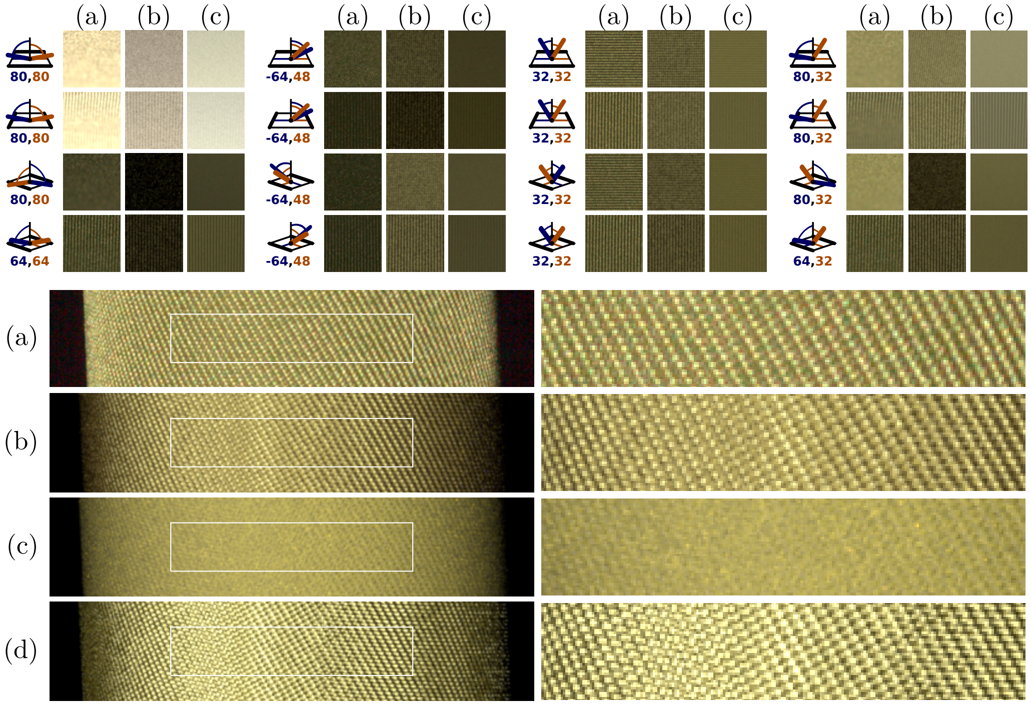
Twill. According to Figure 5.21, our BCSDF models can quantitatively match the training data better than the Volume/microflake. However, there are three problems with the result. The first is that all models generalize rather poorly to the validation configurations.

The second is that our models are worse at reproducing the fabric's texture than the Volume/microflake. The photographs feature alternating bright yellow and dark brown stripes, but our models do not yield as much color contrast. On the other hand, the Volume/microflake produces more contrast between stripes, but does not capture the BRDF well in Figure 5.19.

Low texture contrast was caused by the fibers' being made transparent, as can be seen by low C_R values and high C_{TT} values. However, the fibers still reflect back some light in reality. Indeed, setting C_R higher makes the texture more prominent but worsens the overall color matches. (See Figure 5.17.) A cost function that considers contrast between different image parts might be able to trade BRDF correctness with texture correctness and is left for future work.

The third is that there is a feature in photographs of the cylindrical configurations that none of the models could capture. The photographs get brighter around the cylinder's edges in the 10°, 50°, and 90° configurations while the models get dimmer.

The fact that our models can fit the training data well but not the validation data might be explained by overfitting. However, the fact that we have to trade BRDF correctness with texture contrast and that none of the models can account for the bright cylinder's edges suggest rather that the models are not expressive enough to represent all aspects of Twill's appearance. The reason may be that the warp and weft yarns, while appearing to be dyed the same color, might actually have different optical properties.



The parameters for the models were derived manually from the fitted parameters with C_R changed to $(0.472, 0.460, 0.218)$ and C_{TT} changed to $(0.540, 0.435, 0.310)$. Under configurations used for parameter fitting, we compare (a) the photographs with (b) the alternative parameters and (c) the parameters fitted by our process. Additionally, we provide a photo and renderings of the fabric in the cylindrical configuration where $\phi_o - \phi_i = 10°$. For completeness, we also provide the Volume/microflake model (d) in the cylindrical configuration. While the alternative parameters and the microflake are inferior at the BRDF level, they produce more contrast in the texture.

Figure 5.17: An alternative set of scattering parameters of the Fiber/BCSDF model for Twill.

Cotton. Photographs in Figure 5.22 show that Cotton is a rather diffuse material, and all 3 models can capture this appearance as well as provide realistic geometric details. Nevertheless, measurements in Figure 5.19 indicate that the fabric gets extremely bright at the 80° grazing configurations, and none of the models could imitate this behavior. Moreover, they could not match the color of the photographs in any of the remaining configurations. Compared to the other materials, the recovered Cotton fiber geometry and volume are unusually sparse within the threads of the material, suggesting that too many small fibers were thresholded away by the volume processing, limiting the ability of the models to reflect sufficient light back to the viewer.

Highlights. While we matched against a sparse sample of the fabric's BRDF under flat configurations, Figures 5.23 and 5.24 suggest that our model produced highlights at the right locations and could imitate the complex highlight shapes such as those of Silk and Velvet in the 10° configurations. Our models always match the magnitude of the highlights better in the grazing configurations than the Volume/microflake model. Nonetheless, there are some cases where the latter performed better in non-grazing configurations; for example, Fleece and Silk in the blue and green channels. However, as can be seen in Figure 5.22, cases where our models perform worse than the Volume/microflake do not lead to drastic differences from the reference photographs like the dim grazing highlights of the Volume/microflake.

The lobe widths our system produces may not be accurate. In Table 5.3, many lobe width values are at their extremes, especially the β_R and β_{TT} values of the two BCSDF models. We observed that decreasing β_R and β_{TT} led to brighter images in grazing configurations, so these parameter values tend to get pushed to their lower bounds. We also surmise that the extra degrees of freedom introduced by the albedo α in the Volume/BCSDF model caused more of the model's β_R and β_{TT} values to be at the extremes. As such, adding more training examples might alleviate this problem. Still, note that, while the individual lobe widths might not be accurate, the overall appearances are good in cases as Fleece, Gabardine, and Silk. Moreover, the problems of Twill and Cotton do not seem to be related to lobe widths at all.

We now draw conclusions from the discussions above.

Comparison with microflake model. For Fleece, Gabardine, Twill, and Silk, our new models—the Fiber/BCSDF and the Volume/BCSDF—are able to produce results that match the far-field photographed appearance across the images used for fitting. By comparison, the Volume/microflake model produces renderings that vary less in intensity from image to image, and fail to recover the intense highlights as the camera and light approach grazing angles. Thus, we conclude that our models are superior to the the microflake model. We believe the Fresnel term, which makes the R mode considerably brighter at grazing angles, accounts for much of its advantages.

Fiber versus volume. Except for Velvet, the Fiber/BCSDF and the Volume/BCSDF models produce very similar results. Nevertheless, in Figure 5.19, the Fiber/BCSDF model generally produces slightly brighter images in grazing configurations. Still, the difference in brightness of grazing highlights is very hard to notice in draped configurations of Figures 5.22, 5.25, and 5.26. We therefore conclude that, in practice, the fiber geometry and volume geometry are similarly good when used with our light scattering model.

Areas for Improvement. While we consider Fleece, Gabardine, and Silk to be success cases, the results on the other three fabrics indicate a few potential areas of improvement.

- **Number of training examples.** Velvet indicates the small number of training examples may yield parameters that generalize poorly. Few training examples might lead to lobe width values' being pushed to their extremes, especially when there are extra degrees of freedoms in the scattering model.

- **Texture correctness.** Our objective function only considers correctness at the BRDF level, ignoring texture correctness. As seen in Twill, optimizing according to the function led to parameters that do not reproduce the material's texture contrast.
- **Microgeometry correctness.** We currently have no means to test whether the microgeometry is accurate, and the inaccuracy could have caused problems in Velvet and Cotton.
- **Model correctness.** Our models seem to be not expressive enough to capture Twill's appearance. Moreover, while performing much better than the Volume/microflake, our models still could not precisely reproduce the extremely bright grazing highlights of Silk, Twill, and Cotton.
- **Non-physical behaviors.** The cyan tint in Silk's grazing highlight, while not a major issue on the overall fitted appearance, shows that our light scattering model can produce non-physical appearance. This issue arises from our allowing the R mode to have color to achieve better matching results. A model where expressiveness is retained and physics-based behaviors are strictly observed is a possible future work.

5.7.4 Optical Thickness

As we focused on reflection from fabric rather than transmission, our results do not address the correctness of back-lit appearance. The density scale for the volumes was arbitrarily fixed because it had little effect on the appearance from the illuminated side. (This was also done in [109].) As a result, the renderings may appear optically thicker or thinner than the actual fabrics. Also, while our fiber reconstruction algorithm tries to make the generated fibers cover the micro CT volume well, some volume is always lost because the algorithm throws away fibers to remove noise.

The optical thickness has a significant impact on appearance when the fabric is not draped over an opaque object. Figure 5.8 illustrates the effect of density on the appearance of volume models. Figure 5.18 shows the change in appearance of the draped fiber models when the underlying black meshes are removed. Notice that the fabrics become much brighter after removal due to multiple scattering in and through their layers.

We emphasize that transmission is not a fundamental limitation, but rather a part of the appearance space we have not yet measured. It is entirely within the scope of our optimization and appearance models, but it is currently unknown how well the models will fit under back-lit configurations.

While the fiber and volume microgeometry representations are similar in their abilities to capture fabric reflectance, the optical thickness of the volume geometry can be controlled easily just by setting the density scale. The density of fiber-based geometry can be manipulated by changing the fiber radius, but how to compute derivatives with respect to it is unclear. Therefore it is currently not possible to optimize radius in our system, so the volume geometry is advantageous in this regard. How to control the parameters to achieve the right optical thickness is left for future work.

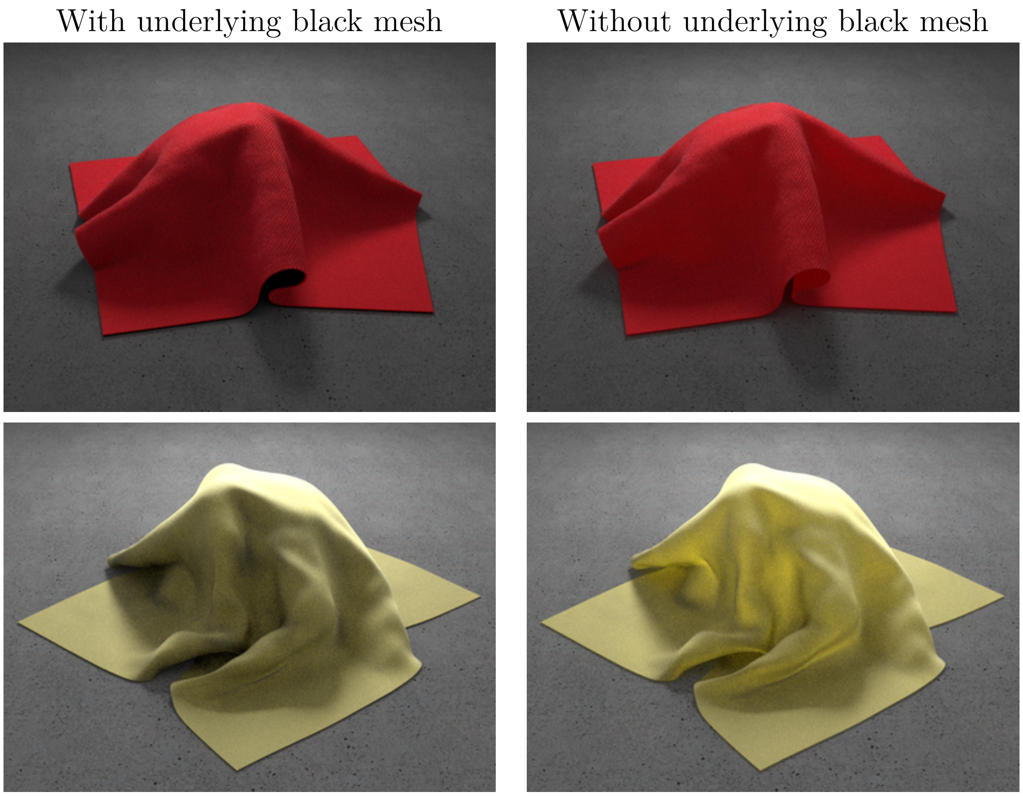
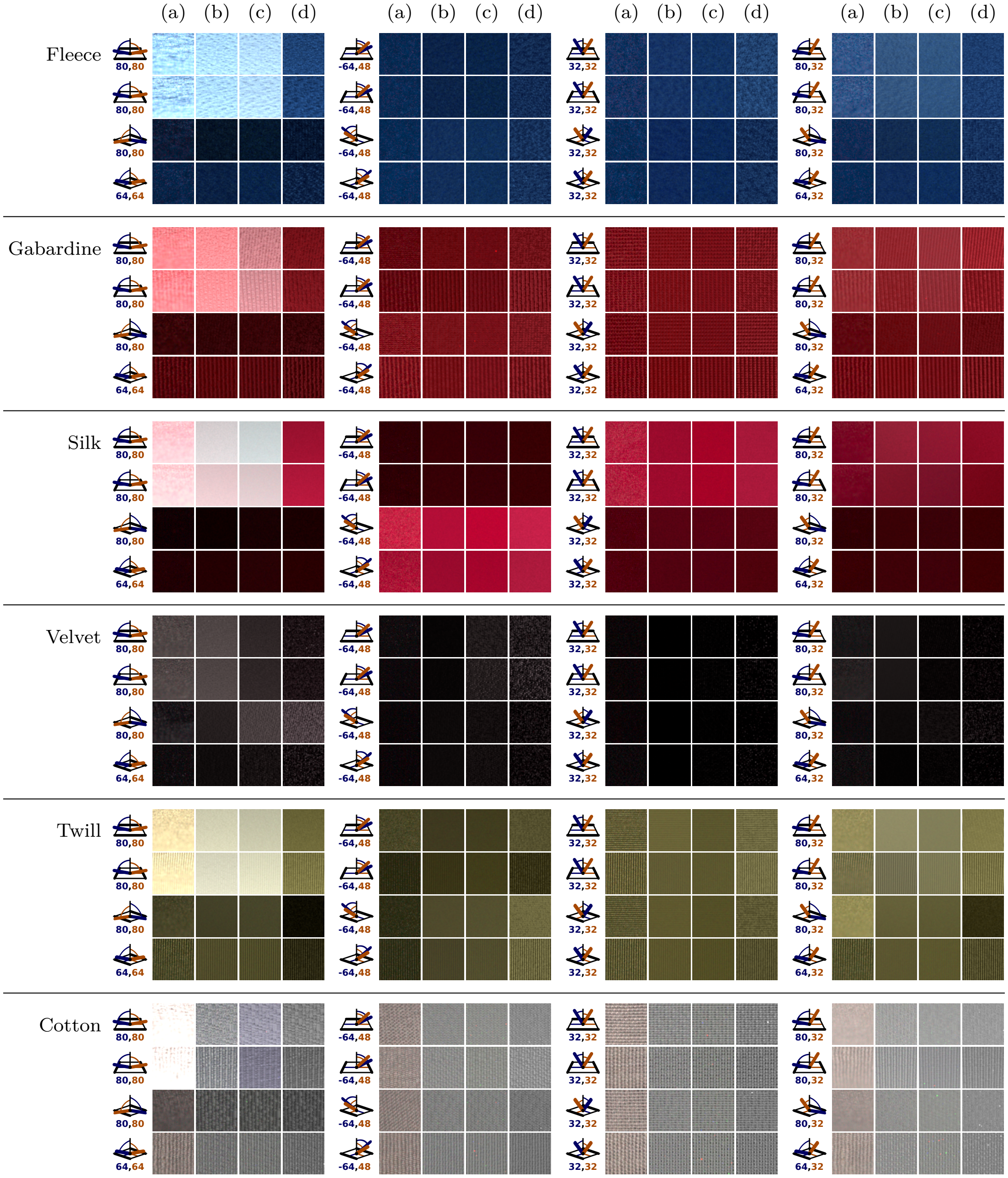


Figure 5.18: Appearance of Gabardine and Twill with and without the underlying black meshes.



For all of the viewing configurations used by the fitting process, we show (a) the photographs, (b) renderings produced by the fiber/BCSDF model, (c) renderings produced by the volume/BCSDF model, and (d) renderings produced by the Volume/microflake model. The icons indicate the orientation of the material (bolded edges), the light source (brown), and the camera (blue); the normal angles to the light and camera are given beneath. Comparisons across all 492 photographed configurations are in the supplementary material.

Figure 5.19: Results for all materials and rendering methods on fitting configurations.

Material	Fiber scattering model, fibers				
	C_R	C_{TT}	β_R	β_{TT}	γ_{TT}
Fleece	0.040, 0.087, 0.087	0.452, 0.725, 0.948	7.238	**10.000**	25.989
Gabardine	0.185, 0.047, 0.069	**0.999**, 0.330, 0.354	2.141	**10.000**	23.548
Silk	0.745, 0.008, 0.070	0.620, 0.553, 0.562	**1.000**	**10.000**	19.823
Velvet	0.044, **0.040**, **0.040**	0.076, 0.058, 0.057	1.577	24.933	44.881
Twill	**0.001**, **0.001**, 0.024	0.987, 0.975, 0.825	1.367	23.509	26.419
Cotton	0.989, 0.959, 0.874	**0.999**, **0.999**, **0.999**	**1.000**	27.197	38.269

Material	Fiber scattering model, volume					
	C_R	C_{TT}	β_R	β_{TT}	γ_{TT}	α
Fleece	0.032, 0.049, 0.055	0.759, 0.622, **0.999**	3.786	**10.000**	21.865	0.631, 0.840, 0.972
Gabardine	0.110, 0.035, 0.048	0.868, 0.633, 0.592	5.034	**10.000**	23.902	0.993, 0.651, 0.698
Silk	0.992, **0.001**, 0.034	0.002, 0.690, 0.570	**1.000**	**10.000**	13.900	0.940, 0.746, 0.773
Velvet	0.969, 0.985, 0.986	0.006, 0.003, 0.003	**10.000**	**10.000**	**45.000**	0.388, 0.310, 0.316
Twill	**0.001**, **0.001**, 0.016	**0.999**, **0.999**, 0.693	**1.000**	19.759	21.156	0.974, 0.969, 0.878
Cotton	0.447, 0.486, 0.251	0.171, 0.125, 0.279	**10.000**	**10.000**	41.464	**0.999**, **0.999**, **0.999**

Material	Microflake, volume	
	α	γ
Fleece	0.137, 0.416, 0.812	0.013
Gabardine	0.967, 0.160, 0.230	0.013
Silk	0.936, 0.051, 0.214	0.013
Velvet	0.490, 0.381, 0.412	**0.005**
Twill	0.843, 0.812, 0.463	0.013
Cotton	**0.999**, **0.999**, **0.999**	0.013

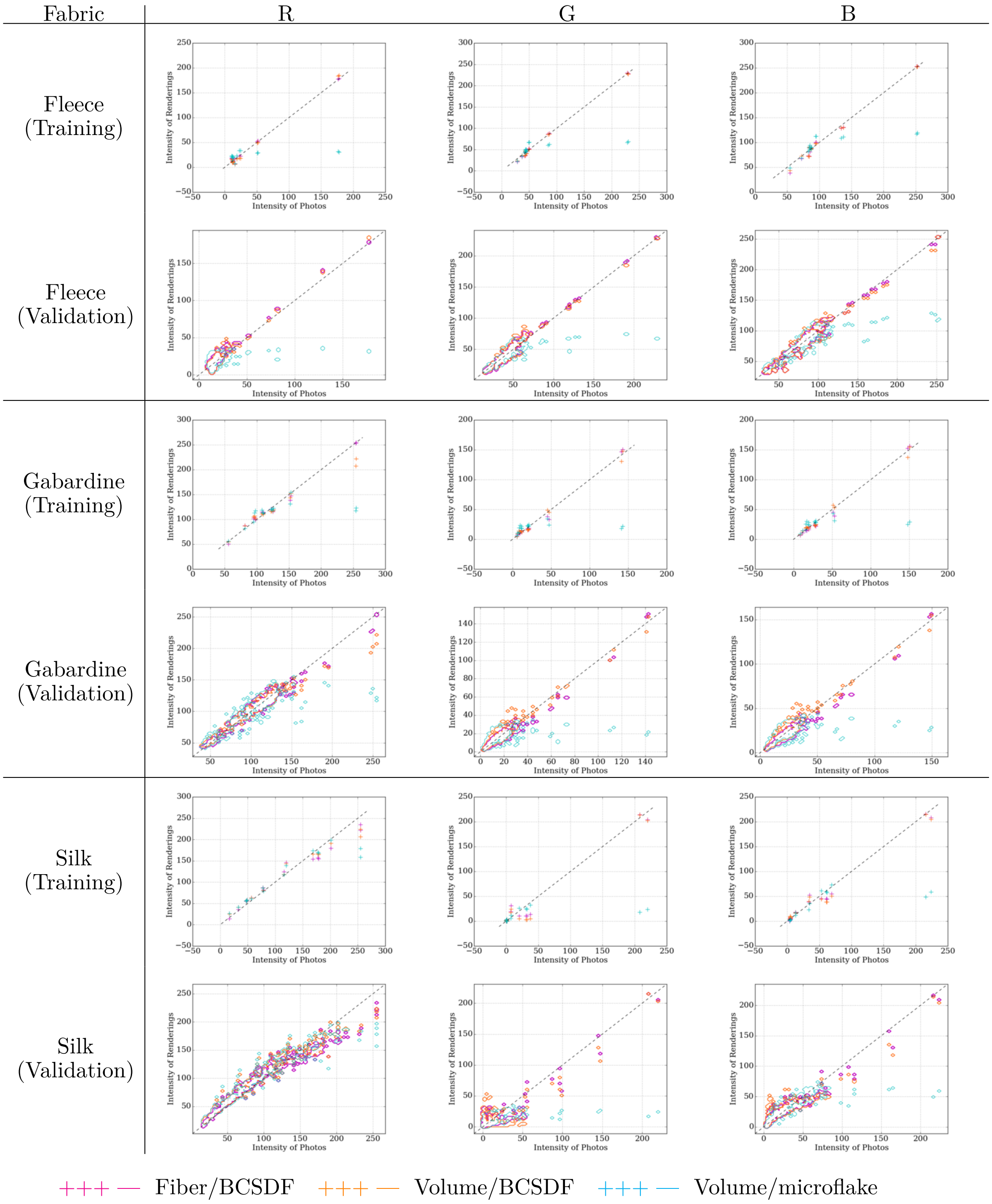
Note: Parameter values that are at an extreme of their permitted range are bolded. For volumes, the density multiplier d is fixed at 4000.

Table 5.3: Scattering model parameters fitted by appearance matching process for all materials and rendering models.

5.8 Conclusions

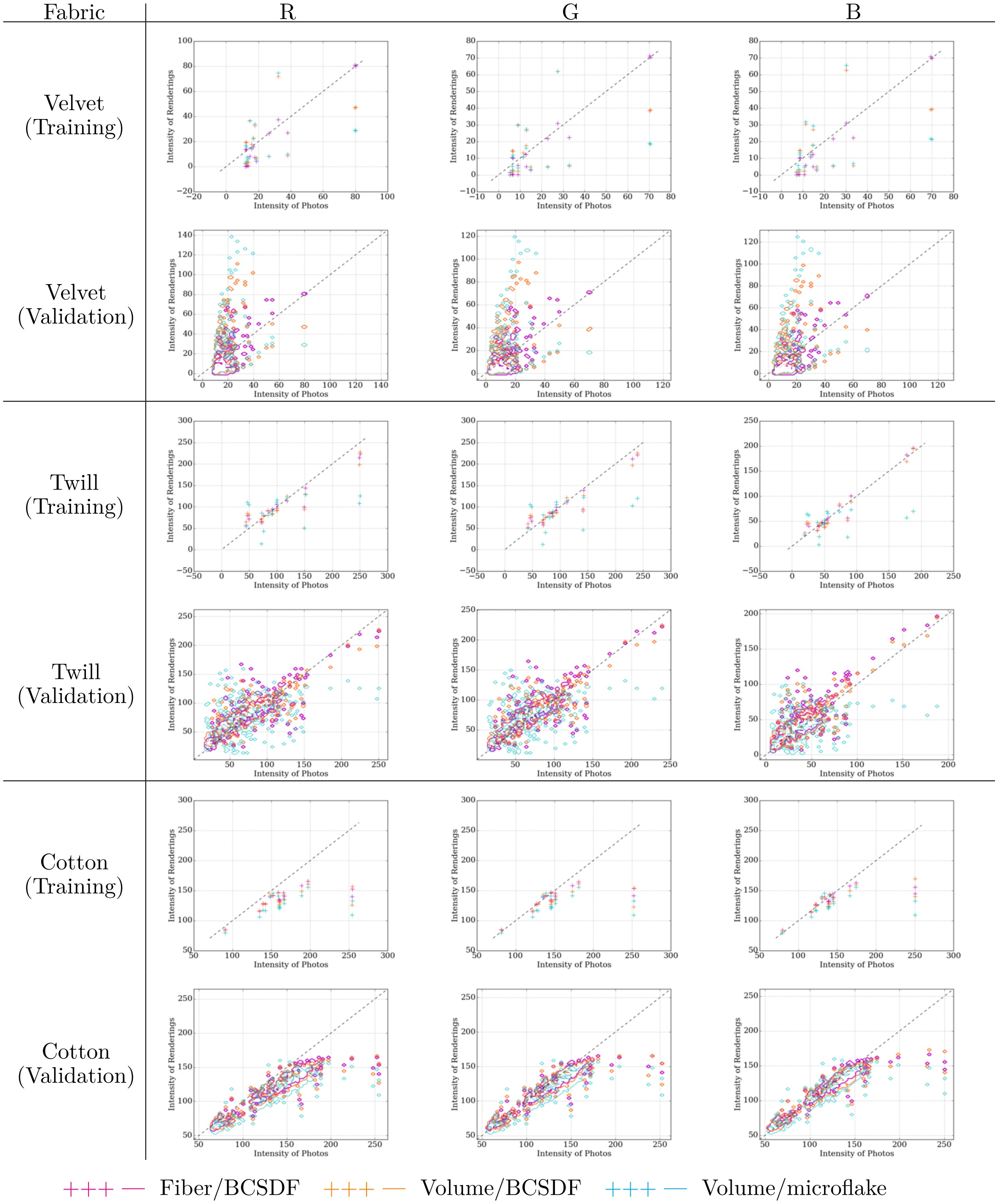
Reproducing the appearance of fabrics is critical for many applications. While progress has been made on increasingly sophisticated appearance models for fabrics, matching the appearance of real fabrics remains very hard. In this chapter, we made two contributions to creating fabric renderings that match real fabrics. First, we introduced an appearance matching framework based on differentiation and optimization to match rendered images with photographs. Second, we introduced a simple fiber-based scattering model (BCSDF), and coupled this with new approaches to reconstruct fiber-based geometry and better volumetric models of fabrics. Finally, we matched the appearance of these new models against real photographs and evaluated their strengths and weaknesses. We found that having a fiber-based BCSDF scattering model was critical to match appearance in grazing configurations. Once we use such a scattering model, both the fiber-based geometry and volume models were approximately similar in quality. Both these approaches proved superior over the prior state-of-art volume based models with microflake scattering.

Additionally we believe the approach we propose maps a way forward in the field of appearance models based on microgeometry. With our new methods for differentiation and optimization, different models can be systematically tested and compared on an equal basis, providing a clear way to identify deficiencies in existing models and to evaluate a range of possible improvements in order to design the next generation of models for a given material. This general approach can be applied to other problems where it is desirable to test the ability of a model to match measurements, but there is a complicated global illumination process in between the parameters and the data.



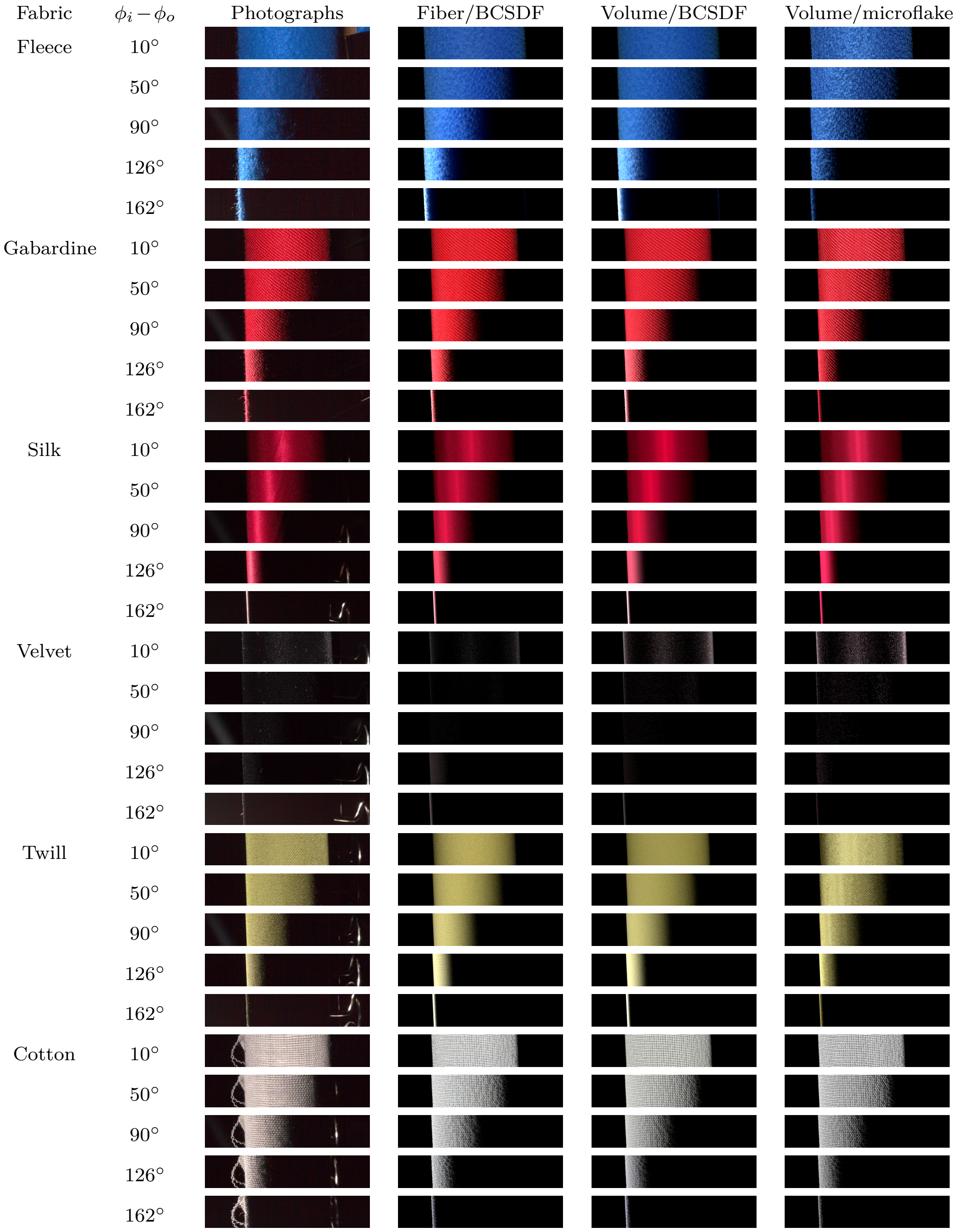
The first row of each material shows the plots for the training configurations, which are the images in Figure 5.19. The second row shows those for the validation configurations, whose images are available in the supplementary material. Because there are 492 validation configurations, we plot the contours of the points instead.

Figure 5.20: Scatter plots of the average values of the photographs versus those of renderings of Fleece, Gabardine, and Silk.



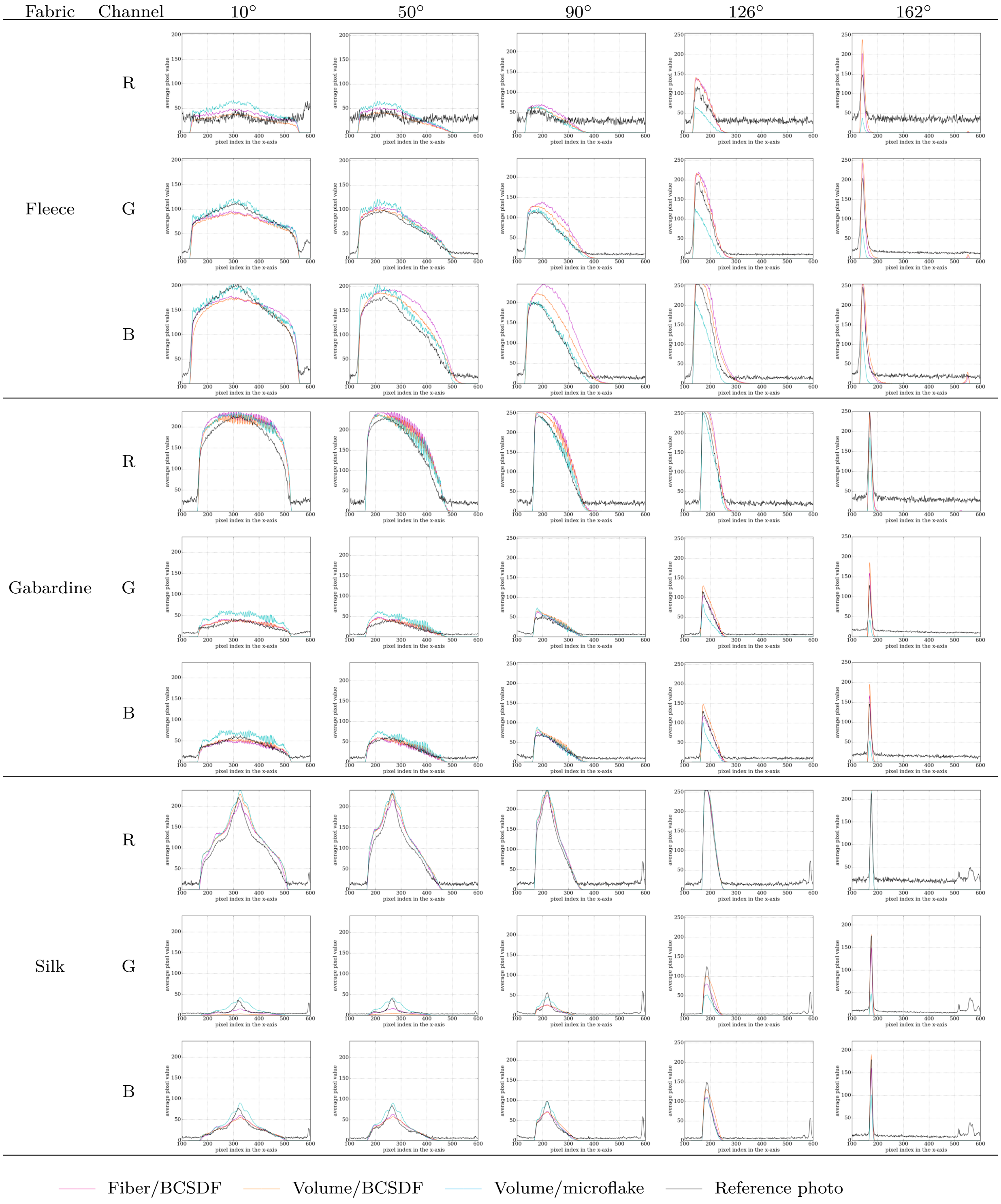
The setting of these plots are the same as in Figure 5.20.

Figure 5.21: Scatter plots of the average values of the photographs versus those of renderings of Twill, Velvet, and Cotton.



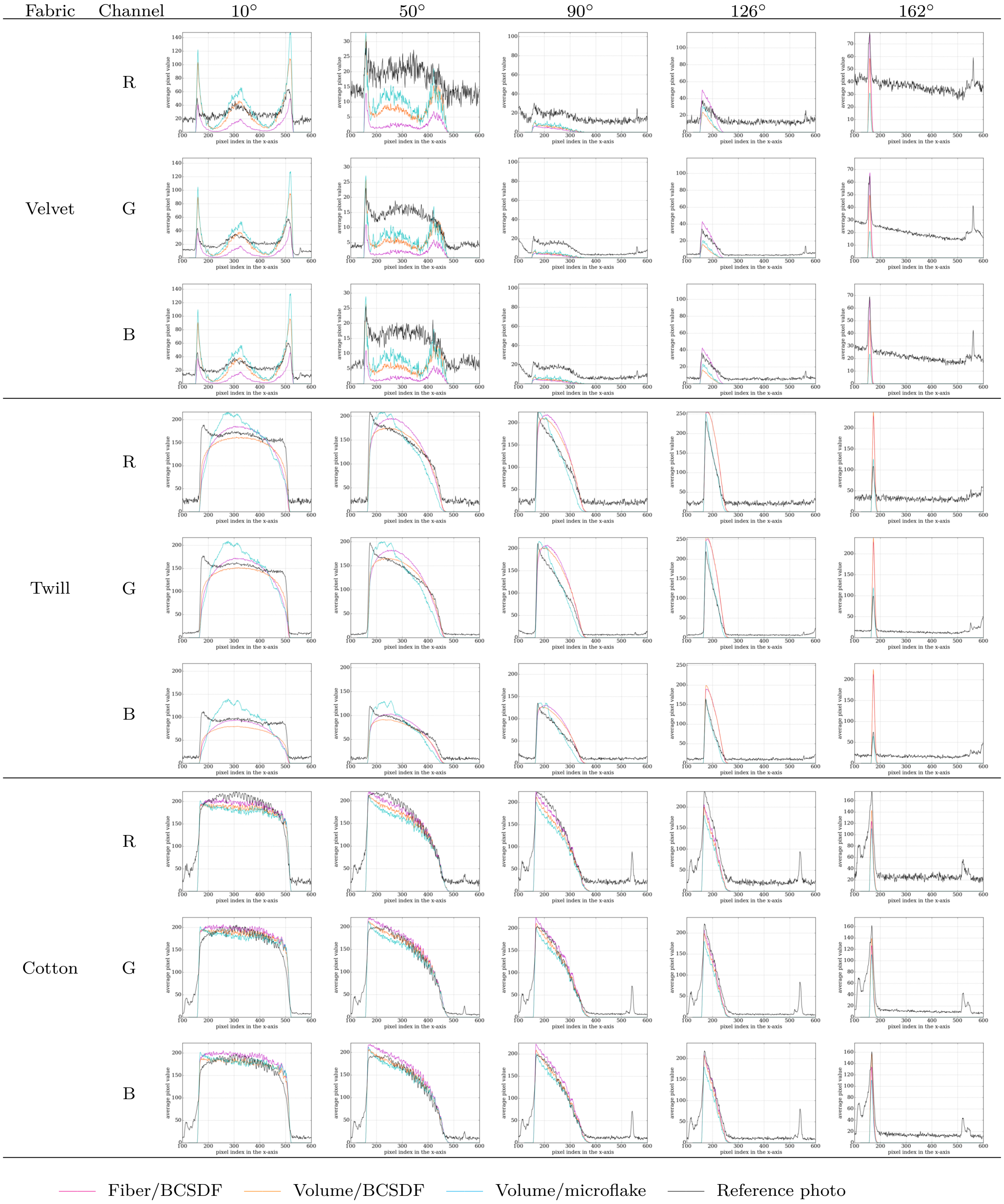
The fabric is wrapped around a cylinder of radius 1.5cm whose axis is vertical and corresponds to the longitudinal angle $\theta = 0°$. The camera was fixed in all images with the light source arranged so that its location spans from a retroreflection configuration ($\phi_i - \phi_o = 10°$) to being close to the opposite of the camera ($\phi_i - \phi_o = 162°$). The longitudinal angles of both the camera and the light source were set to $\theta = 80°$ in all images, except those with $\phi_i - \phi_o = 162°$ where the light source was lifted to $\theta = 76°$ to avoid being seen by the camera.

Figure 5.22: Comparison between photographs and renderings produced by the three fabric rendering methods for the six fabrics.



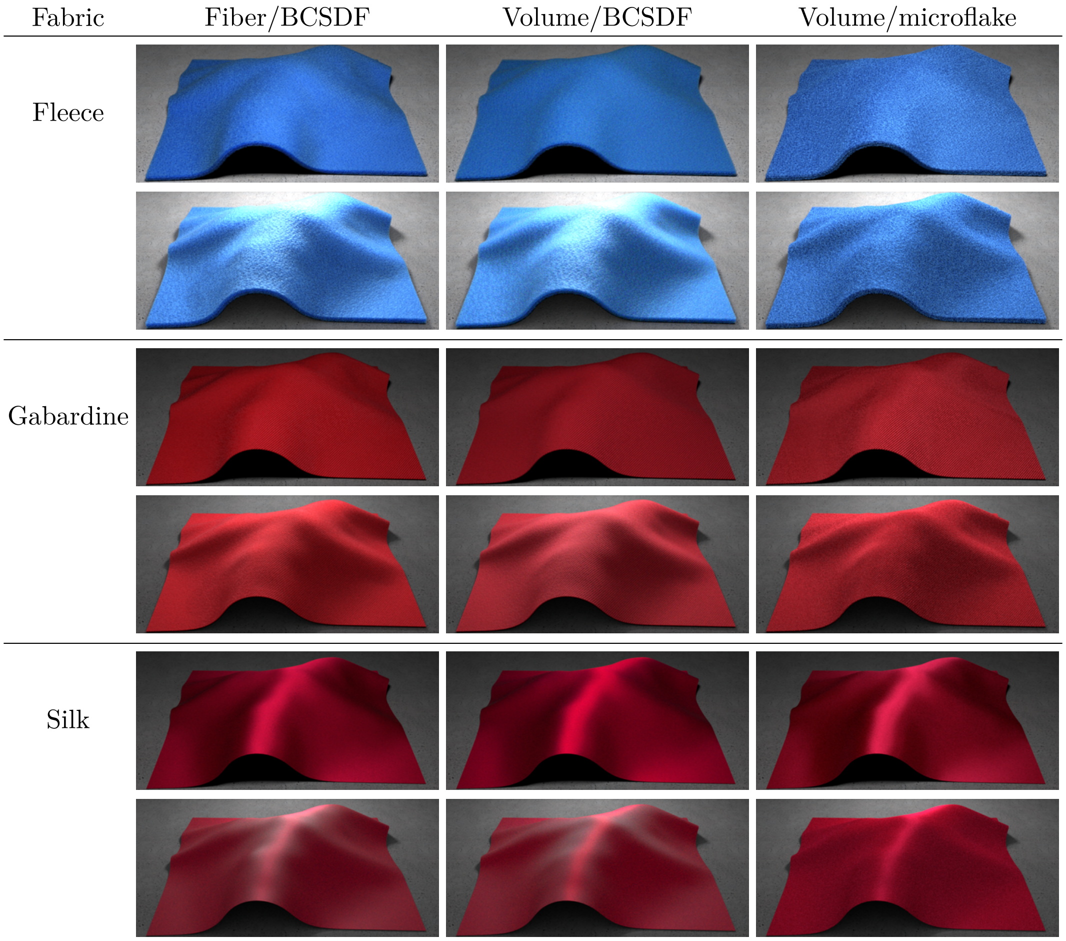
Plots in the same columns are to scale with one another.

Figure 5.23: Plots of per-column average pixel values of photographs and renderings in Figure 5.22 for Fleece, Gabardine, and Silk fabrics. Plots in the same columns are to scale with one another.



Plots in the same columns are to scale with one another.

Figure 5.24: Plots of per-column average pixel values of photographs and renderings in Figure 5.22 for Twill, Velvet, and Cotton.



In all the images, the fabrics are draped over a black mesh to reduce the effect of light passing through the fabrics themselves. In the first row of each fabric, the light is on the same side as the camera while, in the second row, the light is on the opposite side.

Figure 5.25: Renderings of Fleece, Gabardine, and Silk fabric in a simple draped configurations.

The results of our application to cloth appearance also provide crucial knowledge about which models work best, which can be leveraged by future work in this area. In the future our framework can be extended to handle more cases, for instance to reason about parameters, such as fiber radius or other geometric parameters, that cause discontinuous changes to path contributions.

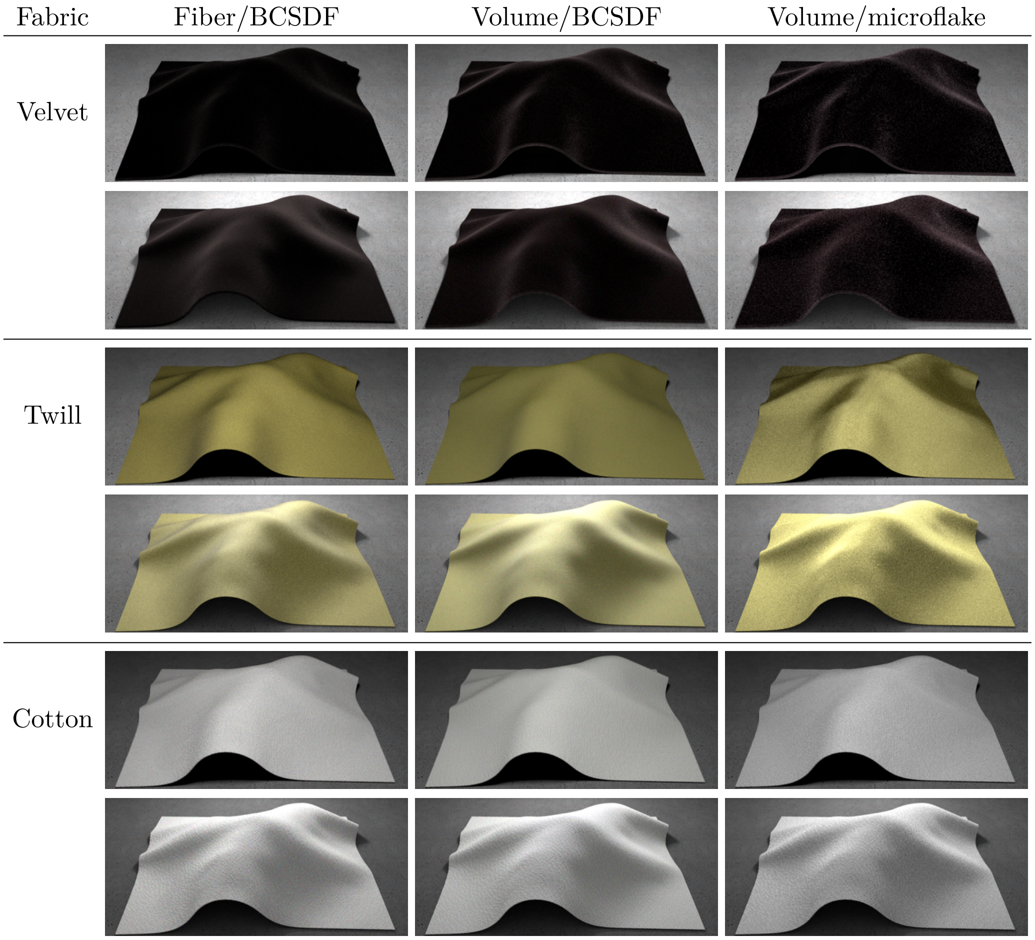


Figure 5.26: Renderings of Twill, Velvet, and Cotton in the same configurations as in Figure 5.25.

CHAPTER 6

FAST, APPROXIMATE RENDERING OF FABRICS

> もっと早く君のもとへ
> たとえ羽が千切れようとも
> 歪む世界走り抜けて
> 感覚のその向うへ
> 「真実」と加速してく
>
> KOTOKO, →*unfinished*→

We have discussed in the previous chapter a complete and practical system for constructing micro-appearance models that imitate the appearance of real fabrics. However, the models do not yet have widespread use due to the immense cost of simulating light transport with them. In this chapter, we present an algorithm for approximately rendering fabric micro-appearance models that significantly lowers rendering cost in practice.

This work originally appears in an unpublished paper "Fast Approximate Rendering of Fabric Micro-Appearance Models Under Directional and Spherical Gaussian Lights." It is joint with Rundong Wu, James Noeckel, Steve Marschner, and Kavita Bala [40].

6.1 Introduction

As we have discussed in Chapter 5, micro-appearance models have achieved high fidelity by coupling fiber-level light scattering models with fiber geometry acquired from micron-resolution CT images of real fabric samples. However, using path tracing is prohibitively slow because tracing rays through the complex microgeometry is costly, and long paths must be evaluated to capture multiple scattering inside the material. The process can take tens of core-hours per frame.

This chapter presents a fast, precomputation-based algorithm for approximately rendering fabric micro-appearance models under directional and/or spherical Gaussian lights, which can be used to approximate environment illumination. Because our algorithm traces only eye rays through microgeometry, it is very suitable for GPU implementation. In particular, our implementation is able to render high-resolution, supersampled images of micron-resolution fabrics in tens of seconds, using only a single commodity GPU.

Our approach is enabled by focusing on fabrics with regular structure and limiting our attention to the effects of multiple scattering *inside* the fabric volume, in contrast to global interreflections.

The key features of our algorithm are (1) it computes shadowing for single scattering using precomputed visibility; and (2) it approximates multiple scattering using precomputed transfer. Precomputation is done once per fabric type and weave pattern, then can be reused across changes to viewpoint, lighting, and cloth geometry.

We identify representations for visibility and multiple scattering transfer functions that perform well while preserving quality. For visibility, we use the spherical signed distance function for sharp spherical Gaussian lights, while for soft lights we introduce a new approximation of visibility as a sum of spherical Gaussians arranged over the sphere. For multiple scattering, we approximate the indirect radiance field by low order spherical harmonics.

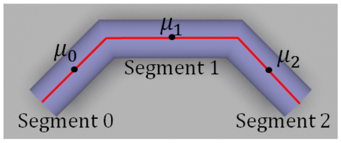
We demonstrate high fidelity results, individually rendered in under a minute, for fabrics ranging from regularly structured woven textiles to a knitted fleece with unorganized surface structure. Our approach makes micro-appearance models practical for applications that require quick turnaround, such as interactive textile design.

6.2 Background and Previous Work

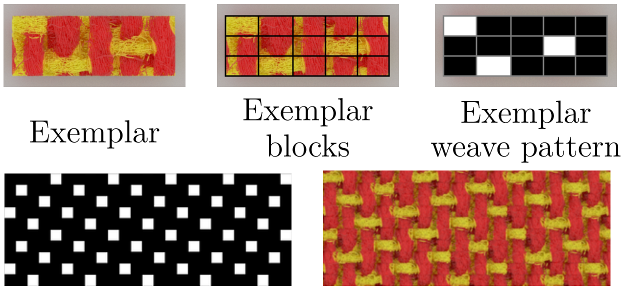
Fabric Modeling

We model fabrics with micro-appearance models whose components were discussed in Chapter 3. Our modeling process (Figure 6.1) starts with an *exemplar*: a small, rectangular piece of fabric modeled with explicit fibers. The exemplar is divided into *blocks*, and larger fabrics are synthesized by tiling instances of these blocks according to weave patterns, then deforming them into arbitrary shapes using shell mapping [110, 65].

We do precomputation and CPU rendering with a surface-based representation, in contrast to the volumetric representation of Zhao et al. however, when using a GPU, we rasterize the fibers into a volume to better match the strengths of that architecture.



A *fiber* is represented by a 3D polyline of multiple straight *segments*. The kth segment's midpoint is denoted by μ_k. Light scattering from fibers is modeled by the two-term BCSDF by Khungurn et al. [39].



Exemplar

Exemplar blocks

Exemplar weave pattern

Fabric weave pattern

Tiled fabric

Figure 6.1: The micro-appearance fabric model that is the target of our rendering algorithm.

Accelerated Rendering of Fiber Assemblies

A fiber assembly is challenging to render because its overall color arises from multiple scattering, requiring a path tracer to recurse to high depths [108]. Two-pass algorithms using photon maps [51] and spherical harmonics [52] have been used to accelerate multiple scattering. Modular flux transfer (MFT) [108] stores flux obtained from photon tracing in a spatial grid. After path tracing 2 to 6 bounces, it switches to a random walk on the transfer matrices before looking up the stored fluxes. While long paths are eliminated, the initial bounces can still take tens of minutes on multi-core machines. Moreover, photon tracing and recursive eye-ray tracing make MFT GPU-unfriendly.

Like dual scattering [116], our algorithm seeks to approximate multiple scattering in one pass without tracing rays other than eye rays and shadow rays. Originally, dual scattering could only handle directional lights. Subsequent work enabled rendering under environment lights [70] and interactive editing of BCSDF parameters [101]. While the algorithm is not well suited to our application, we similarly approximate multiple scattering with smooth functions defined locally.

Iwasaki et al. [31] presented an interactive algorithm for rendering fabrics modeled by the microcylinder model of Sadeghi et al. [72] under environment lights. However, the algorithm does not handle full multiple scattering; and, using essentially a BRDF model, it offers no geometric details.

Precomputed Radiance Transfer

Our algorithm is a PRT algorithm because it precomputes radiance transfer functions and uses them to bypass expensive path tracing at render time. A transfer function is represented as a transformation in some bases such as spherical harmonics [82], piecewise bilinear functions on the outgoing directions [42], wavelets [54, 94], and spherical Gaussians [88, 93]. Because a transfer function typically is precomputed for every mesh vertex in the scene, the whole collection is often too large to reside in GPU memory. As a result, researchers have proposed compression algorithms such as clustered principal component analysis (CPCA) [81] and clustered tensor approximation (CTA) [88]. For a more complete survey of PRT techniques, we refer the reader to the paper by Ramamoorthi [68].

Our algorithm differs from other standard PRT algorithms in two ways. First, standard PRT algorithms consider static scene geometry, but we precompute on a flat fabric that is then warped into the final geometry. In this way, ours is similar to in goals to the local, deformable PRT algorithm [83], but we do not use a special representation to make rotating the incoming light easier.

Second, we separate between single and multiple scattering components of scattered radiance while standard PRT algorithms either do not separate them or only deal with single scattering. As just discussed previously, separating single and multiple scattering is a widely used strategy when rendering fiber assemblies because it allows us to compute different components differently. Two-pass rendering algorithms [51, 52, 108] compute single scattering exactly with ray casing but approximate multiple scattering with photon mapping or low-frequency functions. On the other hand, we approximate single scattering with precomputed visibility functions (which will be discussed in a moment) and multiple scattering with transfer functions specialized to indirect illumination. The transfer functions are represented by a bidirectional spherical harmonics (SH) structure, similar to the interreflection transfer function proposed by Pan et al. [57].

Visibility Decomposition

When computing single scattering, we decompose visibility into *local* and *global* visibility. Local visibility involves occlusion by nearby fiber microgeometry and global visibility involves occlusion by faraway parts of the fabric or other objects in the scene. Only local visibility is precomputed, and we regard it as a univariate function of directions stored at each fiber segment.

Our decomposition is similar to the one used in the work of Schröder et al. [75], which is primarily concerned with representing fabrics with coarse volumes. Visibility around a point in a voxel is decomposed into the local visibility term, which represents occlusion by geometry inside the voxel, and the global visibility term, which represents occlusion by geometry outside the voxel. The local visibility term is a bidirectional function called the *bidirectional visibility distribution function* (BVDF), which can be thought of as an average of visibility functions at individual fiber segments in the voxel. While the BVDF would be the same as our visibility function when the voxels become fine enough, the BVDF's representation in Schröder et al.'s paper only works well with lights that have delta distributions (i.e., directional lights or point lights). We, on the other hand, introduce representations for local visibility that also work with spherical Gaussian lights.

Spherical Harmonics

We denote a real spherical harmonic (SH) basis function with $Y_j(\omega)$ where $\omega \in S^2$, and j serves as a single index. SH functions are classified into orders with $2k+1$ functions having order k. Thus, a representation using SH of order up to k has $(k+1)^2$ coefficients.

Spherical Gaussians

A *spherical Gaussian* (SG) with *axis* $\xi \in S^2$ and *sharpness* $\lambda \in \mathbb{R}$ is given by $G(\omega; \xi, \lambda) = \exp(\lambda(\omega \cdot \xi - 1))$. The *mass function* $M(\theta; \lambda)$ is the integral of an SG with sharpness λ over the spherical cap $\mathcal{C}(\theta, \xi) = \{\omega : \omega \cdot \xi \geq \cos\theta\}$:

$$M(\theta; \lambda) = \int_{\mathcal{C}(\theta,\xi)} G(\omega; \xi, \lambda) \, \mathrm{d}\omega = \frac{2\pi}{\lambda}(1 - e^{\lambda(\cos\theta - 1)}).$$

We call the mass function evaluated at $\theta = \pi$ the *mass* $M(\lambda)$ of the SG, and it is equal to $2\pi(1 - e^{-2\lambda})/\lambda$. Two properties of SGs make them attractive for shading calculations: (1) they can be easily rotated, and (2) the product of two SGs is another SG.

Wang et al. used SGs to represent glossy BSDFs and environment illumination at all frequency scales in static scenes [93]. Later works adapt SG-based representations to dynamic scenes [30], improve shadowing [112], and incorporate anisotropy [102]. Our work similarly represents environment illumination with SGs. Moreover, we introduce sums of SGs as a new visibility representation.

6.3 Overview

The benefits of micro-appearance models depend on simulating multiple scattering, which is responsible for much of the appearance and overall color of fabrics. Doing so with path tracing is expensive. However, since multiple scattering's effects are quite smooth, we aggressively approximate them by precomputing the indirect radiance distribution.

Although global interreflection between distant surfaces is important, this paper focuses on local multiple scattering *within* the fabric volume. We also assume both the camera and the light sources are above the fabric, ignoring cases involving the side and bottom faces. Our algorithm could be adapted to handle these cases.

Figure 6.2 gives an overview of our algorithm. We rely on two main types of precomputed functions to accelerate rendering:

- The *segment visibility function* (SVF) encodes whether a ray starting from a given fiber segment escapes the fabric volume in any given direction.
- The *incoming indirect radiance transfer function* (IIRTF) encapsulates how indirect illumination to a point in the fabric volume depends on the illumination to the fabric as a whole.

While there is a SVF associated with each fiber segment, we precompute the average of the IIRTF over *cells*—further subdivisions of exemplar blocks that may contain multiple fiber segments.

At rendering time, we must first intersect eye rays with the fabric. On the CPU, this involves casting rays through shell maps [65]. While an equivalent

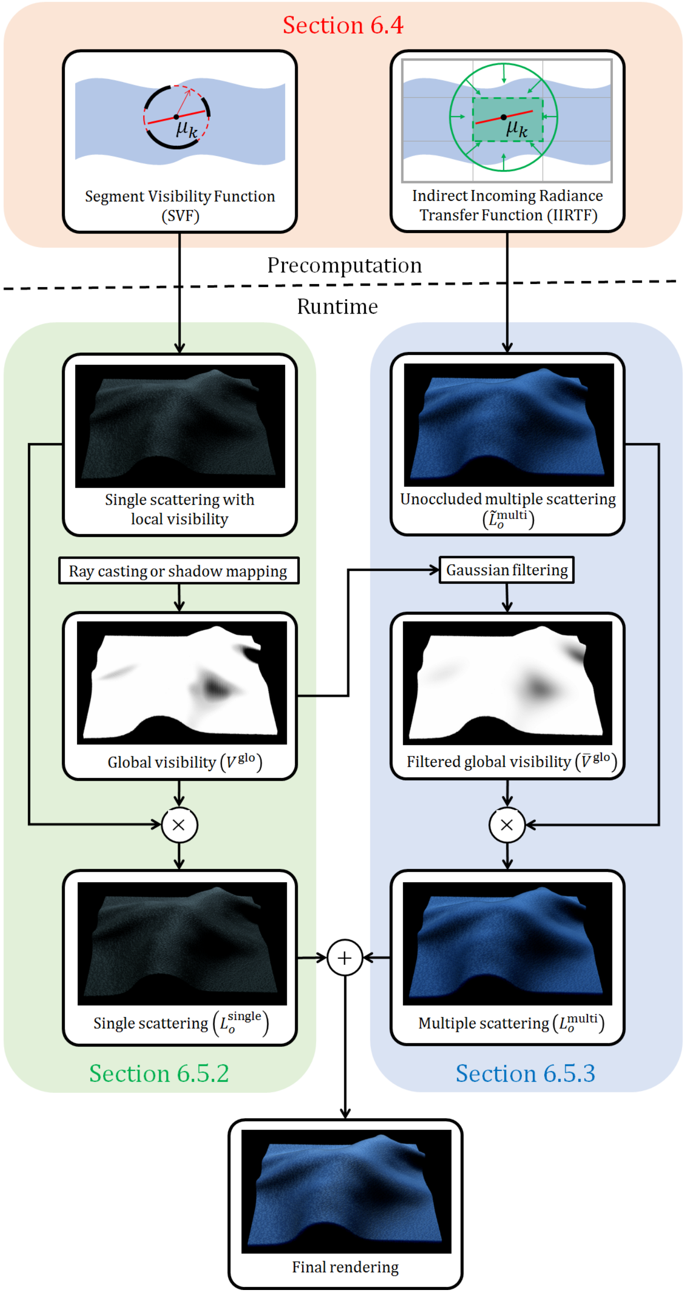

Section 6.4
μ_k
Segment Visibility Function (SVF)
μ_k
Indirect Incoming Radiance Transfer Function (IIRTF)
Precomputation
Runtime
Single scattering with local visibility
Unoccluded multiple scattering ($\tilde{L}_o^{\text{multi}}$)
Ray casting or shadow mapping
Gaussian filtering
Global visibility (V^{glo})
Filtered global visibility ($\bar{V}^{\text{glo}}$)
×
×
+
Single scattering (L_o^{single})
Multiple scattering (L_o^{multi})
Section 6.5.2
Section 6.5.3
Final rendering

Figure 6.2: An overview of our approximate rendering algorithm.

process exists for the GPU [34], Section 6.5.1 discusses our simpler volume ray casting approach. After identifying the hit point, we compute the out-scattered light, splitting it into single and multiple scattering components.

For single scattering, we compute the triple product integral between the SVF, the incoming radiance from the light source, and the BCSDF. This process is nontrivial for spherical Gaussian lights, and we show that our new way of representing the SVF yields good approximation over the whole range of SG sharpness. The integral, however, only accounts for occlusion by nearby fiber microgeometry, so we incorporate macro-scale occlusion by other geometry using ray casting or shadow mapping against the fabric's shell. The single scattering process is discussed in Section 6.5.2.

For multiple scattering, we evaluate the IIRTF with the light source as an argument and convolve the result with the BCSDF to obtain the multiple scattering response to illumination unoccluded by macroscale geometry. To incorporate shadowing and to simulate subsurface scattering effects, we filter the shadow signal used in the single scattering computation with a spatial kernel, and scale the multiple scattering response by the result. Section 6.5.3 covers multiple scattering computation.

6.4 Precomputation and Parameters

The first phase of our algorithm is a series of precomputations carried out once for each type of fabric (that is, once per exemplar and weave pattern) on a large, flat section of fabric synthesized in the same way as the material to be rendered. The main results of these precomputations are the segment visibility function (per fiber segment) and the indirect radiance transfer function (per fabric grid cell). We also precompute tables of projections of the BCSDF and the SG lights into the SH basis. At the end of this section, we list all parameters of our algorithm.

6.4.1 Segment Visibility Function

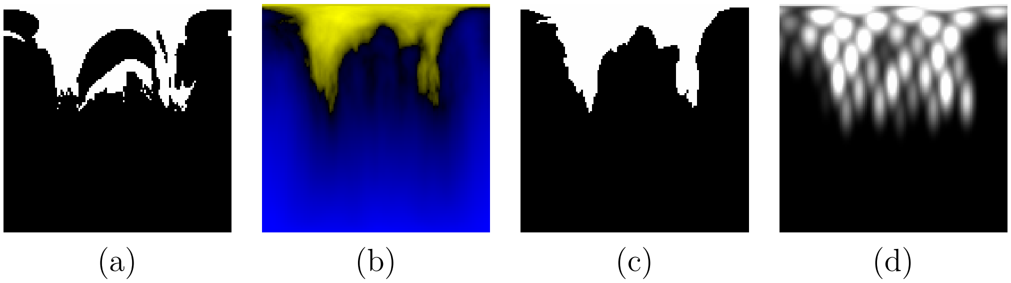
Light occlusion by local fiber microgeometry is approximated using the precomputed segment visibility function $V_k : \mathbb{S}^2 \to \mathbb{R}$ associated with each fiber segment k. $V_k(\omega) = 1$ if the infinite ray originating from μ_k in direction ω does not hit any fiber other than the one segment k is a part of, and $V_k(\omega) = 0$ otherwise.

We use two representations for V_k and choose between them based on whether the fabric is being shaded under a high or low frequency light source. Under high frequency lights (directional lights and SG lights with $\lambda \geq 100$), we use the *spherical signed distance function* (SSDF) [93], denoted by $d_k(\omega)$. Under low frequency lights (SG lights with $\lambda \leq 200$), we represent visibility with a weighted sum of (normalized) SGs:

$$V_k(\omega) \approx V_k^{\mathrm{ssg}}(\omega) = \sum_{q=1}^{Q} w_{k,q} \frac{G(\omega; \xi_{k,q}, \lambda_{\mathrm{svf}})}{M(\lambda_{\mathrm{svf}})}.$$

where the axes $\xi_{k,q}$ are derived from the spherical Fibonacci point sets [48], and all the SGs in the representation has the same sharpness λ_{svf}. Figure 6.3 contains images of the SVF of a fiber segment and its representations.

SSDF implementation. To compute the SSDF of Segment k, we locate the copy of the exemplar block containing it that is the nearest to the center of the flat fabric, so as to avoid any boundary effects. With ray casting, we then render a 128×128 binary visibility image over the (θ, ϕ)-parameterization of the sphere and compute the SSDF from the resulting image by finding, for each pixel, the closest pixel having the opposite value. The whole collection of SSDFs is compressed with PCA using 48 principal components.



In Mercator projection: (a) visibility as a 128 × 128 binary image; (b) the PCA-compressed SSDF (yellow = positive, blue = negative); (c) the sign of the SSDF as a binary image, and (d) the sum of SGs representation.

Figure 6.3: The segment visibility function of a segment in the Silk fabric and its approximate representations.

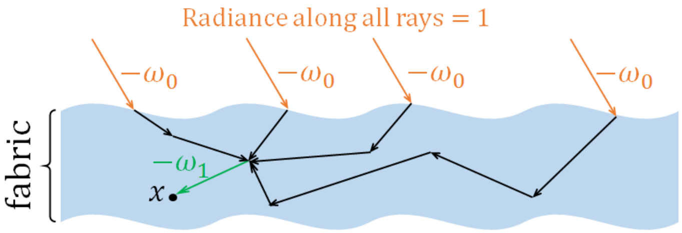
Sum of SGs implementation. The number of SGs, Q, is chosen to be 48, the same as the number of PCA components of the SSDFs. As will be discussed in Section 6.5.2, the sharpness parameter, λ_{svf}, should be at least 100, and we use $\lambda_{\text{svf}} = 128$ for all the renderings in this paper. The axes $\xi_{k,q}$ should ideally be uniformly distributed over the north hemisphere. For this reason, we derive them from the spherical Fibonacci point sets:

$$\begin{aligned} \xi_{k,q} &= R_k \xi'_q \\ \xi'_q &= (\sqrt{1 - z_q^2}\cos(\phi_q), \sqrt{1 - z_q^2}\sin(\phi_q), z_q) \\ z_q &= 1 - (q-1)/Q \\ \phi_q &= 4\pi q/(1 + \sqrt{5}) \end{aligned}$$

where R_k is a random rotation per segment around the z-axis to mask aliasing artifacts that might arise from using the same set of axes for all segments. To compute the weights $w_{k,q}$, we utilize the visibility image used to compute the SSDFs and set $w_{k,q}$ to the total solid angle of all the visible pixels closer to $\xi_{k,q}$ than any other axis.

6.4.2 Indirect Incoming Radiance Transfer Function

We use the IIRTF to convert the raw energy from the light sources to the incoming radiance at a point of interest *after* the light scatters one or more times from fibers. We get the radiance due to local multiple scattering—the quantity we want to compute—by convolving this incoming radiance with the BCSDF. Formally, the IIRTF $T(\omega_0, \omega_1, x)$ equals the incoming radiance arriving at point x in direction $-\omega_1$ through paths that (1) originate from a directional light source that emits radiance of magnitude 1 in direction $-\omega_0$ and (2) contain at least one point on a fabric fiber.



To make sure an IIRTF accurately describes the illumination in all instances of its cell in the fabric, we need a *regularity* assumption: for all instances of a given cell, the weave pattern in neighboring cells should be the same. This is is obviously satisfied when the weave pattern across the whole fabric is the same as in the exemplar, but the method can also be applied to fabrics with varying weave patterns as long as local neighborhoods can be matched. As a result, patterned fabrics can be supported with more precomputation, and handling them is left as future work.

Working in the spherical harmonics domain, we represent the IIRTF—which transforms the SH expansion of a scalar function on $\mathbb{S}^2$ to the expansion of another scalar function on $\mathbb{S}^2$—as a matrix A_x of entries $a_x[\cdot,\cdot]$ where

$$a_x[j_0, j_1] = \int_{\mathbb{S}^2}\int_{\mathbb{S}^2} T(\omega_0, \omega_1, x) Y_{j_0}(\omega_0) Y_{j_1}(\omega_1)\ \mathrm{d}\omega_0 \mathrm{d}\omega_1.$$

Computation. Since the representation is space-consuming ($O(L^4)$ for SH of order L), we cannot afford to store one IIRTF for each fiber. Instead, we divide exemplar blocks into *cells* of equal size and average the IIRTF over each cell. The per-cell IIRTF is the average of the IIRTF at all segment midpoints inside the cell:

$$T(\omega_0, \omega_1, C) = \frac{1}{|\mathscr{C}|}\sum_{k \in \mathscr{C}} T(\omega_0, \omega_1, \mu_k).$$

Here, C is a cell, and $\mathscr{C}$ is the set of segments with midpoints in C. The coefficient $a_C[j_0, j_1]$ of the cell's transfer function can be estimated by sampling k, ω_0, and ω_1 independently and computing:

$$\frac{1}{|\mathscr{C}|}\frac{T(\omega_0, \omega_1, \mu_k) Y_{j_0}(\omega_0) Y_{j_1}(\omega_1)}{p(k) p(\omega_0) p(\omega_1)}.$$

We sample all ks with equal probability ($p(k) = 1/|\mathscr{C}|$), ω_0 uniformly from the upper hemisphere ($p(\omega_0) = 1/(2\pi)$), and ω_1 uniformly from the whole sphere ($p(\omega_1) = 1/(4\pi)$). The value $T(\omega_0, \omega_1, \mu_k)$ is estimated by tracing $50,000$ paths per cell.

Choosing cell dimensions. The cell size is an important parameter to our algorithm. Larger cells require less storage but yield lower quality renderings. As we will see in Section 6.6, too coarse IIRTFs lead to non-smooth and dark renderings.

We found that setting the side lengths of the cells as close as possible to the mean free path ℓ of a path traced through the fabric yielded consistently good results, with only a minor drop in visual quality with cells of size up to 2ℓ. The supplementary material discusses the specifics of how we estimate ℓ.

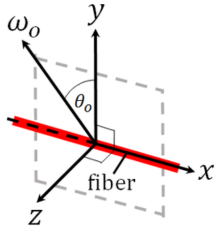
We estimate ℓ by averaging the distances between adjacent vertices from the 5th bounce onward when a flat piece of fabric is rendered under several viewing and illumination conditions. Employing the 16 configurations used for parameter fitting in Chapter 5, we set the image resolution and number of samples per pixel so that we collected about $10,000$ to $20,000$ vertex positions from the 5th bounces for each configuration.

Ideally, the fibers should have the BCSDFs that are used in the final rendered images. However, we found that the ℓ-estimates produced with diffuse BCSDFs are generally larger than the ground truth (thus resulting in smaller IIRTF data) and already yield good rendering results, we simply used them when estimating ℓ.

6.4.3 Function Expansions into the SH Basis

We must compute the convolution between the BCSDF, the light source's radiance distribution, and the IIRTF, the last of which is expressed in the SH basis. This computation can be accelerated by expanding the first two into the same basis.

BCSDF. This expansion is defined in the coordinate system where the shading integral is performed: the fiber-based coordinate system used by Marschner et al. [49]. The x-axis must coincide with the fiber segment's direction, but we are free to choose the y-axis so that the outgoing direction ω_o is in the xy-plane. Let us call this coordinate system the ω_o-space.



We precompute a table $\mathcal{C}_S[\theta_o, j]$ where:

$$\mathcal{C}_S[\theta_o, j] = \int_{\mathbb{S}^2} Y_j(\omega_i) S(\omega_i, \omega_o) \cos\theta_i \; \mathrm{d}\omega_i$$

where S is the BCSDF in Chapter 5. We use 512 equally-spaced values of θ_o from $[-\pi/2, \pi/2]$. For each value of θ_o, we compute the expansion up to the SH order of the IIRTF.

Spherical Gaussians. One of our goals is to render fabrics with local multiple scattering under SG lights, so we also express SGs in the SH basis too. In particular, we precompute a table

$$\mathcal{C}_G[\lambda, j] = \int_{\mathbb{S}^2} \frac{G(\omega; (0,0,1), \lambda)}{M(\lambda)} Y_j(\omega) \; \mathrm{d}\omega,$$

which stores the SH coefficients of normalized SGs with various sharpness aligned with the z-axis. We store only the coefficients of the zonal harmonics because all others are zero. Appendix C.1 discusses how we choose the λ values and how we interpolate the entries.

6.4.4 Parameters

The parameters of our algorithm include the number of PCA components used to compress the SSDFs, the number and sharpness of SGs in the sum-of-SG visibility representation, the size of the fabric grid cells, and the SH order used for the IIRTF.

There is another parameter, σ_{glo}, which is the standard deviation of the 2D Gaussian kernel used to approximate the effect of occlusion on local multiple scattering. Its role is discussed in Section 6.5.3.

6.5 Rendering Algorithm

We now describe how to use the precomputed data to render fabrics. Suppose that the eye ray is along the direction $-\omega_o$. We first intersect the ray with the fabric (Section 6.5.1), giving the hit point x. We then compute the radiance leaving x in direction ω_o, $L_o(x, \omega_o) = L_o^{\mathrm{single}}(x, \omega_o) + L_o^{\mathrm{multi}}(x, \omega_o)$, in separate processes for single (Section 6.5.2) and multiple (Section 6.5.3) scattering.

6.5.1 Eye Ray–Fabric Intersection

On the CPU, primary visibility can be efficiently computed by tracing rays through the shell map and intersecting fibers stored in a spatial hierarchy. On the GPU, the more regular memory access of volume ray casting makes it a better approach, so we rasterize the fabric exemplar into a 3D volume. Each non-empty voxel stores the ID of the fiber segment whose midpoint is nearest to its center. The segment's direction and the ID of its BCSDF are stored in a separate texture indexed by segment ID. This volume is partitioned into blocks, tiled, and shell mapped as described in Section 6.2.

To render a shell-mapped fabric volume, we rasterize the front facing triangles of all tetrahedra in the shell. For each resulting fragment, we transform the eye ray into the flat fabric's space and sample the volume at a fixed number of linearly spaced points along the relevant ray segment to find the first non-empty voxel. (The number of points used can be found in Table 6.1.) If no such voxel is found, we discard the fragment. Otherwise, we save information about the hit point, including its shell texture coordinate and the fiber segment's ID, to a G-buffer for deferred shading.

6.5.2 Single Scattering

Having identified the hit point x on Fiber k, we are now ready to compute the out-scattered light from the fabric. This section describes its single scattering component. We first discuss the directional light case and then continue with the SG light case.

Directional Light

Let ω_d denote the direction toward the light source, and let us say that the radiance along ω_d is 1. The outgoing radiance due to single scattering is given by

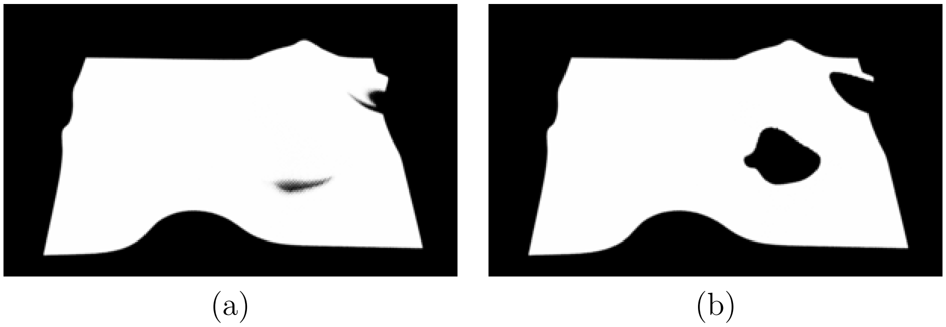
$$L_o^{\text{single}}(x, \omega_o) = V(x, \omega_d) S(\omega_d, \omega_o) \cos \theta_d$$

where S is the BCSDF, and V is the visibility function. Evaluating V by tracing shadow rays is expensive because each ray has to be traced through many fibers. To avoid this, we split V into the *local visibility term*, which deals with occlusion by nearby fiber microgeometry, and the *global visibility term*, which deals with occlusion by macroscopic objects in the scene: $V(x, \omega_d) = V^{\text{loc}}(x, \omega_d) V^{\text{glo}}(x, \omega_d)$.

The local visibility term is approximated with the precomputed SVF: $V^{\text{loc}}(x, \omega_d) \approx V_k(\omega_d)$. In this case, we use the sign of the SSDF (Figure 6.3(c)) to determine visibility.

The global visibility term $V^{\text{glo}}(x, \omega_d)$ is approximated by tracing a shadow ray in direction ω_d, intersecting only against the fabric shell and other macroscopic objects in the scene.

The shadow ray starts at $\underline{x}$, the point on the top surface of the shell that is directly above the fiber hit point x. This choice enables computing global visibility

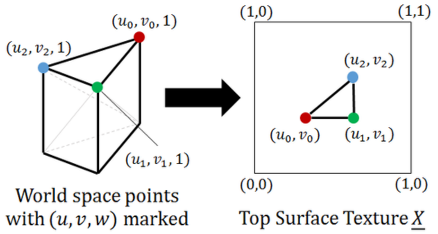


(a) uses the hit point x (and skips the first intersection with the shell), and (b) uses the shell's top surface point that is directly above x.

Figure 6.4: Effects of the choice of shadow ray origin.

on the GPU using a shadow map. It also prevents unintuitive hard shadows that are the result of tracing rays from x itself (See Figure 6.4). Our GPU implementation uses percentage-closer filtering (PCF) for shadow map anti-aliasing [69], but many other more sophisticated techniques are available for this purpose.

On the CPU, $\underline{\mathrm{x}}$ can be computed by a shell map lookup, which involves traversing the bounding volume hierarchy of the shell tetrahedra to identify the tetrahedron that contains the point with the given shell texture coordinate. On the GPU, we render a texture $\underline{\mathrm{X}}[u, v]$ that maps the 2D shell texture coordinate (i.e. ignoring the depth component) to the world position of the top surface of the shell map. Given a hit point x, we can recover its shell texture coordinate (u, v, w) from the G-buffer. We then look up $\underline{\mathrm{X}}[u, v]$ to determine $\underline{x}$.



Generalization to other types of light sources. On the CPU, the single scattering computation can be easily extended to any type of environment light source that can be efficiently sampled. We first sample the incoming direction ω_d and the radiance $\mathcal{L}(\omega_d)$ along it. We then compute the single scattering response and scale it by $\mathcal{L}(\omega_d)/p(\omega_d)$ where $p(\omega_d)$ is the probability of sampling ω_d. The definition of visibility and incoming light may be changed to accommodate point lights and, by extension, area light sources.

Spherical Gaussian Light

While random sampling can convert any arbitrary environment light—including any SG light—into a directional light, it does not work well on the GPU because of the lack of native support for tracing arbitrary shadow rays. Moreover, the approach will yield noisy renderings, particularly when the support of the SG light is large. Our goal is to design a GPU-friendly algorithm that shades micro-appearance models under SG lights *without noise.* We achieve this by exploiting the structure of both the precomputed visibility and the BCSDF.

Let the scene be illuminated by a single normalized SG light $G(\omega; \xi, \lambda)/M(\lambda)$. The single scattering component is given by:

$$\int_{\mathbb{S}^2} V^{\mathrm{glo}}(\omega_i) V^{\mathrm{loc}}(\omega_i) \frac{G(\omega_i; \xi, \lambda)}{M(\lambda)} S(\omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i. \tag{6.1}$$

We will first describe how we approximate global visibility under SG lights to simplify the problem. We will then discuss how to compute single scattering while taking into account the complex occlusion by nearby fibers.

Global visibility under SG lights. We recognize that accurate shadowing under area lights, including SG lights, is a challenging problem that currently has no solution without significant compromises. Approaches based on SSDFs [93, 112] require expensive precomputation of the SSDFs of macroscale geometry, making them impractical in rendering cloth animation. The integral SG approach [30] requires rasterizing thin shell meshes at each fragment and so does not scale well in our setting. As a result, we settle on plausible shadow computation through percentage-closer soft shadow (PCSS) mapping [21], which scalably handles complex, changing geometry while producing visually pleasing shadows. Recent techniques [3, 105, 78] speed up PCSS by enabling prefiltering. However, we only use the original PCSS in our implementation. One drawback of PCSS is the need to determine parameters such as the sizes and positions of the light sources. We picked these parameters manually.

Specifically, we compute the global visibility term $V^{\mathrm{glo}}(x)$ by PCSS. The global visibility term is then used to scale the power of the SG light down without changing the distribution. Namely, (6.1) becomes:

$$\frac{V^{\mathrm{glo}}(x)}{M(\lambda)} \int_{\mathbb{S}^2} V^{\mathrm{loc}}(x, \omega_i) G(\omega_i; \xi, \lambda) S(\omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i.$$

Our problem thus reduces to the triple product integral between the local visibility, an SG, and the BCSDF.

We first describe the solution to the above problem with an approximation to the integral when the SG light is unoccluded and sharp ($\lambda \geq 100$). We then discuss how to use our two SVF representations to incorporate local visibility into the integral.

Unoccluded, sharp SG lights. The BCSDF we use is a sum of two terms: $S(\omega_i, \omega_o) = S_R(\omega_i, \omega_o) + S_{TT}(\omega_i, \omega_o)$ [39]. We first rewrite its convolution with the (unnormalized) SG as:

$$
\begin{aligned}
&\int_{\mathbb{S}^2} G(\omega_i;\xi,\lambda)S(\omega_i,\omega_o)\cos\theta_i \ \mathrm{d}\omega_i \\
&= \int_{-\pi/2}^{\pi/2} g\left(\theta_i;-\theta_o,\frac{1}{2\beta_R^2}\right) g^c(\theta_i;\theta',\lambda) B_R(\theta_i,\lambda\cos\theta') \ \mathrm{d}\theta_i \\
&+ \int_{-\pi/2}^{\pi/2} g\left(\theta_i;-\theta_o,\frac{1}{2\beta_{TT}^2}\right) g^c(\theta_i;\theta',\lambda) B_{TT}(\theta_i,\lambda\cos\theta',\phi') \ \mathrm{d}\theta_i
\end{aligned}
$$

where $g(x;\mu,\lambda) = \exp(-\lambda(x-\mu)^2)$ is the *ordinary Gaussian* function, $g^c(x;\mu,\lambda) = \exp(\lambda(\cos(x-\mu)-1))$ is the *circular Gaussian* function, and β_R and β_{TT} are the standard deviation of the Gaussians in the longitudinal scattering functions of S_R and S_{TT}, respectively. The exact forms of B_R and B_{TT} are given in the Appendix C.2.

When $\lambda \geq 10$, the circular Gaussian $g^c(\theta_i,\theta',\lambda)$ is well approximated by the ordinary Gaussian $g(\theta_i,\theta',\lambda/2)$. Hence, the product between the ordinary Gaussian and the circular Gaussian in the integrand may be approximated by a single ordinary Gaussian:

$$
g\left(\theta_i;-\theta_o,\frac{1}{2\beta_R^2}\right) g^c(\theta_i;\theta',\lambda) \approx a_R \ g(\theta_i;\theta_R,\lambda_R)
$$

where $\lambda_R = \beta_R^{-2}/2 + \lambda/2$,

$$
\theta_R = \frac{-\beta_R^2\theta_o + \lambda\theta'}{\beta_R^{-2}+\lambda}, \qquad a_R = \exp\left(-\frac{\beta_R^{-2}\lambda}{\beta_R^{-2}+\lambda}\frac{(\theta_o+\theta')^2}{2}\right).
$$

Similar equations exist for the TT term.

When this single ordinary Gaussian is sharp, i.e. when $\lambda \geq 100$, the integral of the product between the Gaussian and the B functions is well approximated by factoring B out of the integral:

$$\begin{aligned}
&\int_{-\pi/2}^{\pi/2} g(\theta_i;\theta_R,\lambda_R) B_R(\theta_i,\lambda\cos\theta')\ \mathrm{d}\theta_i \\
&\approx a_R B_R(\theta_R,\lambda\cos\theta') \int_{-\pi/2}^{\pi/2} g(\theta_i;\theta_R,\lambda_R)\ \mathrm{d}\theta_i \\
&= a_R \frac{B_R(\theta_R,\lambda\cos\theta')}{2}\sqrt{\frac{\pi}{\lambda_R}}\Big[\mathrm{erf}(\sqrt{\lambda_R}(\theta_i-\theta_R))\Big]_{-\pi/2}^{\pi/2}.
\end{aligned}$$

In summary, we approximate the convolution between the unoccluded SG and the BCSDF as:

$$\begin{aligned}
&\int_{\mathbb{S}^2} G(\omega_i;\xi,\lambda) S(\omega_i,\omega_o)\cos\theta_i\ \mathrm{d}\omega_i \\
&\approx a_R \frac{B_R(\theta_R,\lambda\cos\theta')}{2}\sqrt{\frac{\pi}{\lambda_R}}\Big[\mathrm{erf}(\sqrt{\lambda_R}(\theta_i-\theta_R))\Big]_{-\pi/2}^{\pi/2} \\
&+ a_{TT} \frac{B_{TT}(\theta_{TT},\lambda\cos\theta',\phi')}{2}\sqrt{\frac{\pi}{\lambda_{TT}}}\Big[\mathrm{erf}(\sqrt{\lambda_{TT}}(\theta_i-\theta_{TT}))\Big]_{-\pi/2}^{\pi/2}.
\end{aligned} \tag{6.2}$$

Let us denote the RHS of (6.2) with $\Gamma(\xi,\lambda)$.

Local visibility under soft SG lights. The approximation Γ only works with unoccluded and sharp SG lights. To shade a soft SG light occluded by nearby fiber segments, we use the fact that the product of two SGs reduces to an SG, which implies that the sum-of-SGs representation of the SVF can "break" the soft SG light into many sharp ones:

$$\begin{aligned}
V^{\mathrm{loc}}(x,\omega_i) G(\omega_i;\xi,\lambda) &\approx V_k^{\mathrm{ssg}}(x,\omega_i) G(\omega_i;\xi,\lambda) \\
&= \sum_{q=1}^{Q} \frac{w_q}{M(\lambda_{\mathrm{svf}})} G(\omega_i;\xi_{k,q},\lambda_{\mathrm{svf}}) G(\omega_i;\xi,\lambda) \\
&= \sum_{q=1}^{Q} \frac{w_q}{M(\lambda_{\mathrm{svf}})} \frac{G(\omega_i;\widehat{\xi_{k,q}^{\mathrm{sum}}},\|\xi_{k,q}^{\mathrm{sum}}\|)}{\exp(\lambda_{\mathrm{svf}}+\lambda-\|\xi_{k,q}^{\mathrm{sum}}\|)}
\end{aligned}$$

where $\xi_{k,q}^{\mathrm{sum}} = \lambda_{\mathrm{svf}}\xi_{k,q} + \lambda\xi$, and $\widehat{\xi_{k,q}^{\mathrm{sum}}} = \xi_{k,q}^{\mathrm{sum}}/\|\xi_{k,q}^{\mathrm{sum}}\|$. So,

$$\int_{\mathbb{S}^2} V_k^{\mathrm{loc}}(x,\omega_i) G(\omega_i;\xi,\lambda) S(\omega_i,\omega_o)\cos\theta_i\ \mathrm{d}\omega_i \approx \sum_{q=1}^{Q} \frac{w_q}{M(\lambda_{\mathrm{svf}})} \frac{\Gamma(\widehat{\xi_{k,q}^{\mathrm{sum}}},\|\xi_{k,q}^{\mathrm{sum}}\|)}{\exp(\lambda_{\mathrm{svf}}+\lambda-\|\xi_{k,q}^{\mathrm{sum}}\|)}$$

A remarkable feature of the above expression is that, when Γ is inaccurate—that is, when $\|\xi^{\mathrm{sum}}_{k,q}\| < 100$—it is divided by $\exp(\lambda_{\mathrm{svf}} + \lambda - \|\xi^{\mathrm{sum}}_{k,q}\|)$, which is large given than λ_{svf} is suitably large. With our choice of $\lambda_{\mathrm{svf}} = 128$, we have that

$$\lambda_{\mathrm{svf}} + \lambda - \|\xi^{\mathrm{sum}}_{k,q}\| > 128 + \lambda - 100 \geq 28.$$

So, the erroneous approximation is scaled down by a factor of at least $e^{-28} \approx 7 \times 10^{-13}$, meaning that we aggressively suppress cases where Γ does not work well.

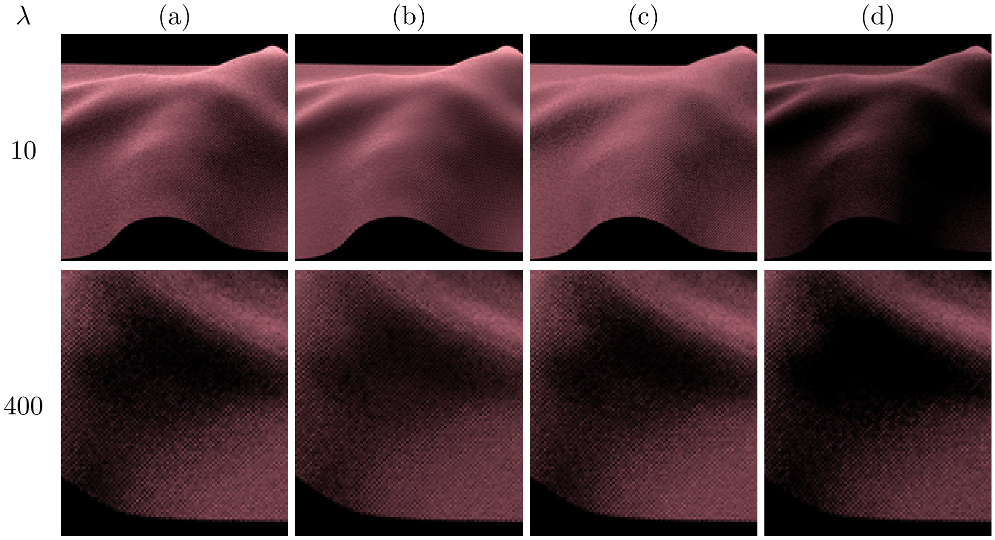
A problem with the sum-of-SGs visibility representation is that it can produce renderings that are too bright when the SG light is sharp, especially in shadowed areas. The reason is that the representation is inherently soft: the Gaussians in the representation is a continuous function. As such, they yield positive function values at directions that are supposed to be occluded in the ground truth visibility function. Figure 6.5 illustrates this problem.

Local visibility under sharp SG lights. To combat the above problem, we gradually fall back to using the SSDFs to represent local visibility as λ increases from 100. Since the SG is sharp, we factor out the visibility term into a new one and approximate the double product integral with Γ as:

$$\begin{aligned}
&\int_{\mathbb{S}^2} V^{\mathrm{loc}}(x, \omega_i) G(\omega_i; \xi, \lambda) S(\omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i \\
&\approx \widetilde{V}^{\mathrm{loc}}(x, \xi) \int_{\mathbb{S}^2} G(\omega_i; \xi, \lambda) S(\omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i \\
&\approx \widetilde{V}^{\mathrm{loc}}(x, \xi) \Gamma(\xi, \lambda).
\end{aligned}$$

The new visibility term $\widetilde{V}^{\mathrm{loc}}(x, \xi)$ is similar to the SSDF calculation for the directional case, but with an error function serving as a smoothed step function:

$$\widetilde{V}^{\mathrm{loc}}(x, \xi) = \frac{1}{2}\left[\mathrm{erf}\left(\sqrt{\frac{\lambda}{2}} d_k(\xi)\right) + 1\right]$$



All images only contain single scattering and do not take into account global visibility. Exposures of all images are set to 8 to make differences clear. Images in Column (a) are reference renderings produced by the CPU version of our algorithm, which accurately convolves the light using Monte Carlo integration. Column (b) uses the sum-of-SGs visibility representation. It works well when $\lambda = 10$ but yields shadowed areas that are too bright when $\lambda = 400$. Column (c) uses the error-function-based visibility term coupled with the sharp SG light approximation, both discussed in Section 6.5.2. The scheme yields wrong results when $\lambda = 10$ but matches the reference well when $\lambda = 400$. Column (d) uses the SSDF coupled with Wang et al.'s visibility term. It yields renderings darker than all other approaches at both λ values. Our conclusion is to use (b) when λ is low and (c) when λ is high.

Figure 6.5: Problematic regions of various single scattering approximations.

where d_k is the SSDF of the fiber segment. The standard deviation of the error function is $\sqrt{2/\lambda}$, motivated by the convolution of a sharp SG with a straight edge visibility function.

We note that the above visibility term is different from the one used by Wang et al. [93], which is equal to $M(\lambda, d_k(\xi))/M(\lambda)$ if $\theta_d(\xi) > 0$ and is equal to 0 otherwise. We empirically found that this term yielded fabric renderings that are too dark. See Figure 6.5.

To make the transition between the sum-of-SGs and the SSDF smooth, we linearly interpolate the single scattering results of the two schemes when $\lambda \in$

$[100, 200]$, using λ itself as the interpolation parameter.

According to the measurements available in the supplementary material, using the SSDF with the error-function-based visibility term can make the algorithm 1% to 3% slower than the directional light case on the GPU. However, using the sum-of-SGs visibility can make it up to 13% slower. As such, when interpolating between the two schemes, the extra cost is dominated by the sum-of-SGs. The worst slowdown we observed is 16%.

6.5.3 Multiple Scattering

While single scattering accounts for non-smooth variation in fabric appearance, most of the fabric color comes from multiple scattering. We approximate multiple scattering as a product between (1) the multiple scattering response $\widetilde{L}_o^{\text{multi}}(x, \omega_o)$ of the fabric to unoccluded illumination from the light source and (2) a visibility factor $\bar{V}^{\text{glo}}(x)$; i.e. $L_o^{\text{multi}}(x, \omega_o) = \widetilde{L}_o^{\text{multi}}(x, \omega_o)\bar{V}^{\text{glo}}(x)$. The former relies on the precomputed IIRTF, and the latter employs Gaussian filtering of the global visibility discussed in the last section. We will now discuss the terms in order.

Multiple Scattering Response to Unoccluded Light

Directional light. Recall that we assume a directional light that emits unit radiance in direction $-\omega_d$. If the light is not occluded in the neighborhood of x, then it causes the incoming radiance field $T(\omega_d, \omega_i, x)$ around x, and the outgoing light due to multiple scattering is:

$$\widetilde{L}_o^{\text{multi}}(x, \omega_o) = \int_{\mathbb{S}^2} T(\omega_d, \omega_i, x) S(\omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i.$$

After identifying the cell C containing the midpoint of the fiber segment on which x lies, we approximate $T(\cdot, \cdot, x)$ using the precomputed IIRTF matrix A_C. The convolution above can be computed as a dot product of SH coefficients. Evaluating the IIRTF involves projecting the directional light into SH basis by evaluating the vector $c_d = (Y_0(\omega_d), Y_1(\omega_d), \dots)$ and then multiplying it with the IIRTF matrix to obtain the vector $c_T = A_C c_d$. Using the precomputed table of BCSDF expansion, we have $S(\omega_i, \omega_o) \cos\theta_i \approx \sum_j \mathcal{C}_S[\theta_o, j] Y(\omega_i)$. Let $c_S = (\mathcal{C}_S[\theta_o, 0], \mathcal{C}_S[\theta_o, 1], \dots)$. Conceptually, $\widetilde{L}_o^{\text{multi}}(x, \omega_o)$ is the dot product between c_S and c_T.

However, we cannot compute the dot product directly because the BCSDF's expansion is defined in the ω_o-space, but the IIRTF's expansion and thus c_T are defined in the fabric's object space. To solve this problem, we transform c_T into the ω_o-space. Let R_{ω_o} be the rotation matrix that transforms spherical harmonics expansion from the fabric's object space to the ω_o-space. Then, we have that $\widetilde{L}_o^{\text{multi}}(x, \omega_o) = c_S \cdot (R_{\omega_o} c_T)$. We compute the matrix using the technique described by Pinchon and Hoggan [64].

Spherical Gaussian light. The computation is essentially unchanged. To compute the fabric's response to the SG light $G(\omega_i; \xi, \lambda)$, we look up the precomputed SG expansion table to get the vector $c_G = (\mathcal{C}_G[\lambda, 0], \mathcal{C}_G[\lambda, 1], \cdots)$, which represents the expansion of the SG with axis $(0, 0, 1)$ and sharpness λ. We then rotate the coefficient vector by a rotation R_ξ to align the axis with ξ. The rest of the process then applies. We multiply the rotated coefficient with the IIRTF matrix, rotate the result to the ω_o-space, and dot the rotated result with the SH expansion of the BCSDF: $\widetilde{L}_o^{\text{multi}}(x, \omega_o) = c_S \cdot (R_{\omega_o} A_C R_\xi c_G)$.

Visibility for Multiple Scattering

We now estimate the effect of occlusion on multiple scattering. Since the IIRTF, by definition, has taken into account occlusion by local geometry, we only need to deal with occlusion by macroscopic geometry. We observe that, around shadowed areas on a piece of cloth, the shadow is not sharp due to light that propagates through the fabric volume into the occluded area; i.e., cloth exhibits subsurface-scattering-like behavior. We approximate this effect by multiplying the response-to-unoccluded-light term with a kernel-smoothed global visibility over the area near the shaded point x:

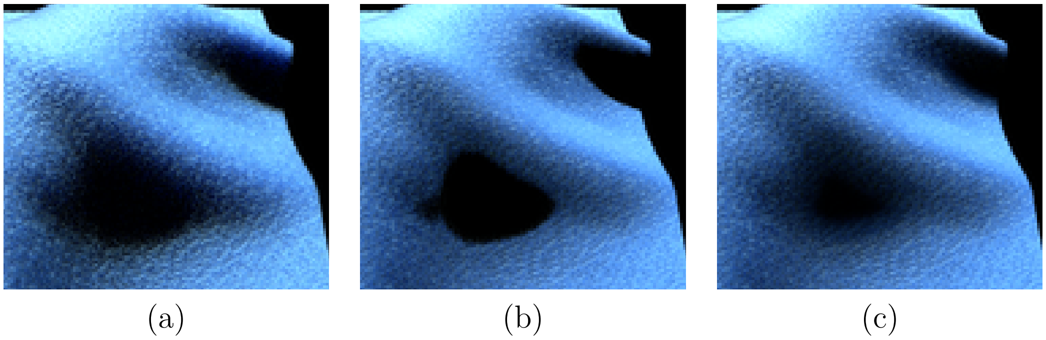
$$\bar{V}^{\mathrm{glo}}(x, \omega_d) = \frac{\int_A K(x, x') V^{\mathrm{glo}}(x', \omega_d)\ \mathrm{d}x'}{\int_A K(x, x')\ \mathrm{d}x'}$$

where A is the area of the (flat, before shell mapping) fabric surface, and K is the kernel function that depends only on the distance between x and x' in the fabric's plane. The global visibility term V^{glo} is computed by ray tracing or standard shadow mapping in the directional light case and by PCSS in the SG light case. The effects of the visibility term can be seen in Figure 6.6.

We choose K to be a 2D Gaussian kernel (with standard deviation σ_{glo}) because it enables efficient implementations. How we implement filtering depends on the target hardware.

On the CPU, we sample x' according to the 2D Gaussian distribution in the flat fabric's space and use the global visibility term $V^{\mathrm{glo}}(x', \omega_d)$ as the unbiased estimate of $\bar{V}^{\mathrm{glo}}(x, \omega_o)$.

On the GPU, we compute the visibility texture $V[u, v] := V^{\mathrm{glo}}(\underline{\mathrm{X}}[u, v])$ by looking up the shadow map at each position stored in the top surface position texture previously discussed in Section 6.5.2. We can then filter V by a 2D Gaussian



We show renderings by (a) path tracing, (b) using the (hard) global visibility term V^{glo} to scale down the multiple scattering response, and (c) using the (soft) global visibility term $\bar{V}^{\mathrm{glo}}$ for the same purpose. Exposure of 4 is used to highlight the difference between shadowed and unshadowed areas. Notice that the shadowed regions in (b) have sharp edges, while those in (a) and (c) have softer edges.

Figure 6.6: The effect of the multiple scattering visibility term $\bar{V}^{\mathrm{glo}}$.

kernel corresponding to K to obtain the average visibility texture $\bar{V}$, which can be done efficiently because the Gaussian kernel is separable. The global visibility term $\bar{V}^{\mathrm{glo}}(x)$ is simply a lookup into $\bar{V}$ using the shell texture coordinate of x. (However, our implementation actually performs Monte Carlo integration with 40 samples per fragment with the help of a random number texture.)

6.6 Results

We implemented two versions of our algorithm according to the hardware they run on. The CPU version employs Monte Carlo sampling of the light source discussed at the end of Section 6.5.2. As a result, it treats environment light sources as if they were directional lights, and does not use approximation specific to SG lights in Section 6.5.2. The GPU version employs all GPU-specific computation and the entirety of Section 6.5.2. The algorithms were implemented in Java and the OpenGL Shading Language (GLSL). We performed experiments on 8 fabrics, which are derived from micro CT scans except for the shot silks, which are procedural

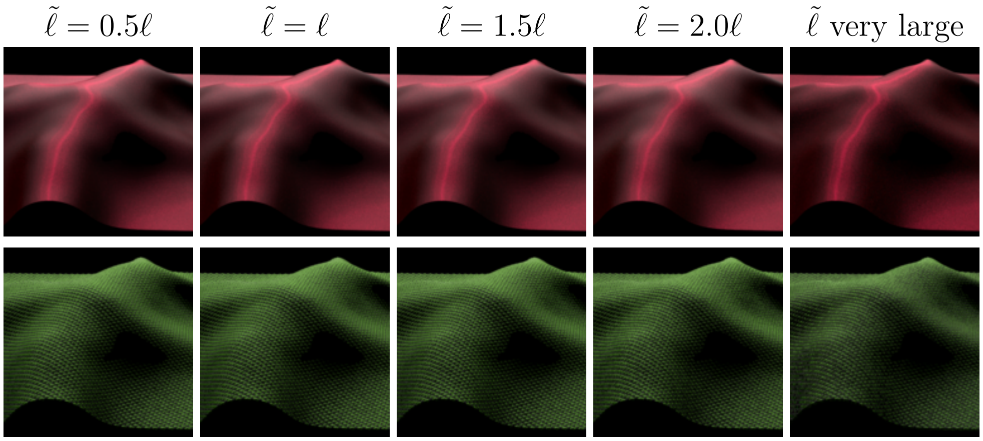


Figure 6.7: The effects of the dimensions of IIRTF cells on renderings of the Silk (top) and the 2/3 Satin fabrics (bottom).

fiber models. Their details are given in Table 6.1.

We will first describe the effects of the two parameters that are the most important to the appearance of the rendered fabrics: the dimensions of the IIRTF cells and the SH order used in the IIRTF. After fixing these parameters, we compare our algorithm against other algorithms. We then lastly discuss its limitations.

Effects of the dimensions of the IIRTF cells. As discussed in Section 6.4.2, we subdivide an exemplar block into cells whose side lengths are as close as possible to a number which we now call the *target cell length* $\tilde{\ell}$. We consider a sequence of 5 target cell length values for each fabric. The first four are 0.5ℓ, ℓ, 1.5ℓ, and 2ℓ, where ℓ is the mean free path. We picked the fifth so that the numbers of cells in all dimensions are close to 1 in order to see effects of very coarse subdivisions. We then generated 5 IIRTF data according to the sequence and used them to render a draped piece of fabric with the CPU version of our algorithm. The SH order used was 5 in all renderings, which is the maximum SH order that we use in this paper. The renderings of the Silk and the 2/3 Satin fabrics are shown in Figure 6.7. The

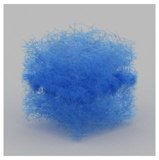
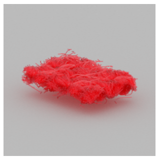
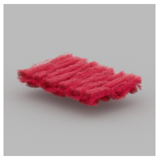
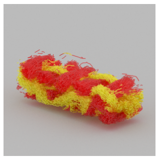
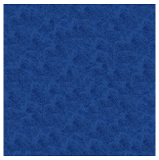
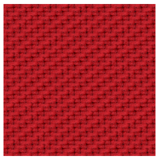
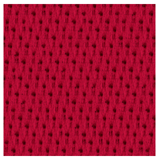
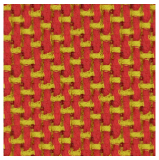
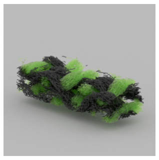
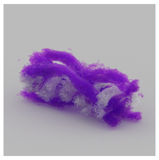
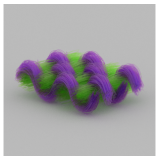
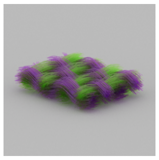
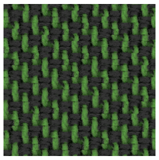
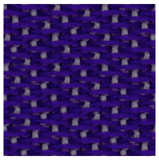
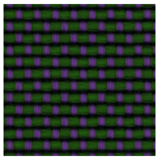
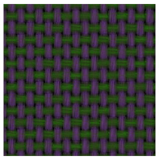
	Fleece	Gabardine	Silk	4/1 Satin
Swatch				
Tiled				
#Fibers	31,091	8,377	7,681	21,953
#Segments	580,660	120,121	112,294	318,882
Fiber radius	0.00087 cm	0.00161 cm	0.00047 cm	0.00150 cm
Mean free path ℓ	0.03550 cm	0.00668 cm	0.00250 cm	0.01257 cm
Target cell length $\tilde{\ell}$	ℓ	2.0ℓ	2.0ℓ	2.0ℓ
SH Order used	4	4	5	4
IIRTF file size	7.87 MB	11.38 MB	9.01 MB	11.44 MB
SSDF file size	218.71 MB	50.05 MB	47.19 MB	122.84 MB
Sum-of-SGs file size	54.27 MB	11.23 MB	10.50 MB	29.80 MB
#Samples/tetrahedron	32	32	32	32
	2/3 Satin	1/4 Satin	Shot Silk A	Shot Silk B
Swatch				
Tiled				
#Fibers	16,357	18,507	2,005	2,005
#Segments	244,420	290,149	75,550	56,518
Fiber radius	0.00150 cm	0.00150 cm	0.00010 cm	0.00010 cm
Mean free path ℓ	0.01438 cm	0.01693 cm	0.00135 cm	0.00119 cm
Target cell length $\tilde{\ell}$	1.5ℓ	1.5ℓ	2.0ℓ	2.0ℓ
SH Order used	4	5	5	5
IIRTF file size	10.52 MB	21.05 MB	25.97 MB	29.96 MB
Sum-of-SGs file size	95.57 MB	112.32 MB	33.73 MB	26.76 MB
Sum-of-SGs file size	22.84 MB	27.12 MB	7.06 MB	5.28 MB
#Samples/tetrahedron	32	32	64	64

Table 6.1: The fabric models.

complete set of results are given in the supplementary material.

We observed that, in all fabrics, there are virtually no differences between the renderings of the target cell lengths of 0.5ℓ and ℓ, showing that the mean free path provides a good starting point for finding the right target cell length. The renderings generally become darker as cell dimensions become larger. This is because the IIRTFs of fiber segments deeper below the fabric surface are averaged with those near the top. Larger cell sizes also yield blockier, less smooth renderings in fabrics with multiple yarn colors.

For each fabric, we choose the coarsest subdivision that yields smooth renderings that are similar in color to the renderings of the finest subdivisions. The choices we made are listed in Table 6.1.

Effects of IIRTF SH order. We used the CPU version of our algorithm to render all the fabrics, varying the SH order of the IIRTF from 0 to 5. We show renderings of the 4/1 Satin and Shot Silk A fabrics in Figure 6.8. The complete set of results can be found in the supplementary material.

In general, as we increase the SH order, we see more "directionality." That is, the highlights become more defined and sharper, and shadows also become darker. Color changes start to stabilize after SH of order 3.

The fabrics can be divided into two groups based on their responses to the SH order. In the Fleece, Gabardine, 4/1 Satin, and 2/3 Satin fabrics, we observe smaller changes between consecutive pairs of SH orders as we move through higher SH orders. In this group, differences between renderings of SH order 4 and 5 are minor and limited to highlight structures. In the 1/4 Satin, the Silk, and the Shot

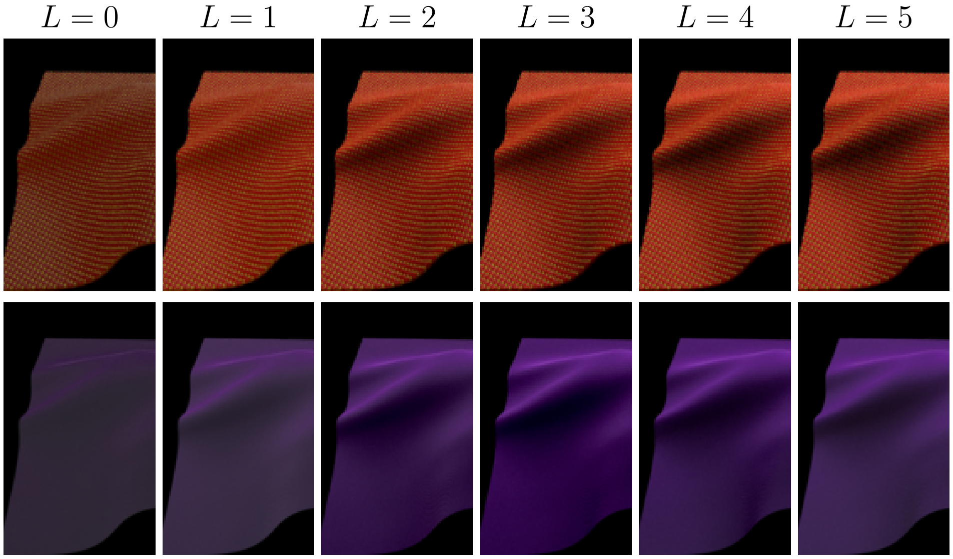


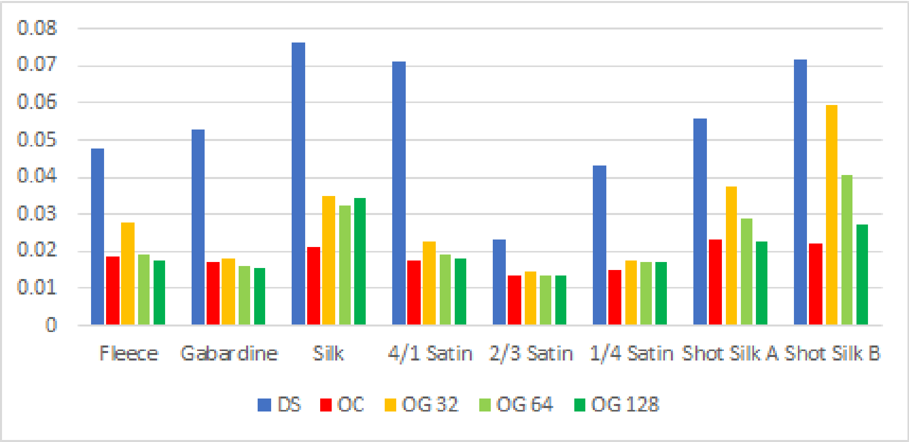
Figure 6.8: Effects of the IIRTF's SH order, denoted by L, on renderings of the 4/1 Satin (top) and Shot Silk A (bottom) fabrics.

Silk fabrics, however, significant changes can be observed between SH order 4 and 5. For this reason, we choose to use SH order 4 for the first group, and SH order 5 for the second group.

Quantitative match in flat configurations. We compare between dual scattering[1] and the two version of our algorithm. We rendered flat pieces of the 8 fabrics under 492 scene configurations used to validate fitted models in [39]. We rendered each configuration as a 64 × 64 images with 128 samples per pixel. For the GPU version of our algorithm, we also vary the number of volume samples per shell tetrahedron (32, 64, and 128).

For each image rendered, we computed the average intensity, resulting in 3 × 492 = 1,476 values per algorithm and per fabric. The root mean squared errors

[1] See the details of our implementation of dual scattering in the supplementary material. We use the value 1.0 for all scattering density factors, but it still produced images that are not as bright as the path tracing references for most of the fabrics.



We show RMSEs when compared to path tracing references of the average intensities of the 492 images rendered by dual scattering (DS), the CPU version of our algorithm (OC), and three runs of GPU version (OG) using 32, 64, and 128 volume samples per tetrahedron. The supplementary material of [40] contains numerical values of the RMSEs.

Figure 6.9: Accuracy of dual scattering and the two versions of our algorithm.

(RMSE) of these values when compared to those produced by path tracing are graphed in Figure 6.9. The data show that, if the GPU version uses enough volume samples, both versions of our algorithm are more accurate than dual scattering. The result can be attributed to the assumptions that dual scattering (which was designed for rendering hair) makes—for example, that all fibers have the same BCSDF and nearby fibers are parallel to one another—which are violated in fabrics. The CPU version of our algorithm is generally more accurate than the GPU one because of its accurate ray intersection. As we increase the number of volume samples per tetrahedron, the GPU version becomes more accurate, except when rendering the Silk fabric, which might be because the rasterized Silk volume is the coarsest among all the volumes.

Renderings. We rendered the 8 fabrics in a draped configuration under 5 different lightings. To prevent algorithms from picking up illumination from underneath the fabric, we put a black mesh of the same shape as the shell underneath the fabrics. The images are rendered with 256 samples per pixel at resolution 1024×1024. We compare between four algorithms: path tracing, dual scattering, and both versions of our algorithm. Some of the renderings are available in Figure 6.11.

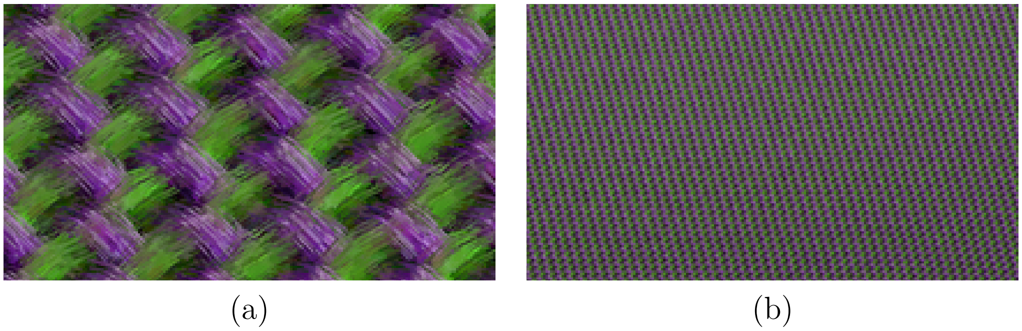
Our algorithm generally produced images whose colors are similar to path tracing references, while dual scattering yielded images that are generally too dim. (For the Shot Silk fabrics, however, it overly amplified the green color.) The renderings of our GPU algorithm under the SG light with $\lambda = 10$ are darker than those of other algorithms because PCSS overestimates shadows when the light source is large. Our algorithm also produced noticeable changes in highlights in all fabrics, especially in the Fleece and the Shot Silks. Subtleties in the shadows yarns cast on one another are missed in the Satin fabrics. These deviations might be caused by low order spherical harmonics' inability to capture all the features of the incoming radiance field in the fabric volumes. We emphasize, though, that our renderings look plausible and have similar colors to the ground truth. Moreover, our GPU algorithm produced essentially noise-less images as it does not rely heavily on stochastic sampling.

We ran the first three algorithms on a cluster of 5 machines equipped with a total of 192 cores in Intel Xeon X7560 processors, each clocked at 2.27 GHz. Our GPU algorithm was run on an Nvidia GeForce GTX 980 graphics card with 4 GB of RAM. Both the cluster and the GPU are controlled by a PC equipped with a 4-core Intel Xeon E5-2637 processor clocked at 3.5 GHz. We report the wall clock time, measured on the controller PC, used by all algorithms in Table 6.2. The wall clock time does not include loading and preparation of data (e.g. construction acceleration structures) but does include network communication with the cluster while rendering.

The table clearly shows that our GPU algorithm could render in tens of seconds the images a compute cluster needs several minutes to render, all with a single commodity GPU.

Our CPU algorithm is the fastest among all the CPU-based algorithms, but its speedup over path tracing depends on the rendered fabrics. Path tracing is efficient in fabrics that result in low average path length. For example, the Silk has a strong reflective component, so light tends to reflects off it rather than going inside. Moreover, the 2/3 Satin has a black yarn, so Russian roulette tends to terminate paths early. Even in these cases, our CPU algorithm achieves a speedup of 2. For the other fabrics, light tends to remain in the fabric volume due to the fibers' strong transmission, so multiple scattering is more visually important for these fabrics. When rendering them, our algorithm achieves significant speedups. As we did not implement MFT, we could not compare to it directly. Instead, we compare against conservative estimates of MFT's running time, computed as described in Table 6.2. In the fabrics that are difficult for path tracing, our running times are 2 to 4 times better than MFT's estimates. We also include running times of dual scattering for completeness. However, as we did not implement the GPU version of dual scattering, they do not represent how efficient the algorithm can be.

Limitations. The memory usage and speed of our algorithm depend strongly on the size of the data used. In particular, the GPU implementation does not perform well when rendering dense fabrics: those whose mean free path is small relative to the exemplar's size. A short mean free path leads to large IIRTF data, which can

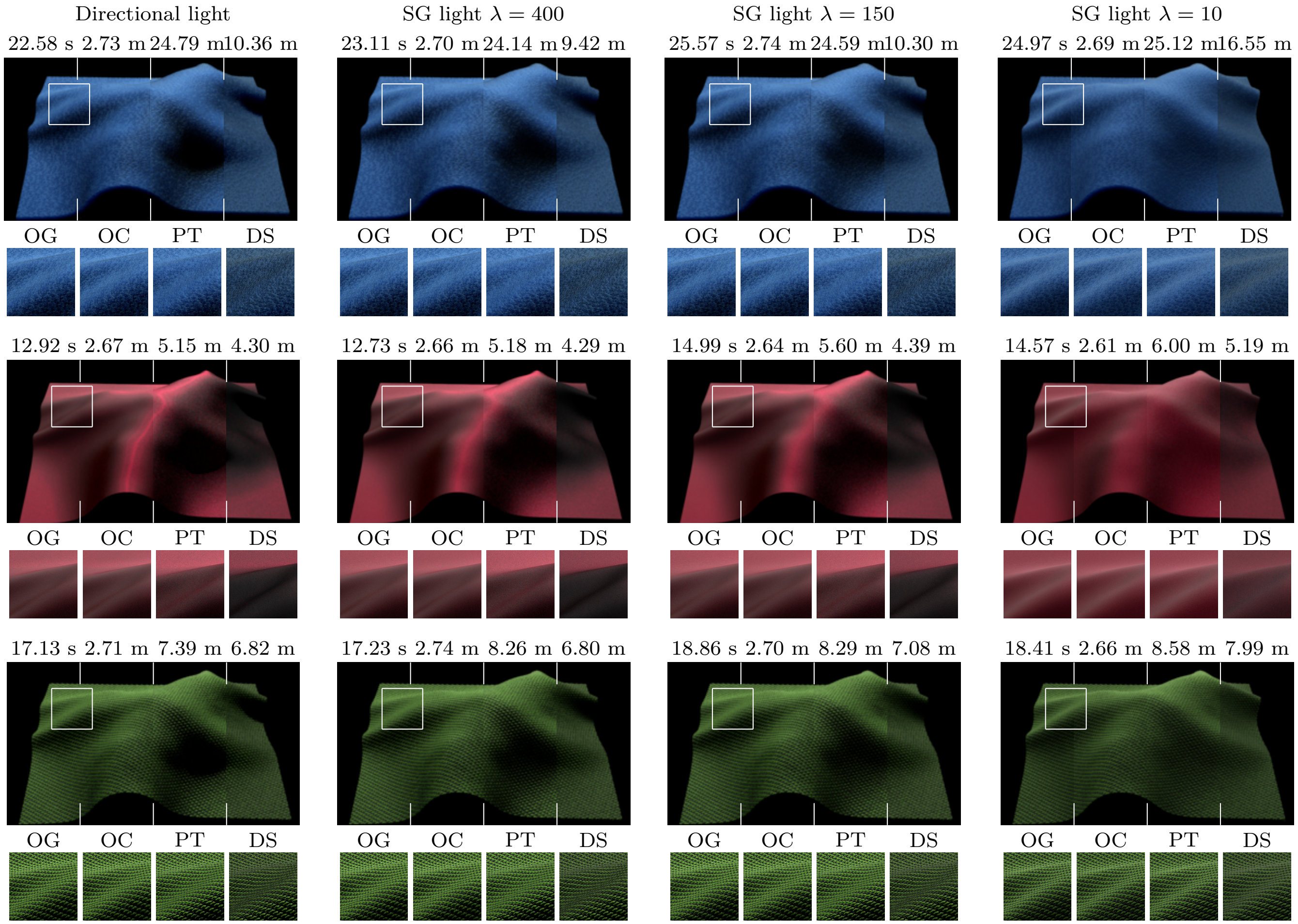


(a) In a microscope view, which is beyond our intended range of application, it is plain to see that on each segment, all pixels in the same block have the same color. (b) However, at a magnification closer to what can be seen by the naked eye, the blockiness is invisible. The supplementary material of [40] contains an animated version of (b), showing that the artifacts do not cause temporal flickering.

Figure 6.10: The Shot Silk B fabric at two magnifications.

thrash the GPU memory bandwidth or might not fit in GPU memory altogether. We note that this is a problem faced by any algorithm that performs volumetric precomputation on a uniform grid: dense material means appearance changes fast spatially, so such an algorithm needs a fine grid to be accurate.

Our algorithm yields blocky artifacts at zoom levels where fiber segments occupy multiple pixels because all fragments from the same segment use the same SVF and IIRTF. However, these artifacts are at the level of fiber segments, so they are not visible as long as the zoom level is not that of a microscope (see Figure 6.10). Moreover, they do not cause any flickering on animated fabrics (see supplementary videos).


Directional light
SG light $\lambda = 400$
SG light $\lambda = 150$
SG light $\lambda = 10$
22.58 s 2.73 m 24.79 m 10.36 m
23.11 s 2.70 m 24.14 m 9.42 m
25.57 s 2.74 m 24.59 m 10.30 m
24.97 s 2.69 m 25.12 m 16.55 m
OG OC PT DS
12.92 s 2.67 m 5.15 m 4.30 m
12.73 s 2.66 m 5.18 m 4.29 m
14.99 s 2.64 m 5.60 m 4.39 m
14.57 s 2.61 m 6.00 m 5.19 m
OG OC PT DS
17.13 s 2.71 m 7.39 m 6.82 m
17.23 s 2.74 m 8.26 m 6.80 m
18.86 s 2.70 m 8.29 m 7.08 m
18.41 s 2.66 m 8.58 m 7.99 m
OG OC PT DS

From top to bottom, renderings of the Fleece, Silk, and 2/3 Satin fabrics under 4 lighting configurations. We compare results by path tracing (PT), dual scattering (DS), and the CPU and GPU versions of our algorithms (OC and OG, respectively). The times on top of the renderings are the wall clock times used to render the full images. The supplementary material of [40] contains the complete set of results.

Figure 6.11: Comparison of renderings of a draped fabric model produced by our algorithms, a path tracer, and the dual scattering algorithm.

	Fleece		Gabardine		Silk		4/1 Satin	
	Time	Speedup	Time	Speedup	Time	Speedup	Time	Speedup
PT	24.59 m	1.00x	16.21 m	1.00x	5.60 m	1.00x	25.74 m	1.00x
DS	10.30 m	2.39x	6.88 m	2.36x	4.39 m	1.28x	9.94 m	2.59x
MFT*	11.88 m	2.07x	7.70 m	2.10x	5.09 m	1.10x	9.72 m	2.65x
Ours (CPU)	2.74 m	8.99x	2.54 m	6.38x	2.64 m	2.12x	2.72 m	9.46x

	2/3 Satin		1/4 Satin		Shot Silk A		Shot Silk B	
	Time	Speedup	Time	Speedup	Time	Speedup	Time	Speedup
PT	8.29 m	1.00x	19.48 m	1.00x	11.11 m	1.00x	8.17 m	1.00x
DS	7.08 m	1.17x	10.46 m	1.86x	8.11 m	1.37x	6.32 m	1.29x
MFT*	6.12 m	1.35x	9.47 m	2.06x	6.33 m	1.75x	6.16 m	1.33x
Ours (CPU)	2.70 m	3.07x	2.81 m	6.93x	2.67 m	4.16x	2.71 m	3.02x

	Fleece	Gabardine	Silk	4/1 Satin	2/3 Satin	1/4 Satin	Shot Silk A	Shot Silk B
Ours (GPU) time	25.57 s	13.98 s	14.99 s	18.86 s	18.86 s	22.47 s	29.26 s	25.52 s

🟥 Smaller values are better 🟩 Larger values are better

In addition to path tracing (PT), dual scattering (DS), the CPU version of our algorithm, and the GPU version, we also provide a conservative estimate of the time used by MFT. (*MFT's running time is estimated by running path tracing up to 6 accurate scattering events. We do not estimate the time MFT needs for photon tracing and stochastic matrix inversion and simply set it to 0.) The CPU-based algorithms ran on a 192-core cluster while the GPU version of our algorithm ran on a single GPU. We separate out the timings of the GPU algorithm since it uses different hardware from the rest. Note that rendering is very fast, taking from 14 seconds to 30 seconds on a single GPU compared to minutes on 192 CPU cores.

Table 6.2: The wall clock time used to render the draped fabrics under an SG light with $\lambda = 150$.

6.7 Conclusion

We have described a GPU-friendly algorithm for approximate rendering of micro-appearance models of fabrics. It allows a commodity GPU to render, in tens of seconds, high-quality images with multiple scattering that a sizable CPU cluster needs to spend several minutes on. The efficiency gain is possible through the use of appropriate precomputation. In particular, we employ the IIRTF to compute indirect illumination in a single step. We also introduce a new representation for visibility around a fiber segment—the sum of spherical Gaussians on Fibonacci points—that can exploit both the structure of the fiber's scattering function and the spherical Gaussian light source.

Future directions. Our algorithm requires regular weave patterns, so a precomputation scheme that allows multiple scattering to be approximated in fabrics with complex weave patterns would expand its applicability. An efficient compression scheme for precomputed data is necessary to enable multiple fabrics to be rendered all at once. It is also interesting to see whether the sum-of-SGs visibility representation can be applied in other rendering contexts.

CHAPTER 7

CONCLUSION

> จบแล้ว รักนี้ที่ทนมา
> เหนื่อยล้า เพราะรักที่ยาวไกล
> หนึ่งคำที่อาจจะฝืนใจ
> แต่วันนี้ต้องพูดมันออกไป ลาก่อน
>
> อัสนี วสันต์ โชติกุล, *ลาก่อน*

Hair and textiles play a major role in human appearance and therefore are important materials to model well in computer graphics. While simplified models that abstract them to flat surfaces often suffice for media production, they are not enough to capture the complex shapes and optics that arise from the multitude of fibers that constitutes the materials. To capture all such complexity, researchers have proposed micro-appearance models in which individual fibers are modeled explicitly. The hope is that, by distilling the materials to their fundamental elements, the correct aggregate appearance would arise automatically from physical simulation. Moreover, the behavior of these elements would be simpler to specify.

Creating and deploying micro-appearance models, of course, are not straightforward, and many research questions have to be answered. At the level of individual fibers, we need to know how exactly to represent the fibers' geometric and optical properties. At the level of the fiber assemblies, we need to know how to control and efficiently reproduce the aggregate appearance that arises from the fibers' interaction with light.

In this dissertation, we have presented research that contributed answers to the above questions. We hope that we have expanded insights on micro-appearance models for hair and fabrics and have made them more practical in computer graphics applications.

Chapter 4 confirms that cross-sectional shape has significant impact on a hair fiber's light scattering behavior. An elliptical cross section gives rise to features such as the bright lobes in the TT mode that cannot be predicted by models that are based on circular cross section. We have also observed the E mode—a bright, perfect specular reflection that occur only at grazing angles—that cannot be explained by geometric optics. It is thus imperative to take into account both cross-sectional shape and wave properties of light if we would like to create more accurate light scattering models of hair and other types of fibers. Additionally, while our new measurement device is not accurate enough to extract BCSDF values, it can serve as a guide on which BCSDF features are important to model.

Chapter 5 contributes a number of tools for modeling appearance of fabrics. We present an optimization-based algorithm for parameter fitting that works on any models whose contribution can be differentiated with respect to model parameters, which allows a large class of models to be fitted to photographed appearance of fabrics in a systematic way. We also expand the means to create micro-appearance models by introducing (1) an algorithm for converting micro CT scans of cloth samples to fiber meshes and (2) a new and simple BCSDF for textile fibers. Comparison between the BCSDF and the microflake phase function shows that accounting for Fresnel reflectance is important to faithfully modeling light scattering behavior of fabrics.

Chapter 6 presents a GPU-friendly algorithm for rendering a micro-appearance models of textiles under environment illumination. While rendering such a model with multiple scattering typically takes multiple core-hours with CPU-based algorithms, our algorithm shortens the rendering time to under a minute using only a commodity GPU, thus making micro-appearance models practical for interactive design. In designing the algorithm, we make use of two insights. The first is that multiple scattering inside a fabric volume is smooth and can be approximated with smooth functions such as low-order spherical harmonics. The second is that shading under large spherical Gaussian lights can be approximated efficiently if we represent visibility as a sum of spherical Gaussians with appropriate sharpness. These insights should prove useful in designing other fast rendering algorithms.

7.1 Future Directions

We see many opportunities for building on our proposed models and algorithms to increase their effectiveness and capability.

The light scattering model we proposed in Chapter 4 is not yet practical for production rendering. It requires a large precomputed table that is specific to a setting for the hair's cross section. However, a head of hair contains fibers with different colors and cross sections, and these attributes might also change along a fiber. As a result, modeling hair with all its complexity would require us to precompute the tables for all attribute configurations. A model that requires minimal storage while still being able to handle a wide range of parameters is needed. For many tasks such as rendering for media production, it is enough that the model be plausible instead of being highly accurate. Hence, it is interesting to explore how we may trade accuracy for efficiency when designing models under the constraint that the features due to ellipticity are still conveyed in a plausible manner.

The appearance matching process in Chapter 5 is hard to deploy outside a laboratory because it requires a gonioreflectometer. It is also inefficient in the sense that an input photograph only yields a single data point to fit against. How may we design an acquisition process that requires only off-the-shelf equipments, is deployable in the field, and yields higher data density per photograph? An interesting approach to explore is to wrap a piece of fabric around a curved surface like in the works of Marschner et al. [50] and Zinke et al. [114] so that one photograph contains multiple lighting configurations.

A real piece of fabric can have multiple types of yarn, but we only tested our appearance matching process on fabrics containing a single yarn type. While the process is theoretically applicable to the multiple-yarn-type case, what is actually required to apply it—including how to align photographs with 3D models and what objective function to use—is unknown. An answer to this problem would greatly expand the types of materials the algorithm can handle in practice.

Our rendering algorithm in Chapter 6 also suffers the same problem as the elliptical hair scattering model: it needs precomputed data that are specific to a certain parameter setting. This requirement makes it difficult to render multiple types of fabrics at once. Moreover, an interactive design session that uses our algorithm can change only the fabric's macrogeometry but not the color or other light scattering behavior. This problem may be attacked on multiple fronts. We may research better compression schemes for the precomputed data, or we may design novel precomputations that allow us to approximate multiple scattering on the fly as some fabric parameters change. There already exists a body of works that allow editing of light scattering model parameters in real time [5, 101, 25], and it would be interesting to see whether we can achieve similar goals when rendering fabric micro-appearance models.

Lastly, the main reason that our rendering algorithm is still not fast enough to be real-time is that it explicitly renders all microgeometry details. Consequently, it needs high sample count per pixel to avoid aliasing when the fabric is viewed from far away. This problem calls for a level-of-detail representation of fabric appearance that takes into account internal multiple scattering. Such a representation would greatly expand the realism afforded by interactive applications. It would make environments in games and virtual reality applications more believable and immersive, and it would also make virtual objects integrate to real-world visuals more seamlessly in augmented reality. Moreover, it would bring real-time feedback with accurate visuals to interactive prototyping of textiles, making the process greatly more responsive and rapid to designers. Due to these wide-ranging impacts, we think that more research should be done on not only the level-of-detail representation but also on efficient simulation the appearance of micro-appearance in general.

APPENDIX A

SUPPLEMENTARY MATERIAL FOR CHAPTER 4

A.1 Problems with d'Eon's Longitudinal Scattering Function

d'Eon et al. [17] introduced the LSF

$$M_p(\theta_i, \theta_o) = \frac{\operatorname{csch}(1/\beta_p^2)}{2\beta_p^2} \exp\left(\frac{\sin(-\theta_i)\sin\theta_o}{\beta_p^2}\right) I_0\left(\frac{\cos(-\theta_i)\cos(\theta_o)}{\beta_p^2}\right)$$

which satisfies the following energy conservation property:

$$\int_{-\pi/2}^{\pi/2} M_p(\theta_i, \theta_o) \cos\theta_o \, \mathrm{d}\theta_o = 1.$$

The exponent of the cosine factor is 1, not 2, because their LSF already has a cosine factor folded in. d'Eon et al. briefly noted that $M_p(\theta_i, \theta_o - \alpha_p)$ can be used to incorporate the longitudinal shift. However, if $\alpha_p \neq 0$, it is not true that

$$\int_{-\pi/2}^{\pi/2} M_p(\theta_i, \theta_o - \alpha_p) \cos\theta_o \, \mathrm{d}\theta_o \leq 1.$$

For example, if $\theta_i = -\pi/2$, $\beta_p = \pi/15$, and $\alpha_p = -\pi/15$, then the integral above evaluates to 2.662.

The extra energy problem can be fixed by using $M_p^*(\theta_i, \theta_o) = M_p(\theta_i - \alpha_p, \theta_o)$ instead, and this might have been what d'Eon et al. intended. Nevertheless, this definition causes the function to achieve its maximum at unintended locations when $\alpha_p \neq 0$. Here, the sensible behavior of the M_p^* as a function of θ_o should be that the function's peak should occur at $\theta_o = -\theta_i + \alpha_p$. In case $-\theta_i + \alpha_p > \pi/2$, the peak should occur at $\theta_o = \pi/2$ because it is the boundary of θ_o's range.

d'Eon's LSF, however, does not follow the above property. For example, when $\alpha_p = \pi/9$ and $\beta_p = \pi/15$, we have that $M_p^*(-\pi/2, \theta_o)$, which is equal to $M(-\pi/2 - \pi/9, \theta_o)$, achieves its maximum at another location which is not $\pi/2$. In general, when $\alpha_p > 0$, as θ_i approaches $-\pi/2$, the peak of the LSF approaches $\pi/2$ and then "bounces back" from $\theta_o = \pi/2$ towards $\theta_o = 0$. We believe this behavior is not physically plausible. As a result, we use the LSF detailed in Chapter 4, which yields sensible peak locations when $\alpha_p \neq 0$.

A.2 Properties of Azimuthal Scattering Function

In Section 4.4.2, we define the *blurred response function as:*

$$\begin{aligned} R_p(\phi_i, \phi_o) &= \int_0^{2\pi} \int_{-D(\phi'_o)/2}^{D(\phi'_o)/2} A_p(\phi'_o, s_o) K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi'_o, s_o) - \phi_i)\ \mathrm{d}s_o\ K^{\gamma_p}(\phi_o - \phi'_o)\ \mathrm{d}\phi'_o \\ &= \int_0^{2\pi} \int_{-D(\phi'_o)/2}^{D(\phi'_o)/2} A_p(\phi'_o, s_o) K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi'_o, s_o) - \phi_i) K^{\gamma_p}(\phi_o - \phi'_o)\ \mathrm{d}s_o \mathrm{d}\phi'_o. \end{aligned}$$

In this section, we show that the function is energy conserving and *almost* reciprocal.

To reduce clutter in the proofs, we introduce compact notations for integrals. we shall abbreviate $\int_0^{2\pi}$ as $\int_{\circledcirc}$, and $\int_{-D(\phi)/2}^{D(\phi)/2}$ as $\int_\phi$. So, the function can be written as follows:

$$R_p(\phi_i, \phi_o) = \int_{\circledcirc} \int_{\phi'_o} A_p(\phi'_o, s_o) K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi'_o, s_o) - \phi_i) K^{\gamma_p}(\phi_o - \phi'_o)\ \mathrm{d}s_o \mathrm{d}\phi'_o.$$

One important property of this function is that it is *reciprocal.*

Lemma A.1. *$R_p(\phi_i, \phi_o) = R_p(\phi_o, \phi_i)$ for all values of ϕ_i and ϕ_o.*

Proof. We first note that

$$\begin{aligned}&A_p(\phi'_o, s_o) K^{\gamma_p}(\phi^{\mathrm{e}}_p(\phi'_o, s_o) - \phi_i) \\ &= \int_{\circledcirc} \int_{\phi'_i} A(\phi'_i, s_i) K^{\gamma_p}(\phi'_i - \phi_i) \delta(\phi'_i - \phi^{\mathrm{e}}_p(\phi'_o, s_o)) \delta(s_i - s^{\mathrm{e}}_p(\phi'_o, s_o)) \ \mathrm{d}s_i \mathrm{d}\phi'_i\end{aligned}$$

where $\delta(x)$ is Dirac delta function. As a result,

$$\begin{aligned}&R_p(\phi_i, \phi_o) \\ &= \int_{\circledcirc} \int_{\phi'_o} \left(\int_{\circledcirc} \int_{\phi'_i} A_p(\phi'_i, s_i) K^{\gamma_p}(\phi'_i - \phi_i) \delta(\phi'_i - \phi^{\mathrm{e}}_p(\phi'_o, s_o)) \delta(s_i - s^{\mathrm{e}}_p(\phi'_o, s_o)) \ \mathrm{d}s_i \mathrm{d}\phi'_i \right) K^{\gamma_p}(\phi'_o - \phi_o) \ \mathrm{d}s_o \mathrm{d}\phi'_o \\ &= \int_{\circledcirc} \int_{\phi'_o} \int_{\circledcirc} \int_{\phi'_i} A_p(\phi'_i, s_i) K^{\gamma_p}(\phi'_i - \phi_i) K^{\gamma_p}(\phi'_o - \phi_o) \delta(\phi'_i - \phi^{\mathrm{e}}_p(\phi'_o, s_o)) \delta(s_i - s^{\mathrm{e}}_p(\phi'_o, s_o)) \ \mathrm{d}s_i \mathrm{d}\phi'_i \mathrm{d}s_o \mathrm{d}\phi'_o \\ &= \int_{\circledcirc} \int_{\phi'_i} A_p(\phi'_i, s_i) \left(\int_{\circledcirc} \int_{\phi'_o} K^{\gamma_p}(\phi'_o - \phi_o) \delta(\phi'_i - \phi^{\mathrm{e}}_p(\phi'_o, s_o)) \delta(s_i - s^{\mathrm{e}}_p(\phi'_o, s_o)) \ \mathrm{d}s_o \mathrm{d}\phi'_o \right) K^{\gamma_p}(\phi'_i - \phi_i) \ \mathrm{d}s_i \mathrm{d}\phi'_i.\end{aligned}$$

Consider the middle integral, the delta functions are not zero only when $\phi'_i = \phi^{\mathrm{e}}_p(\phi'_o, s_o)$ and $s_i = s^{\mathrm{e}}_p(\phi'_o, s_o)$. By reversibility of path, the previous two conditions imply that $\phi'_o = \phi^{\mathrm{e}}_p(\phi'_i, s_i)$ and $s_o = s^{\mathrm{e}}_p(\phi'_i, s_i)$. So, the middle integral evaluates to simply $K^{\gamma_p}(\phi^{\mathrm{e}}_p(\phi'_i, s_i) - \phi_o)$, and

$$R_p(\phi_i, \phi_o) = \int_{\circledcirc} \int_{\phi'_i} A_p(\phi'_i, s_i) K^{\gamma_p}(\phi^{\mathrm{e}}_p(\phi'_i, s_i) - \phi_o) K^{\gamma_p}(\phi'_i - \phi_i) \ \mathrm{d}s_i \mathrm{d}\phi'_i = R_p(\phi_o, \phi_i)$$

as required. □

To prove that our ASF is energy conserving, we first interpret the blurred response function as a response of the cross section to some incoming light distribution convolved with the kernel (and thus its name).

Lemma A.2. *Let ϕ_i and ϕ_o be fixed azimuthal angles. Let L_i be the incoming radiance distribution in the FRS such that $L_i(\phi'_i, s_i) = K^{\gamma_p}(\phi'_i - \phi_i)$ for all ϕ'_i and s_i. Let $L_o(\phi'_o, s_o)$ be the outgoing radiance distribution in the FRS that is the result of the fiber scattering L_i, and let $\overline{L}_o(\phi'_o)$ be its associated curve radiance* *distribution. Then,*

$$\begin{aligned} R_p(\phi_i, \phi_o) = \overline{L}_o * K^{\gamma_p} &= \int_{\circledcirc} \overline{L}_o(\phi_o') K^{\gamma_p}(\phi_o' - \phi_o) \ \mathrm{d}\phi_o' \\ &= \int_{\circledcirc} \int_{\phi_o'} L_o(\phi_o, s_o) K^{\gamma_p}(\phi_o' - \phi_o) \ \mathrm{d}s_o \mathrm{d}\phi_o'. \end{aligned}$$

Proof. Because L_o is the response of the cross section to L_i, we have that

$$\begin{aligned} L_o(\phi_o', s_o) &= \int_{\circledcirc} \int_{\phi_i'} A_p(\phi_i', s_i) \delta(\phi_o' - \phi_p^{\mathrm{e}}(\phi_i', s_i)) \delta(s_o - s_p^{\mathrm{e}}(\phi_i', s_i)) L_i(\phi_i', s_i) \ \mathrm{d}s_i \mathrm{d}\phi_i \\ &= \int_{\circledcirc} \int_{\phi_i'} A_p(\phi_i', s_i) K^{\gamma_p}(\phi_i' - \phi_i) \delta(\phi_o' - \phi_p^{\mathrm{e}}(\phi_i', s_i)) \delta(s_o - s_p^{\mathrm{e}}(\phi_i', s_i)) \ \mathrm{d}s_i \mathrm{d}\phi_i. \end{aligned}$$

From the last proof, we know that $L_o(\phi_o', s_o) = A_p(\phi_o', s_o) K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi_o', s_o) - \phi_i)$. Substituting, we have

$$\begin{aligned} &\int_{\circledcirc} \int_{\phi_o'} L_o(\phi_o, s_o) K^{\gamma_p}(\phi_o' - \phi_o) \ \mathrm{d}s_o \mathrm{d}\phi_o' \\ &= \int_{\circledcirc} \int_{\phi_o'} A_p(\phi_o', s_o) K^{\gamma_p}(\phi_p^{\mathrm{e}}(\phi_o', s_o) - \phi_i) K^{\gamma}(\phi_o' - \phi_o) \ \mathrm{d}s_o \mathrm{d}\phi_o' \\ &= R_p(\phi_i, \phi_o) \end{aligned}$$

as claimed. □

The next lemma gives the normalization constant we need to define the ASF. Before proceeding, we note that in the 2D world setting, the curve irradiance $\overline{E}$ is given by

$$\overline{E} = \int_{\circledcirc} \overline{L}(\phi) \ \mathrm{d}\phi = \int_{\circledcirc} \int_{\phi} L(\phi, s) \ \mathrm{d}s \mathrm{d}\phi.$$

Lemma A.3.

$$\int_{\circledcirc} R_p(\phi_i, \phi_o) \ \mathrm{d}\phi_o \leq D^{\gamma_p}(\phi_i).$$

Proof. From Lemma A.2, we have that

$$\begin{aligned}\int_{\circledcirc} R_p(\phi_i,\phi_o)\ \mathrm{d}\phi_o &= \int_{\circledcirc}\int_{\circledcirc} \overline{L}_o(\phi_o')K^{\gamma_p}(\phi_o'-\phi_o)\ \mathrm{d}\phi_o'\mathrm{d}\phi_o \\ &= \int_{\circledcirc} \overline{L}_o(\phi_o')\bigg(\int_{\circledcirc} K^{\gamma_p}(\phi_o'-\phi_o)\mathrm{d}\phi_o\bigg)\ \mathrm{d}\phi_o' \\ &= \int_{\circledcirc} \overline{L}_o(\phi_o')\ \mathrm{d}\phi_o' = \overline{E}_o\end{aligned}$$

where $\overline{E}_o$ is the outgoing curve irradiance. Because the attenuation function never increases energy A_p, we have that the outgoing radiance must be less than the incoming curve radiance E_i. Hence,

$$\begin{aligned}\int_{\circledcirc} R_p(\phi_i,\phi_o)\ \mathrm{d}\phi_o \leq E_i &= \int_{\circledcirc}\int_{\phi_i'} L_i(\phi_i',s_i)\ \mathrm{d}s_i\mathrm{d}\phi_i' \\ &= \int_0^{2\pi}\int_{-D(\phi_i')/2}^{D(\phi_i)} K^{\gamma_p}(\phi_i'-\phi_i)\ \mathrm{d}s_i\mathrm{d}\phi_i' \\ &= \int_0^{2\pi} D(\phi_i')K^{\gamma_p}(\phi_i'-\phi_i)\ \mathrm{d}\phi_i' \\ &= D^{\gamma_p}(\phi_i).\end{aligned}$$

Theorem A.4. *The ASF $N_p(\phi_i,\phi_o) = R_p(\phi_i,\phi_o)/D^{\gamma_p}(\phi_i)$ is energy conserving in the sense that*

$$\int_0^{2\pi} N_p(\phi_i,\phi_o)\ \mathrm{d}\phi_o \leq 1$$

for all ϕ_i. It is also approximately reciprocal in the sense that

$$\frac{N_p(\phi_i,\phi_o)}{D^{\gamma_p}(\phi_o)} = \frac{N_p(\phi_o,\phi_i)}{D^{\gamma_p}(\phi_i)}.$$

Proof. By Lemma A.3, we have that

$$\int_0^{2\pi} N_p(\phi_i,\phi_o)\ \mathrm{d}\phi_o = \frac{1}{D^{\gamma_p}(\phi_i)}\int_0^{2\pi} R_p(\phi_i,\phi_o)\ \mathrm{d}\phi_o \leq \frac{D^{\gamma_p}(\phi_i)}{D^{\gamma_p}(\phi_i)} = 1.$$

Moreover, because R_p is reciprocal, we have

$$\frac{N_p(\phi_i,\phi_o)}{D^{\gamma_p}(\phi_o)} = \frac{R_p(\phi_i,\phi_o)}{D^{\gamma_p}(\phi_i)D^{\gamma_p}(\phi_o)} = \frac{R_p(\phi_o,\phi_i)}{D^{\gamma_p}(\phi_o)D^{\gamma_p}(\phi_i)} = \frac{N_p(\phi_o,\phi_i)}{D^{\gamma_p}(\phi_i)}.$$

as claimed. □

A.3 Origin of Perpendicular Blobs in the TT Mode

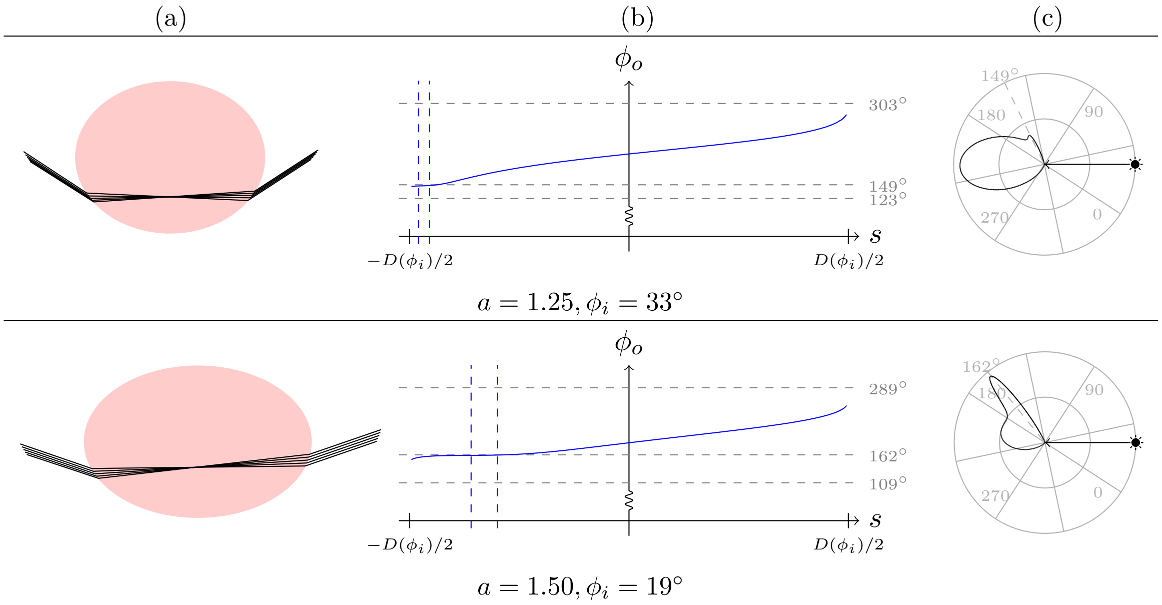
In Section 4.5, we observe that the TT mode of an elliptical fiber contains two features: the "parallel strips" (⫽) and "perpendicular blobs" (⋱) that appear for limited ranges of ϕ_i. This perpendicular blobs can be explained by the fact that elliptical cross sections causes a group of incoming rays to have very similar outgoing directions as seen in Column (a) of Figure A.1. The figure also shows plots of s versus the outgoing direction ϕ_o of the ray (ϕ_i, s) in Column (b). We can see that the two s-versus-ϕ_o plots in Figure A.1 have intervals where the curves are relatively flat, implying light energy will concentrate at the azimuthal angle that is the vertical offset of the flat area. Looking at the polar plots in Column (c), we can verify that the location of the peak of the blob coincides with the ϕ_o at which the s-versus-ϕ_o is flat.

While the perpendicular blobs are similar to caustic lobes as defined in [49], we note though that the perpendicular blob is not actually a caustic. The reason is that Marschner et al. defined a caustic as a location where $\mathrm{d}\phi_o/\mathrm{d}s$ is zero. However, the relatively flat areas in the graphs of Figure A.1 do not have slope of exactly 0.

A.4 Image Formation Model for Photographic Acquisition

As noted in Section 4 of the paper, we take three photos at different exposure times and merge the photos into a single HDR image. To do so, we assume that the camera has a linear response curve, and employ the following image formation model:

$$Y = ty + N_r + N_d + N_s$$



We show two situations in which the perpendicular blobs are present in the TT mode. For each situation, we give (a) an illustration of ray paths that have very similar outgoing directions, (b) the plot of s versus $\phi_o = \phi_1^{\mathrm{e}}(\phi_i, s)$ along with the interval where the curve is flat, which roughly corresponds to the group of rays in (a), and (c) the polar plot of the ASF as a function of ϕ_o.

Figure A.1: Origin of the perpendicular blobs in the TT mode.

where

- Y denotes the measured pixel value,
- y denotes the power coming into the sensor, which is the quantity we want to infer,
- N_r denotes the read noise,
- N_d is the dark current noise, and
- N_s denotes the shot noise.

We chose to use the above image formation model because our photographs are usually dark and thus contaminated significantly by dark current noise and read noise. The noise terms are modeled as follows:

- The read noise N_r is modeled as a Gaussian random variable with mean μ_r and variance σ_r^2.
- The dark current noise N_d is modeled as a Gaussian random variable with mean $\mu_d t$ and variance $\sigma_d^2 t$.
- The shot noise is modeled as another Gaussian random variable with mean 0 and variance $\sigma_s^2 ty$.

We performed a number of experiments and numerical optimizations to determine values of the above parameters. The values are: $\mu_r = 1.708274$, $\sigma_r^2 = 4.454967$, $\mu_d = 0.014256$, $\sigma_d^2 = 0.039172$, and $\sigma_s^2 = 0.423542$. The units of these values are based on pixel intensity. (For example, μ_d has the unit of pixel intensity per second.)

We now describe the HDR merging process. Suppose we take n photographs with exposure times t_1, t_2, $\ldots$, t_n with the corresponding readouts y_1, y_2, $\ldots$, y_n. Also, let Y_i denote the random variable corresponding to the ith exposure. Assuming that the measurements are independent, the probability of the readouts given the parameters is

$$\Pr(y_1, \ldots, y_n | y) = \prod_{i=1}^{n} \Pr(Y_i = y_i) \Pr(y_1, \ldots, y_n | y) = \prod_{i=1}^{n} \Pr(N_i = y_i - t_i y)$$

where N_i is the noise component of Y_i, which is a Gaussian random variable with mean $\mu_r + t_i \mu_d$ and variance $\sigma_r^2 + \sigma_d^2 t + \sigma_s^2 t_i y$ as noted above. As a result,

$$\Pr(y_1, \ldots, y_n | y) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi(\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 t_i y)}} \exp\left(\frac{-(y_i - t_i y - \mu_r - t_i \mu_d)^2}{2(\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 t_i y)} \right).$$

Merging the photos into a single HDR image is just finding y that yields the maximum value of the above conditional probability. This can be done through a numerical optimization at each pixel, but can be very time consuming given the number of pixels and photos we have to process. To make the calculation faster, we assume that $y_i \approx t_i y$ (i.e., the noise is small compared to the readouts) and changes the variances of the noise components to be $\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i$. This leads to the following approximation:

$$\Pr(y_1, \ldots, y_n | y) \approx \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi(\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i)}} \exp\left(\frac{-(y_i - t_i y - \mu_r - t_i \mu_d)^2}{2(\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i)}\right).$$

Thus, the log likelihood has the form:

$$\log \Pr(y_1, \ldots, y_n | y) = \sum_{i=1}^{n} \frac{-(y_i - t_i y - \mu_r - t_i \mu_d)^2}{2(\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i)} + C.$$

Taking the derivative with respect to y, we have:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}s}(\log \Pr(y_1, \ldots, y_n | y)) &= -\sum_{i=1}^{n} \frac{(y_i - t_i y - \mu_r - t_i \mu_d) t_i}{\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i} \\ &= \left(\sum_{i=1}^{n} \frac{t_i^2}{\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i}\right) y - \left(\sum_{i=1}^{n} \frac{(y_i - \mu_r - t_i \mu_d) t_i}{\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i}\right).\end{aligned}$$

Setting the derivative equal to 0, we have that

$$y = \frac{\sum_{i=1}^{n} (y_i - \mu_r - t_i \mu_d) t_i / (\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i)}{\sum_{i=1}^{n} t_i^2 / (\sigma_r^2 + \sigma_d^2 t_i + \sigma_s^2 y_i)}.$$

Note that this estimate is biased because we made the approximation $y_i \approx t_i y$, but we use it because of its expediency and ease of implementation.

APPENDIX B

SUPPLEMENTARY MATERIAL FOR CHAPTER 5

B.1 Evaluation Points for Parameter Rescaling Curves

In Section 5.5.5, we define a rescaling function r_p for each parameter p in the hope that, after applying each to the corresponding parameter, it becomes easier to find a good learning rate for the optimization. The definition of r_p requires an increasing sequence of values c_1, c_2, $\ldots$, c_k, where $[c_1, c_k]$ should cover the domain of parameter p. In this section, we discuss what this increasing sequence is for each parameter.

In general, c_i takes the form $c_i = a + b \cdot m^{i-1}$ for some constants a, b, and m. We fix $k = 9$. We use the aforementioned formula to compute the elements only from c_2 to c_8. We set c_1 and c_9 to be the lower bound and the upper bound of the parameter's domain, respectively. The values of the constants are given in the table below:

Parameter	a	b	m	c_1	c_9
C_R^*	1.02296	-0.032255	1.5403	0.001	0.999
β_R	$-7.0°$,	$8.0°$	1.1	$1°$	$10.0°$
C_{TT}	1.02296,	-1.02197	0.64922	0.001	0.999
β_{TT}	$-0.91°$,	$1.9°$	1.5	$1°$	$45.0°$
γ_{TT}	$-0.91°$,	$1.9°$	1.5	$1°$	$45.0°$

*We use $1 - (a + b \cdot m^{i-1})$ instead of $a + b \cdot m^{i-1}$.

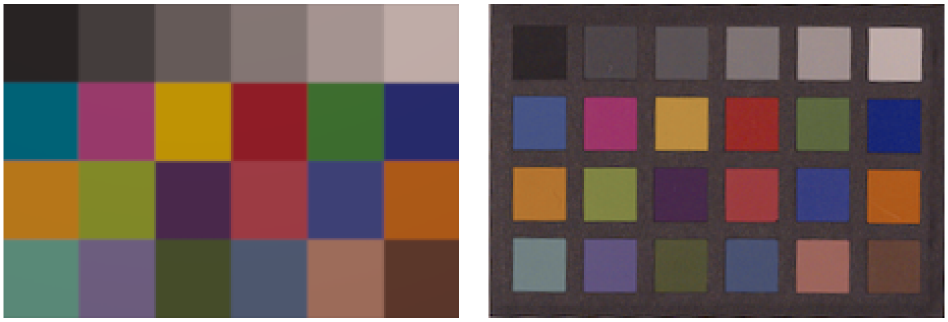
The values computed from the above process are given in the following table:

	C_R	β_R	C_{TT}	β_{TT}	γ_{TT}
c_1	0.001000	1.000000	0.001000	1.000000	1.000000
c_2	0.026722	1.800000	0.359470	1.940000	1.940000
c_3	0.053566	2.680000	0.592206	3.365000	3.365000
c_4	0.094913	3.648000	0.743304	5.502500	5.502500
c_5	0.158599	4.712800	0.841401	8.708750	8.708750
c_6	0.256696	5.884080	0.905087	13.518125	13.518125
c_7	0.407794	7.172488	0.946434	20.732187	20.732187
c_8	0.640530	8.589737	0.973278	31.553281	31.553281
c_9	0.999000	10.000000	0.999000	45.000000	45.000000

However, we stated that the range of β_{TT} is from 10° to 45°. This means that, when performing optimization, we clamp the β_{TT} value to 10° if it gets lower than 10°. The scaling curve is independent of this clamping and was computed using the above sequence of numbers.

B.2 Processing of Cloth Photographs

Here, we briefly describe the processing we performed on our photographs so that they could be compared directly to renderings of our cloth models. We began by exporting linearized images from the camera raw data for each photograph. After normalizing for differences in exposure and ISO between photographs, we derived a color matrix to account for characteristics of the camera sensor and the light source as follows. We photographed a Macbeth chart placed on our measurement apparatus and rendered a corresponding model of the chart under the same geometric configuration with accurate spectral reflectance.



Rendering Photograph

Figure B.1: Rendered Macbeth chart and photograph after applying color matrix.

We then fit a matrix to apply to our photographs to minimize the sum of squared differences between the chart colors in the photo (after applying the matrix) and the render, constrained to exactly match the color of the white square of the rendered chart. The resulting color match is shown in Figure B.1.

APPENDIX C

SUPPLEMENTARY MATERIAL FOR CHAPTER 6

C.1 SG Sharpness Values for Precomputation

In Section 6.4.3, we generate a table $\mathcal{C}_G$ of projections of SG lights of various sharpness λ into the SH basis. This section details how we choose the sequence of λ values to create the table.

We do so with the help of the mass of an SG. Recall that the mass of the SG is given by the integral of the SG over the sphere:

$$M(\lambda) = \int_{\mathbb{S}^2} G(\omega; \xi, \lambda)\ \mathrm{d}\omega = \begin{cases} \frac{2\pi}{\lambda}(1 - e^{-2\lambda}), & \lambda > 0 \\ 4\pi, & \lambda = 0 \end{cases}.$$

We pick a sequence of λ so that the masses of the SGs roughly follow an arithmetic sequence from $M(0) = 4\pi$ to $M(\infty) = 0$. That is, if the table has $m + 1$ entries, the sequence $\lambda_0, \lambda_1, \ldots, \lambda_m$ is given by:

$$M(\lambda_i) \approx \frac{m - i}{m} M(0).$$

That is,

$$\lambda_i \approx M^{-1}\biggl(\frac{4\pi(m - i)}{m}\biggr).$$

Computing the inverse of M is not trivial when the given mass is small. So, instead we approximate M with the following function:

$$\widetilde{M}(\lambda) = \begin{cases} 2\pi[0.5 + 1.5((\lambda - 2)/2)^2], & 0 \le \lambda \le 2 \\ 4\pi/\lambda, & \lambda > 2 \end{cases}.$$

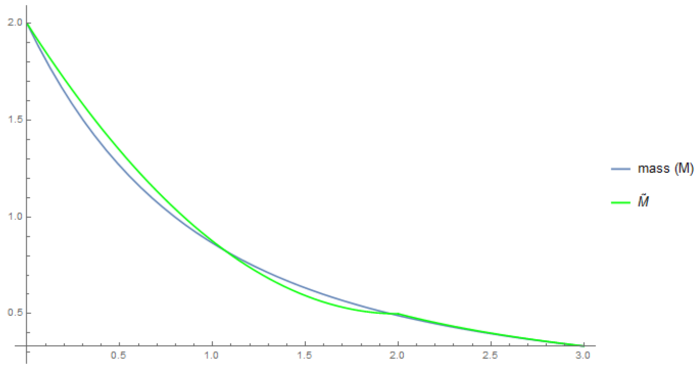


Figure C.1: The graph of $M(\lambda)$ and $\widetilde{M}(\lambda)$.

The graph of M and $\widetilde{M}$ are given in Figure C.1. We see that $\widetilde{M}$ approximates M well when $\lambda \geq 2$, but not quite so when $\lambda < 2$. Nevertheless, the inverse of $\widetilde{M}$ is very easy to compute:

$$\widetilde{M}^{-1}(y) = \begin{cases} 2\pi/y, & y < \pi \\ 2 - 2\sqrt{(y/(2\pi) - 0.5)/1.5}, & y \geq \pi \end{cases}.$$

More concretely, when precomputing the table $\mathcal{C}_G$, we first compute the sequence λ_0, λ_1, $\dots$, λ_m where:

$$\lambda_i = \widetilde{M}^{-1}\left(\frac{4\pi(m-i)}{m}\right).$$

Then, we produce to compute the table entries

$$\mathcal{C}_G[\lambda_i, j] = \int_{\mathbb{S}^2} \frac{G(\omega; (0,0,1), \lambda_i)}{M(\lambda_i)} Y_j(\omega)\ \mathrm{d}\omega$$

with Monte Carlo integration.

At render time, we're given an unseen λ. We solve for the index i of the table cell such that λ_i is the greatest lower bound of λ:

$$i = \mathrm{floor}\left(m - \frac{m}{4\pi}\widetilde{M}(\lambda)\right).$$

We then linearly interpolate the table entries with the following equation:

$$\text{interpolated value} \approx (1-\alpha)\mathcal{C}_G[\lambda_i, j] + \alpha \mathcal{C}_G[\lambda_{i+1}, j]$$

where

$$\alpha = \frac{M(\lambda_i) - M(\lambda)}{M(\lambda_i) - M(\lambda_{i+1})}.$$

Note that we use the masses instead of the λs to compute the interpolation factor.

C.2 Shading Unoccluded, Sharp Spherical Gaussian

Section 6.5.2 discusses how to compute the triple product integral between an SG light, the BCSDF, and the segment visibility function (SVF). It starts the discussion with an approximation to the double product integral between the BCSDF and a sharp SG light with $\lambda \geq 100$. In this section, we give details on how the convolution is carried out. In particular, we give the full forms of the functions B_R and B_{TT} that appear in the approximation.

First, let us establish the BCSDF that we are going to convolve. We use the one described in Chapter 5, and it has the following parameters:

- C_R: the color of the R mode,
- β_R: the longitudinal lobe width of the R mode,
- C_{TT}: the color of the TT mode,
- β_{TT}: the longitudinal lobe width of the TT mode, and
- γ_{TT}: the azimuthal lobe width of the TT mode.

The BCSDF is given by:

$$S(\omega_i,\omega_o) = S_R(\omega_i,\omega_o) + S_{TT}(\omega_i,\omega_o)$$

where

$$\begin{aligned}
S_R(\omega_i,\omega_o) &= \frac{1}{2\pi}\frac{\mathscr{F}_R(\theta_i)}{\mathscr{G}(-\theta_i;\beta_R^{-2}/2))}\frac{g(\theta_o;-\theta_i,\beta_R^{-2}/2)}{\sqrt{2\pi}\beta_R}\\
S_{TT}(\omega_i,\omega_o) &= \frac{(1-\mathscr{F}_R(\theta_i))C_{TT}}{\mathscr{G}(-\theta_i;\beta_{TT}^{-2}/2)}\frac{g(\theta_o;-\theta_i,\beta_{TT}^{-2}/2)}{\sqrt{2\pi}\beta_{TT}}\frac{\exp(\gamma_{TT}^{-2}\cos(\phi_o-\phi_i-\pi))}{2\pi I_o(\gamma_{TT}^{-2})}\\
\mathscr{F}_R(\theta_i) &= C_R - (1-C_R)(1-\cos\theta_i)^5\\
\mathscr{G}(\mu;\lambda) &= \sqrt{\frac{\lambda}{\pi}}\int_{-\pi/2}^{\pi/2} g(\theta;\mu,\lambda)\cos^2\theta\ \mathrm{d}\theta\\
g(x;\mu,\lambda) &= \exp(-\lambda(x-\mu)^2)
\end{aligned}$$

Note that I_0 denotes the 0th modified Bessel function of the first kind.

Now, we would like to compute the double product integral:

$$\begin{aligned}
&\int_{\mathbb{S}^2} G(\omega_i,\xi,\lambda)S(\omega_i,\omega_o)\cos\theta_i\ \mathrm{d}\omega_i\\
&= \int_{\mathbb{S}^2} G(\omega_i,\xi,\lambda)S_R(\omega_i,\omega_o)\cos\theta_i\ \mathrm{d}\omega_i + \int_{\mathbb{S}^2} G(\omega_i,\xi,\lambda)S_{TT}(\omega_i,\omega_o)\cos\theta_i\ \mathrm{d}\omega_i.
\end{aligned}$$

where $\xi = (\theta',\phi')$ when written in the fiber-based spherical coordinate system [49].

C.2.1 R Mode

We have that:

$$\begin{aligned}
&\int_{\mathbb{S}^2} S_R(\omega_i,\omega_o)G(\omega_i;\xi,\lambda)\cos\theta_i\ \mathrm{d}\omega\\
&= \int_{\mathbb{S}^2} \frac{\mathscr{F}_R(\theta_i)}{2\pi\mathscr{G}(-\theta_i;\beta_R^{-2}/2)}\frac{g(\theta_i;-\theta_o,\beta_R^{-2}/2)}{\sqrt{2\pi}\beta_R}G(\omega_i;\xi,\lambda)\cos\theta_i\ \mathrm{d}\omega_i\\
&= \int_{-\pi/2}^{\pi/2}\int_0^{2\pi} \frac{\mathscr{F}_R(\theta_i)}{2\pi\mathscr{G}(-\theta_i;\beta_R^{-2}/2)}\frac{g(\theta_i;-\theta_o,\beta_R^{-2}/2)}{\sqrt{2\pi}\beta_R}G(\omega_i;\xi,\lambda)\cos^2\theta_i\ \mathrm{d}\phi_i\mathrm{d}\theta_i.
\end{aligned}$$

Using Lemma C.8, we have that

$$\begin{aligned}
&\int_{\mathbb{S}^2} S_R(\omega_i,\omega_o)G(\omega_i;\xi,\lambda)\cos\theta_i \ \mathrm{d}\omega_i \\
&= \int_{-\pi/2}^{\pi/2}\int_0^{2\pi} \frac{\mathscr{F}_R(\theta_i)}{2\pi\mathscr{G}(-\theta_i;\beta_R^{-2}/2)}\frac{g(\theta_i;-\theta_o,\beta_R^{-2}/2)}{\sqrt{2\pi}\beta_R} g^c(\theta_i;\theta',\lambda)g^c(\phi_i;\phi',\lambda\cos\theta_i\cos\theta')\cos^2\theta_i \ \mathrm{d}\phi_i\mathrm{d}\theta_i \\
&= \int_{-\pi/2}^{\pi/2} g(\theta_i;-\theta_o,\beta_R^{-2}/2)g^c(\theta_i;\theta',\lambda)\frac{\mathscr{F}_R(\theta_i)\cos^2\theta_i}{2\pi\sqrt{2\pi}\beta_R\mathscr{G}(-\theta_i;\beta_R^{-2}/2)}\left(\int_0^{2\pi} g^c(\phi_i;\phi',\lambda\cos\theta_i\cos\theta') \ \mathrm{d}\phi_i\right) \mathrm{d}\theta_i
\end{aligned}$$

By Lemma C.1,

$$\int_0^{2\pi} g^c(\phi_i;\phi',\lambda\cos\theta_i\cos\theta') \ \mathrm{d}\phi_i = \frac{2\pi I_0(\lambda\cos\theta_i\cos\theta')}{e^{\lambda\cos\theta_i\cos\theta'}}.$$

So,

$$\begin{aligned}
&\int_{\mathbb{S}^2} S_R(\omega_i,\omega_o)G(\omega_i;\xi,\lambda)\cos\theta_i \ \mathrm{d}\omega_i \\
&= \int_{-\pi/2}^{\pi/2} g(\theta_i;-\theta_o,\beta_R^{-2}/2)g^c(\theta_i;\theta',\lambda)\frac{\mathscr{F}_R(\theta_i)\cos^2\theta_i}{2\pi\sqrt{2\pi}\beta_R\mathscr{G}(-\theta_i;\beta_R^{-2}/2)}\frac{2\pi I_0(\lambda\cos\theta_i\cos\theta')}{e^{\lambda\cos\theta_i\cos\theta'}} \ \mathrm{d}\theta_i \\
&= \int_{-\pi/2}^{\pi/2} g(\theta_i;-\theta_o,\beta_R^{-2}/2)g^c(\theta_i;\theta',\lambda)\frac{\mathscr{F}_R(\theta_i)\cos^2\theta_i}{\sqrt{2\pi}\beta_R\mathscr{G}(-\theta_i;\beta_R^{-2}/2)}\frac{I_0(\cos\theta_i(\lambda\cos\theta'))}{e^{\cos\theta_i(\lambda\cos\theta')}} \ \mathrm{d}\theta_i.
\end{aligned}$$

Let

$$B_R(\theta_i,\lambda) = \frac{\mathscr{F}_R(\theta_i)\cos^2\theta_i}{\sqrt{2\pi}\beta_R\mathscr{G}(-\theta_i;\beta_R^{-2}/2)}\frac{I_0(\lambda\cos\theta_i)}{e^{\lambda\cos\theta_i}}.$$

We have that the integral becomes:

$$\int_{-\pi/2}^{\pi/2} g(\theta_i;-\theta_o,\beta_R^{-2}/2)g^c(\theta_i;\theta',\lambda)B_R(\theta_i,\lambda\cos\theta') \ \mathrm{d}\theta_i.$$

C.2.2 TT Mode

We have that:

$$
\begin{aligned}
&\int_{\mathbb{S}^2} S_{TT}(\omega_d, \omega_o) G(\omega_d; \xi, \lambda) \cos\theta_i \ \mathrm{d}\omega_d \\
&= \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} \frac{(1 - \mathscr{F}_R(\theta_i))C_{TT}}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2)} \frac{g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2)}{\sqrt{2\pi}\beta_{TT}} \frac{\exp(\gamma_{TT}^{-2}\cos(\phi_o - \phi_i - \pi))}{2\pi I_o(\gamma_{TT}^{-2})} G(\omega_d; \xi, \lambda) \cos^2\theta_i \ \mathrm{d}\phi_i \mathrm{d}\theta_i \\
&= \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} \frac{(1 - \mathscr{F}_R(\theta_i))C_{TT}}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2)} \frac{g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2)}{\sqrt{2\pi}\beta_{TT}} \frac{\exp(\gamma_{TT}^{-2}\cos(\phi_i - (\phi_o - \pi)))}{2\pi I_o(\gamma_{TT}^{-2})} G(\omega_d; \xi, \lambda) \cos^2\theta_i \ \mathrm{d}\phi_i \mathrm{d}\theta_i \\
&= \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} \frac{(1 - \mathscr{F}_R(\theta_i))C_{TT}}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2)} \frac{g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2)}{\sqrt{2\pi}\beta_{TT}} \frac{\exp(\gamma_{TT}^{-2}\cos(\phi_i - (\phi_o - \pi)) - 1)}{2\pi I_o(\gamma_{TT}^{-2}) e^{-\gamma_{TT}^2}} G(\omega_d; \xi, \lambda) \cos^2\theta_i \ \mathrm{d}\phi_i \mathrm{d}\theta_i \\
&= \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} \frac{(1 - \mathscr{F}_R(\theta_i))C_{TT}}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2)} \frac{g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2)}{\sqrt{2\pi}\beta_{TT}} \frac{g^c(\phi_i; \phi_o - \pi, \gamma_{TT}^{-2})}{2\pi I_o(\gamma_{TT}^{-2}) e^{-\gamma_{TT}^2}} G(\omega_d; \xi, \lambda) \cos^2\theta_i \ \mathrm{d}\phi_i \mathrm{d}\theta_i \\
&= \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} \frac{(1 - \mathscr{F}_R(\theta_i))C_{TT}}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2)} \frac{g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2)}{\sqrt{2\pi}\beta_{TT}} \frac{g^c(\phi_i; \phi_o - \pi, \gamma_{TT}^{-2})}{2\pi I_o(\gamma_{TT}^{-2}) e^{-\gamma_{TT}^2}} \\
&\qquad\qquad g^c(\theta_i; \theta', \lambda) g^c(\phi_i; \phi', \lambda\cos\theta_i\cos\theta') \cos^2\theta_i \ \mathrm{d}\phi_i \mathrm{d}\theta_i \\
&= \int_{-\pi/2}^{\pi/2} \frac{(1 - \mathscr{F}_R(\theta_i))C_{TT}\cos^2\theta_i}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2) 2\pi\sqrt{2\pi}\beta_{TT} I_o(\gamma_{TT}^{-2}) e^{-\gamma_{TT}^2}} g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2) g^c(\theta_i; \theta', \lambda) \\
&\qquad\qquad \left(\int_0^{2\pi} g^c(\phi_i; \phi_o - \pi, \gamma_{TT}^{-2}) g^c(\phi_i; \phi', \lambda\cos\theta_i\cos\theta') \ \mathrm{d}\phi_i \right) \mathrm{d}\theta_i.
\end{aligned}
$$

Let's work on the integral involving ϕ_i. To simplify matters, let's $\phi_0 = 0$. Using Lemma C.6, we have that:

$$
\begin{aligned}
&g^c(\phi_i; -\pi, \gamma_{TT}^{-2}) g^c(\phi_i; \phi', \lambda\cos\theta_i\cos\theta') \\
&\quad = e^{\lambda_m(\theta_i, \lambda\cos\theta', \phi') - (\gamma_{TT}^{-2} + \lambda\cos\theta'\cos\theta_i)} g^c\big(\phi_i; \phi_m, \lambda_m(\theta_i, \lambda\cos\theta', \phi')\big)
\end{aligned}
$$

where ϕ_m is an angle which doesn't matter in the resulting integral, and λ_m is given by:

$$
\lambda_m(\theta_i, \lambda, \phi) = \sqrt{\gamma_{TT}^{-4} + \lambda^2\cos^2\theta_i - 2\gamma_{TT}^{-2}\lambda\cos\theta_i\cos\phi}.
$$

So,

$$\begin{aligned}
&\int_0^{2\pi} g^c(\phi_i; -\pi, \gamma_{TT}^{-2}) g^c(\phi_i; \phi', \lambda \cos\theta_i \cos\theta')\ \mathrm{d}\phi_i \\
&= e^{\lambda_m(\theta_i, \lambda\cos\theta', \phi') - (\gamma_{TT}^{-2} + \lambda\cos\theta'\cos\theta_i)} \int_0^{2\pi} g^c(\phi_i; \phi_m, \lambda_m(\theta_i, \lambda\cos\theta', \phi'))\ \mathrm{d}\phi_i \\
&= e^{\lambda_m(\theta_i, \lambda\cos\theta', \phi') - (\gamma_{TT}^{-2} + \lambda\cos\theta'\cos\theta_i)} \frac{2\pi I_0(\lambda_m(\theta_i, \lambda\cos\theta', \phi'))}{\exp(\lambda_m(\theta_i, \lambda\cos\theta', \phi'))} \\
&= \frac{2\pi I_0(\lambda_m(\theta_i, \lambda\cos\theta', \phi'))}{\exp(\gamma_{TT}^{-2} + \lambda\cos\theta'\cos\theta_i)}.
\end{aligned}$$

As a result,

$$\begin{aligned}
&\int_{\mathbb{S}^2} S_{TT}(\omega_d, \omega_o) G(\omega_d; \xi, \lambda) \cos\theta_i\ \mathrm{d}\omega_d \\
&= \int_{\mathbb{S}^2} g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2) g^c(\theta_i; \theta', \lambda) \\
&\quad \left(\frac{(1 - \mathscr{F}_R(\theta_i)) C_{TT} \cos^2\theta_i}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2) 2\pi\sqrt{2\pi}\beta_{TT} I_o(\gamma_{TT}^{-2}) e^{-\gamma_{TT}^{2}}} \frac{2\pi I_0(\lambda_m(\theta_i, \lambda\cos\theta', \phi'))}{\exp(\gamma_{TT}^{-2} + \lambda\cos\theta'\cos\theta_i)} \right) \mathrm{d}\theta_i.
\end{aligned}$$

In the same way as the previous section, we let:

$$\begin{aligned}
B_{TT}(\theta_i, \lambda, \phi) &= \frac{(1 - \mathscr{F}_R(\theta_i)) C_{TT} \cos^2\theta_i}{\mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2) 2\pi\sqrt{2\pi}\beta_{TT} I_o(\gamma_{TT}^{-2}) e^{-\gamma_{TT}^{2}}} \frac{2\pi I_0(\lambda_m(\theta_i, \lambda, \phi))}{\exp(\gamma_{TT}^{-2} + \lambda\cos\theta_i)} \\
&= \frac{(1 - \mathscr{F}_R(\theta_i)) C_{TT} \cos^2\theta_i}{\sqrt{2\pi}\beta_{TT} \mathscr{G}(-\theta_i; \beta_{TT}^{-2}/2) I_o(\gamma_{TT}^{-2})} \frac{I_0(\lambda_m(\theta_i, \lambda, \phi))}{\exp(\lambda\cos\theta_i)}
\end{aligned}$$

so that the integral becomes:

$$\int_{\mathbb{S}^2} g(\theta_i; -\theta_o, \beta_{TT}^{-2}/2) g^c(\theta_i; \theta', \lambda) B_{TT}(\theta_i, \lambda\cos\theta', \phi')\ \mathrm{d}\theta_i.$$

C.3 Some Useful Mathematical Identities

In this section, we list some identities involving Gaussian-like functions and prove some of them. These identities are used throughout this appendix and in Section 6.5.2. Important lemmas include:

- Lemma C.1 gives the masses of spherical Gaussians and circular Gaussians.
- Lemma C.5 shows that the product of two spherical Gaussians is another spherical Gaussian. It is used in Section 6.5.2.
- Lemma C.6 shows that the product of two circular Gaussians is another circular Gaussian. It is used in Section C.2.
- Lemma C.7 shows that the product of two (ordinary) Gaussians is another (ordinary) Gaussian. It is used in Section 6.5.2.
- Lemma C.8 allows us to decompose a spherical Gaussian to a product of two circular Gaussians. It is what makes the algebraic manipulation in Section C.2 possible.
- Lemma C.9 states that a circular Gaussian may be approximated by an ordinary Gaussian. It is used in Section 6.5.2.

These lemmas are used in many previous works such as [93] and [31], often without proof. We collect them here in the hopes that other practitioners might find them useful.

In this dissertation, we deal with a number of Gaussian functions. They are:

$$G(\omega; \xi, \lambda) = \exp(\lambda(\omega \cdot \xi - 1)) \qquad \text{(spherical Gaussian)}$$

$$g^c(x; \mu, \lambda) = \exp(\lambda(\cos(x - \mu) - 1)) \qquad \text{(circular Gaussian)}$$

$$g(x; \mu, \lambda) = \exp(-\lambda(x - \mu)^2) \qquad \text{(ordinary Gaussian)}$$

There are also related directional probability distributions.

$$f_{\text{vMF}}(\omega; \xi, \lambda) = \frac{\lambda \exp(\lambda(\omega \cdot \xi))}{4\pi \sinh(\lambda)} \qquad \text{(von Mises–Fisher distribution)}$$

$$f_{\text{vM}}(x; \mu, \lambda) = \frac{\exp(\lambda \cos(x - \mu))}{2\pi I_0(\lambda)} \qquad \text{(von Mises distribution)}$$

The first thing to notice is that the above distributions are Gaussians in disguise:

$$\begin{aligned}
G(\omega;\xi,\lambda) &= \exp(\lambda(\omega\cdot\xi-1)) = e^{-\lambda}\exp(\lambda(\omega\cdot\xi)) \\
&= \frac{4\pi\sinh(\lambda)e^{-\lambda}}{\lambda}\frac{\lambda\exp(\lambda(\omega\cdot\xi))}{4\pi\sinh(\lambda)} = \frac{4\pi}{\lambda}\frac{e^{\lambda}-e^{-\lambda}}{2}e^{-\lambda}f_{\mathrm{vMF}}(\omega;\xi,\lambda) \\
&= \frac{2\pi}{\lambda}(1-e^{-2\lambda})f_{\mathrm{vMF}}(\omega;\xi,\lambda) \\
g^c(x;\mu,\lambda) &= \exp(\lambda(\cos(x-\mu)-1)) = e^{-\lambda}\exp(\lambda\cos(x-\mu)) \\
&= 2\pi I_0(\lambda)e^{-\lambda}\frac{\exp(\lambda\cos(x-\mu))}{2\pi I_0(\lambda)} \\
&= 2\pi I_0(\lambda)e^{-\lambda}f_{\mathrm{vM}}(x;\mu,\lambda)
\end{aligned}$$

Lemma C.1. *We have that*

$$\begin{aligned}
\int_{\mathbb{S}^2} G(\omega;\xi,\lambda)\ \mathrm{d}\omega &= \frac{2\pi}{\lambda}(1-e^{-2\lambda}) \\
\int_0^{2\pi} g^c(x;\mu,\lambda)\ \mathrm{d}x &= 2\pi I_0(\lambda)e^{-\lambda}
\end{aligned}$$

Lemma C.2.

$$G(\omega;\xi,\lambda) = \exp\left(-\frac{\lambda}{2}\|\omega-\xi\|^2\right) = \exp\left(-\frac{\lambda}{2}(\omega-\xi)\cdot(\omega-\xi)\right)$$

Proof. We have that

$$\begin{aligned}
\omega\cdot\xi-1 = -(1-\omega\cdot\xi) &= -\frac{2-2(\omega\cdot\xi)}{2} = -\frac{(\omega\cdot\omega)+(\xi\cdot\xi)-2(\omega\cdot\xi)}{2} \\
&= -\frac{(\omega-\xi)\cdot(\omega-\xi)}{2} = -\frac{\|\omega-\xi\|^2}{2}.
\end{aligned}$$

The rest is obvious. □

Definition C.3. *Let $\mathfrak{p}(\theta)$ denote the unit vector*

$$\begin{bmatrix} 0 \\ \cos\theta \\ \sin\theta \end{bmatrix}.$$

Lemma C.4.

$$g^c(x;\mu,\lambda) = G(\mathfrak{p}(x);\mathfrak{p}(\mu),\lambda)$$

Proof. The lemma follows from the fact that:

$$\mathfrak{p}(x)\cdot\mathfrak{p}(\mu) - 1 = \cos x\cos\mu + \sin x\sin\mu - 1 = \cos(x-\mu) - 1.$$

Lemma C.5.

$$G(\omega_1;\xi_1,\lambda_1)G(\omega_2,\xi_2,\lambda_2) = e^{\|\lambda_1\xi_1+\lambda_2\xi_2\|-(\lambda_1+\lambda 2)\|}G\bigg(\omega;\frac{\lambda_1\xi_1+\lambda_2\xi_2}{\|\lambda_1\xi_1+\lambda_2\xi_2\|},\|\lambda_1\xi_1+\lambda_2\xi_2\|\bigg).$$

Proof.

$$\begin{aligned}
&G(\omega;\xi_1,\lambda_1)G(\omega,\xi_2,\lambda_2)\\
&=\exp(\lambda_1(\omega\cdot\xi_1-1))\exp(\lambda_2(\omega\cdot\xi_2-1))\\
&=\exp(\lambda_1(\omega\cdot\xi_1-1)+\lambda_2(\omega\cdot\xi_2-1))\\
&=\exp(\omega\cdot(\lambda_1\xi_1)+\omega\cdot(\lambda_2\xi_2)-\lambda_1-\lambda_2)\\
&=\exp\left(\|\lambda_1\xi_1+\lambda_2\xi_2\|\bigg(\omega\cdot\frac{\lambda_1\xi_1+\lambda_2\xi_2}{\|\lambda_1\xi_1+\lambda_2\xi_2\|}\bigg)-(\lambda_1+\lambda_2)\right)\\
&=\exp\left(\|\lambda_1\xi_1+\lambda_2\xi_2\|\bigg(\omega\cdot\frac{\lambda_1\xi_1+\lambda_2\xi_2}{\|\lambda_1\xi_1+\lambda_2\xi_2\|}-1\bigg)+\|\lambda_1\xi_1+\lambda_2\xi_2\|-(\lambda_1+\lambda_2)\right)\\
&=e^{\|\lambda_1\xi_1+\lambda_2\xi_2\|-(\lambda_1+\lambda_2)}G\bigg(\omega;\frac{\lambda_1\xi_1+\lambda_2\xi_2}{\|\lambda_1\xi_1+\lambda_2\xi_2\|},\|\lambda_1\xi_1+\lambda_2\xi_2\|\bigg).
\end{aligned}$$

Lemma C.6.

$$\begin{aligned}
&g^c(x;\mu_1,\lambda_1)g^c(x;\mu_2,\lambda_2)\\
&=e^{\sqrt{\lambda_1^2+\lambda_2^2+2\lambda_1\lambda_2\cos(\mu_1-\mu_2)}-(\lambda_1+\lambda_2)}\\
&\qquad g^c\bigg(x;\tan^{-1}\left(\frac{\lambda_1\sin\mu_1+\lambda_2\sin\mu_2}{\lambda_1\cos\mu_1+\lambda_2\cos\mu_2}\right),\sqrt{\lambda_1^2+\lambda_2^2+2\lambda_1\lambda_2\cos(\mu_1-\mu_2)}\bigg)
\end{aligned}$$

Proof.

$$\begin{aligned}
&g^c(x;\mu_1,\lambda_1)g^c(x;\mu_2,\lambda_2)\\
&\quad= G(\mathfrak{p}(x);\mathfrak{p}(\mu_1),\lambda_1)G(\mathfrak{p}(x);\mathfrak{p}(\mu_2),\lambda_2)\\
&\quad= e^{\|\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)\|-(\lambda_1+\lambda_2)}G\Bigg(\mathfrak{p}(x);\frac{\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)}{\|\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)\|},\|\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)\|\Bigg).
\end{aligned}$$

We have that:

$$\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)=\begin{bmatrix}0\\ \lambda_1\cos\mu_1+\lambda_2\cos\mu_2\\ \lambda_1\sin\mu_1+\lambda_2\sin\mu_2\end{bmatrix}.$$

As such,

$$\begin{aligned}
&\|\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)\|^2\\
&= (\lambda_1\cos\mu_1+\lambda_2\cos\mu_2)^2+(\lambda_1\sin\mu_1+\lambda_2\sin\mu_2)^2\\
&= \lambda_1^2\cos^2\mu_1+\lambda_2^2\cos^2\mu_2+2\lambda_1\lambda_2\cos\mu_1\cos\mu_2+\lambda_1^2\sin^2\mu_1+\lambda_2^2\sin^2\mu_2+2\lambda_1\lambda_2\sin\mu_1\sin\mu_2\\
&= \lambda_1^2+\lambda_2^2+2\lambda_1\lambda_2(\cos\mu_1\cos\mu_2+\sin\mu_1\sin\mu_2)\\
&= \lambda_1^2+\lambda_2^2+2\lambda_1\lambda_2\cos(\mu_1-\mu_2).
\end{aligned}$$

So,

$$\|\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)\|=\sqrt{\lambda_1^2+\lambda_2^2+2\lambda_1\lambda_2\cos(\mu_1-\mu_2)}.$$

Moreover, consider now the unit vector $\frac{\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)}{\|\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)\|}$. The angle that it makes with the y-axis is given by:

$$\tan^{-1}\frac{\lambda_1\sin\mu_1+\lambda_2\sin\mu_2}{\lambda_1\cos\mu_1+\lambda_2\cos\mu_2}.$$

Hence,

$$\frac{\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)}{\|\lambda_1\mathfrak{p}(\mu_1)+\lambda_2\mathfrak{p}(\mu_2)\|}=\mathfrak{p}\Bigg(\tan^{-1}\frac{\lambda_1\sin\mu_1+\lambda_2\sin\mu_2}{\lambda_1\cos\mu_1+\lambda_2\cos\mu_2}\Bigg).$$

In conclusion,

$$
\begin{aligned}
&g^c(x;\mu_1,\lambda_1)g^c(x;\mu_2,\lambda_2) \\
&= e^{\sqrt{\lambda_1^2+\lambda_2^2+2\lambda_1\lambda_2\cos(\mu_1-\mu_2)}-(\lambda_1+\lambda_2)} \\
&\qquad g^c\Bigg(x;\tan^{-1}\left(\frac{\lambda_1\sin\mu_1+\lambda_2\sin\mu_2}{\lambda_1\cos\mu_1+\lambda_2\cos\mu_2}\right),\sqrt{\lambda_1^2+\lambda_2^2+2\lambda_1\lambda_2\cos(\mu_1-\mu_2)}\Bigg)
\end{aligned}
$$

Lemma C.7.

$$
g(x;\mu_1,\lambda_1)g(x;\mu_2,\lambda_2)=\exp\Bigg(-\frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}(\mu_1-\mu_2)^2\Bigg)g\Bigg(x;\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2},\lambda_1+\lambda_2\Bigg)
$$

Proof.

$$
\begin{aligned}
&g(x;\mu_1,\lambda_1)g(x;\mu_2,\lambda_2) \\
&=\exp(-\lambda_1(x-\mu_1)^2)\exp(-\lambda_2(x-\mu_2)^2) \\
&=\exp(-(\lambda_1x^2-2\lambda_1\mu_1x+\lambda_1\mu_1^2+\lambda_2x^2-2\lambda_2\mu_2x+\lambda_2\mu_2^2)) \\
&=\exp\big(-[(\lambda_1+\lambda_2)x^2-2(\lambda_1\mu_1+\lambda_2\mu_2)x+\lambda_1\mu_1^2+\lambda_2\mu_2^2]\big) \\
&=\exp\big(-[(\lambda_1+\lambda_2)x^2-2(\lambda_1\mu_1+\lambda_2\mu_2)x+\lambda_1\mu_1^2+\lambda_2\mu_2^2]\big) \\
&=\exp(-(\lambda_1\mu_1^2+\lambda_2\mu_2^2))\exp\big(-[(\lambda_1+\lambda_2)x^2-2(\lambda_1\mu_1+\lambda_2\mu_2)x]\big) \\
&=\exp(-(\lambda_1\mu_1^2+\lambda_2\mu_2^2))\exp\Bigg(-(\lambda_1+\lambda_2)\Big[x^2-2\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}x\Big]\Bigg) \\
&=\exp(-(\lambda_1\mu_1^2+\lambda_2\mu_2^2))\exp\Bigg(-(\lambda_1+\lambda_2)\Big[x^2-2\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}x+\Big(\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}\Big)^2-\Big(\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}\Big)^2\Big]\Bigg) \\
&=\exp(-(\lambda_1\mu_1^2+\lambda_2\mu_2^2))\exp\Bigg(-(\lambda_1+\lambda_2)\Big[\Big(x-\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}\Big)^2-\Big(\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}\Big)^2\Big]\Bigg) \\
&=\exp(-(\lambda_1\mu_1^2+\lambda_2\mu_2^2))\exp\Bigg(-(\lambda_1+\lambda_2)\Big(x-\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}\Big)^2+\frac{(\lambda_1\mu_1+\lambda_2\mu_2)^2}{\lambda_1+\lambda_2}\Bigg) \\
&=\exp\Bigg(-(\lambda_1\mu_1^2+\lambda_2\mu_2^2)+\frac{(\lambda_1\mu_1+\lambda_2\mu_2)^2}{\lambda_1+\lambda_2}\Bigg)\exp\Bigg(-(\lambda_1+\lambda_2)\Big(x-\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2}\Big)^2\Bigg) \\
&=\exp\Bigg(\frac{-\lambda_1^2\mu_1^2-\lambda_1\lambda_2\mu_1^2-\lambda_1\lambda_2\mu_2^2-\lambda_2^2\mu_2^2+\lambda_1^2\mu_1^2+\lambda_2^2\mu_2^2+2\lambda_1\lambda_2\mu_1\mu_2}{\lambda_1+\lambda_2}\Bigg)g\Bigg(x;\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2},\lambda_1+\lambda_2\Bigg) \\
&=\exp\Bigg(\frac{-\lambda_1\lambda_2\mu_1^2-\lambda_1\lambda_2\mu_2^2+2\lambda_1\lambda_2\mu_1\mu_2}{\lambda_1+\lambda_2}\Bigg)g\Bigg(x;\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2},\lambda_1+\lambda_2\Bigg) \\
&=\exp\Bigg(-\frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}(\mu_1-\mu_2)^2\Bigg)g\Bigg(x;\frac{\lambda_1\mu_1+\lambda_2\mu_2}{\lambda_1+\lambda_2},\lambda_1+\lambda_2\Bigg).
\end{aligned}
$$

□

Lemma C.8.

$$G(\omega;\xi,\lambda) = g^c(\theta;\theta',\lambda)g^c(\phi;\phi',\lambda\cos\theta\cos\theta').$$

Proof. First, let us write:

$$\omega = \begin{bmatrix} \sin\theta \\ \cos\theta\cos\phi \\ \cos\theta\sin\phi \end{bmatrix} \qquad \text{and} \qquad \xi = \begin{bmatrix} \sin\theta' \\ \cos\theta'\cos\phi' \\ \cos\theta'\sin\phi' \end{bmatrix}.$$

We have that

$$\begin{aligned}
\omega\cdot\xi - 1 &= \sin\theta\sin\theta' + \cos\theta\cos\phi\cos\theta'\cos\phi' + \cos\theta\sin\phi\cos\theta'\sin\phi' - 1 \\
&= \sin\theta\sin\theta' + \cos\theta\cos\theta'\cos\phi\cos\phi' + \cos\theta\cos\theta'\sin\phi\sin\phi' - 1 \\
&= \sin\theta\sin\theta' + \cos\theta\cos\theta'(\cos\phi\cos\phi' + \sin\phi\sin\phi') \\
&= \sin\theta\sin\theta' + \cos\theta\cos\theta'\cos(\phi-\phi') - 1 \\
&= \sin\theta\sin\theta' + (\cos\theta\cos\theta' - \cos\theta\cos\theta') + \cos\theta\cos\theta'\cos(\phi-\phi') - 1 \\
&= (\sin\theta\sin\theta' + \cos\theta\cos\theta' - 1) + [\cos\theta\cos\theta'\cos(\phi-\phi') - \cos\theta\cos\theta'] \\
&= (\sin\theta\sin\theta' + \cos\theta\cos\theta' - 1) + \cos\theta\cos\theta'(\cos(\phi-\phi') - 1) \\
&= (\cos(\theta-\theta') - 1) + \cos\theta\cos\theta'(\cos(\phi-\phi') - 1).
\end{aligned}$$

As a result:

$$\begin{aligned}
\lambda(\omega\cdot\xi - 1) &= \lambda(\cos(\theta-\theta') - 1) + \lambda\cos\theta\cos\theta'(\cos(\phi-\phi') - 1) \\
\exp(\lambda(\omega\cdot\xi - 1)) &= \exp(\lambda(\cos(\theta-\theta') - 1)) + \exp(\lambda\cos\theta\cos\theta'(\cos(\phi-\phi') - 1)) \\
G(\omega;\xi,\lambda) &= g^c(\theta;\theta',\lambda)g^c(\phi;\phi',\lambda\cos\theta\cos\theta')
\end{aligned}$$

as desired. □

Lemma C.9. $g^c(x;\mu,\lambda) \approx g(x;\mu,\lambda/2)$

Proof. By graphing. The graphs become closer as λ increases. We have very good approximation when $\lambda \geq 10$, but poor approximation otherwise. See Figure C.2.□

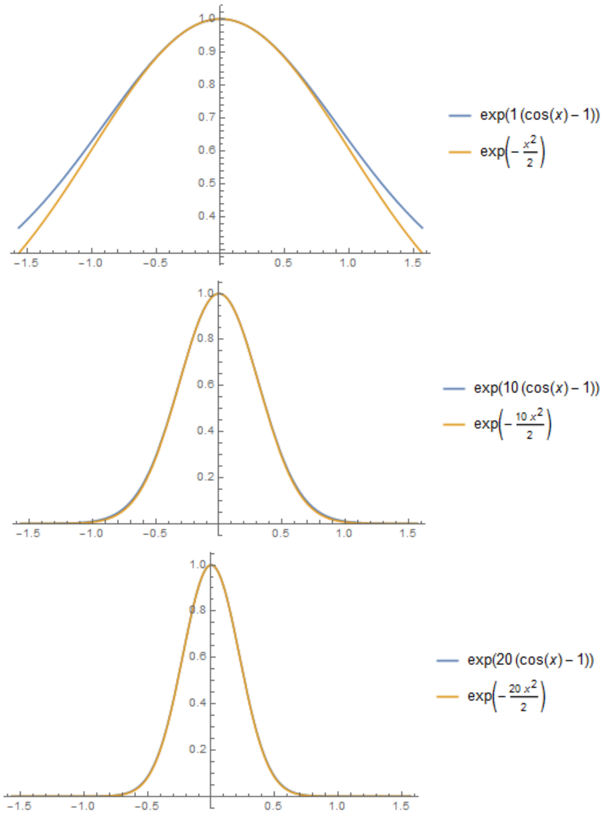


Figure C.2: Circular Gaussians and ordinary Gaussians of various sharpness parameters.

BIBLIOGRAPHY

[1] Neeharika Adabala, Nadia Magnenat-Thalmann, and Guangzheng Fei. Visualization of woven cloth. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pages 178–185, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[2] Charles L. Adler, James A. Lock, and Bradley R. Stone. Rainbow scattering by a cylinder with a nearly elliptical cross section. *Appl. Opt.*, 37(9):1540–1550, Mar 1998.

[3] Thomas Annen, Zhao Dong, Tom Mertens, Philippe Bekaert, Hans-Peter Seidel, and Jan Kautz. Real-time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph.*, 27(3):1–8, 2008.

[4] Peter J. Basser, Sinisa Pajevic, Carlo Pierpaoli, Jeffrey Duda, and Akram Aldroubi. In vivo fiber tractography using DT-MRI data. *Magn. Reson. Med*, 44:625–632, 2000.

[5] Aner Ben-Artzi, Kevin Egan, Frédo Durand, and Ravi Ramamoorthi. A precomputed polynomial representation for interactive brdf editing with global illumination. *ACM Trans. Graph.*, 27(2):13:1–13:13, May 2008.

[6] Nicolas Bonneel, Sylvain Paris, Michiel Van De Panne, Frédo Durand, and George Drettakis. Single photo estimation of hair appearance. *Computer Graphics Forum*, 28(4):1171–1180, 2009.

[7] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France, August 2010. Springer.

[8] Willem F. Bronsvoort and Fopke Klok. Ray tracing generalized cylinders. *ACM Trans. Graph.*, 4(4):291–303, October 1985.

[9] Helen K. Bustard and Robin W. Smith. Investigation into the scattering of light by human hair. *Appl. Opt.*, 30(24):3485–3491, Aug 1991.

[10] Menglei Cai, Changxi Zheng, and Kun Zhou. A reduced model for interactive hairs. *ACM Transactions on Graphics (SIGGRAPH 2014)*, 33(4), August 2014.

[11] Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J. Seron, and François X. Sillion. A survey on participating media rendering techniques. *Vis. Comput.*, 21(5):303–328, June 2005.

[12] Subrahmanyan Chandrasekhar. *Radiative transfer.* Dover publications, New York, 1960. Unabridged and slightly revised version of the work first published in 1950.

[13] Johnny T. Chang, Jingyi Jin, and Yizhou Yu. A practical model for hair mutual interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pages 73–80, New York, NY, USA, 2002. ACM.

[14] D. Cline, A. Razdan, and P. Wonka. A Comparison of Tabular PDF Inversion Methods. *Computer Graphics Forum*, 2009.

[15] Michael F. Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. Wang tiles for image and texture generation. *ACM Trans. Graph.*, 22(3):287–294, July 2003.

[16] Eugene d'Eon, Guillaume Francois, Martin Hill, Joe Letteri, and Jean-Marie Aubry. An energy-conserving hair reflectance model. In *Proceedings of the Twenty-second Eurographics Conference on Rendering*, EGSR '11, pages 1181–1187, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association.

[17] Eugene d'Eon, Guillaume Francois, Martin Hill, Joe Letteri, and Jean-Marie Aubry. An energy-conserving hair reflectance model. *Computer Graphics Forum*, 30(4):1181–1187, 2011.

[18] Eugene d'Eon, Steve Marschner, and Johannes Hanika. A fiber scattering model with non-separable lobes. In *SIGGRAPH 2014 Talks*, pages 46:1–46:1, August 2014.

[19] Doratha E. Drake and Stefan Hougardy. A simple approximation algorithm for the weighted matching problem. *Information Processing Letters*, 85:211–213, 2003.

[20] Phil Dutre, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination, 2nd Edition.* A K Peters, Natick, MA, 2006.

[21] Randima Fernando. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[22] Andrew Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):476–480, May 1999.

[23] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Trans. Graph.*, 32(6):162:1–162:13, November 2013.

[24] Dan B. Goldman. Fake fur rendering. In *SIGGRAPH '97 Proceedings*, pages 127–134, 1997.

[25] Miloš Hašan and Ravi Ramamoorthi. Interactive albedo editing in path-traced volumetric materials. *ACM Trans. Graph.*, 32(2):11:1–11:11, April 2013.

[26] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. *SIGGRAPH Comput. Graph.*, 25(4):175–186, July 1991.

[27] Tomas Lay Herrera, Arno Zinke, and Andreas Weber. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Trans. Graph.*, 31(6):146:1–146:9, November 2012.

[28] Christophe Hery and Ravi Ramamoorthi. Importance sampling of reflection from hair fibers. *Journal of Computer Graphics Techniques (JCGT)*, 1(1):1–17, June 2012.

[29] Piti Irawan and Steve Marschner. Specular reflection from woven cloth. *ACM Trans. Graph.*, 31(1):11:1–11:20, February 2012.

[30] Kei Iwasaki, Wataru Furuya, Yoshinori Dobashi, and Tomoyuki Nishita. Real-time rendering of dynamic scenes under all-frequency lighting using integral spherical gaussian. *Comput. Graph. Forum*, 31(2pt4):727–734, May 2012.

[31] Kei Iwasaki, Kazutaka Mizutani, Yoshinori Dobashi, and Tomoyuki Nishita. Interactive Cloth Rendering of Microcylinder Appearance Model under Environment Lighting. *Computer Graphics Forum*, 2014.

[32] Wenzel Jakob, Adam Arbree, Jonathan T. Moon, Kavita Bala, and Steve Marschner. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Trans. Graph.*, 29(4):53:1–53:13, July 2010.

[33] Wenzel Jakob, Jonathan T. Moon, and Steve Marschner. Capturing hair assemblies fiber by fiber. *ACM Trans. Graph.*, 28(5):164:1–164:9, December 2009.

[34] Stefan Jeschke, Stephan Mantler, and Michael Wimmer. Interactive Smooth and Curved Shell Mapping. In Jan Kautz and Sumanta Pattanaik, editors, *Rendering Techniques*. The Eurographics Association, 2007.

[35] J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. In *SIGGRAPH '89 Proceedings*, pages 271–280, 1989.

[36] James T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986.

[37] Ning Kang and Jun Zhang. White matter fiber tractography via anisotropic diffusion simulation in the human brain. *Medical Imaging, IEEE*, 24(9):1127–37, September 2005.

[38] Pramook Khungurn and Steve Marschner. Azimuthal scattering from elliptical hair fibers. *ACM Trans. Graph.*, 36(2):13:1–13:23, April 2017.

[39] Pramook Khungurn, Daniel Schroeder, Shuang Zhao, Kavita Bala, and Steve Marschner. Matching real fabrics with micro-appearance models. *ACM Trans. Graph.*, 35(1):1:1–1:26, December 2015.

[40] Pramook Khungurn, Rundong Wu, James Noeckel, Steve Marschner, and Kavita Bala. Fast approximate rendering of fabric micro-appearance models under directional and spherical gaussian lights. Unpublished manuscript, 2017.

[41] Tae-Yong Kim. *Modeling, rendering and animating human hair*. PhD thesis, University of Southern California, Los Angeles, CA, USA, 2002.

[42] Jaakko Lehtinen and Jan Kautz. Matrix radiance transfer. In *Proceedings of the 2003 Symposium on Interactive 3D Graphics*, I3D '03, pages 59–64, New York, NY, USA, 2003. ACM.

[43] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:79–116, 1998.

[44] Albert Liu, Steve Marschner, and Noah Snavely. Caliber: Camera localization and calibration using rigidity constraints. *International Journal of Computer Vision*, pages 1–21, 2015.

[45] Albert Julius Liu, Zhao Dong, Miloš Hašan, and Steve Marschner. Simulating the structure and texture of solid wood. *ACM Trans. Graph.*, 35(6):170:1–170:11, November 2016.

[46] Linjie Luo, Hao Li, and Szymon Rusinkiewicz. Structure-aware hair capture. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4), July 2013.

[47] D. Marcuse. Light scattering from elliptical fibers. *Appl. Opt.*, 13(8):1903–1905, Aug 1974.

[48] R. Marques, C. Bouville, M. Ribardière, L. P. Santos, and K. Bouatouch. Spherical fibonacci point sets for illumination integrals. *Computer Graphics Forum*, 32(8):134–143, 2013.

[49] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM Trans. Graph.*, 22(3):780–791, July 2003.

[50] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. Image-based brdf measurement including human skin. In *Proceedings of the 10th Eurographics Conference on Rendering*, EGWR'99, pages 131–144, Aire-la-Ville, Switzerland, Switzerland, 1999. Eurographics Association.

[51] Jonathan T. Moon and Stephen R. Marschner. Simulating multiple scattering in hair using a photon mapping approach. *ACM Trans. Graph.*, 25(3):1067–1074, July 2006.

[52] Jonathan T. Moon, Bruce Walter, and Steve Marschner. Efficient multiple scattering in hair using spherical harmonics. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 31:1–31:7, New York, NY, USA, 2008. ACM.

[53] Ivan Neulander and Michiel van de Panne. Rendering generalized cylinders with paintstrokes. In *Graphics Interface*, volume 98, pages 233–242, 1998.

[54] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph.*, 23(3):477–487, August 2004.

[55] Shinji Ogaki, Yusuke Tokuyoshi, and Sebastian Schoellhammer. An empirical fur shader. In *ACM SIGGRAPH ASIA 2010 Sketches*, SA '10, pages 16:1–16:2, New York, NY, USA, 2010. ACM.

[56] J. Ou, F. Xie, P. Krishnamachari, and F. Pellacini. ISHair: importance sampling for hair scattering. In *Computer Graphics Forum*, volume 31, pages 1537–1545, 2012.

[57] Minghao Pan, Rui Wang Xinguo Liu, Qunsheng Peng, and Hujun Bao. Precomputed radiance transfer field for rendering interreflections in dynamic scenes. In *Computer Graphics Forum*, volume 26, pages 485–493. Wiley Online Library, 2007.

[58] Sylvain Paris, Hector M. Briceño, and François X. Sillion. Capture of hair geometry from multiple images. *ACM Trans. Graph.*, 23(3):712–719, August 2004.

[59] Sylvain Paris, Will Chang, Oleg I. Kozhushnyan, Wojciech Jarosz, Wojciech Matusik, Matthias Zwicker, and Frédo Durand. Hair photobooth: Geometric and photometric acquisition of real hairstyles. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 30:1–30:9, New York, NY, USA, 2008. ACM.

[60] G. J. Parker, C. A. Wheeler-Kingshott, and G. J. Barker. Estimating distributed anatomical connectivity using fast marching methods and diffusion tensor imaging. *IEEE Trans Med Imaging*, 21(5):505–512, May 2002.

[61] Lena Petrovic, Mark Henne, and John Anderson. Volumetric methods for simulation and rendering of hair. Technical report, Pixar Animation Studios, 2006.

[62] Gustavo Pfeiffer and Ricardo Marroquim. Brdf fitting using inverse global illumination and stochastic optimization. In *Proceedings of the Conference on Graphics, Patterns and Images (SIBGRAPI 2012)*, Minas Gerais, Brazil, August 2012.

[63] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory To Implementation.* Morgan Kaufmann, 3nd edition, 2016.

[64] Didier Pinchon and Philip E Hoggan. Rotation matrices for real spherical harmonics: general rotations of atomic orbitals in space-fixed axes. *Journal of Physics A: Mathematical and Theoretical*, 40(7):1597, 2007.

[65] Serban D. Porumbescu, Brian Budge, Louis Feng, and Kenneth I. Joy. Shell maps. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 626–633, New York, NY, USA, 2005. ACM.

[66] Rudolph W. Preisendorfer. *Hydrologic Optics, Volume I: Introduction.* U.S. Dept. of Commerce, 1976.

[67] Matthias Raab, Daniel Seibert, and Alexander Keller. *Unbiased Global Illumination with Participating Media*, pages 591–605. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[68] Ravi Ramamoorthi. *Precomputation-Based Rendering.* NOW Publishers Inc, 2009.

[69] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. *SIGGRAPH Comput. Graph.*, 21(4):283–291, August 1987.

[70] Zhong Ren, Kun Zhou, Tengfei Li, Wei Hua, and Baining Guo. Interactive hair rendering under environment lighting. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pages 55:1–55:8, New York, NY, USA, 2010. ACM.

[71] Clarence R. Robbins. *Chemical and Physical Behavior of Human Hair.* Springer-Verlag, New York, third edition, 1994.

[72] Iman Sadeghi, Oleg Bisker, Joachim De Deken, and Henrik Wann Jensen. A practical microcylinder appearance model for cloth rendering. *ACM Trans. Graph.*, 32(2):14:1–14:12, April 2013.

[73] Iman Sadeghi, Heather Pritchett, Henrik Wann Jensen, and Rasmus Tamstorf. An artist friendly hair shading system. In *SIGGRAPH 2010 Proceedings*, pages 56:1–56:10, 2010.

[74] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Efficient and realistic visualization of cloth. In *Eurographics Symposium on Rendering 2003*, jun 2003.

[75] Kai Schröder, Reinhard Klein, and Arno Zinke. A volumetric approach to

predictive rendering of fabrics. In *Proceedings of the Twenty-second Eurographics Conference on Rendering*, EGSR '11, pages 1277–1286, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association.

[76] Kai Schröder, Arno Zinke, and Reinhard Klein. Image-based reverse engineering and visual prototyping of woven cloth. *IEEE Transactions on Visualization and Computer Graphics*, PP(99), 2014.

[77] Shaded0. Orange adult American shorthair cat. `https://commons.wikimedia.org/wiki/File:OrangeCat_hair.jpg`, December 2016. This image is licensed under the Creative Commons Attribution-Share Alike 4.0 International license.

[78] Li Shen, Jieqing Feng, and Baoguang Yang. Exponential Soft Shadow Mapping. *Computer Graphics Forum*, 2013.

[79] Toshihiro Shinohara, Jun ya Takayama, Shinji Ohyama, and Akira Kobayashi. Extraction of yarn positional information from a three-dimensional ct image of textile fabric using yarn tracing with a filament model for structure analysis. *Textile Research Journal*, 80(7):623–630, 2010.

[80] M. Shinya, M. Shiraishi, Y. Dobashi, K. Iwasaki, and T. Nishita. A simplified plane-parallel scattering model and its application to hair rendering. In *2010 18th Pacific Conference on Computer Graphics and Applications*, pages 85–92. IEEE, 2010.

[81] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph.*, 22(3):382–391, July 2003.

[82] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.*, 21(3):527–536, July 2002.

[83] Peter-Pike Sloan, Ben Luna, and John Snyder. Local, deformable precomputed radiance transfer. *ACM Trans. Graph.*, 24(3):1216–1224, July 2005.

[84] K. Sriprateep and E. L. J. Bohez. A new computer geometric modelling approach of yarn structures for the conventional ring spinning process. *The Journal of The Textile Institute*, 100(3):223–236, 2009.

[85] Robert F. Stamm, Mario L. Garcia, and Judith J. Fuchs. The optical prop-

erties of human hair I. fundamental considerations and goniophotometer curves. *J. Soc. Cosmet. Chem.*, (28):571–599, September 1977.

[86] Tomomarusan. Washi paper (Sugihara paper). `https://en.wikipedia.org/wiki/File:Washi(Sugihara_paper).JPG`, August 2005. This image is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.

[87] R. A. R. Tricker. *Introduction to Meteorological Optics.* Mills & Boon, London, 1970.

[88] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.*, 25(3):967–976, July 2006.

[89] User:Matte. Dirndl aus Seide mit Lichtschimmer, gesehen in einem Schaufenster. `https://commons.wikimedia.org/wiki/File:Salzburg_Mattes_2013-08-03_(31).JPG`, August 2013. This image is licensed under the Creative Commons Attribution-Share Alike 2.0 Generic license.

[90] H.C. van de Hulst. *Light Scattering by Small Particles.* Dover Books on Physics Series. Dover Publications, 1957.

[91] Nicky van Foreest. Fitting an ellipse to a set of data points. `http://nicky.vanforeest.com/misc/fitEllipse/fitEllipse.html`, September 2012.

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

[93] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 133:1–133:10, New York, NY, USA, 2009. ACM.

[94] Rui Wang, John Tran, and David Luebke. All-frequency relighting of glossy objects. *ACM Trans. Graph.*, 25(2):293–318, April 2006.

[95] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R. Marschner, Marie-Paule Cani, and Ming C. Lin. A survey on hair modeling: Styling, simulation,

and rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):213–234, March 2007.

[96] Kelly Ward, Ming C. Lin, Joohi Lee, Susan Fisher, and Dean Macri. Modeling hair using level-of-detail representations. In *Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, CASA '03, pages 41–, Washington, DC, USA, 2003. IEEE Computer Society.

[97] Yasuhiko Watanabe and Yasuhito Suenaga. A trigonal prism-based method for hair image generation. *IEEE Comput. Graph. Appl.*, 12(1):47–53, January 1992.

[98] Yichen Wei, Eyal Ofek, Long Quan, and Heung-Yeung Shum. Modeling hair from multiple views. *ACM Trans. Graph.*, 24(3):816–820, July 2005.

[99] E Woodcock, T Murphy, P Hemmings, and S Longworth. Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Conf. Applications of Computing Methods to Reactor Problems*, volume 557, 1965.

[100] Kui Wu and Cem Yuksel. Real-time fiber-level cloth rendering. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D 2017)*, New York, NY, USA, 2017. ACM.

[101] Kun Xu, Li-Qian Ma, Bo Ren, Rui Wang, and Shi-Min Hu. Interactive hair rendering and appearance editing under environment lighting. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 173:1–173:10, New York, NY, USA, 2011. ACM.

[102] Kun Xu, Wei-Lun Sun, Zhao Dong, Dan-Yong Zhao, Run-Dong Wu, and Shi-Min Hu. Anisotropic spherical gaussians. *ACM Trans. Graph.*, 32(6):209:1–209:11, November 2013.

[103] Ying-Qing Xu, Yanyun Chen, Stephen Lin, Hua Zhong, Enhua Wu, Baining Guo, and Heung-Yeung Shum. Photorealistic rendering of knitwear using the lumislice. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 391–398, New York, NY, USA, 2001. ACM.

[104] Ling-Qi Yan, Chi-Wei Tseng, Henrik Wann Jensen, and Ravi Ramamoorthi. Physically-accurate fur reflectance: modeling, measurement and rendering.

ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2015), 34(6):185, 2015.

[105] Baoguang Yang, Zhao Dong, Jieqing Feng, Hans-Peter Seidel, and Jan Kautz. Variance soft shadow mapping. *Computer Graphics Forum*, 29(7):2127–2134, 2010.

[106] Cem Yuksel and John Keyser. Deep Opacity Maps. *Computer Graphics Forum*, 2008.

[107] Jiahua Zhang, G. Baciu, Dejun Zheng, Cheng Liang, Guiqing Li, and Jinlian Hu. IDSS: A novel representation for woven fabrics. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):420–432, 2013.

[108] Shuang Zhao, Miloš Hašan, Ravi Ramamoorthi, and Kavita Bala. Modular flux transfer: Efficient rendering of high-resolution volumes with repeated structures. *ACM Trans. Graph.*, 32(4):131:1–131:12, July 2013.

[109] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Building volumetric appearance models of fabric using micro ct imaging. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 44:1–44:10, New York, NY, USA, 2011. ACM.

[110] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Structure-aware synthesis for predictive woven fabric appearance. *ACM Trans. Graph.*, 31(4):75:1–75:10, July 2012.

[111] Shuang Zhao, Fujun Luan, and Kavita Bala. Fitting procedural yarn models for realistic cloth rendering. *ACM Trans. Graph.*, 35(4):51:1–51:11, July 2016.

[112] Kun Zhou, Hujun Bao, Wei Hua, Zhong Ren, Weifeng Chen, Minghao Pan, and Rui Wang. Analytic double product integrals for all-frequency relighting. *IEEE Transactions on Visualization and Computer Graphics*, 19(undefined):1133–1142, 2013.

[113] Arno Zinke. *Photo-Realistic Rendering of Fiber Assemblies.* PhD thesis, University of Bonn, Bonn, Germany, 2008.

[114] Arno Zinke, Tomas Lay Herrrera, Anton Andriyenko, Martin Rump, Andreas Weber, and Reinhard Klein. A practical approach for photometric acquisition

of hair color. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 28(5):165, 2009.

[115] Arno Zinke and Andreas Weber. Light scattering from filaments. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):342–356, March 2007.

[116] Arno Zinke, Cem Yuksel, Andreas Weber, and John Keyser. Dual scattering approximation for fast multiple scattering in hair. *ACM Trans. Graph.*, 27(3):32:1–32:10, August 2008.