

ACQUISITION AND REPRESENTATION OF MATERIAL
APPEARANCE FOR EDITING AND RENDERING

JASON DAVIS LAWRENCE

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
COMPUTER SCIENCE

SEPTEMBER 2006

© Copyright by Jason Davis Lawrence, 2006. All rights reserved.

Abstract

Providing computer models that accurately characterize the appearance of a wide class of materials is of great interest to both the computer graphics and computer vision communities. The last ten years has witnessed a surge in techniques for measuring the optical properties of physical materials. As compared to conventional techniques that rely on hand-tuning parametric light reflectance functions, a data-driven approach is better suited for representing complex real-world appearance. However, incorporating these representations into existing rendering algorithms and a practical production pipeline has remained an open research problem.

One common approach has been to fit the parameters of an analytic reflectance function to measured appearance data. This has the benefit of providing significant compression ratios and these analytic models are already fully integrated into modern rendering algorithms. However, this approach can lead to significant approximation errors for many materials and it requires computationally expensive and numerically unstable non-linear optimization.

An alternative approach is to compress these datasets, using algorithms such as Principal Component Analysis, wavelet compression or matrix factorization. Although these techniques provide an accurate and compact representation, they do have several drawbacks. In particular, existing methods do not enable efficient importance sampling for measured materials (and even some complex analytic models) in the context of physically-based rendering systems. Additionally, these representations do not allow editing.

In this thesis, we introduce techniques for acquiring and representing real-world material appearance that address these research challenges. First, we introduce the Inverse Shade Trees (IST) framework. This is a conceptual framework for representing high-dimensional measured appearance data as a tree-structured collection of simpler masks and functions. We use it to provide an intuitive representation of the Spatially-Varying Bidirectional Reflectance Distribution Function (SVBRDF) that is automatically computed from measured data. Like other data-driven techniques, ISTs are more accurate than fitting parametric BRDFs to measured appearance data, but are intuitive enough to support direct editing. We also introduce a factored model of the BRDF optimized to support efficient importance sampling in the context of global illumination rendering. We demonstrate that our technique provides more efficient sampling than previous methods that sample a best-fit parametric model.

Dedicated to Jeff and Susy.

Contents

Abstract	iii
List of Figures	ix
List of Tables	xiii
1 Introduction	1
1.1 Material Appearance	1
1.2 Fitting Parametric Models to Measured Data	3
1.3 A Data-Driven Approach	4
1.4 Contributions	4
1.5 Overview	5
2 Background and Related Work	7
2.1 The Bi-directional Reflectance Distribution Function	7
2.2 Parametric BRDF Models	10
2.2.1 The Lambertian BRDF	10
2.2.2 The Phong BRDF	11
2.2.3 The Cook-Torrance-Sparrow BRDFs	13
2.2.4 The Lafortune BRDF	17
2.2.5 The Ward BRDF	18
2.2.6 Anisotropic BRDF Models	19
2.2.7 Virtual Goniorelectometry	19
2.2.8 Summary	20
2.3 The Spatially-Varying BRDF	21

2.4	Texture Maps	21
2.5	Shade Trees	22
2.6	Appearance Acquisition	24
2.6.1	Goniorelectometry	25
2.6.2	Spatially-Varying Reflectance	26
2.6.3	Translucent Materials	28
2.6.4	Summary	29
2.7	Fitting Analytic BRDFs to Reflectance Data	29
2.8	Factored BRDF Models	30
2.9	Rendering	33
2.9.1	Physically-Based Rendering	34
2.9.2	Interactive Rendering	35
2.10	Appearance Editing	36
3	Appearance Acquisition	38
3.1	Spherical Gantry	38
3.2	High Dynamic Range	39
3.2.1	Sampling	40
3.3	Calibration	41
3.3.1	Image Alignment	42
3.3.2	Photometric Calibration	45
3.4	Reconstruction	45
3.5	Datasets	46
3.6	Alternative Acquisition Setups	46
4	Inverse Shade Trees	49
4.1	Introduction	49
4.2	Relationship to Previous Work	52
4.3	System Overview	53
4.4	Algorithms for Matrix Factorization	57
4.4.1	Evaluation of Existing Algorithms	58

4.5	Our Method: Alternating Constrained Least Squares	62
4.5.1	Non-Negative Factorization	62
4.5.2	Sparsity	63
4.5.3	Domain-Specific Constraints	66
4.5.4	SVBRDF Constraints: Energy Conservation	66
4.5.5	BRDF Constraints: Value and Monotonicity	69
4.5.6	Practical Considerations	70
4.5.7	Missing Data and Confidence Weighting	70
4.5.8	Subsampling for Large Datasets	71
4.5.9	Initialization of ACLS	72
4.6	Normal and Tangent Estimation	72
4.7	Results: Editing	73
4.7.1	SVBRDF Editing	74
4.7.2	BRDF Editing	75
4.8	Comparison to Analytic Models	78
4.9	Limitations	84
4.10	Shade Trees for Heterogeneous Subsurface Scattering	84
4.11	Conclusions and Future Work	87
5	Global Illumination Rendering	89
5.1	Introduction	89
5.2	Previous BRDF Models	90
5.3	Importance Sampling	91
5.4	A New BRDF Representation for Sampling	91
5.4.1	Factorization	94
5.4.2	Sampling	97
5.4.3	Results	102
5.4.4	Limitations	109
5.5	Sampling n-Dimensional Measured Functions	111
5.5.1	Related Work	113

5.5.2	Background	114
5.5.3	Numerical CDF Compression	115
5.5.4	Polygonal Curve Approximation	116
5.5.5	Applying Curve Approximation to CDFs	117
5.5.6	Multidimensional CDFs: The Cascading Douglas-Peucker Algorithm	119
5.5.7	Evaluation of Algorithm	120
5.5.8	Environment Map Sampling	123
5.5.9	BRDF Sampling	124
5.5.10	Novel Applications	126
5.5.11	Local Environment Map Sampling	128
5.5.12	Multiple Importance Sampling	130
5.5.13	Conclusions and Future Work	130
6	Conclusion and Future Work	133
6.1	Conclusion	133
6.2	Areas of Future Work	134

List of Figures

1.1	example rendering that uses analytic appearance models	2
1.2	examples of real-world materials with complex appearance	3
1.3	images of a wallpaper material taken at different viewing positions and under illumination from different incoming directions	4
2.1	BRDF geometry and notation	8
2.2	2D slices of a BRDF for different incident directions	10
2.3	images rendered using the Phong BRDF	12
2.4	images rendered using the Cook-Torrance BRDF	16
2.5	image of brushed aluminum captured with Ward’s gonioreflectometer	18
2.6	two shade tree diagrams from Rob Cook’s seminal paper	22
2.7	visualization of the different levels in a shade tree for a realistic bowling pin	23
2.8	using a natural image as a texture map	24
2.9	comparison of two gonioreflectometer designs	25
2.10	illustration of image-based gonioreflectometry	26
2.11	image of Stanford’s spherical gantry	27
2.12	example images from the CURET BTF database	28
2.13	illustration of the half/difference parameterization proposed by Rusinkiewicz	32
3.1	several raw images of a wallpaper sample captured with a spherical gantry	39
3.2	stack of images used to compute a high-dynamic range image	40
3.3	two screenshots of a simple application for visualizing the positions of the gantry’s arms during acquisition	41

3.4	illustration of the coordinate system used for calibration	42
3.5	illustration of image alignment procedure	43
3.6	several images of the Spectralon [©] sample used for photometric calibration	44
3.7	visual illustration of the five SVBRDF datasets	47
3.8	acquisition setup designed for measuring the reflectance properties of human skin .	48
4.1	diagram of our Inverse Shade Tree (IST) framework	50
4.2	diagram showing how our SVBRDF decomposition relates to matrix factorization .	57
4.3	diagram showing how our BRDF decomposition relates to matrix factorization . .	58
4.4	ideal separation for the “Season’s Greetings” dataset	59
4.5	blending weights computed from the “Season’s Greetings” datasets using existing factorization algorithms	61
4.6	blending weights computed from the “Season’s Greetings” dataset using the ACLS matrix algorithm	65
4.7	visual comparison of the spatial blending weights computed by several linear factor- ization algorithms on the “Wood+Tape” dataset	67
4.8	accuracy of representing four different SVBRDFs with four possible linear decom- position algorithms	68
4.9	visualization of the importance of using a monotonicity constraint provided by ACLS for factoring the BRDF	69
4.10	visualization of how the ACLS algorithms handles confidence weighted and missing data in the input	71
4.11	normal and tangent maps computed for the “Wallpaper #2” and “Dove” datasets	74
4.12	screenshot of a prototype system that enables interactive rendering and editing of shade trees computed from measured data	75
4.13	example of editing the spatial distribution of the component materials in the “Sea- son’s Greetings” dataset	76
4.14	example of editing the spatial distribution of the component materials in the “Wood+Tape” dataset	77
4.15	example of performing material transfer with the “Dove” dataset	78

4.16	examples of different BRDF edits provided by our system	78
4.17	quantitative comparison of representing a measured SVBRDF using the Ward BRDF and k -means clustering and our method	80
4.18	visual comparison between representing measured SVBRDF data with the Ward model and our method	81
4.19	visual comparison of the separation achieved by applying the ACLS algorithm . . .	82
4.20	analysis of the error introduced by several levels of our tree-structured decomposition for BRDFs, and comparison with Ward fits	83
4.21	qualitative analysis of our compact factored representation applied to several real- world datasets	86
4.22	a shade tree computed from a wax candle material model applied to the Stanford dragon	87
5.1	diagram of the steps in factoring a BRDF into our representation	92
5.2	visualization of the factorization applied to each 2d function	95
5.3	visualization of how our factored model can be used to sample the BRDF	98
5.4	visual comparison of the accuracy of our factored model	103
5.5	comparison of the RMS error of an anisotropic Ward BRDF, as a function of the number of terms in the factorization	105
5.6	image variance as a function of sample count	105
5.7	visual comparison of sampling a BRDF according to a best-fit parametric model and our factored representation	109
5.8	visual comparison between sampling a BRDF according to the tabular method of Matusik [73] and our factored representation	110
5.9	equal time and equality quality visual comparisons for a global illumination scene rendered by importance sampling a best-fit parametric model and our representation	111
5.10	visualization of representing a continuous 1D function with a set of non-uniformly spaced samples	112
5.11	visual illustration of the Douglas-Peucker polyline approximation algorithm	116

5.12	probability density function along with a piecewise linear and piecewise constant approximation for uniformly and non-uniformly placed samples	118
5.13	didactic illustration of how we incorporate the function gradient into our sample placement algorithm	121
5.14	visualizations of a spherical probability density function computed from a light probe along with the our approximation at different resolutions	122
5.15	two different probability distribution functions and the RMS error in approximating them using different numbers of points and different sampling strategies	123
5.16	test scenes rendered by importance sampling the BRDF using our adaptive numerical CDF	125
5.17	visualization of the inefficiency of sampling from the same lighting-based importance function at every pixel	126
5.18	visualizations of several importance functions represented with our adaptive technique and computed at different surface orientations	127
5.19	comparison of the variance of a Monte Carlo estimator computed according to jittered sampling, sampling using a single uniformly-sampled CDF and using our <i>local environment map sampling</i> algorithm	129
5.20	multiple importance sampling using adaptive numerical CDFs computed from both BRDFs and image-based lighting	131

List of Tables

3.1	properties of the different datasets we acquired	47
4.1	qualitative comparison of different matrix factorization algorithms	60
4.2	statistics and details regarding acquired and factored subsurface scattering materials	85
5.1	resolution and numerical accuracy of our factored BRDF representation	99
5.2	comparison of the efficiency of importance sampling the BRDF using a best-fit parametric model and our factored representation under constant illumination . . .	106
5.3	comparison of the efficiency of importance sampling the BRDF using a best-fit parametric model and our factored representation under complex illumination . . .	108

Acquisition and Representation of Material Appearance for
Editing and Rendering

Jason Davis Lawrence

August 11, 2006

Chapter 1

Introduction

1.1 Material Appearance

Providing computer models that accurately characterize the optical properties of different materials is of great interest to both the computer graphics and computer vision communities. Indeed, a material's appearance depends on the way in which it absorbs, transmits and reflects incident light energy. For example, consider the differences in the appearance of red plastic, copper, satin, brushed aluminum and chalk.

Light that strikes an opaque material is either absorbed or reflected back into the surrounding environment. This is true for most plastics, metals and many textiles. This is not true for translucent materials like milk, human skin and marble, in which cases light is scattered below the surface and then re-emitted into the environment at a different surface position. In both cases, a different amount (and color) of light is reflected into each possible outgoing direction. This distribution of reflected light, which depends on the direction of incident illumination, results in a 4D *directional component* of surface reflectance. This directional dependence is responsible for our perception of certain materials being perfect mirrors while others are best described as glossy, polished, matte or dusty. In general, the reflectance and transmittance of a material will also vary across its surface. For example, consider the *spatial component* of the reflectance of rusting metal, a marble chessboard, a bruised banana, oak wood or roof shingles.

The directional component of surface reflectance can be specified with the Bidirectional Re-



Figure 1.1: One frame from an image sequence rendered at 30 hertz using consumer graphics hardware. The different materials in this complex scene were specified with analytic BRDFs and texture maps.

reflectance Distribution Function (BRDF) [82]. Ignoring wavelength-dependence, the BRDF is a 4D function that defines the amount of light transported between any pair of incident and reflected directions with each represented as a unit-length vector on the hemisphere. Traditionally, the BRDF has been represented with analytic expressions based on phenomenological hypotheses or derived from the physical properties of light-matter interaction for a particular class of materials. These analytic BRDFs are typically designed to have parameters that correspond to salient aspects of the material's appearance. One example is the *shininess* parameter of a dielectric represented with the widely popular Phong BRDF [89].

The spatial component of reflectance has traditionally been modeled by mapping a 2D function onto the surface of a 3D shape. These functions are either procedurally specified or stored in raster images called *texture maps*. They can be either derived from natural images or created by hand.

Indeed, designers familiar with existing analytic BRDFs and texture mapping techniques can

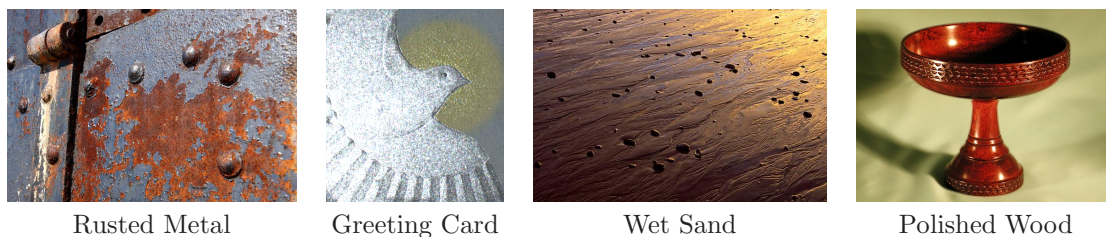


Figure 1.2: Real-world materials with complex appearance not easily modeled using existing analytic BRDFs and hand-generated texture maps. Image credits: Casey Danek (rusted metal), Jason Lawrence (dove greeting card), QT Luong (wet sand at sunset), Ray Ford (mahogany compote).

produce quite compelling synthetic imagery (see Figure 1.1). Additionally, these methods are fully integrated into state-of-the-art interactive and physically-based rendering algorithms and, due to their small number of intuitive parameters, allow direct editing. Nevertheless, there are still many real-world materials whose complex appearance cannot be easily modeled using these techniques (see Figure 1.2).

Over the last ten years, several techniques have been developed for measuring the optical properties of real-world materials. In general, this requires recording thousands of images of a material sample from different viewing directions and under illumination from varying incident angles. Figure 1.3 shows a few images from such a dataset of a green and gold wallpaper material. These images can be stored in a large table and subsequent synthesis involves performing simple look-ups within this table. Although these data-driven *tabular models* provide the most accurate representation of a material’s appearance, they require expensive measurement devices, time-consuming acquisition and careful calibration of the raw input. More importantly, the large size of these datasets presents several research challenges related to incorporating them into existing interactive and physically-based rendering algorithms and providing a practical production pipeline.

1.2 Fitting Parametric Models to Measured Data

One common approach to incorporating measured appearance data into computer-generated images has been to fit the parameters of an analytic BRDF model to the measurements. This provides a useful representation because most analytic BRDFs are already fully integrated into existing interactive and physically-based rendering algorithms and commercially-available production

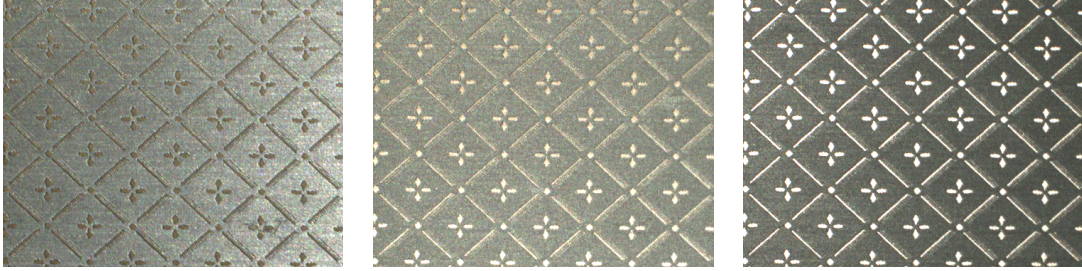


Figure 1.3: Three images (from a set of 5,000) of a wallpaper material taken at different viewing positions and under illumination from different incoming directions. These images provide a data-driven representation of this material’s appearance.

software.

Nevertheless, fitting measured appearance data to an analytic model can lead to significant approximation errors for many materials (see, for example, Ngan et al. [81] and Chapters 4 and 5). This fitting process also requires computationally expensive and numerically unstable non-linear optimization and is susceptible to undesirable local minima.

1.3 A Data-Driven Approach

An alternative approach is to compress these tabular datasets using basis function decomposition [21] and standard matrix rank-reduction algorithms like Principal Component Analysis (PCA) [53, 30, 106], Independent Component Analysis (ICA) [102], k -means clustering [66], and Non-Negative Matrix Factorization (NMF) [16]. Although existing techniques are effective in providing an accurate and compact representation they do not allow several important operations. In particular, these representations do not enable efficient importance sampling for the materials in a 3D scene in the context of physically-based rendering algorithms. Additionally, these compressed representations cannot be directly edited.

1.4 Contributions

This thesis addresses several open research problems related to incorporating measured appearance data into physically-based and interactive rendering algorithms. We also incorporate the design goal of *editability*, making these representations useful in a practical production pipeline.

We introduce the Inverse Shade Trees (IST) framework. This is a conceptual framework for representing high-dimensional measured appearance data as a tree-structured collection of simpler masks and functions. Within this framework, we develop a new data-driven representation of the spatially-varying BRDF (SVBRDF) that is automatically computed from measured data. Like previous data-driven techniques, our shade trees are compact enough to support interactive rendering and are more accurate than fitting a parametric BRDF to measured appearance data. Unlike previous data-driven models, however, they contain intuitive components that can be edited, allowing a designer to change both the spatial and directional behavior of surface reflectance. Within the IST framework, we also introduce a representation of the spatial component of the Bidirectional Subsurface Scattering Reflectance Distribution Function (BSSRDF) for translucent materials. Although this representation is not optimized to support editing, it provides a more compact and accurate representation than previous approaches. We evaluate the performance and accuracy of both of these representations on real-world data.

We also introduce a factored model of the BRDF optimized to support efficient importance sampling in the context of global illumination rendering. We demonstrate that our technique is more efficient than previous methods that sample a best-fit parametric model. Lastly, we introduce a representation designed for compressing and sampling measured functions of arbitrary dimension. We show this representation is useful for sampling image-based illumination and reflectance within a physically-based rendering systems.

1.5 Overview

Chapter 2 contains background material related to appearance models and rendering algorithms. We first review the radiometry of several important surface reflectance functions and discuss existing parametric models designed to represent the appearance of various classes of materials. We then discuss the role texture maps and shade trees play in specifying complex appearance functions before reviewing existing techniques for acquiring and representing real-world material appearance. We conclude with a brief overview of the principles behind physically-based and interactive rendering algorithms.

We discuss our pipeline for acquiring the spatially-varying surface reflectance of real-world ma-

materials in Chapter 3. This includes a description of our acquisition setup along with the techniques we use for geometric and photometric calibration of a camera and light-source pair. We list the qualitative properties of the five datasets we have acquired along with their respective size and sampling densities. This chapter ends with a discussion of alternative acquisition setups.

In Chapter 4 we introduce the Inverse Shade Tree (IST) framework. This is a framework for representing high-dimensional measured appearance functions as a tree-structured collection of simpler masks and functions. We describe how to compute intuitive shade trees of the SVBRDF from measured data by computing a sequence of matrix factorizations. Motivated by a review of existing factorization algorithms, we introduce a new algorithm based on linearly constrained least squares. This algorithm, which we call Alternating Constrained Least Squares (ACLS), can incorporate both general and domain-specific constraints, making it suitable for automatically computing *intuitive* decompositions. We demonstrate how these shade trees support interactive rendering and editing and compare their accuracy with previous techniques that fit a parametric BRDF to the input measurements. Within this shade tree framework, we also introduce a novel representation of the spatial component of the Bidirectional Subsurface Scattering Reflectance Distribution Function (BSSRDF). This representation is not optimized to support editing, but it is more compact and accurate than previous approaches. This work originally appeared in [59, 86].

In Chapter 5 we introduce a new factored model of the BRDF designed to provide efficient importance sampling in the context of physically-based rendering algorithms. We compare the efficiency and accuracy of this representation to previous techniques that draw samples from a best-fit parametric model. We also introduce a representation that provides compression and sampling of arbitrary high-dimensional measured functions. This representation is based on the Douglas-Peucker polyline approximation algorithm [27] and is useful for compressing and sampling image-based illumination and measured BRDFs. We compare the sampling efficiency provided by our technique to existing methods. The work was originally published in [60, 61].

In Chapter 6 we conclude with a discussion of our contributions and suggest several possible directions of future research in this area.

Chapter 2

Background and Related Work

2.1 The Bi-directional Reflectance Distribution Function

A material's appearance is related to the way it absorbs, transmits and reflects incident light. For opaque materials, we can assume that the light arriving at a point on its surface is absorbed and/or reflected back into the surrounding environment. Examples of opaque materials include metals, plastics, painted surfaces and some textiles. Their appearance can be characterized by a function that records the amount of light energy reflected from any incoming direction into any outgoing direction for a given wavelength. This function is called the Bi-directional Reflectance Distribution Function (BRDF) [82]. For computational efficiency, we typically represent colorized functions as a projection onto a tristimulus color space (e.g. RGB). Therefore we can ignore the dependence on wavelength and the BRDF is a 4D function that maps any pair of directions over the unit hemisphere to a non-negative real number:

$$f_r(\omega_i, \omega_o) : \Omega_+ \times \Omega_+ \rightarrow \mathbb{R}_+ \quad (2.1)$$

where f_r is the BRDF, Ω_+ is the upper unit hemisphere and ω_i and ω_o are unit-length vectors of the incoming and reflected directions respectively. We will sometimes write the BRDF in terms of the spherical angles of the incoming and reflected directions: $f_r(\omega_i, \omega_o) = f_r(\theta_i, \phi_i; \theta_o, \phi_o)$.

The BRDF was introduced in the field of radiometry [82] and is the ratio of outgoing radiance

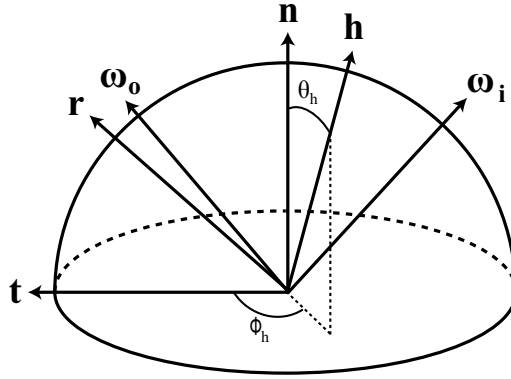


Figure 2.1: The geometry of the BRDF domain labeled according to our notation.

(L) along the reflected direction to the irradiance (E) arriving at the surface from the incident direction:

$$f_r(\omega_i, \omega_o) = \frac{dL(\omega_o)}{dE(\omega_i)}. \quad (2.2)$$

Recall that radiance is the amount of light energy traveling along a ray through space and is proportional to the power (Φ) per solid angle (ω) per area perpendicular to the ray's direction (A) and is measured in units of $W/m^2/sr$:

$$L = \frac{d\Phi}{dA d\omega}. \quad (2.3)$$

Irradiance is the incident flux per unit area and can be thought of as the amount of energy striking an oriented surface patch from a particular direction:

$$E(\omega_i) = L(\omega_i) \cos(\theta_i) d\omega_i. \quad (2.4)$$

Because irradiance is measured in units of W/m^2 , the units of the BRDF are inverse steradians, sr^{-1} .

Of great interest to the field of computer graphics is the fact that the BRDF can be used to compute the amount of radiance that is *reflected* from any incoming direction into any outgoing direction:

$$L(\omega_o) = f_r(\omega_i, \omega_o) L(\omega_i) \cos \theta_i, \quad (2.5)$$

where $\cos \theta_i$ converts the radiance arriving from ω_i into the irradiance incident to the surface. Our notation, along with the geometry of the BRDF domain, is illustrated in Figure 2.1.

The fact that the BRDF can take only non-negative values follows from the laws of physics: it makes no sense to consider a surface reflecting a *negative* amount of light. Another physical property of light is that it is reciprocal. This means the BRDF is invariant to exchanging the incident and reflected directions:

$$f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i). \quad (2.6)$$

As with any physical system, the BRDF conserves energy. In other words, the sum of reflected light cannot exceed the amount of energy incident to the surface:

$$\int_{\Omega_+} f_r(\omega_i, \omega_o) \cos \theta_o d\omega_o \leq 1. \quad (2.7)$$

We will make use of these properties in Chapter 4 when we derive an algorithm for factoring a matrix that contains measured BRDF data.

There is an important class of materials with scattering functions that do not require the full four degrees of freedom of the BRDF. For many materials, the amount of light reflected between any pair of directions is invariant to rotation around the surface normal. These *isotropic* materials include many metals, plastics and painted surfaces. They can be parameterized with respect to a 3D domain:

$$f_r(\theta_i, \phi_i, \theta_o, \phi_o) = f_r(\theta_i, \theta_o, \phi_i - \phi_o). \quad (2.8)$$

Materials whose reflectance does depend on the orientation of the surface around the normal direction are called *anisotropic*. Their reflectance requires the complete 4D BRDF domain and examples include brushed aluminum, most fabrics, wood and human hair.

It is convenient to visualize 2D slices of the BRDF for a fixed incoming (or reflected) direction as shown in Figure 2.2. However, we will commonly visualize BRDFs with lit sphere images similar

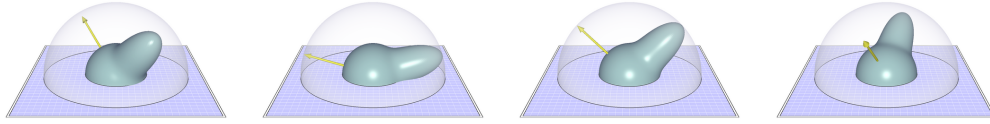


Figure 2.2: Slices of a BRDF at different incoming (or reflected) directions. These visualizations show the magnitude of light reflected into each direction over the upper hemisphere for a fixed incident direction. In this case, the region centered around the mirror direction reflects more light than at other regions. This corresponds to a material having a *specular highlight*.

to those in Figure 2.3.

It is important to keep in mind that the BRDF is appropriate for characterizing the appearance of only a certain class of materials. For example, representing translucent materials or materials with time-varying optical properties requires using more general scattering functions. Additionally, we will only consider interference that occurs at scales much larger than the wavelength of visible light. A more complete treatment of the wave properties of light would be necessary for recovering effects like thin-film interference, dispersion or diffraction.

2.2 Parametric BRDF Models

Computer models of material appearance should be accurate, concise and inexpensive to evaluate. Traditionally, this has been accomplished by using analytic expressions of the BRDF that depends on a small number of parameters. These parametric models are either phenomenological or derived from the physics of light-matter interaction at various scales for different classes of materials. In both cases, the underlying parameters generally relate to salient aspects of a material’s appearance. For example, we will see that different parameters control perceptual properties of material appearance like its “glossiness” and “color” and how it behaves as the incident and reflected directions approach grazing angles. Using these parameters, a designer is able to directly edit the properties of a BRDF. We review the most important parametric BRDF models in this section.

2.2.1 The Lambertian BRDF

One of the earliest BRDFs is based on Lambert’s law [58]. It states that an equal amount of light is reflected from a surface for every pair of incoming and outgoing directions:

$$f_r(\omega_i, \omega_o) = \frac{k_d}{\pi}, \quad (2.9)$$

where k_d is the single parameter of this BRDF and relates to the *surface albedo* of the material. Recall that f_r is a colorized function so k_d is technically a function of wavelength, but is typically represented with a three-tuple in a tristimulus color space (e.g. RGB or HSV).

Although perfectly Lambertian surfaces do not exist in nature, this BRDF is useful for approximating the appearance of dusty or matte materials. It is also commonly used to represent the *diffuse* component of surface reflectance. In other words, a certain percentage of the light striking a surface is immediately reflected back into the environment, but some light is *transmitted* below the surface. This light is scattered among the particles suspended within the material before being re-emitted into the environment. After being scattered multiple times, this light has a roughly Lambertian distribution and is commonly modeled by Equation 2.9. The light that is reflected directly off the surface is called the *specular* component. The next section introduces a popular phenomenological model intended to represent the specular reflectance of dielectric materials.

2.2.2 The Phong BRDF

Another ubiquitous parametric model is the Phong BRDF named after its inventor Bui Tuong Phong [89]:

$$f_r(\omega_i, \omega_o) = k_s \frac{n+2}{2\pi} \cos^n \alpha, \quad (2.10)$$

where α is the angle between the surface normal \mathbf{n} and the vector halfway between ω_i and ω_o (also called the half-angle vector \mathbf{h}) as shown in Figure 2.1. The parameter k_s defines the magnitude and color of the overall reflectance and n is a scalar value that controls the width of the specular highlight. Increasing n sharpens the specular highlight and produces a glossier or shinier appearance. Decreasing n widens the specular highlight and produces a more matte appearance. Figure 2.3 demonstrates the effect of this parameter on the material appearance.

Equation 2.10 is not the original form proposed by Phong in 1975. We have incorporated two modifications. We scale the BRDF by $\frac{n+2}{2\pi}$ as proposed by [67]. This is necessary for the BRDF to conserve energy (the original model did not conserve energy). Second, the original model defined

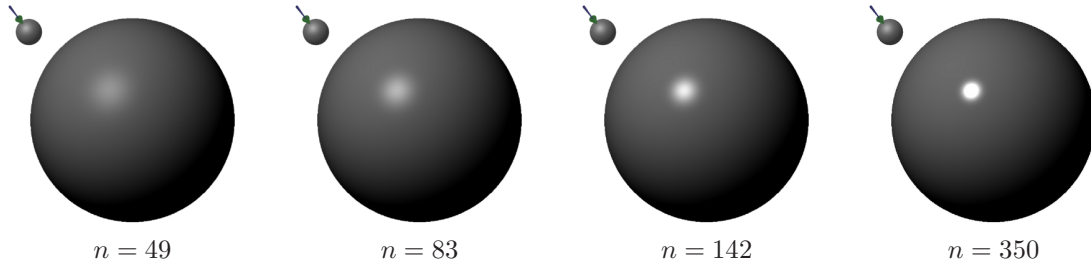


Figure 2.3: Images rendered using the Phong model, showing the effect of the *glossiness* parameter n on the material’s appearance. Note that the diffuse component of the surface reflectance is modeled with the Lambertian BRDF where $k_d = 0.2$.

α as the angle between ω_o and the direction of perfect mirror reflection from ω_i (i.e. the vector \mathbf{r} in Figure 2.1). Defining α as the angle between the normal and half-angle vector as done here was originally proposed by [9]. These two versions of the BRDF are known as the “R dot V” variant and physically-plausible “H dot N” variant or Blinn-Phong model respectively. Recent work has demonstrated that defining α with respect to the half-angle vector produces a better match to the reflectance of real-world materials [81].

Like the Lambertian BRDF model, the Phong BRDF is a phenomenological model. In other words, it was not derived from the physical laws governing light-matter interaction. Instead it is one possible mathematical explanation of the phenomena we observe when light strikes a particular type of surface. Specifically, the Phong model is intended to describe the specular reflection of plastic and other dielectric surfaces.

Typically the Phong and Lambertian BRDFs are used together to model the specular and diffuse components of a material’s reflectance respectively (see Figure 2.3). The main benefit of this approach (referred to simply as the “Phong reflectance model”) is its computational simplicity. Evaluating this model requires only 5 additions, 6 multiplications and 1 exponentiation (assuming the simplest functional form). Due to its simplicity, efficient rendering algorithms and widely adopted graphics APIs have been designed around the Phong reflectance model. The fact that it is still ubiquitous today (30 years after it’s introduction) indicates its practical utility.

The main drawback of the Phong model is its lack of visual fidelity. The appearance represented by this model tends to have a “plastic” look regardless of the designer’s intent—indeed it was intended to represent plastic materials. Not surprisingly, this simple mathematical formula lacks sufficient expressiveness to capture the appearance of many different materials. We will review

several additional parametric BRDF models that attempt to address this shortcoming. In each case they achieve a greater level of realism at the expense of greater computational complexity.

2.2.3 The Cook-Torrance-Sparrow BRDFs

Some of the earliest analytic BRDFs derived from the physics of light-matter interaction are the Torrance-Sparrow [101] and Cook-Torrance [20] models. They are based on a microfacet theory that describes a surface as a collection of tiny Fresnel mirrors oriented at random angles with respect to the average surface normal over a small area. Light traveling along ω_i toward a point x on the surface can undergo three possible events:

- It is blocked by a microfacet before it arrives at x . This is called *shadowing*.
- It is blocked by a microfacet after being reflected from x . This is called *masking*.
- It enters the environment after reflecting off the Fresnel mirror at x .

Within this framework the spectral distribution of light reflected into ω_o is proportional to the number of microfacets with an orientation parallel to $\omega_o + \omega_i$, the reflectance off each appropriately oriented microfacet and the number of incoming rays that were neither shadowed nor masked. I will refer the reader to [101, 20] for the derivation of the aggregate surface reflectance and instead list the final Cook-Torrance BRDF:

$$f_r(\omega_i, \omega_o) = d \frac{k_d}{\pi} + s \frac{F}{\pi} \frac{DG}{(\mathbf{n} \cdot \omega_i)(\mathbf{n} \cdot \omega_o)}, \quad (2.11)$$

where d and s are scalar valued parameters that control the relative amount of diffuse and specular reflectance, k_d is the surface albedo, F is a Fresnel term that describes how light is scattered from each smooth microfacet, D controls the distribution of the surface microfacets and G is a geometry term that accounts for the shadowing and masking effects. The geometry term is defined as:

$$G = \min \left[1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \omega_o)}{(\omega_i \cdot \omega_o)}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \omega_i)}{\omega_o \cdot \mathbf{h}} \right], \quad (2.12)$$

where \mathbf{h} is the half-angle vector defined as $\mathbf{h} = \frac{\omega_o + \omega_i}{|\omega_o + \omega_i|}$ and \mathbf{n} is the surface normal. Note that this term approaches 1 as $\omega_i = \omega_o$ and 0 as ω_i and ω_o point away from one another. For a detailed derivation consult [101].

The D term defines the statistical distribution of microfacet orientations. The two most common distributions are the Gaussian distribution whose use in this context was first proposed by Blinn [9, 10]:

$$D = ce^{-(\alpha/m)^2}, \quad (2.13)$$

and the Beckmann distribution [7] designed to characterize the distribution of microfacets for rough surfaces:

$$D = \frac{1}{m^2 \cos^4 \alpha} e^{-[(\tan \alpha)/m]^2}. \quad (2.14)$$

In both cases, the parameter m defines the RMS slope of the distribution and consequently controls the shape of the specular highlight (this is akin to the n parameter of the Phong BRDF). A larger or smaller value of m leads to a broader or sharper fall-off respectively. The value α is the angle made by the surface normal \mathbf{n} and half-angle vector \mathbf{h} (i.e. $\cos \alpha = \mathbf{n} \cdot \mathbf{h}$). In Equation 2.13, c is an arbitrary constant that should be set to guarantee energy conservation. One advantage of the Beckmann distribution is that it does not have such a constant at the expense of requiring more effort to compute.

Some surfaces have multiple scales of roughness. For example, a common Christmas decoration is a colored metallic sphere coated with a shiny glaze. For these objects, light is reflected off both the glaze and the underlying metallic paint. This results in the material's appearance having two specular highlights of different widths. It is appropriate to model these type of phenomena with a weighted combination of multiple distributions:

$$D = \sum_{j=1}^N w_j D(m_j), \quad (2.15)$$

where w_j are the weights for each distribution ($\sum_{j=1}^N w_j = 1$) and the $D(m_j)$ are the microfacet distribution functions each with RMS slope of m_j . As discussed in Section 2.7 fitting multiple reflectance lobes to measured data is numerically unstable.

The Fresnel term F describes the dependency of surface reflectance on the angle of incident illumination. It is based on the Fresnel equation from classical optics which relates surface reflectance

to a material’s index of refraction (n), extinction coefficient (k) and the angle of incoming light (θ_i). Note that n and k both depend on wavelength. Because these values are not readily available for most materials, Cook and Torrance [20] propose a simplification that depends only on a material’s reflectance at normal incidence under the assumption that $k = 0$. Although this is true for dielectrics, most metals have a non-zero extinction coefficient. Nevertheless, the simplified Fresnel equation provides a suitable approximation of the angular and wavelength dependence of surface reflectance for many materials:

$$F = \frac{1}{2} \frac{(g - c)^2}{(g + c)^2} \left\{ 1 + \frac{[c(g + c) - 1]^2}{[c(g - c) + 1]^2} \right\}, \quad (2.16)$$

where $c = \omega_o \cdot \mathbf{h}$ and $g^2 = n^2 + c^2 - 1$. Equation 2.16 can be used to extrapolate the reflectance at normal incidence (F_0) to values of F at arbitrary incident directions (see [20] for details). Because F_0 has been measured for most materials [103, 104, 105], this provides a useful approach for modeling the reflectance of many real-world materials.

In general, the reflectance of dielectrics varies greatly with θ_i . They tend to reflect a greater amount of light from their surfaces as the illumination angle approaches 90 degrees. This is the reason that plastics exhibit mirror-like behavior when viewed from near grazing directions while being largely diffuse when viewed from above. A nice exercise is to compare the appearance of a polished wooden table-top when viewed from above and from a near-grazing angle.

Conductors, on the other hand, reflect a more constant amount of light over the range of incident directions. Unlike dielectrics, however, they exhibit colorized specular highlights with hues that can vary with viewing and illumination angle. Along with increased specular reflectance toward grazing and non-perfect Lambertian diffuse reflectance, these color shifts are collectively referred to as *Fresnel effects*. They are faithfully reproduced by the Cook-Torrance model as seen in Figure 2.4.

Because it is based on the physics of light-matter interaction, the Cook-Torrance BRDF is able to represent a wider class of materials than the Phong or Lambertian models. However, this also makes it more computationally expensive to evaluate. Not until recent improvements in consumer graphics hardware has the Cook-Torrance model been successfully incorporated into interactive rendering algorithms [44].

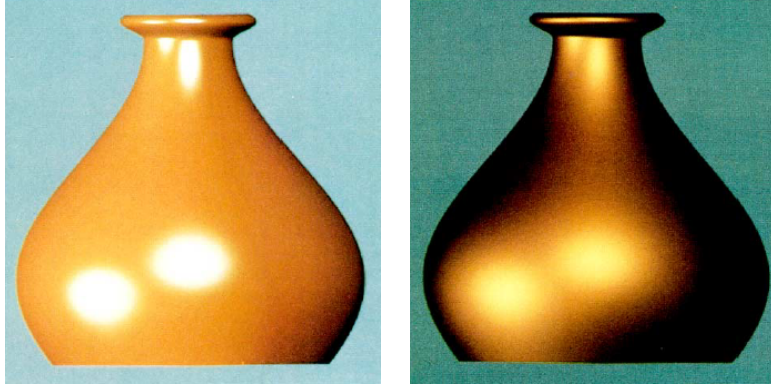


Figure 2.4: The vase’s appearance is modeled with a Cook-Torrance BRDF. The scene is lit by two white point light sources. Left: the Fresnel term is set to resemble copper-colored plastic. Right: the Fresnel term is set to resemble copper metal. Note the colorized highlight in the metal vase. This is a key property of the appearance of conductors that is represented by the Cook-Torrance BRDF. Figure reproduced from [20].

One key drawback of the Cook-Torrance BRDF is that it cannot be directly sampled in the context of physically-based rendering algorithms. This follows from the fact that integrating Equation 2.11 over ω_i for a fixed ω_o does not have a closed-form solution. We propose a data-driven representation that enables sampling of arbitrary analytic and measured BRDFs in Chapter 5.

Although the Cook-Torrance BRDF can represent a wider class of materials than the Phong model, there are many optical properties that it cannot express as well. First, it is intended to represent only isotropic materials. Additionally, it only considers interferences at scales much larger than the wavelength of light. This perspective ignores effects such as dispersion and diffraction. Extensions of this BRDF that do consider the wave properties of visible light have been proposed [42, 97]. In fact, the HTSG BRDF [42] is widely considered the most sophisticated physically-based model in existence today although its computational complexity has prevented it from being incorporated into most production systems.

Models have also been proposed that generalize the reflectance properties of the individual microfacets that make up a material’s surface. Unlike the Cook-Torrance BRDF that assumes these microfacets behave like Fresnel mirrors, the Oren-Nayar [84] BRDF assumes each microfacet is a perfectly Lambertian reflector. This model has proved useful for describing the reflectance of dusty surfaces. The Hapke-Lommel BRDF [39] was designed to describe the appearance of lunar surfaces. It also generalizes perfectly Lambertian reflectance with the addition of a retro-reflective

or back-scattering reflectance lobe. It too is useful for modeling matte or dusty materials.

Ashikhmin et al. [4] propose an extension to the Cook-Torrance BRDF that allows a general characterization of the distribution term D in Equation 2.11. Specifically, D is represented as a discretized function over the half-angle \mathbf{h} and is stored as a 2D image. The flexibility of this model has recently proved useful for fitting to measured reflectance data [81]. These type of *hybrid models* that combine both analytic and sampled or *non-parametric* functions are closely related to the factored BRDF models discussed in Section 2.8.

2.2.4 The Lafortune BRDF

The Lafortune BRDF [56] represents surface reflectance as a weighted combination of variable-width cosine lobes, each centered around a direction related to ω_i by an arbitrary 3x3 matrix:

$$f_r(\omega_i, \omega_o) = \sum_{j=1}^N k_s^j [\omega_i^T M_j \omega_o]^{n_j} \quad (2.17)$$

where k_s^j controls the magnitude of the j^{th} lobe, M_j defines the center of the lobe with respect to the reflected direction and n_j controls its fall-off. The full generality of a 3x3 matrix is unnecessary to represent most useful transformations. For this reason, a more restricted formulation is typically used:

$$f_r(\omega_i, \omega_o) = \sum_{j=1}^N k_s^j [C_x^j \omega_{ix} \omega_{ox} + C_y^j \omega_{iy} \omega_{oy} + C_z^j \omega_{iz} \omega_{oz}]^{n_j}, \quad (2.18)$$

where C_x^j, C_y^j and C_z^j control both the magnitude and direction of the j^{th} cosine lobe.

As discussed by [56], the Lafortune BRDF is able to represent a variety of reflectance properties. Although it does not explicitly include a Fresnel term, the C_x, C_y and C_z parameters can be adjusted to model non-Lambertian diffuse reflection and increased specularity at grazing angles. It can also represent retroreflective materials and off-specular highlights. Another advantage of this model is that it can be sampled in the context of global illumination rendering as discussed in Section 2.9.1.

Nevertheless, the Lafortune BRDF has several shortcomings. First, it is unable to represent a half-angle parameterization of the upper hemisphere since this cannot be encoded as a linear



Figure 2.5: An image of brushed aluminum reflecting light from a small source. Notice the elliptical highlight that results from uneven scattering along and against the orientation of the surface’s microcylinders. Figure reproduced from [109].

operator on ω_i . Generalizing this model to allow a “H dot N” parameterization remains an interesting direction of future research because this parameterization was recently shown to be superior at representing measured materials [81]. Second, although it is a function of four dimensions, the generalized cosine model is not suitable for representing the elliptical shape of most anisotropic specular highlights. Lastly, fitting more than two cosine lobes to measured data is often numerically unstable.

2.2.5 The Ward BRDF

Another model developed in the context of fitting to measured data is the Ward BRDF [109]:

$$f_r = \frac{k_d}{\pi} + k_s \frac{1}{\sqrt{\cos \theta_i \cos \theta_o}} \frac{\exp[-\tan^2 \theta_h (\cos^2 \phi_h / \alpha_x^2 + \sin^2 \phi_h / \alpha_y^2)]}{4\pi \alpha_x \alpha_y}, \quad (2.19)$$

where k_d and k_s control the amount and color of the diffuse and specular reflectance respectively. The parameters α_x and α_y define the RMS slope of a bivariate Gaussian distribution. Recall from Figure 2.1 that θ_h and ϕ_h are the elevation and azimuthal angles of the half-angle vector \mathbf{h} in the local coordinate system defined by the surface normal \mathbf{n} and tangent vector \mathbf{t} . Therefore, Equation 2.19 describes an elliptical-shaped specular highlight with major and minor axes defined by the parameters α_x and α_y (for isotropic materials $\alpha_x = \alpha_y$).

Although the Ward BRDF is a phenomenological model, it is physically plausible. Assuming $k_d \geq 0, k_s \geq 0, k_d + k_s \leq 1$ and α_x and α_y are not “too large,” this BRDF is non-negative, reciprocal and conserves energy [109]. Its mathematical form also allows efficient importance sampling within physically-based rendering algorithms (see Section 2.9.1).

The Ward BRDF was designed to be fit to measurements of isotropic and anisotropic reflectance functions [109]. One important characteristic of anisotropic materials is that their appearance depends on rotation about the surface normal for a fixed view and light direction. This property is not true for isotropic BRDFs and is tied to differences between the microgeometry of these two classes of materials. The Ward BRDF assumes anisotropic materials consist of densely packed fibers or cylinders that are oriented along a common direction called the *direction of anisotropy*. Materials such as wood and brushed aluminum have similar microgeometries. When light strikes these cylinders it produces an elliptical highlight due to uneven scattering along and across the orientation of these microcylinders. This results in an elliptical-shaped highlight that resembles a bivariate Gaussian distribution over the half-angle as proposed by the Ward model (see Figure 2.5).

The Ward BRDF is appropriate for modeling certain isotropic and anisotropic materials. It has also been shown to be well suited for fitting to measured data [81]. Nevertheless, there are many anisotropic materials that the Ward BRDF is not able to express. The next section reviews additional anisotropic BRDFs based on alternative microfacet theories.

2.2.6 Anisotropic BRDF Models

There are several BRDFs designed to describe anisotropic surface reflectance. One common approach is to model a material's surface as a collection of densely packed cylinders oriented along a common direction. Such physically-based models are suitable for modeling materials like brushed aluminum and wood. The Poulin-Fournier BRDF [90] is one such model that includes parameters related to the spacing, height and size of these microcylinders. Adjusting these parameters effects the overall appearance of the material.

The Banks BRDF [6] and Kajiya-Kay BRDF [52] are based on how light reflects from a 1D cylindrical fiber. These models are effective in reproducing the appearance of metallic and furry materials. Recently, these fiber scattering models have been extended to describe the appearance of human hair [71] and polished wood [72].

2.2.7 Virtual Gonioreflectometry

Westin et al. [111] take a different perspective on modeling surface reflectance. They explicitly model the microgeometry of a material as a height field and simulate its aggregate reflectance at

different incoming and outgoing directions. They store the resulting estimate of its BRDF in a spherical harmonic basis. Although virtual gonioreflectometry is computationally expensive and requires explicitly modeling the 3D microgeometry, it is capable of reproducing complex appearance like, for example, that of satin and leather.

2.2.8 Summary

There are many different analytic BRDFs, each designed with a particular class of materials in mind. When evaluating the utility of any one model, it is important to consider several key properties we would like to be true of any representation:

- **Accuracy:** A model should accurately describe the appearance of a wide variety of materials.
- **Physical-Plausibility:** A BRDF should meet physical constraints based on the underlying physics of light-matter interaction. Specifically, a BRDF should be reciprocal, non-negative and conserve energy.
- **Importance Sampling:** Another criterion for incorporating BRDFs into global illumination rendering algorithms is that they allow efficient importance sampling. This is further discussed in Section 2.9.1.
- **Simplicity:** For interactive applications it is important that a BRDF be inexpensive to evaluate. The definition of “inexpensive” is changing as the performance of consumer graphics hardware continues to improve, but a model should still avoid unnecessary complexity.
- **Editing:** A useful model should allow a designer to efficiently edit the salient aspects of its appearance.

In Chapters 4 and 5 we will introduce a new data-driven representation of surface appearance. We will show that current analytic models do not in general possess these key properties and how this is addressed by our approach.

2.3 The Spatially-Varying BRDF

Most objects that we encounter in the world do not scatter light the same way across their entire surface. Figure 1.2 shows a few examples of objects with *spatially-varying* appearance. In order to represent these types of materials, we need a bi-directional reflectance function that also depends on surface position. This gives rise to the Spatially-Varying Bi-directional Reflectance Distribution Function (SVBRDF):

$$S(u, v, \omega_o, \omega_i) : [0, 1] \times [0, 1] \times \Omega_+ \times \Omega_+ \rightarrow \mathbb{R}_+, \quad (2.20)$$

where the coordinate vector (u, v) encodes a position along a parameterized 2-manifold. Note the BRDF is a special case of the SVBRDF. Moreover, it is often helpful to interpret the SVBRDF as consisting of a unique 4D BRDF at each surface position. The same properties of the BRDF we previously introduced apply to the SVBRDF as well. It is non-negative, reciprocal and conserves energy.

Although it is more general than the BRDF, the SVBRDF still cannot represent the appearance of all types of materials. In particular, the SVBRDF still assumes the surface is opaque, passive (i.e. it does not emit energy) and static. In Chapter 4 and 6 we mention more general light transport functions that relax some of these assumptions. As a rule, increasing the generality of an appearance function allows us to represent a broader class of materials, but requires having to consider a higher-dimensional domain. We will see that this has tremendous consequences on the feasibility of measuring and storing these functions.

2.4 Texture Maps

The conventional approach for modeling spatially-varying properties of a material's appearance is to map a 2D function (typically stored as an image and called a *texture map*) onto an object's 3D surface [13]. A texture map can be interpreted, for example, as encoding the spatially-varying diffuse albedo of a Lambertian BRDF (see Figure 2.8). However, texture maps provide a general technique for representing any property that varies with surface position and has been extended to represent spatially-varying surface orientation [11], surface displacement [18], surface roughness

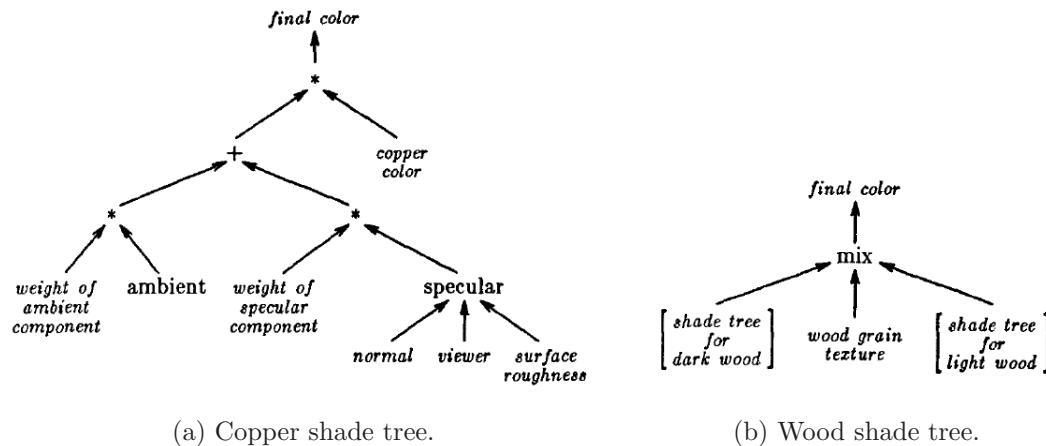


Figure 2.6: Shade trees describe the color of each pixel as a tree-structured collection of simpler functions and geometric properties of the underlying 3D scene. This figure is reproduced from the seminal paper by Rob Cook [18]

and even the surrounding environment’s color at mirror reflection [12]. A thorough survey of the computational issues related to texture maps along with a discussion of their extensive applications is available [43].

The challenge in representing real-world appearance using texture maps is three-fold. First, a designer must identify the spatially-varying properties of the target material (i.e. normal variation, albedo, surface roughness, etc.). Second, the characteristics of each property must be encoded into a separate texture map. These maps are either generated manually, defined by a procedural operation [87], or derived from natural images. Finally, there must exist a parameterization that maps points on the surface to points in the texture map.

2.5 Shade Trees

One of the challenges designers’ face when modeling the materials in a 3D scene is expressing complex appearance or shading functions using the low-level tools available to them. An important breakthrough was the introduction of *Shade Trees* by Rob Cook in 1984 [18]. Shade trees express a complex appearance function as a tree-structured collection of simpler masks and functions. As shown in Figure 2.6, the leaves of the tree correspond to both geometric properties of the 3D scene (e.g. normal direction, view direction, etc.) as well as particular aspects of a material’s



Figure 2.7: The appearance of this bowling pin is specified with a shade tree. Left to right: images rendered as the shade tree is traversed bottom-up. Note how its complex appearance is achieved by applying a sequence of relatively simple modifications that correspond to each leaf in the shade tree. This figure is reproduced from [92].

overall appearance (e.g. diffuse color texture map, rust pattern texture map, displacement map for surface wrinkles, etc.). The internal nodes of the tree describe how its sub-trees are combined in order to yield a more complex function. For example, a node may combine two texture maps in order to describe a material’s spatially-varying diffuse color **and** surface roughness. Shade trees give designers a powerful framework for modeling complex appearance from an ensemble of simple pieces that are more suitable for direct manipulation. They typically specify the leaves of the tree manually (e.g. create bump maps, displacement maps, albedo maps, etc.) along with its internal topology and composition nodes in order to achieve the desired appearance at the root node. Figure 2.7 shows a 3D scene of a bowling pin whose appearance was modeled with a shade tree. The advent of modern graphics hardware has brought renewed interest in shade trees as they can be implemented within small programs that execute in parallel at each pixel in the image.

Because there are potentially many ways to describe a complex appearance function, creating shade trees that meet a particular design goal require significant human expertise. Today, computer generated images typically contain tens to hundreds of shade trees. The art of their design lies in determining how to best characterize a complex function as a combination of simpler functions that can be expressed using available BRDF models, masks and images. In Chapter 4 we introduce a set of techniques suitable for inferring a shade tree from measurements of a complex appearance function. This *Inverse Shade Tree* (IST) framework is one of the key contributions of this thesis. We show that ISTs enable interactive rendering and editing of complex appearance functions

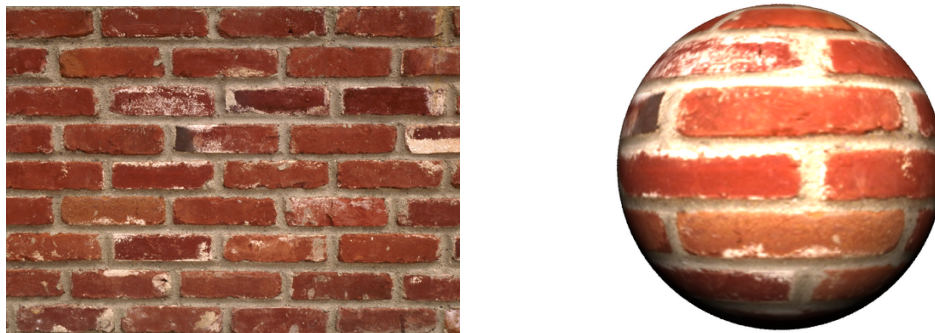


Figure 2.8: Using a photograph as a texture map is one of the earliest examples of appearance acquisition. Left: an image of bricks can be used to define the spatially-varying diffuse albedo of a material. Right: a sphere with this brick texture map applied to its surface.

derived from multi-gigabyte input datasets of measured SVBRDFs. Although artists will never be replaced by machines, we believe the IST framework presents a new and exciting design tool and suggests a promising direction of future work in data-driven appearance modeling.

2.6 Appearance Acquisition

Deriving analytic expressions for realistic surface reflectance is often a tedious (if not impossible) task. Although state-of-the-art BRDF models can express the appearance of a wide variety of materials, there are still many materials whose appearance is not well captured by existing techniques. Moreover, it's difficult to manually adjust the parameters of a particular analytic BRDF in order to achieve a desired result.

One way to address this challenge is to directly measure the appearance of a material. The earliest example of this approach is using a digitized photograph as a texture map. Figure 2.8 shows a 3D scene where an image of a brick wall has been used to modulate the diffuse color of a sphere. Although we get the sense that this object is intended to be made of brick, the rendered image lacks visual realism. This is because a single image captures the appearance of these bricks from a single view under fixed lighting conditions. In order to recover a complete representation of this material's appearance we would need many images of its appearance at different viewing directions and under different lighting conditions. More precisely, we would need knowledge of its SVBRDF.

In this section, we review previous research aimed at acquiring high-dimensional appearance

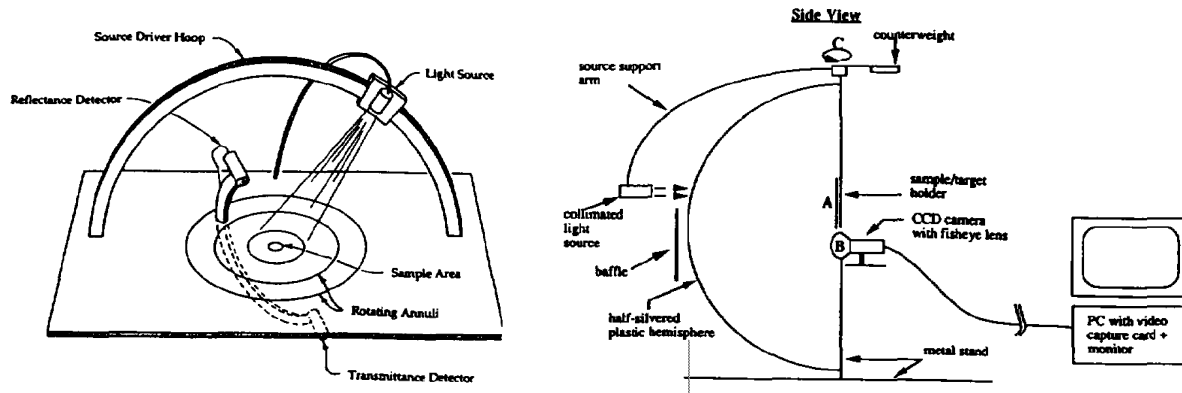


Figure 2.9: Two different gonioreflectometer designs. Left: the BRDF of a material is measured using a computerized photosensor and light-source. Right: using a half-silvered hemispherical mirror enables a single image to record the distribution of reflected light into the entire hemisphere. This design speeds up acquisition because it only requires exploring the 2D space of light-source positions. Figure reproduced from [109].

functions of real-world materials. We will see that this approach provides the most accurate representation of a material’s appearance, but requires time-consuming acquisition procedures, delicate calibration procedures and large storage costs. For this reason, providing efficient techniques to both acquire and store high-dimensional measured appearance datasets remains an active area of research.

2.6.1 Gonioreflectometry

A gonioreflectometer is a device for measuring the BRDF of a material. Traditional gonioreflectometers [82, 100] consist of a computerized photosensor and light source that can be positioned anywhere over the upper hemisphere of a homogeneous material sample. As a result, these devices are capable of sampling the full four-dimensional domain of the BRDF, but require long acquisition times because the photosensor and light-source must be sequentially positioned to cover the space of light and view directions.

An important development was the setup proposed by Ward [109]. It captures images of a hemispherical one-way mirror under variable light positions (see Figure 2.9). This reduces acquisition time because a single image measures the amount of light reflected into many outgoing directions. A similar approach to reducing acquisition time is taken by “image-based” setups [70, 74, 75]. As Figure 2.10 illustrates, these techniques capture multiple measurements of a BRDF simultaneously by imaging a curved convex object of a homogeneous material. Image-based devices have also been

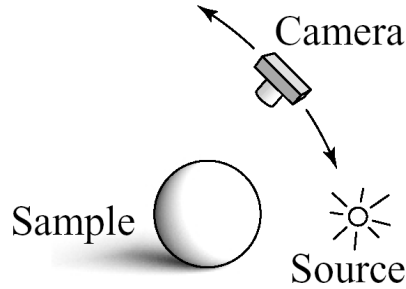


Figure 2.10: Image-based goniorelectrometry refers to using a digital camera as the photosensor. Because a digital camera records the amount of light striking each individual pixel, it can be used to acquire many measurements of the BRDF of a curved object simultaneously. This reduces the overall acquisition time by requiring exploration of only a 2D space of either light-source or camera position. Figure reproduced from [70].

developed to record measurements of spatially-varying reflectance [23, 112].

The measurements obtained from a goniorelectrometer can be stored in a large 4D table (3D for isotropic materials). Assuming the measurements are taken from a sufficiently dense sampling pattern, this approach provides the most accurate representation of a BRDF. However, this is still a time-consuming process dominated by tedious calibration procedures and the latency of positioning the light-source and photometer. Furthermore, a tabular representation is expensive to store and thus cannot be easily integrated into interactive and physically-based rendering algorithms nor can it be edited. In Chapters 4 and 5, we introduce two new representations of the SVBRDF and BRDF derived from measured data that address these shortcomings.

Because of the engineering challenges related to BRDF acquisition, there are still very few high-quality and high-resolution datasets in existence. A notable exception is the Matusik database [74]. It contains measurements of 55 different isotropic BRDFs including metals, plastics and several textiles. Each BRDF requires 33MB of storage. Obtaining a database of anisotropic BRDFs measured at a comparable resolution remains an open research problem.

2.6.2 Spatially-Varying Reflectance

The spatially-varying appearance of an object can be measured with a computerized light-source and digital camera. A spherical gantry is one such device and is shown in Figure 2.11. It consists of two arms that can be positioned to point toward a rotating base platform from any direction. If we

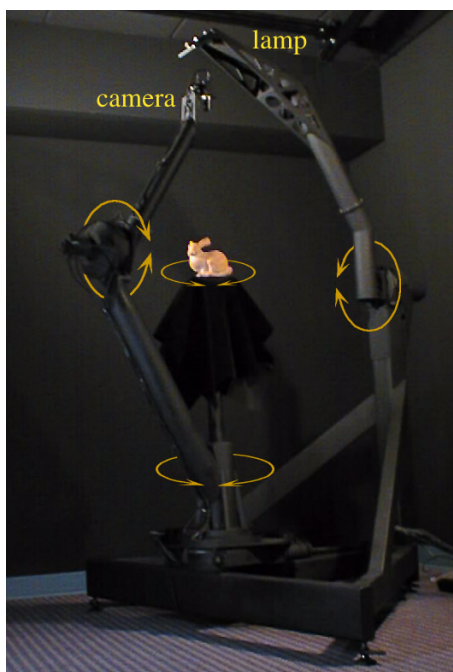


Figure 2.11: An image of the Stanford spherical gantry showing its four degrees of freedom. (Image courtesy of Stanford Graphics Laboratory.)

mount a digital camera and light source on the two arms as shown in Figure 2.11, we can measure various directionally- and spatially-dependent scattering functions. Dana et al. [21] measured the reflectance of 61 different materials at ~ 200 different view and light positions using a similar setup. Their *CUReT* database contains materials with spatially-varying appearance like wood, gravel, peacock feathers, etc. (see Figure 2.12).

As with measurements of a BRDF, these images can be stored in a large table. During image synthesis, we can look-up the appropriate value in this table for a particular light and view direction and surface position. Note that the scale and structure of the geometry of these surfaces does not match the microfacet theories used to derive many physically-based analytic BRDFs. However, if we still assume that these surfaces are planar we can rely on these images to encode the effect their actual geometry has on defining their appearance. This class of appearance functions with extended textures are commonly called Bi-Directional Texture Functions (BTFs) in contrast to the spatially-varying BRDF described in Section 2.3 for which the true surface geometry (at least on the scale of the wavelength of light) is available.

Malzbender et al. [69] record images of a spatially-varying material from a *fixed* view, but

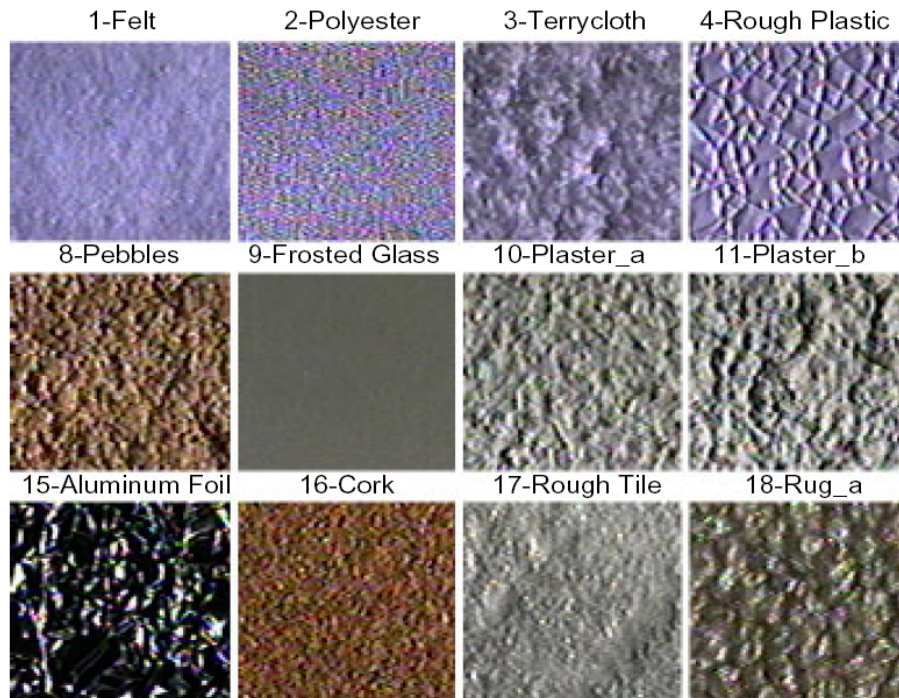


Figure 2.12: A few datasets in the CURET database [21] that contains reflectance measurements of 61 different spatially-varying materials. Figure reproduced from [21].

under different directions of point-source illumination. Surprisingly, this 4D subset of the full 6D appearance function is capable of reproducing visually compelling images. Nevertheless, the restriction to a single view limits their generality.

There has also been previous work in acquiring the SVBRDF of real-world materials. Lensch et al. [63] and Debevec et al. [23] record images of a curved object with spatially-varying reflectance at different view and light-source positions. Because they estimate an accurate representation of the object’s 3D geometry these images provide measurements of its spatially-varying BRDF. More recently, Marschner et al. [72] acquired the SVBRDF of different types of polished wood. In Chapter 3 we discuss a technique for acquiring the SVBRDF of several real-world materials.

2.6.3 Translucent Materials

Several techniques have been proposed for acquiring the appearance of translucent materials. When light strikes a translucent material, a significant percentage scatters beneath its surface and is re-emitted into the surrounding environment at a different position. Modeling these materials

requires resolving their Bi-Directional Subsurface Scattering Reflectance Distribution Function (BSSRDF). The BSSRDF is an 8d function that records the amount of light transported between any pair of incident and reflected directions at different surface positions. Because of its high-dimensionality, it is infeasible to exhaustively measure its domain and current approaches measure only a subspace [33, 86].

2.6.4 Summary

Efficiently acquiring the appearance of many real-world materials remains an open research problem. With the exception of 3D isotropic BRDFs, the high-dimensionality of more general scattering functions prohibits designs that densely sample their complete domain. One promising approach are acquisition setups that enable a single image to record multiple measurements of a light transport function simultaneously. Another interesting direction are techniques that acquire only a subset or lower-dimensional projection of the full high-dimensional function. The challenge with these techniques is determining which subsets carry the most information about the material’s appearance and, of course, providing a useful physical setup.

2.7 Fitting Analytic BRDFs to Reflectance Data

One common approach for incorporating measured appearance data into existing rendering algorithms is to fit the parameters of an analytic BRDF model to the measurements. In fact, several BRDFs were designed in the context of fitting to measured data [109, 56]. In general, this process requires performing non-linear optimization over the BRDF parameters in order to recover a locally optimal fit. Ngan et al. [81] evaluate the performance of fitting several popular analytic BRDF models to a database of 100 measured isotropic BRDFs [74]. They evaluate seven different analytic models: Ward [109], Blinn-Phong [9], Cook-Torrance [20], Lafortune [56], Ashikhmin-Shirley [5] and HTSG [42]. They achieved the most accurate results by fitting a single specular lobe of the Cook-Torrance, Ashikhmin-Shirley and HTSG models, although this approach still introduces significant errors for many materials. For materials with multiple layers of finish, the accuracy of the fit is improved with an additional specular lobe. In all cases, they report that a half-angle parameterization of the specular lobe better matches measured data than one based

on the direction of perfect mirror reflection. They also note that the required optimization is computationally expensive and often numerically unstable (especially for multiple specular lobes) and that great care is needed to avoid local minima. Lastly, they recorded sparse measurements of several anisotropic materials and found no existing analytic BRDF model was able to accurately fit their reflectance and instead used a hybrid representation similar to that proposed by [4].

One possible approach for representing measured SVBRDF data is to fit parameters of an analytic BRDF at each surface location [63, 76, 31]. Such a representation provides for easy editing of materials, and with the addition of a clustering step [64] allows editing a single material everywhere it appears on a surface. This approach, however, has several key drawbacks. As with the BRDF, reducing a dense set of measurements to a handful of parameters may introduce significant error. Moreover, it requires non-linear optimization, which is computationally expensive and numerically unstable. Finally, clustering the values of the BRDF parameters across the surface does not generate a desirable separation of the component materials in the presence of blending on the surface (even, in some cases, the trivial pixel-level blending present at antialiased material edges). This is because the problem is underconstrained (i.e., there are many possible cluster allocations with comparable approximation error). Moreover, these techniques have only been demonstrated on relatively simple isotropic materials. In Chapter 4 we introduce a new approach for addressing this *material separation* problem. We pose it as computing the factorization of a matrix and introduce a new set of algorithms designed to produce intuitive decompositions. We demonstrate the effectiveness of our approach on datasets with complex spatial blending that include both anisotropic and retro-reflective surface reflectance.

2.8 Factored BRDF Models

Both analytic and measured BRDFs can be represented as a large 4D table of values (3D if the BRDF is isotropic). Although a tabular model provides a very accurate representation of the BRDF it does so at the expense of substantial storage requirements (e.g. 35MB/3GB for isotropic/anisotropic at resolutions comparable to the Matusik database [74]). For example, interactive applications require that all the geometry, texture maps and materials for a scene must not exceed the memory available on consumer graphics hardware. As a result, it is important to

compress these representations.

One key property of the BRDF is that it is typically *factorizable* or *separable*. A factored function is one that is written as the product of lower-dimensional functions. The work of Neumann & Neumann [80] seems to mark the first time this property of the BRDF was exploited for computational efficiency. They were concerned with computing radiosity solutions for scenes with non-Lambertian materials. They introduced the notion of *separable reflectance* and described a (very limited) class of BRDFs that can be written as the product of two 2D functions dependent on the incoming and outgoing directions respectively. However, the majority of BRDFs are not separable into this form. For example, the location of the specular lobe *depends* on the position of the incoming and outgoing angles. To address this, Fournier [29] and DeYoung et al. [26] used Singular Value Decomposition (SVD) to factor the BRDF into a sum of multiple products of these 2D functions:

$$f_r(\omega_i, \omega_o) = \sum_{j=1}^J F_j(\omega_i)G_j(\omega_o), \quad (2.21)$$

where the BRDF f_r is approximated by the sum of J products of the 2D functions F_j and G_j that depend on the incoming and outgoing angles respectively. Equation 2.21 allows an arbitrary number of terms and can, in theory, represent the original BRDF to arbitrary precision. However, because most BRDFs are not factorizable into this form, accurate factorizations typically require many terms. Also note that this numerical factorization presents significant computational challenges related to generating, storing and factoring a very large and dense matrix.

Heidrich et al. [44] observed that certain analytic BRDFs, such as the Cook-Torrance and Banks model, can be written as the product of lower-dimensional functions. These *analytic factorizations* perfectly match the original BRDF and typically depend on mixed parameterizations of the domain. They show how a slightly modified version of the Cook-Torrance BRDF can be written as the product of analytic pieces that depend on the outgoing, incoming and half-angle directions. These individual factors can be sampled and stored in texture maps making more complex analytic BRDFs available to interactive rendering systems. Of course, for certain analytic models and all measured BRDFs this type of analysis is not possible.

Rusinkiewicz [94] introduced a parameterization of the BRDF that better aligns the main fea-

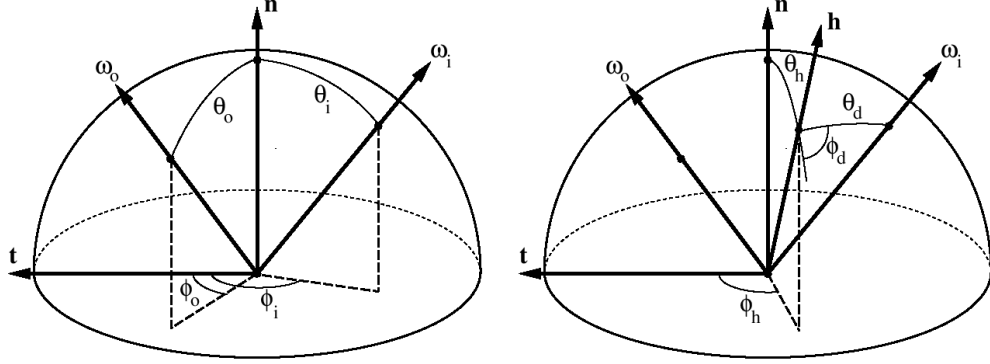


Figure 2.13: Rusinkiewicz [94] describes a parameterization of the BRDF based on the half- and difference-angles. Left: standard spherical parameterization of the incoming and outgoing directions, $(\theta_i, \phi_i, \theta_o, \phi_o)$. Right: each pair of incoming and outgoing directions is represented by the spherical coordinates of the corresponding half and difference angles, $(\theta_h, \phi_h, \theta_d, \phi_d)$. Figure reproduced from [94].

tures of its reflectance (i.e. retro-reflective, specular and diffuse lobes) than the vanilla spherical parameterization. He describes a parameterization of the BRDF with respect to the half-angle as in [9] and the difference-angle which is the incoming direction represented in a frame defined by the half-angle. This half/difference frame is illustrated in Figure 2.13. The main advantage of this parameterization is that it makes the inherent redundancy of most BRDFs available for subsequent compression and factorization techniques. Kautz et al. [53] note the important role parameterizations plays in efficiently factoring a BRDF. They also use SVD to compute the factorization like in [29], but build a matrix of BRDF values that are uniformly spaced in this half/difference frame. This allows efficient factorization (i.e. higher accuracy for fewer terms) for very glossy isotropic and anisotropic BRDFs.

McCool et al. [77] also note the significant impact the choice of parameterization has on the accuracy and compactness of factored BRDF models. They introduce Homomorphic Factorization (HF): an algorithm for computing for factoring a BRDF into a single product of an arbitrary number of terms. The main advantage of HF is that it can handle scattered and sparse measurements of the BRDF without requiring the construction of a regularly spaced matrix. It also guarantees the factors are strictly non-negative through a logarithmic transformation. Non-negativity is important due to the lack of support for signed arithmetic on traditional consumer computer graphics hardware (this restriction has been removed with the latest generation of graphics hard-

ware). They are able to compute accurate factored representations with very few terms (i.e. 1 or 2) of several different analytic and measured BRDFs using the half/difference parameterization and the HF algorithm¹.

When measured BRDF data is available, computing a factored model provides a more accurate representation than fitting a small number of parameters of an analytic function [77, 60]. Not surprisingly, the accuracy of these models comes at the expense of larger storage requirements (although they are still compact enough to support interactive rendering) and engineering challenges related to generating and factoring a large dense matrix.

An important contribution of this thesis is the development of several new factored representations of the SVBRDF and BRDF that address previously open problems related to editing and importance sampling these models. In Chapter 4 we introduce a tree-structured representation of the SVBRDF that is based on a sequence of matrix factorizations of measured surface reflectance data. We evaluate the performance of existing algorithms and conclude that no single technique guarantees *physically-plausible* separations. To address this, we introduce a new set of matrix factorization algorithms in Section 4.5. By incorporating general and domain-specific constraints, they guarantee a physical-plausible factorization of the SVBRDF and BRDF into *intuitive* factors. In Chapter 5 we introduce a new factored BRDF model that is designed to provide efficient importance sampling in the context of global illumination rendering. We will see that this application places interesting constraints on both the parameterization and the properties of the individual factors.

2.9 Rendering

Computer graphics is concerned with the synthesis of images of a 3D scene – a process known as *rendering*. Rendering algorithms take as input computer models of the 3D geometry of the shapes in the scene, illumination sources and material or appearance models and compute an image of the scene from a specific viewpoint according to these specifications. These algorithms have rightfully gained considerable research attention in the last 30 years. Broadly speaking, they fall into two different categories: physically-based and interactive.

¹<http://www.cgl.uwaterloo.ca/Projects/rendering/Shading/database.html>

2.9.1 Physically-Based Rendering

Physically-based or global illumination rendering algorithms are concerned with maximizing the visual realism of the synthetic image. These algorithms are appropriate, for example, when the final image will be composited onto live action footage.

To recover important visual effects, these algorithms simulate the way light is transported through the scene before striking the virtual image plane. Ultimately, they are interested in computing the radiance traveling along the group of rays that intersect each image pixel. Due to the reciprocal nature of light, this simulation is equivalent to tracing paths from the camera into the scene and simulating the way they are transmitted, absorbed, refracted and reflected by different materials before striking the light sources. Suppose one such camera ray intersects the scene at a point x with associated surface normal n . The total amount of radiance traveling along this ray back toward the image plane can be computed through numerical integration of the *rendering equation*, first formulated by [51]:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega_+} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i d\omega_i, \quad (2.22)$$

where L_o is the radiance leaving point x along the direction ω_o toward the image pixel, L_e is the radiance that is emitted by the material at point x , L_i is the radiance arriving at point x along the direction ω_i and f_r is the BRDF of the material at x . Recall the cosine term is necessary to convert the radiance traveling along ω_i into the irradiance incident to the differential surface patch at point x .

The domain of integration in Equation 2.22 is over the space of incoming directions at x or, if we consider only opaque surfaces, the upper hemisphere Ω_+ aligned with the surface normal n . Therefore, $L_i(x, \omega_i)$ will, in general, require evaluating Equation 2.22 at a separate point in the scene. Equations of this form are called *Fredholm equations of the second kind*. These equations have no known analytic solutions thus requiring numerical estimation of this recursive function and suggests the difficulty in efficiently computing $L_o(x, \omega_o)$.

The approach of Monte Carlo rendering algorithms is to evaluate the incoming radiance L_i by recursively casting rays through the scene to simulate light transport paths. In order to reduce the variance in the estimate of Equation 2.22, it is desirable to *importance sample* reflected rays by

preferentially considering paths carrying high energy. We refer the reader to textbooks that cover this fascinating area of numerical integration [91, 28].

One possibility is to consider drawing samples according to an importance function derived from the lighting L_i , as in sampling light sources for direct illumination. In the special case when L_i is described by an environment map [12], effective illumination sampling methods have recently been demonstrated [1, 55]. These methods are intended for diffuse or rough semi-glossy materials, and typically require a few hundred samples for good results. However, at low to medium sample counts, they may miss important details in the illumination, especially for glossy materials and slowly-varying environments. For these cases, it is desirable to importance sample according to the product of the BRDF f_r and the incident cosine term $\cos \theta_i$. This can be achieved using the inverse of a Cumulative Distribution Function (CDF) derived from the BRDF. This inverse function transforms a uniformly distributed random variable into a random variable distributed according to the energy in the BRDF. There exists a closed form expression of the inverse of the CDF for several analytic BRDFs such as the Blinn-Phong, Ward and Lafortune models. For more complex analytic BRDFs like the Cook-Torrance model, however, no such analytic expression exists and generating samples according to these models has remained an open area of research.

In Chapter 5 we introduce a new factored BRDF model that enables efficient importance sampling of measured and complex analytic material models. We also introduce a representation of general sampled functions that provides compression and enables efficient importance sampling. In Section 5.5 we use this representation to generate samples according to both measured BRDFs and image-based illumination.

2.9.2 Interactive Rendering

Interactive rendering algorithms are designed to compute images at real-time frame rates (e.g. ≥ 20 Hz). These algorithms are appropriate for applications like video games, virtual reality simulations and interactive data visualization. In order to render a complete image within 30^{-1} seconds, these algorithms must restrict their evaluation of Equation 2.22 to consider only a subset of the full lighting interactions within a scene. Typically they work by considering only *local* lighting effects: light that is transported directly from a point source, reflected off of a single surface and into the camera. This can be interpreted as approximating the rendering equation:

$$L_o(x, \omega_o) \approx L_e(x, \omega_o) + \sum_{j=1}^N L_i^j f_r(x, \omega_i^j, \omega_o) (n \cdot \omega_i^j), \quad (2.23)$$

where N is the number of point light sources, L_i^j is the radiance emitted from the j^{th} source, ω_i^j is a unit-length vector pointing toward the light source from position x and n is the surface normal at x . Note that the cosine term in Equation 2.22 is equal to the dot product $(n \cdot \omega_i^j)$.

The complexity of evaluating Equation 2.23 is proportional to the scene geometry, number of light sources and the cost of evaluating each BRDF in the scene as opposed to computing the high-dimensional integral required for full global illumination. Although computing Equation 2.23 can be accomplished at interactive rates using specialized graphics hardware, it clearly ignores many important visual effects. Also note that graphics hardware has a limited amount of available memory (e.g. ~ 512 MB) and must accommodate the geometry, lighting models and BRDFs in each scene. For this reason, it is important to *compress* measured BRDFs before they can be incorporated into an interactive setting.

In Chapter 4 we introduce a novel data-driven representation of the SVBRDF that provides compression ratios sufficient to support interactive rendering. This work is most closely related to previous techniques for rendering scenes containing measured or non-parametric BRDFs at interactive rates [53, 77].

2.10 Appearance Editing

For an appearance model to be useful within a production pipeline it is important that it supports editing. In many cases, analytic BRDF models were designed specifically with this goal in mind. For example, the Lambertian, Phong, Ward and Cook-Torrance BRDFs all have parameters that correspond to salient aspects of appearance. For example, adjusting k_d in Equation 2.9 controls the albedo or diffuse color of a material modeled with the Lambertian BRDF. The parameter n in Equation 2.10, m in Equation 2.14 and α_x and α_y in Equation 2.19 all control the “glossiness” or “shininess” of the resulting appearance for the Phong, Cook-Torrance and Ward BRDFs respectively.

In Chapter 4 we introduce a data-driven BRDF representation that allows editing. The BRDF is represented with a set of sampled curves computed automatically from measured data. We

develop matrix factorization algorithms that generate intuitive curves that can be directly edited by a user familiar with surface reflectance functions.

A data-driven approach to BRDF editing has been proposed by Matusik et al. [74], in which a user labels directions (e.g., “shininess”, “rustiness”, etc.) in a high-dimensional space of measured materials. The large number of materials, and the uncompressed representation of each BRDF, contribute to large storage requirements and inhibit interactive control. Though our BRDF editing system is also based on measured data, we provide for direct manipulation of curves controlling the reflectance instead of focusing on higher-level behaviors. In addition, our compressed representation and interactive renderer permit real-time manipulation of materials, including spatial variation.

To our knowledge, only two systems have demonstrated material editing via direct manipulation of low-dimensional factors of the BRDF [4, 48]. In these systems the user edits a 2D image, which is then used as a component in a parametric or homomorphic-factored BRDF model. We generalize this by allowing interactive control over all the 1D and 2D factors necessary to specify a full SVBRDF.

Chapter 3

Appearance Acquisition

Although the data-driven representations we present in Chapter 4 could in principle represent many types of appearance data, including BTFs, BSSRDFs, light fields, and time-varying textures, we focus on SVBRDFs. In this chapter we introduce our acquisition pipeline, describe the five datasets that we have acquired and conclude by discussing alternative setups.

3.1 Spherical Gantry

A spherical gantry is designed to measure the appearance of a 3D object. It consists of two computerized arms that can be positioned anywhere on a sphere centered around a rotating base platform. The design restricts these arms to always point at the rotating platform. The gantry's physical configuration along with its four degrees of freedom are illustrated in Figure 2.11. If we mount a digital camera and light source on these arms as shown in Figure 2.11, we can measure various directionally- and spatially-dependent scattering functions. In particular, we can place a spatially-varying, static and opaque material on the base platform and obtain measurements of its SVBRDF at different incident and reflected directions by adjusting the position of the camera and light-source. We show several images of two wallpaper samples captured with such a setup in Figure 3.1.

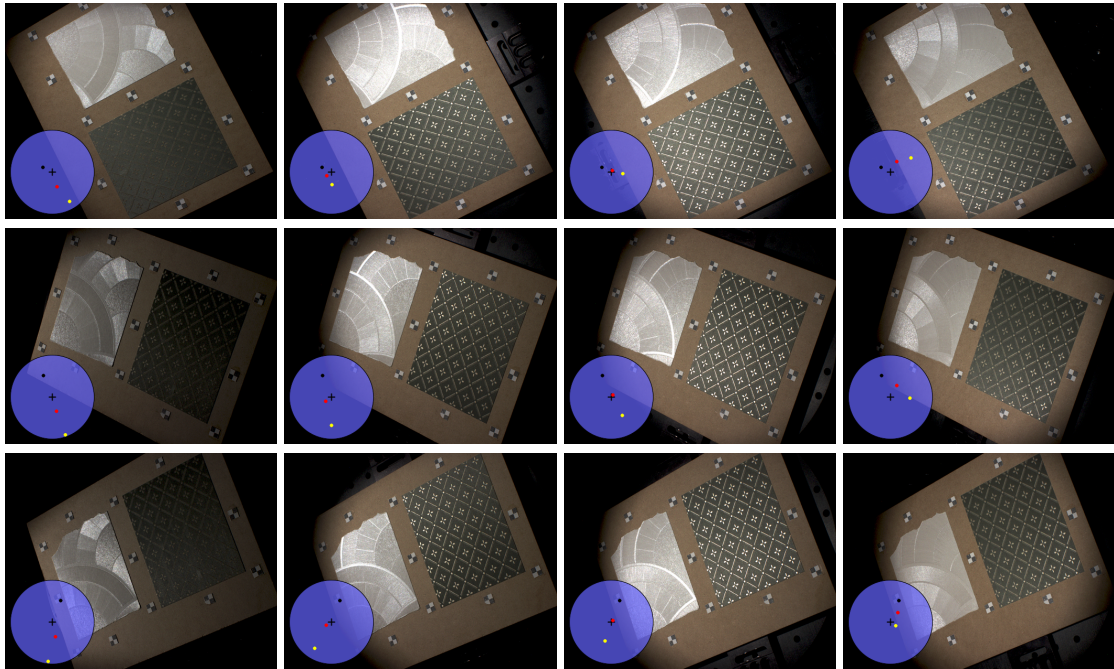


Figure 3.1: Several raw images of the wallpaper samples acquired with a spherical gantry. The diagrams in the bottom-left corner indicate the hemispherical position of the camera (black dot), light-source (yellow dot) and the associated half-angle vector (red dot) relative to the surface normal of the base platform (black cross).

3.2 High Dynamic Range

Modern digital cameras have a dynamic range of around 1:100. Within a single exposure they can resolve intensities whose relative values lie within this ratio. However, the scattering functions (i.e. BRDFs and SVBRDFs) for most materials exceed this limited range. For example, some materials in our dataset have a dynamic range in excess of 1:10,000. Following the work of [25], we combine several exposures at different shutter speeds for each camera position in order to recover a high-dynamic-range (HDR) image (see Figure 3.2). Although this image provides a better estimate of the radiance reflected at each position, it increases the acquisition time. For the datasets that we acquired, we typically recorded four or five exposures with shutter speeds ranging from 2ms to 1s. The total amount of time to acquire a single HDR image was between 2-3s (this includes both the time to capture each image plus the time to re-position the arms of the gantry).

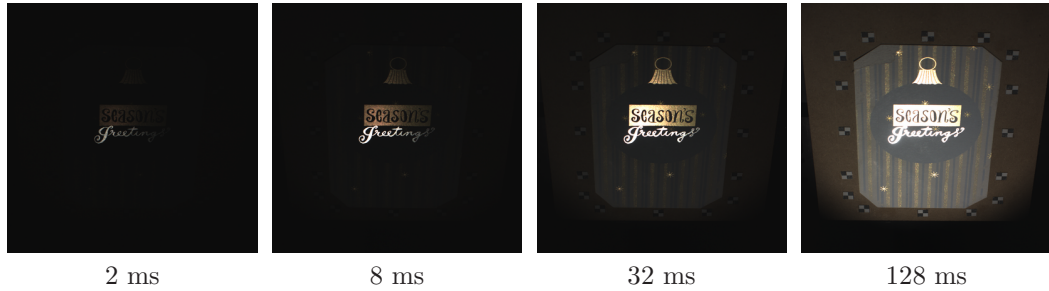


Figure 3.2: Images of the Season’s Greetings SVBRDF at different shutter speeds for a fixed camera and light position. These images are combined to form a high-dynamic-range (HDR) image that records the amount of radiance reflected into the camera at each surface position using the technique of Debevec et al. [25].

3.2.1 Sampling

Due to the high-dimensionality of the functions we are interested in measuring it is infeasible to densely measure their entire domain. As an example, suppose we wished to measure the directional variation of a SVBRDF to within a single degree in each of its four angular dimensions. For a planar sample, this would require taking $360 \times 180 \times 360 \times 180$ measurements. In our setup, acquiring a single HDR image requires ~ 2 s and ~ 8 MB of storage. Therefore, the complete dataset would require 15 petabytes of storage and take 266 years to acquire. After this long, you would not expect the material to have the same optical properties! Therefore, we need a technique for reducing the total number of images that we must acquire without compromising the accuracy of our record of a material’s reflectance.

Depending on the specific material, we expect variation of its appearance to be localized to certain regions of the SVBRDF domain. For the materials we measured, their reflectance varied quickly around the directions of specular and retro-reflection while changing smoothly across other regions. Therefore, we measured these functions more densely around their specular highlights and at backscattering configurations. Two of the datasets exhibit isotropic reflectance (see Table 3.1). For these cases, we further reduced acquisition time by measuring only 3 angular dimensions. Specifically, we fixed the azimuthal angle of the viewing direction to a constant value.

With these optimizations we were able to decrease the acquisition time while providing a suitable reconstruction of the material’s appearance. We implemented a simple program that allows a user to manually specify the size and location of different sampling regions and the sampling

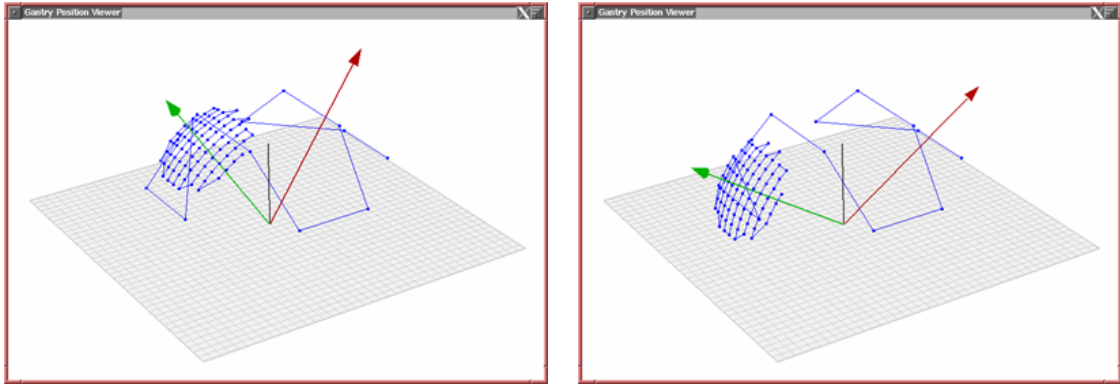


Figure 3.3: Two screenshots of a simple application for visualizing the location of the camera and lamp arms of a spherical gantry. The red arrow is the position of the camera and the green arrow shows the direction of perfect specular reflection for this camera position. The gantry records a separate image for the light-source at each position marked by a blue point. The edges connecting these points illustrate the *path* of the lamp arm.

density within each region. Figure 3.3 contains screenshots of this program showing two different sampling patterns. Although this process was effective in reducing acquisition times, the general problem of *view-planning* should be addressed with more theoretically motivated techniques. Certainly, issues related to optimal adaptive sampling from the field of information theory would apply to this problem as well. Although there has been recent work on this topic for a particular choice of parametric BRDF, [65], a more generic treatment of this problem is still an open problem.

3.3 Calibration

As seen in Figure 3.1, we obtain images of mostly planar spatially-varying materials for different camera and light positions. One option is to use these images as the final representation. However, this image-based approach would not allow mapping these materials to arbitrary geometry. Moreover, using these images directly would not support interactive rendering, sampling in the context of physically-based rendering nor editing. Imagine trying to adjust the “diffuse color” of the green material in the wallpaper sample by consistently changing 6,000 images!

In order to provide a more useful representation we must first organize these raw images into scattered samples of the BRDF at each surface position. In general, this task requires knowing the position of the light-source and camera for each image along with the geometry of the object. In our case, we assume the geometry is a perfectly flat plane. Although these objects are not perfectly

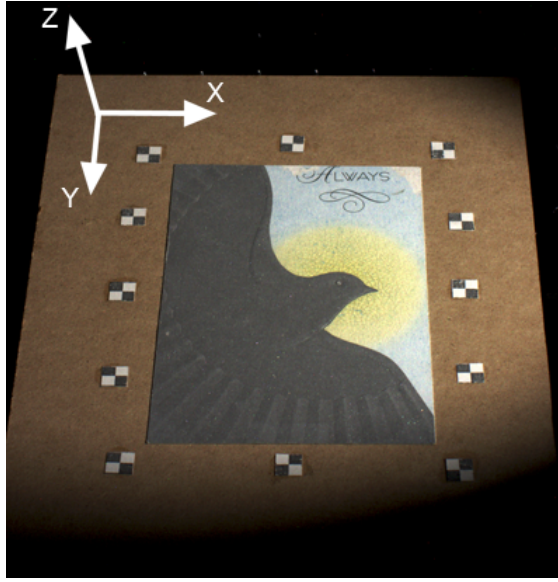


Figure 3.4: Image from the *Dove* dataset with superimposed canonical coordinate frame.

flat due to physical irregularities and subtle embossing effects, this is a reasonable assumption. In Section 4.6, we discuss a procedure for estimating the fine geometric detail of these objects thus relaxing our original assumption.

The process of estimating the position of the camera and light source at each image is known as *calibration*. The servo motors that control the arms of the gantry indicate their position relative to the base platform within 1 or 2 degrees. Although this provides a helpful estimate of their position, using these values to warp the raw images into a canonical image plane would result in significant loss of spatial and angular detail. The result would be blurry reconstructions of the material with poorly defined boundaries.

3.3.1 Image Alignment

The purpose of camera calibration is to place the input images into a common coordinate system. In our case, a useful coordinate system is one with its x-y plane coincident to the face of our planar sample and with a z-axis pointing upward (see Figure 3.4). Each image can be thought of as a perspective projection of the x-y plane in this coordinate system. If we knew the parameters of this warp (also called a homography) we could easily map each input image into this common coordinate system. However, determining these parameters requires knowledge of the precise position of the

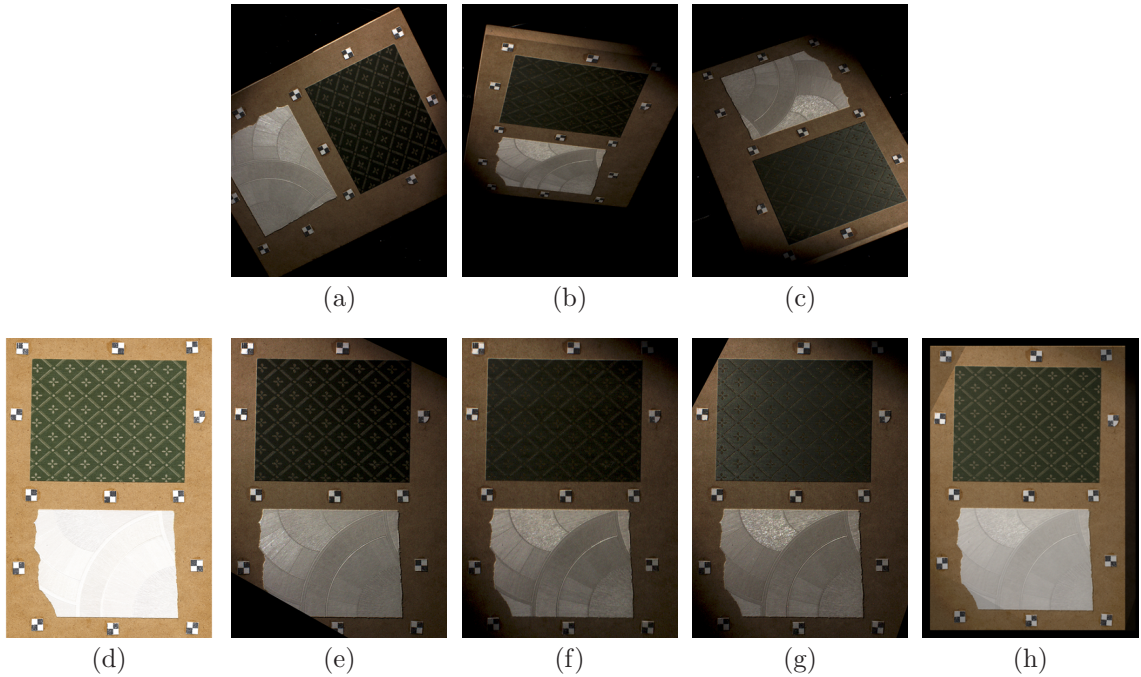


Figure 3.5: An illustration of our alignment procedure using images from the *Wallpaper* dataset. (a-c) Source images acquired with a spherical gantry at different camera and light-source positions are aligned to (d) a target image generated by scanning the planar sample using a flat-bed scanner. (e-g) The aligned images are accurate to within $1/8$ of a pixel with respect to the target. (h) We can gain a sense of the precision of our alignment procedure by showing the warped images (e-g) superimposed atop one another.

camera relative within this coordinate system. We solve this problem in two steps. First, we *define* a coordinate system relative to an image of our planar sample obtained from a flat-bed scanner. The scanner returns a near-orthographic view of the sample and we *assume* the pixels in this scanned image are uniformly spaced along the x-y plane of our hypothetical coordinate system shown in Figure 3.4. Although these planes do not align exactly, the margin of error is within the resolution of our imaging equipment. Furthermore, errors introduced by this assumption can be corrected when we estimate the fine geometric detail of the surface (see Section 4.6). The second step is to estimate the parameters of a homography that best aligns each image obtained with the gantry (called the *source image*) to this scanned image (called the *target image*). Computing this alignment effectively recovers the position of the camera relative to this coordinate system and allows us to combine our raw input into a representation suitable for further processing.

Image alignment is a well studied problem in computer vision [8, 68] and graphics [99]. A

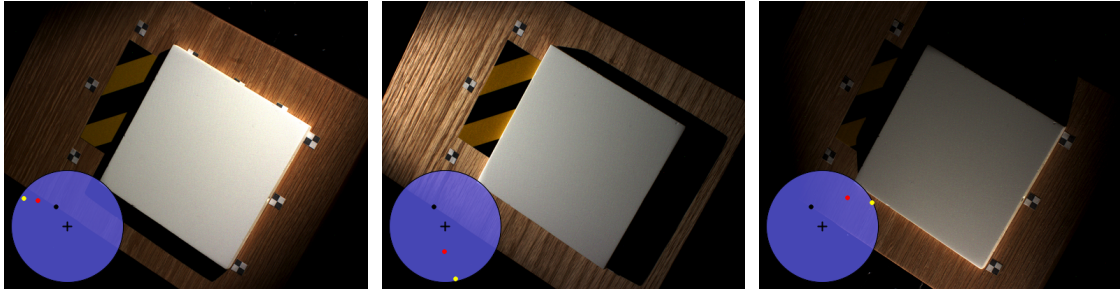


Figure 3.6: Images of Spectralon[®] at a fixed view for different light positions. These images are used to estimate the irradiance arriving at each point on the surface for an arbitrary light-source position. Computing the SVBRDF amounts to dividing the outgoing radiance recorded at each pixel of the image by this estimate of the incoming irradiance

common approach is to align a small set of features common to all images and visible from all viewing directions [68, 99]. Therefore, it is important to select features in the scene that produce highly-correlated pixel regions in each HDR image regardless of the position of the camera and light-source when the image was acquired (assuming the light-source is above the horizon). Diffuse materials with sharp corners are good image features [68]. We used a laser printer to print a checkerboard pattern onto thick white paper. As seen in Figure 3.4, these paper squares were placed in a ring around each planar sample and are manually detected within the target image. We use the position of the camera reported by the gantry to construct an initial homography for each source image. Using the Nelder-Mead simplex algorithm [79], we perform non-linear optimization over the parameters of this homography to best align the feature locations in the target image with those in the source image. Although this procedure guarantees recovering only a locally optimal homography, our good starting position along with using stable features allows this process to work very well in practice. For each dataset, we aligned all of the source images to the target image without a single failure. We performed this alignment to within $\frac{1}{8}$ of a pixel using modern graphics hardware to accelerate the computation of each perspective warp and to compute the normalized cross-correlation between a pair of pixel windows centered around features in the source and target image. The entire processing time was roughly 5 seconds per source image or about 9 hours for each dataset.

3.3.2 Photometric Calibration

We need to associate the pixel values in each HDR image with values of the underlying BRDF. This can be accomplished by acquiring images of a material with known reflectance properties. We record HDR images of a planar sample of Spectralon[®] from a fixed camera position and 100 different positions of the light source uniformly distributed over the upper hemisphere. If we assume the Spectralon[®] is a perfectly white Lambertian reflector, these images encode the colorized intensity of the surface irradiance produced by placing the light source at these different locations in units proportional to *pixel values*. Several images of this irradiance calibration dataset are shown in Figure 3.6.

During acquisition, the position of the light-source will not perfectly coincide with one of these images so we interpolate the values between the three nearest images to obtain an estimate of the incident irradiance. We can now divide each value in the image of our target material by this *irradiance estimate image* and obtain values of the BRDF in units of inverse steradians at each pixel. This is performed in each color band, thus accounting for the non-uniformity of the spectral intensity of the spherical gantry’s light source.

3.4 Reconstruction

After aligning and color balancing the raw images, we have samples of the SVBRDF that are uniformly spaced in the spatial dimensions, but still non-uniformly spaced in the four angular dimensions. In order to carry out the factorizations discussed in Chapter 4, we need to interpolate the scattered SVBRDF measurements into a uniformly spaced grid. The scattered data are conceptually resampled onto a uniform grid using a push-pull algorithm [36] with a Gaussian reconstruction kernel (as we shall see later, we avoid a complete reconstruction by using a subsampling method). The standard deviation of the Gaussian kernel is manually set based on the observed density of the scattered measurements and the resolution of the table (one could imagine estimating this automatically). This interpolation scheme maintains a sum of the reflectance values and weights at each cell at coarse-to-fine resolutions. After estimating the values in the matrix, we associate a confidence value with each cell equal to $\text{MIN}(1.0, \text{total_weight_at_top_level})$. ”Unmeasured regions” are those that correspond to incident or reflected directions below the horizon, and their

confidence is set to zero. The matrix factorization algorithm that we introduce in Section 4.5 uses these confidences to weight the contribution of each element in the matrix toward the error. This has the effect of interpolating values across missing regions in the final factorization.

Reconstructing scattered measurements into a uniformly spaced matrix is one of the most expensive operations in our pipeline. This underscores the utility of acquiring very densely sampled datasets as these would require less expensive interpolation procedures. For example, the isotropic BRDF database acquired by Matusik et al. [74] has multiple measurements for each “cell” in their table so no interpolation is needed. Instead, they use a simple binning procedure that requires computing only the median of a small set of numbers at each grid position. The challenge, of course, is recording a dense set of measurements within feasible acquisition times.

3.5 Datasets

We used a spherical gantry [72] to record a set of high-dynamic-range images of planar samples with spatially-varying material under many illumination and view directions. The choice to measure mostly-planar samples simplified the calibration and registration (Section 3.3.1), but the techniques presented in Chapter 4 apply to arbitrary geometry. We acquired five different SVBRDFs that include layered materials, anisotropy, retro-reflection, and spatially-varying normal and tangent directions. The complete processed data along with source code for file I/O can be downloaded from a public web server¹. Several images from each dataset are shown in Figure 3.7 and a description of their qualitative and quantitative properties is provided in Table 3.1.

3.6 Alternative Acquisition Setups

One key limitation of a spherical gantry is that the camera and/or light-source must be re-positioned between each exposure. This results in long acquisition times and restricts its use to stationary and static samples.

An interesting alternative is to use multiple sensors to parallelize the acquisition process. One such design was recently proposed by Weyrich et al. [112] for acquiring the reflectance of human skin (shown in Figure 3.8). Similar setups have recently been described [24, 110]. All these systems

¹<http://www.ist.cs.princeton.edu>

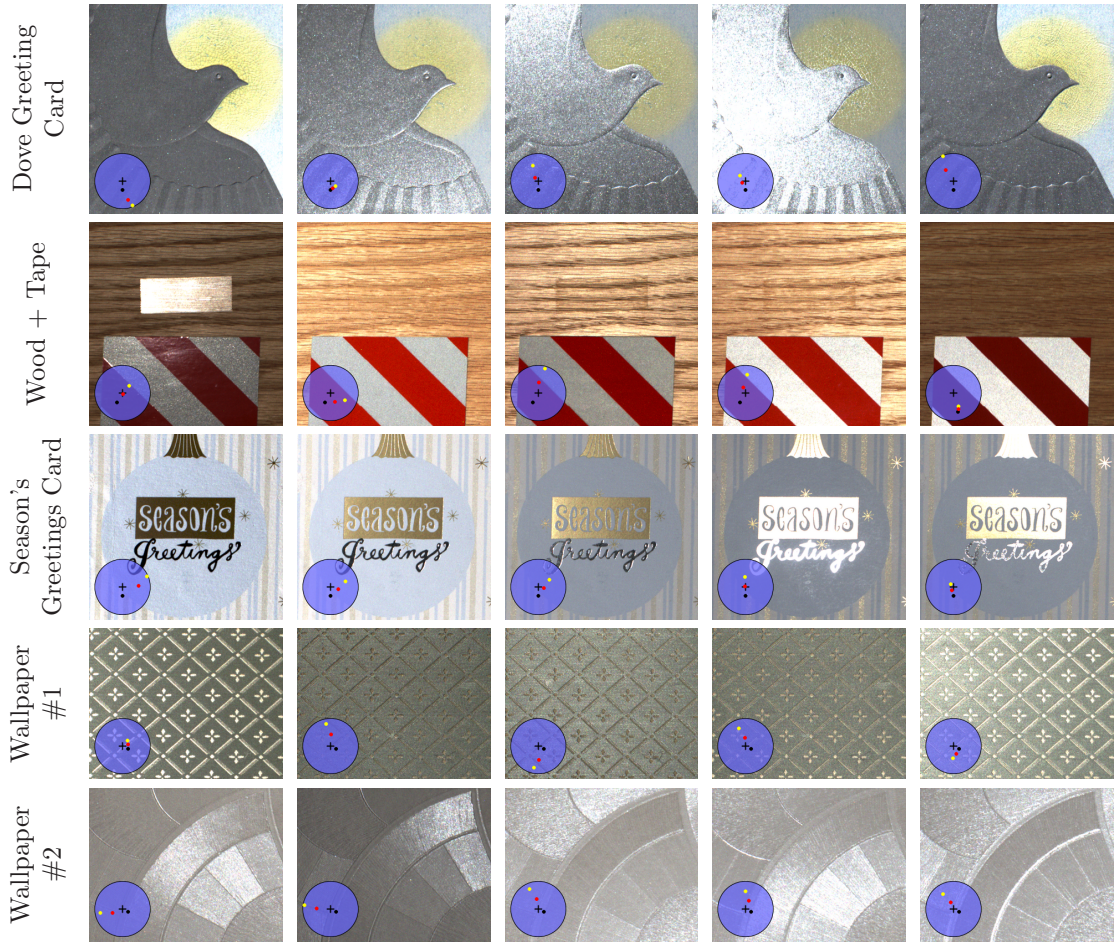


Figure 3.7: The five SVBRDF datasets we acquired.

Dataset	Material Count	Spatial Res.	Ang. Res. $V! \times L$	Size (GB)	Reflectance Properties	Normal Variation	Tangent Variation
Season Greetings	4	523×500	5×350	5.5	Iso	Yes	N/A
Wood+Tape	~ 5	380×400	12×480	10	Aniso/Retro	Yes	Yes
Dove	~ 3	470×510	5×400	5.5	Iso	Yes	N/A
Wallpaper 1	~ 2	470×510	28×175	10	Aniso	Yes	No
Wallpaper 2	~ 3	470×510	28×175	10	Aniso	Yes	Yes

Table 3.1: Properties of the different datasets we acquired. We list the approximate number of component materials (i.e. basis BRDFs), the spatial resolution of the SVBRDF, the number of measurements and how these were distributed between camera and light-source positions, the combined size of the aligned and cropped images and an indication of which datasets contain anisotropic, isotropic and retro-reflective scattering properties along with the presence of variation in the normal and tangent directions.



Figure 3.8: Acquisition setup designed for measuring the reflectance properties of human skin [112].

acquire a single HDR image from each camera for each light activated in rapid succession. The entire acquisition process is on the order of a few seconds, as compared to a spherical gantry that would need several hours to obtain a comparable set of measurements.

The key drawback of these setups, however, is that the position of the cameras and light-sources are fixed. An interesting direction of future research would be providing a parallelizable and non-rigid acquisition device.

Chapter 4

Inverse Shade Trees

The ideal is for things to be intuitive and responsive, making it easier for people to express their intent. —Rob Cook

4.1 Introduction

The use of measured surface reflectance has the potential to bring new levels of photorealism to renderings of complex materials. Such datasets are becoming common, with recent work on acquiring dense measurements of both individual materials [70, 74] and spatially-dependent reflectance [21, 76, 64, 38, 72]. The availability of such data, however, has highlighted the difficulty of representing complex materials accurately using conventional analytic reflectance models [81]. Nonparametric representations provide greater accuracy and generality, but so far have not incorporated the important design goal of *editability*. That is, in order to be useful in a practical production pipeline, an appearance representation must let the designer change both the spatial and directional behavior of surface reflectance. We propose a compact tree-based representation (Figure 4.1) that provides the intuitive editability of parametric models while retaining the accuracy and flexibility of general linear decomposition methods.

As discussed in Section 2.5, the concept of composing a complex shading function from a tree-structured collection of simpler functions and masks was first introduced in the seminal “shade trees” work of Cook [18]. We develop an *Inverse Shade Tree* (IST) framework that takes as input

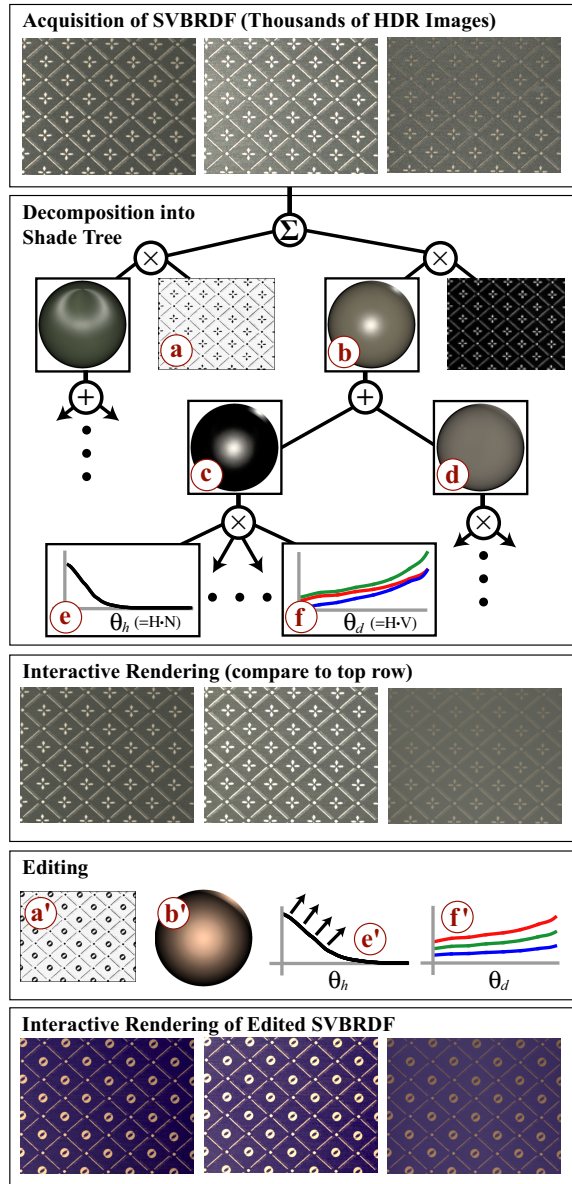


Figure 4.1: We introduce a non-parametric framework for decomposing measured SVBRDF data into a set of (a) spatially-varying blending weight maps and (b) basis BRDFs. The basis BRDFs are factored into sampled 2D functions corresponding to (c) specular and (d) diffuse components of reflectance (we show lit spheres rendered with these factors, not the 2D factors). These 2D functions are further decomposed into (e & f) 1D curves. In addition to providing accurate interactive rendering of the original SVBRDF, this representation also supports editing either (a') the spatial distribution of the component materials or (b') individual material properties. The latter is accomplished by editing (e' & f') the sampled curves.

a measured appearance dataset and a (user-supplied) tree structure, and fills in the leaves of the tree. In our trees, the leaves are sampled curves and maps (i.e., 1-D and 2-D) representing intuitive concepts such as specular highlight shape or texture maps. They are combined at interior nodes, of which the most common is a “mixing” node (other node types such as normal and tangent mapping are also supported). For example, in the first level of the tree in Figure 4.1, the Spatially-Varying Bidirectional Reflectance Distribution Function or SVBRDF is composed of a sum of products of spatial mixing weights (a) and basis BRDFs (b). The IST decomposition proceeds top-down, at each stage decomposing the current dataset according to the type of node encountered in the tree. Most notably, an “unmixing” algorithm is used to factor a high-dimensional dataset into a sum of products of lower-dimensional functions.

The editability of the resulting shade trees depends on having their leaves correspond to pieces that are meaningful to the user. For example, when decomposing an SVBRDF, we would like the resulting BRDFs to correspond to our intuitive notion of separate materials, instead of being arbitrary linear combinations. For this reason Section 4.4 focuses on the “unmixing” problem, showing how to map it to matrix factorization. We introduce flexible algorithms based on linear constrained least squares that are designed to produce intuitive decompositions. These algorithms can incorporate constraints such as non-negativity, and provide control over the sparsity in the decomposition (resulting in a continuous tradeoff between pure factorization and clustering). As compared to existing methods, we maintain accuracy while producing editable *parts-based* separations. When the original function is a SVBRDF, these “parts” correspond to different materials; when the function is a BRDF, these “parts” correspond to different scattering phenomena, such as diffuse reflection, specularity, or back-scattering. In addition, our algorithms incorporate domain-specific constraints such as energy conservation, and deal with practical issues such as large datasets and confidence weighting. We expect these techniques to be generally applicable to data dimensionality reduction applications, beyond the task of material representation addressed here.

We explore inverse shade trees in a prototype system that begins with densely measured spatially-varying reflectance (with raw dataset sizes of several gigabytes), and generates compact and intuitive trees. We demonstrate that the resulting trees permit real-time non-parametric editing (Section 4.7) of materials and their spatial distribution, and analyze the accuracy (Section 4.8) of both the material separation and BRDF decomposition stages.

4.2 Relationship to Previous Work

Parametric Models for Reflectance: Fitting analytic reflectance models to data has been a widely-adopted approach, and some models were in fact developed specifically for fitting to measurements [109, 56]. Thus, one possible representation of a measured SVBRDF is a collection of analytic BRDF parameters at each surface location [76, 31]. Such a representation provides for easy editing of materials, and with the addition of a clustering step [64] allows editing a single material everywhere it appears on a surface.

These approaches, however, have several key drawbacks. Reducing a dense set of measurements to a handful of parameters may introduce significant error [81]. Moreover, it requires non-linear optimization, which is computationally expensive and numerically unstable. Finally, clustering the values of the BRDF parameters [64] does not generate a desirable separation of the component materials in the presence of blending on the surface (even, in some cases, the trivial pixel-level blending present at antialiased material edges). This is both because the problem is underconstrained (i.e., there are many possible cluster allocations with comparable approximation error) and because the parameters of most BRDF models are not linearly related.

The work of Goldman et al. [34] is most similar to our own. They fit a convex combination of two analytic BRDFs (along with surface normals) at each surface location. This results in a sparse, non-negative representation, although their sparsity constraint is less general than the one we introduce in Section 4.5.2. Moreover, they use an isotropic Ward BRDF model which is less general than our data-driven approach.

In this section, we solve the *material separation* problem using the measurements directly, before fitting any secondary models to individual BRDFs. This allows for arbitrary blending of materials, giving correct results when parametric approaches fail (see section 4.8 for a comparison of accuracy). In addition, we use a non-parametric representation of BRDFs based on a small set of intuitive curves, providing both generality and greater accuracy for some classes of materials that exhibit anisotropy or retroreflection.

Nonparametric Models and Matrix Decomposition: Nonparametric approaches, including basis function decomposition [21] and standard matrix rank-reduction algorithms such as PCA, can retain high fidelity to the original data. In the context of appearance representation, researchers

have explored a variety of rank reduction algorithms, including variants of PCA [53, 30, 106], homomorphic factorization [77], ICA [102], k -means clustering [66], and NMF [16, 60].

Though our approach also falls in the category of dimensionality reduction, we build on prior work by performing a multi-level sequence of decompositions, rather than just a single one. More importantly, motivated by an evaluation of existing methods, we introduce a set of new matrix factorization algorithms (Section 4.4) specifically designed to provide *editable* decompositions, a criterion for which existing methods are not optimized.

Non-Parametric Material Editing: A data-driven approach to BRDF editing has been proposed by Matusik et al. [74], in which a user labels directions (e.g., “shininess”, “rustiness”, etc.) in a high-dimensional space of measured materials. The large number of materials, and the uncompressed representation of each BRDF, contribute to large storage requirements and inhibit interactive control. Though our BRDF editing system is also based on measured data, we provide for direct manipulation of curves controlling the reflectance instead of focusing on higher-level behaviors. In addition, our compressed representation and interactive renderer permit real-time manipulation of materials, including spatial variation.

To our knowledge, only two systems have demonstrated material editing via direct manipulation of low-dimensional factors of the BRDF [4, 48]. In these systems the user edits a 2D image, which is then used as a component in a parametric or homomorphic-factored BRDF model. We generalize this by allowing interactive control over all the 1D and 2D factors necessary to specify a full SVBRDF.

4.3 System Overview

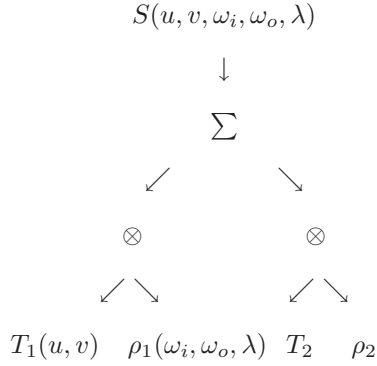
Although the shade tree framework could in principle represent many types of appearance data, including BTFs, light fields, and time-varying textures, this section focuses on SVBRDFs. We have also developed techniques for inferring shade trees of the BSSRDF of real-world materials [86] which we briefly discuss in Section 4.11. Here we provide a brief overview of our full pipeline, ranging from measurement of a SVBRDF through representation, rendering, and editing.

Appearance Acquisition: Chapter 3 provides a detailed description of our acquisition pipeline. Recall that we used a spherical gantry with computer control over camera and light source direction [72] to record a set of high-dynamic-range images of planar samples with spatially-varying material under many illumination and view directions. The choice to measure mostly-planar samples simplified the calibration and registration, but the techniques presented here apply to arbitrary geometry. We acquired five different SVBRDFs that include layered materials, anisotropy, retro-reflection, and spatially-varying normal and tangent directions. The complete raw and processed data, together with a description of acquisition and calibration procedures, are available on a web server (URL to be announced).

After geometric and photometric calibration, the images are reprojected onto the best-fit plane of the surfaces, yielding a uniform spatial sampling (at approximately 500×500 points) of reflectance measurements. Because the SVBRDF is a function of six variables (i.e., it can be written $S(u, v, \omega_i, \omega_o, \lambda)$, where λ is discretized into RGB or HSV bands), sampling the illumination and view directions uniformly and densely is impractical. Therefore, we sampled the forward- and backward-scattering lobes of the reflectance more densely than the other regions of the domain, yielding a total of between 2000 and 6000 reflectance measurements for each point on the surface. The scattered data are conceptually resampled onto a uniform grid using a push-pull algorithm [36] with a Gaussian reconstruction kernel (as we shall see later, we avoid a complete reconstruction by using a subsampling method).

Decomposition: We produce a shade tree with a series of decompositions of the SVBRDF and, using the same algorithms, the component BRDFs. For example, consider Figure 4.1(top), which shows a few images from a dense set of measurements of the SVBRDF of an anisotropic wallpaper sample. The first level of our decomposition separates the data into 4D functions that depend on directions of incidence and reflection (“basis BRDFs,” shown in Figure 4.1b as lit spheres) and 2D functions of spatial position (“blending weights,” shown in Figure 4.1a). We represent the decomposition with this tree diagram:

Tree 1

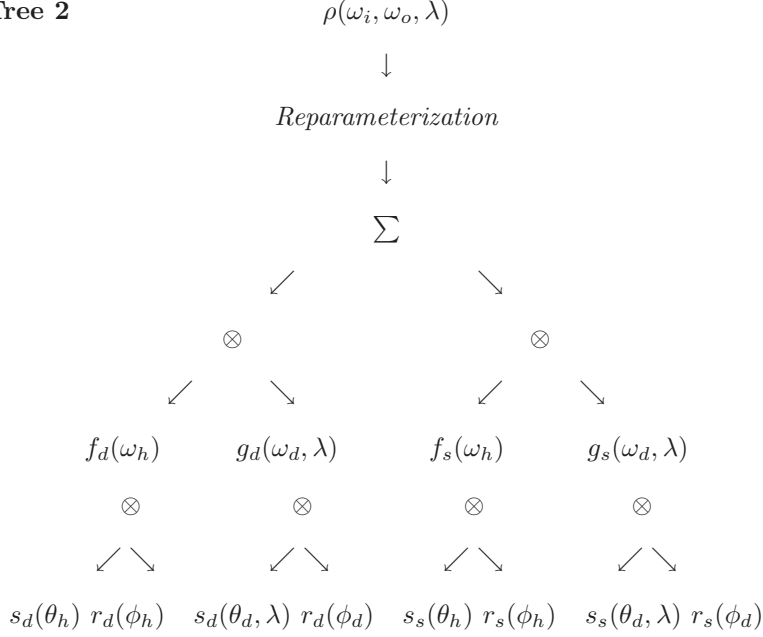


Note that we have chosen to associate color (i.e., λ) with the BRDFs. However, if it were more convenient for later editing, we could have associated color with the spatial blending weights instead, resulting in a *color* texture and *colorless* basis BRDFs.

Although we have reduced the size of the original SVBRDF, these basis BRDFs are still tabular representations of a 4D function, making them unsuitable for interactive rendering or editing. To address this, we further reduce the basis materials through a series of decompositions into 2D functions and eventually into 1D curves. Although we introduce these decompositions in the context of representing the materials contained within a SVBRDF, they provide, in general, an editable non-parametric representation of the BRDF.

For the example in Figure 4.1, each basis BRDF is decomposed into two terms, each a product of 2D functions of half (ω_h)- and difference (ω_d) angles [94]. For the shiny gold material, one term corresponds roughly to the specular component of the reflectance (Figure 4.1c), while the other represents near-Lambertian diffuse (Figure 4.1d). For improved editability, we may further factor the 2D functions into products of 1D curves defined on the corresponding elevation and azimuthal angles (Figures 4.1e and f):

Tree 2



While the decomposition into 1D curves results in a simpler representation and is desirable for isotropic BRDFs (which are invariant to ϕ_h), we have found that it may reduce accuracy for some complex anisotropic materials. In these cases, we terminate the decomposition one level higher, resulting in 2D maps resembling those of Ashikhmin et al. [4] and Ngan et al. [81].

Rendering and Editing: The measured materials may now be rendered by re-composing the shade trees in a pixel shader; this provides interactive feedback during editing. In most cases, the 1D functions or curves at the leaves of the tree correspond naturally to physical phenomena controlled by parameters in existing analytic models. For example, the $s_s(\theta_h)$ curve is related to the distribution of microfacets on the surface, and hence determines the shape of the specular highlight. The $s_s(\theta_d, \lambda)$ curve describes the behavior of the specular lobe as the view direction moves from normal incidence to grazing, capturing Fresnel effects such as color shifts, increased specular reflection, and a reduced diffuse term. Color variation, such as in $s_s(\theta_d, \lambda)$, can be represented with separate curves for each RGB color component (Figure 4.1f), or in an alternate colorspace such as HSV.

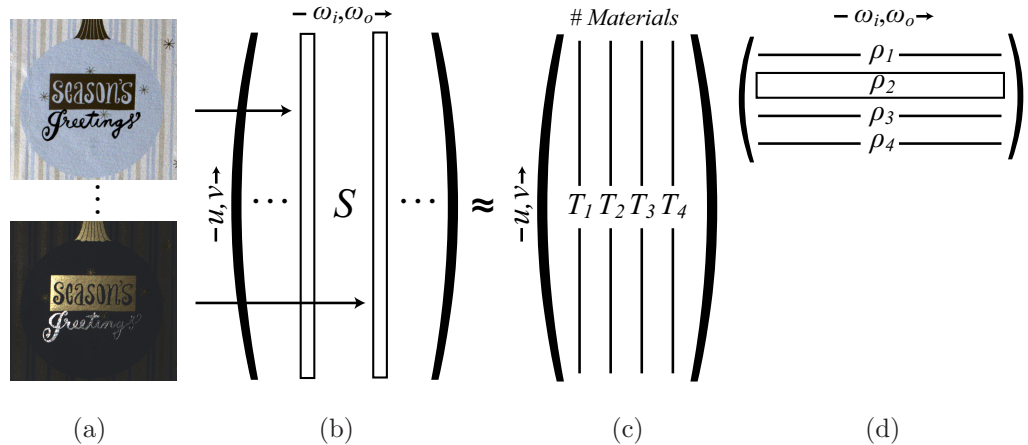


Figure 4.2: We cast the decomposition depicted in Tree 1 as the factorization of a matrix. (a) High-dynamic range images of a SVBRDF captured with a spherical gantry are (b) organized into a matrix with rows that vary along spatial position and columns that vary along incident and outgoing angles. This matrix is factored into the outer product of (c) functions of spatial position (“blending weights”) and (d) functions of incident and reflected directions (i.e., “basis BRDFs” in tabular form). In this example, we factor the SVBRDF of a holiday greeting card into four terms.

4.4 Algorithms for Matrix Factorization

We can cast the tree-structured decompositions described previously as a sequence of matrix factorizations. At the top-level (Tree 1), we organize samples of a SVBRDF into a matrix that is factored into the outer product of 2D blending weights and 4D basis BRDFs, as illustrated in Figure 4.2. At the second-level (Tree 2), we decompose each basis BRDF into appropriate 2D factors by computing another matrix factorization, as shown in Figure 4.3.

There are a variety of matrix algorithms available for these factorizations. In Section 4.4.1, we compare existing approaches and discuss the conditions under which they fail to provide a meaningful decomposition. In Section 4.5, we introduce a new factorization algorithm based on linearly constrained optimization that improves the separation in challenging cases. One key benefit of this new algorithm is that it can incorporate domain-specific constraints for decomposing appearance data (Section 4.5.3). Lastly, we address practical considerations related to scattered input data and matrices whose sizes exceed the capacity of main memory (Section 4.5.6).

We will adopt the notation of [62] to discuss matrix factorizations. Specifically, a $n \times m$ input matrix V is approximated as the product of a $n \times k$ matrix W and a $k \times m$ matrix H ($V \approx WH$). Rank reduction occurs when k is smaller than n and m , and we are most interested in cases of

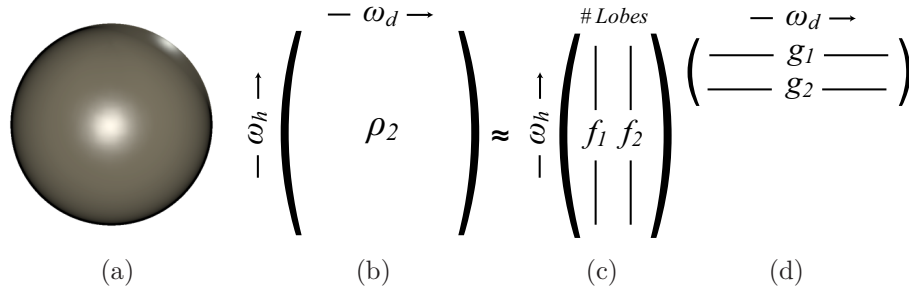


Figure 4.3: We cast the BRDF decomposition in Tree 2 to matrix factorization. Each (a) tabular BRDF (shown as a lit sphere) is (b) first re-parameterized and then rasterized into a matrix with rows that vary along the half-angle and columns that vary with difference-angle. We factor this matrix into (c) functions of the half-angle and (d) functions of the difference angle.

extreme compression (e.g., n and m are hundreds or thousands, while k is between 1 and 5). We consider factorizations that minimize the Euclidean error,

$$\|V - WH\|^2 = \sum_{ij} (V_{ij} - (WH)_{ij})^2. \quad (4.1)$$

To begin, we focus on SVBRDF decomposition, so that the original data is in matrix V and the mixing weights and basis BRDFs end up in W and H , respectively. Later, we will consider BRDFs, and W and H hold sampled half-angle and difference-angle maps.

4.4.1 Evaluation of Existing Algorithms

We compare several classes of factorization algorithms suitable for accurately representing measured appearance data, evaluating their performance in representing the *Season’s Greetings* dataset shown in Figure 4.4. This measured SVBRDF is of a holiday greeting card with four materials (blue and white paper, and gold and silver foil), as shown in the ideal separation at bottom. Note that the materials are smoothly blended over the surface. For example, the gold foil is present in different amounts at the boundary between the gold border around the word “Season’s” and the paper background (Figure 4.4, T_2 and T_3). Also, the stripes in the background were created by halftoning the gold material over the paper background. This is visible as spatial blending between materials T_2 and T_4 .

To represent these effects, while providing an editable decomposition, we have identified three key properties of a factorization algorithm. First, it should allow for a basis consisting of linear

Original Images



Ideal Decomposition



Figure 4.4: Two images (originals are HDR) from the “Season’s Greetings” dataset, together with hand-generated mixing masks that would be produced by an ideal decomposition. Notice that the separation is **soft**, with significant blending between the gold foil and both the blue and white paper. Since the blending weights in the bottom row are colorless scalars (the color for this shade tree is in the BRDFs), we use grayscale images to visualize them.

combinations of the input to resolve the blending of different materials. Second, the algorithm should guarantee non-negativity to produce a physically plausible result and favor parts-based decompositions. Third, the algorithm should provide control over the sparsity of the solution, favoring a representation that uses individual materials, where possible, to represent the SVBRDF (as opposed to blending between materials across the *entire* surface). Table 4.1 summarizes these properties for different algorithm classes.

PCA/ICA: Two popular rank reduction algorithms are Principal Component Analysis (PCA) and Independent Component Analysis (ICA), along with extensions such as multilinear tensor factorization [106]. The main advantage of PCA is that it yields a *global* minimum of Equation 4.1.

Algorithm Groups	Properties:		
	Linear	Positive	Sparse
SVD / ICA	Yes	No	No
Homomorphic	No	Yes	No
Clustering	No	Yes	Yes
NMF / pLSI	Yes	Yes	No
Our Method: ACLS	Yes	Yes	Yes

Table 4.1: Comparison of matrix factorization algorithms. Existing methods do not satisfy the three crucial properties of linearity, positivity, and control over sparsity, which are critical for a meaningful editable decomposition.

Although these algorithms are guaranteed to recover a basis that is orthonormal (for PCA) or statistically independent (for ICA), these restrictions are not sufficient to produce a meaningful description of the data. In particular, both algorithms allow negative values in W and H , resulting in a representation whose terms cannot be edited independently (Figure 4.5, top).

Homomorphic Factorization: Introduced in the context of representing non-parametric BRDFs, Homomorphic Factorization [77], decomposes a high-dimensional function into a single *product* of an arbitrary number of lower dimensional functions. Although it can support an arbitrary number of factors, it does not allow linear combinations. Hence, this algorithm is not appropriate for representing the SVBRDF as a *sum of products* of basis materials and spatial blending weights, or decomposing a BRDF into a *sum* of diffuse, retroreflective and specular lobes.

Clustering: One popular method for clustering data is the k -means algorithm [40]. Like all clustering algorithms, k -means partitions the input into disjoint sets, associating each point with a representative point called the *cluster center*. This can be interpreted as a factorization of the SVBRDF, where the cluster centers are stored in the matrix H and W is computed by re-projecting the data onto this basis (using gradient descent for example). In our experiments, clustering performs well on input with a small basis that is well-separated over the surface. However, when the SVBRDF exhibits blending of its component materials, clustering typically fails to recover a useful basis. For example, in Figure 4.5, middle, k -means has incorrectly assigned a single cluster to the *combination* of the gold foil and white paper (T_2) while grouping the gold and silver foils into a separate cluster (T_1).

Principal Component Analysis (RMS=0.016)



k -means with Re-projection (RMS=0.084)



Non-Negative Matrix Factorization (RMS=0.017)



Figure 4.5: Blending weights computed from the “Season’s Greetings” dataset using the factorization algorithms discussed in Section 4.4. For PCA, these terms are visualized as images where red and green correspond to positive and negative values with luminance proportional to the magnitude. For k -means and NMF, all values are non-negative and are visualized as grayscale images. Note that neither PCA nor NMF provide a separation of the data into distinct *parts* suitable for editing. Although clustering performs better, it too fails to recover the desirable separation into the four component materials present in this sample (Figure 4.4, bottom row). In particular, k -means assigns both the gold and silver foil to a single cluster (T_1) and combines the gold foil and white paper into a separate term (T_2).

Non-Negative Matrix Factorization: Another matrix decomposition approach is Non-Negative Matrix Factorization (NMF), introduced by Lee and Seung [62]. Together with similar algorithms such as Probabilistic Latent Semantic Indexing [46], NMF guarantees that both resulting factors contain only non-negative values. One motivation for this constraint is to encourage the algorithm to describe the input data as the sum of positive *parts*, thereby producing a more meaningful

factorization. In our experiments, however, the character of the decomposition is sensitive to small changes in the data (including those due to measurement noise and misalignment), and the non-negativity constraint is not always enough to guarantee an editable separation (see Figure 4.5, bottom).

4.5 Our Method: Alternating Constrained Least Squares

We have seen that existing matrix factorization methods do not fulfill the three properties (linearity, positivity, sparsity) needed to produce meaningful, editable decompositions. We now describe a new suite of factorization algorithms that allow for all of these, while being flexible enough to support additional domain-specific constraints such as energy conservation in the SVBRDF and monotonicity of the BRDF (Section 4.5.3).

Our algorithm is built upon efficient numerical methods for solving linear constrained least squares (LCLS) problems of the form:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \frac{1}{2} \|b - Mx\|^2 \quad \text{subject to} \quad l \leq \begin{Bmatrix} x \\ Ax \end{Bmatrix} \leq u \quad (4.2)$$

The n -element vector x is called the *vector of unknowns*, M is called the *least-squares matrix* and b is the *vector of observations*. The vectors u and l provide the upper and lower *bound constraints* of both x and the linear combinations encoded in the matrix A , called the *general constraints*. There are numerous algorithms available for efficiently solving these types of convex quadratic programming problems. We use an implementation of an inertia-controlling method that maintains a Cholesky factorization of the reduced Hessian of the objective function [32]. We use the implementation of this algorithm in the Numerical Algorithms Group (NAG) library set, called `nag-opt-lin-lsq` [78].

4.5.1 Non-Negative Factorization

As with NMF, we initialize W and H to contain positive random values, and minimize Equation 4.1 by alternately updating these two matrices. This problem is known to be convex in either W or H separately, but not simultaneously in both. As a consequence, we will present an algorithm that

finds a *local* minimum of Equation 4.1.

Without loss of generality, we consider the case where both V and W are row vectors ($v \approx wH$). We later extend the discussion to consider the entire matrix W . For a fixed H , we update our current estimate of w by minimizing Equation 4.1, subject to the linear constraint $w \geq 0$. To accomplish this, we solve the LCLS problem in Equation 4.2, with $M = H^T$, $b = v^T$, and $x = w^T$. To constrain the solution to be non-negative, we set $l = 0$ and $u = \infty$.

We update the entire matrix W by computing the above solution for each of its rows in turn. Similarly, we can transpose the problem, take W to be the *least-squares matrix* M , and update our estimate of H one column at a time. By alternating between estimating W and H , we achieve a non-negative factorization of the input matrix V . Because we are guaranteed never to increase Equation 4.1 after either update, this algorithm, which we call Alternating Constrained Least Squares (ACLS), is guaranteed to converge to a local minimum.

Compared to NMF, for which each iteration (i.e. one update of both W and H) costs only a few matrix multiplications, each iteration of ACLS is considerably more expensive. On the other hand, each iteration of ACLS results in a more significant decrease in error, leading to convergence with an order of magnitude fewer iterations. In our experiments, we have found the overall computation time for these two algorithms to be comparable.

4.5.2 Sparsity

A non-negativity constraint is frequently not enough to provide an intuitive *parts-based* decomposition. We introduce a modification that considers the *sparsity* of the solution, providing a continuous tradeoff between non-negative matrix factorization and clustering. We have found this flexibility to be effective for decomposing SVBRDFs exhibiting complex blending of multiple materials.

In order to define sparsity, consider the SVBRDF factorization shown in Figure 4.2. A sparse decomposition is one in which there is a linear combination of relatively few basis materials at each surface location. That is, each *row* of W has few non-zero entries. Although there are several expressions that quantify this notion, we require one that leads to a linear least-squares problem: it must be quadratic in the elements of the row. Therefore, we define the *sparsity penalty* for a row w as the sum of the squares of all but one of the coordinates of w (i.e., $\sum_{i \neq j} w_i^2$, where the

selection of j is discussed below). For a fixed H , we can combine this sparsity penalty, weighted by a parameter λ , with the approximation error (4.1), which gives a new error to be minimized:

$$\|v - wH\|^2 + \lambda \sum_{i \neq j} w_i^2. \quad (4.3)$$

One potential problem with this formulation is that we can decrease the overall error simply by decreasing the magnitude of *all* the elements of w . This is particularly true for large values of λ . To address this, we introduce an additional soft constraint that the L_1 norm of w should be *close* to unity. As before, we can add this penalty, weighted by the parameter μ , to the error:

$$\|v - wH\|^2 + \lambda \sum_{i \neq j} w_i^2 + \mu \left(1 - \sum_i w_i\right)^2. \quad (4.4)$$

Starting from Equation 4.4, we can write out the corresponding least-squares matrix and observation vector to be used in Equation 4.2:

$$b = \begin{pmatrix} v^T \\ \sqrt{\mu} \\ 0 \end{pmatrix} \quad M = \begin{pmatrix} \text{---} & H^T & \text{---} \\ \text{---} & \sqrt{\mu} \dots \sqrt{\mu} & \text{---} \\ \sqrt{\lambda} \dots \sqrt{\lambda} & 0 & \sqrt{\lambda} \dots \sqrt{\lambda} \end{pmatrix}. \quad (4.5)$$

The 0 in the bottom row of M will be at the j^{th} position.

Putting things together, we estimate w by iterating over the possible values of j (in practice this corresponds to the rank of the approximation and is small) and minimizing Equation 4.4; we retain the w that corresponds to the selection of j with the smallest error. The entire matrix W is estimated one row at a time in this fashion. We alternate between updating our estimate of W and H until the error converges to a local minima. Assuming that both W and H are initially within the feasible region, each iteration cannot increase Equation 4.4, so this algorithm, which we call Sparse Alternating Constrained Least Squares (SACLS), is guaranteed to converge. This is a critical property, and one not shared by some previous approaches to sparse factorization (such as



Figure 4.6: The blending weights computed from the “Season’s Greetings” dataset using SACLS with different settings of λ and μ . Increasing values of λ force the algorithm to behave more like clustering, trading reconstruction accuracy for a more meaningful separation. RMS errors refer directly to the quality of reconstruction, without taking into account the additional error terms for sparsity or other constraints.

that of Hoyer [47]), which include a normalization step that can increase error.

The parameter λ influences the sparsity of the factorization, ranging from pure matrix factorization ($\lambda = 0$) to pure clustering ($\lambda \rightarrow \infty$). The parameter μ , in contrast, determines the extent to which we insist that the sum of material contributions at each location is 1. We have found the algorithm to be somewhat less sensitive to the selection of this parameter. As with previous work in low entropy coding [83], we define both λ and μ in units of the variance in V . This provides more intuitive control over these values, but trial and error is still required to determine their appropriate settings. Figure 4.6 illustrates the impact of different settings of λ on the decomposition.

4.5.3 Domain-Specific Constraints

One advantage of the ACLS algorithm is that it can be easily extended to include additional linear constraints beyond non-negativity and sparsity. In this section, we introduce several useful constraints in the context of representing the SVBRDF and BRDF, including energy conservation and monotonicity.

4.5.4 SVBRDF Constraints: Energy Conservation

When factoring the SVBRDF, we can extend ACLS to guarantee that the basis BRDFs conserve energy. For convenience, suppose that H contains values of the BRDF for different light positions and a *single* viewing direction (these techniques can readily be extended to multiple viewing directions). In this simplified case, we can constrain the BRDF at the j^{th} row of H to conserve energy by bounding the sum of its values, each weighted by the solid angle:

$$\sum_i H_{ji} \Delta\omega_i \leq 1 \quad (4.6)$$

This constraint is incorporated into the ACLS framework by first linearizing the matrices V and H into column vectors $\tilde{v} = (V_{11} \ V_{12} \ \dots \ V_{mn})^T$ and $\tilde{h} = (H_{11} \ H_{12} \ \dots \ H_{km})^T$. From Equation 4.2, we set $b = \tilde{v}$, $x = \tilde{h}$, and define M and A as follows:

$$M = \begin{pmatrix} w_{11} & 0 & \dots & 0 & w_{12} & 0 & \dots & 0 \\ & & & \cdot & & & & \\ & & & \cdot & & & & \\ & & & \cdot & & & & \\ & & & \cdot & & & & \\ 0 & \dots & 0 & w_{m1} & 0 & \dots & w_{mk} & \end{pmatrix} \quad A = \begin{pmatrix} \Delta\omega_1 & \dots & \Delta\omega_m & 0 & \dots & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ 0 & \dots & 0 & \Delta\omega_1 & \dots & \Delta\omega_m \end{pmatrix} \quad (4.7)$$

Finally, we set the boundary constraints (i.e., l and u) to guarantee that H is non-negative and the sums encoded in the matrix A lie between 0 and 1. By solving Equation 4.2 under these substitutions, we guarantee that the BRDFs encoded in H conserve energy. With the added constraint on the parameterization of the BRDF that $\phi_d + \pi \rightarrow \phi_d$, we can also guarantee reciprocity. To our knowledge, this is the first factorization algorithm that guarantees an SVBRDF decomposition into *physically valid* non-parametric BRDFs.

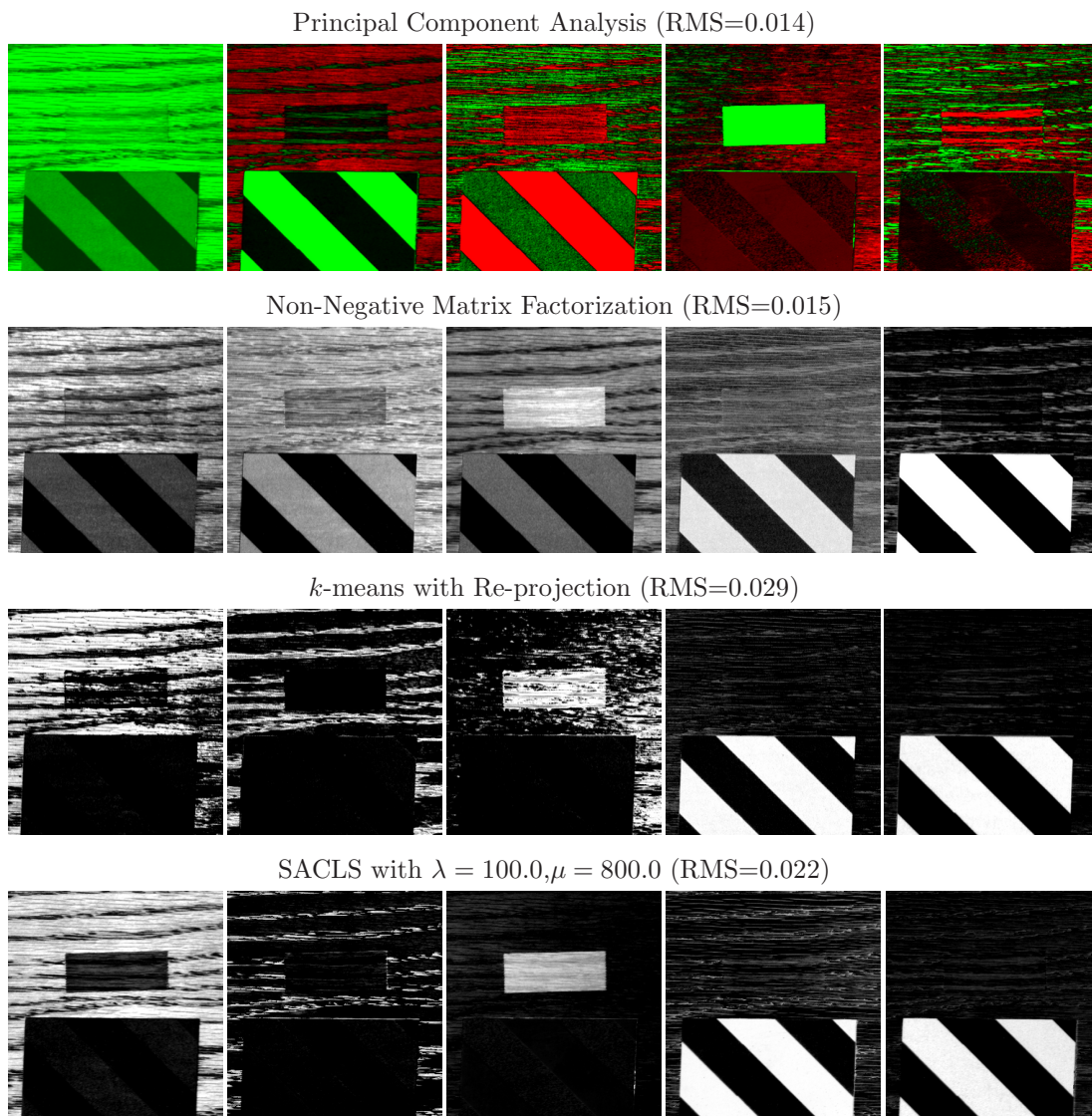


Figure 4.7: Visual comparison of the spatial blending weights computed by several linear factorization algorithms on the “Wood+Tape” dataset. Our method (bottom row) provides control over sparsity and guarantees the component BRDFs are physically valid (energy conserving and reciprocal). This aids in providing automatic separation of the measured data into its component materials and provides a final representation that can be edited (Figure 4.14).

Reciprocity and energy conservation constraints were used to perform material separation on all the samples considered here. Figure 4.7 shows our separation for the *Wood+Tape* dataset. This particularly challenging SVBRDF consists of a piece of oak partially covered by semi-transparent tape and retro-reflective tape. Note that the tape completely disappears at certain incident and

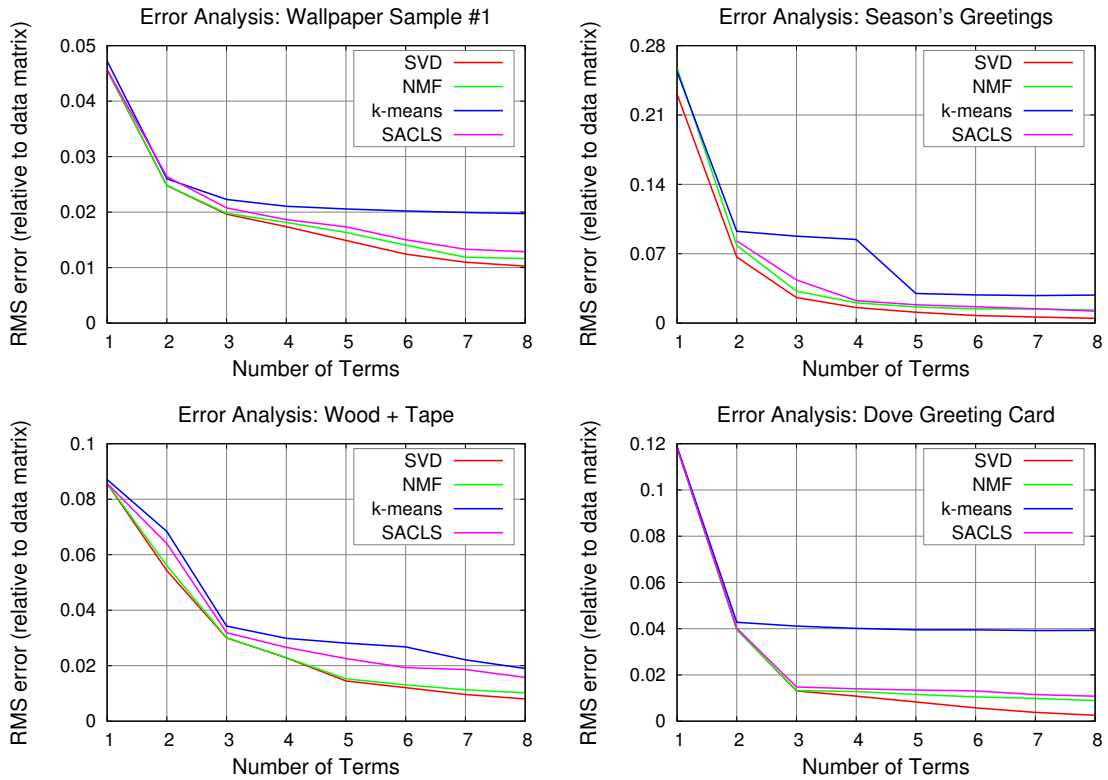


Figure 4.8: Accuracy of representing four different SVBRDFs with four possible linear decomposition algorithms. For each dataset, the SACLs algorithm provides a representation with comparable numerical accuracy to existing data-driven approaches.

reflected directions (Figure 4.14, left column). On this data, PCA and NMF produce decompositions with significant mixing, while clustering improperly groups regions of the wood grain with the tape (Figure 4.7). On the other hand, SACLs correctly separates the SVBRDF into two different types of wood grain, a tape layer smoothly blended over the wood, and two separate terms for the retroreflective materials. We have observed similarly intuitive material separation results for all the datasets. Moreover, this intuitive separation comes at little or no decrease in the numerical accuracy. Figure 4.8 shows the cosine-weighted RMS error (defined as the square root of the sum of squared differences between the original images and the reconstruction, weighted by $\cos(\theta_i)$) produced by four decomposition algorithms, for a range of different terms (number of materials).

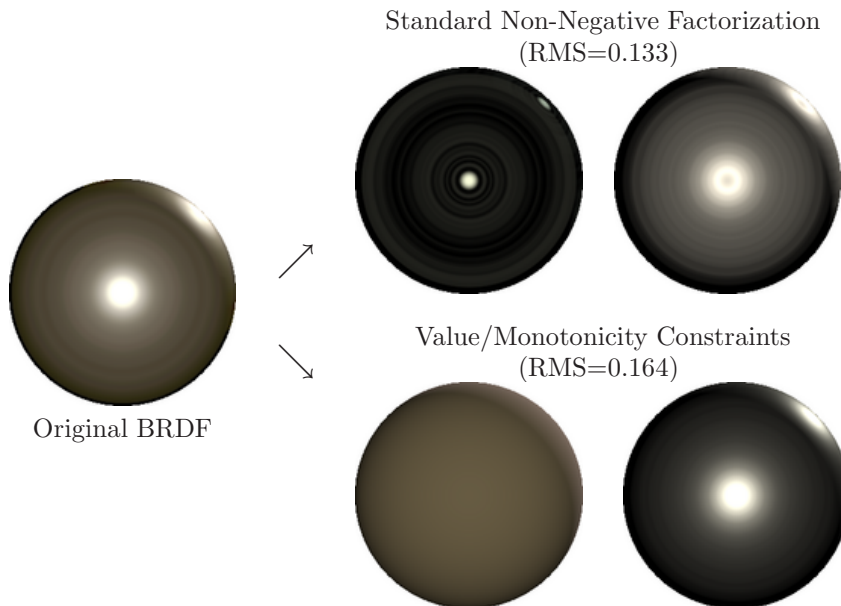


Figure 4.9: The ACLS algorithm can be extended to incorporate value and monotonicity constraints. We factor a tabular BRDF (left) into the sum of two terms. At top, we use basic non-negative factorization (Section 4.5). At bottom, the two terms are computed by constraining one term to have uniform θ_h dependence while the other is monotonically decreasing in θ_h .

4.5.5 BRDF Constraints: Value and Monotonicity

At the second level in our tree-structured decomposition (Tree 2), we factor a tabular BRDF into the sum of terms, each a product of functions of half- and difference-angle. As with the SVBRDF, this is equivalent to factoring a matrix (Figure 4.3).

Factoring the BRDF into multiple 2D terms using standard non-negative factorization (Section 4.5) generally yields factors that are arbitrary linear combinations whose values should not be independently edited. To address this, we allow for two types of constraints on these factors. First, we can constrain one of the half-angle terms in Figure 4.3c to remain at a constant value while allowing the rest of the factorization to update normally. This has the effect of separating the BRDF into a lobe with uniform θ_h dependence (typically diffuse-like terms, though not restricted to be perfectly Lambertian) plus a lobe with arbitrary half-angle distribution (usually a specular lobe). In all cases, the dependence on the difference angle is retained, allowing for Fresnel effects such as color shifts, increased specular reflection, and a reduced diffuse term.

Additionally, we can constrain the half-angle dependence of the “specular” term to be monotonically decreasing in θ_h , resulting in more physically plausible highlights. We constrain the

derivative of f_i to be negative at each sample. Because this is a linear operator (e.g., central differences), it can be directly encoded into the matrix A along with the settings $l = -\infty$ and $u = 0$ in Equation 4.2. Figure 4.9 provides an example of using both value and monotonicity constraints for the shiny gold foil ρ_2 from Figures 4.2 and 4.3.

4.5.6 Practical Considerations

To make these factorization algorithms practical, there are several issues we must consider. First, for most real-world data, the level of confidence of each measurement is not uniform across the input matrix V . For example, reflectance measurements from grazing angles will be less accurate than those from perpendicular angles. Additionally, some regions of the domain are not measured, producing “holes” in the input. We would like to associate a “confidence” with each value in the matrix in order to allow scattered data interpolation across missing regions in the input. Second, because of the high dimensionality and resolution of our datasets, the sizes of the matrices we factor often exceed the capacity of main memory. Finally, to help avoid incorrect local minima we initialize ACLS from multiple starting positions.

4.5.7 Missing Data and Confidence Weighting

We incorporate a confidence matrix C into our objective function to weight the contribution of each element in V towards the error:

$$\sum_{ij} \left(C_{ij} (V_{ij} - (WH)_{ij}) \right)^2. \quad (4.8)$$

An element with a confidence of 0 will have no effect on the factorization, seamlessly allowing for missing data. We can incorporate C into any ACLS variant through a simple modification to the least-squares matrix M and the observation vector b in Equation 4.2. For convenience, consider estimating a single row in W (denoted w) for a fixed H according to the corresponding rows in V and C (denoted v and c respectively). The related linear constrained problem from Equation 4.2 will have $b_j = c_j v_j$ and $M_{jk} = c_j H_{kj}$. Note that this reduces to standard ACLS for $c_j = 1$. Figure 4.10 shows the performance of confidence-weighted ACLS on a controlled example, where 50% of samples of a BRDF are removed.

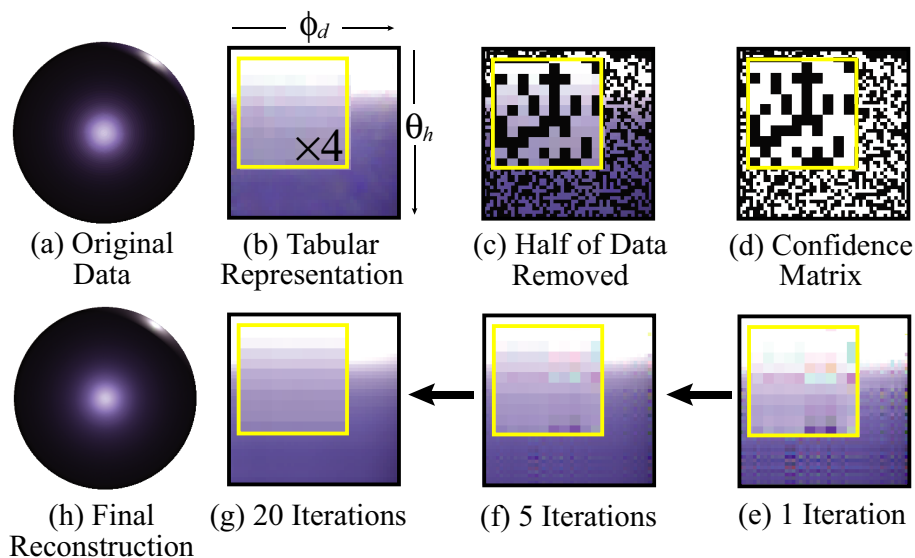


Figure 4.10: Using a measured BRDF as input (a) we construct a data matrix indexed by θ_h and ϕ_d shown in (b). (Note: we show only one section of the complete data matrix by omitting variation in θ_d). For this test, we removed 50% of the data values to produce the matrix in (c) and compute a confidence matrix (d) where measured values have a confidence of 1 and missing values have a confidence of 0. We show the factorization computed by confidence-weighted ACLS after one (e) and five (f) iterations. After 20 iterations (g), we produce a matrix that approximates the original.

In practice, we compute the confidence matrix by setting unmeasured regions in the input matrix to zero and assigning a relative confidence to each view according to its observed calibration error.

4.5.8 Subsampling for Large Datasets

Due to the high-dimensionality of the datasets we are interested in factoring, V often exceeds main memory. However, its rank is significantly smaller than the resolution of our measurements (i.e., $k \ll m$). We exploit this rank-deficiency by computing a basis for a subset of the complete data (call this V'). We use whichever variant of ACLS is appropriate to compute: $V' \approx W'H$. Because we discarded only *complete rows* of V , the matrix H can be thought of as an estimate of its basis. The original data is projected onto H using ACLS to estimate W while holding H fixed. This procedure requires storing only one row of V , one row of W and the complete H matrix in main memory at any given time. We can similarly reduce V by computing a factorization over a subset of the columns.

This strategy converges at least as quickly as standard ACLS even for aggressive downsampling. If the sample size is too small, however, the basis of V' will not accurately represent V , and the error will increase. For the datasets we consider, this situation only arises when using less than 0.01% of the original matrix.

In practice, we reconstruct the SVBRDF datasets containing isotropic materials at an angular resolution of $100 \times 30 \times 15$ ($\theta_h \times \theta_d \times \phi_d$), while representing anisotropic datasets with a 64×64 parabolic map [44] for the half-angle term at 30×15 different positions of θ_d and ϕ_d respectively. The spatial resolution of the samples are approximately 500^2 . If we were to represent the SVBRDF matrix explicitly, this would require 125GB of memory for the isotropic case, and 5,149GB for anisotropic samples. Instead, we rely on subsampling: we compute blending weights using 50 columns of the original matrix at qualitatively different positions (i.e., specular highlights, back-scattering, perpendicular and grazing angles), and reconstruct full resolution tabular BRDFs at 100 randomly selected positions.

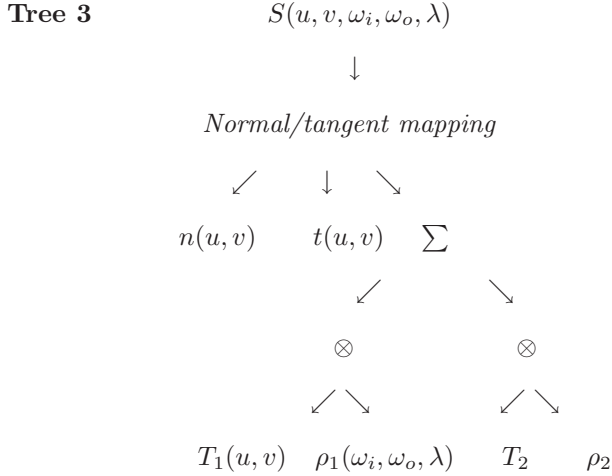
4.5.9 Initialization of ACLS

While the simplest strategy for all ACLS variants is to initialize the matrices W and H with random positive values (subject, of course, to any additional desired constraints such as monotonicity), the fact that ACLS performs local minimization leaves it susceptible to local minima. We have found that a more robust strategy is to first run k -means clustering with a relatively large k (for example, 20), then initialize ACLS with a random subset of the cluster centers. For even greater robustness, we repeat the ACLS minimization with many randomly-chosen subsets of cluster centers, and take as our final result the one with smallest error. In our experiments, this strategy is robust in avoiding incorrect local minima, and ameliorates the undesirable lack of provable global convergence (shared by all algorithms considered here, except PCA).

4.6 Normal and Tangent Estimation

For materials containing normal variation or (for anisotropic materials) variation in tangent direction, we can augment our SVBRDF decomposition tree with the addition of normal and/or tangent maps. If both are present, we are effectively estimating a full rotated coordinate system at

each spatial location, thus capturing effects similar to those recently demonstrated by Marschner et al. [72]. All these effects are captured with the following tree, which we use instead of Tree 1:



We estimate normal and tangent directions at each spatial position in three stages. First, we fit a generic BRDF with an anisotropic Gaussian specular lobe (i.e., a Ward BRDF) at each location, with the rotation angles defining the coordinate system as free parameters to the fit. Using these initial orientation estimates, we build the matrix described in the previous section and compute its k -term factorization. We then refine our estimates using this factorization, again solving for the best-fit normal and tangent. We can repeat these steps until the overall error converges, though in practice we found that two iterations are sufficient to accurately recover the fine geometric surface detail present in our samples. We show the final normal maps and tangent maps for two datasets in Figure 4.11.

4.7 Results: Editing

The benefit of obtaining a decomposition of the SVBRDF into a meaningful shade tree is that any leaf node may be independently edited. In this section, we describe several possible edits at both the material level (altering the spatial texture of which component material appears where) and at the BRDF level (changing salient aspects of a material’s reflectance using our curve-based model). The supplementary video shows further real-time editing results. While most of these edits are straightforward given our intuitive shade tree representation, they are to our knowledge the first demonstration of non-parametric editing of spatially-varying measured materials, and would not be

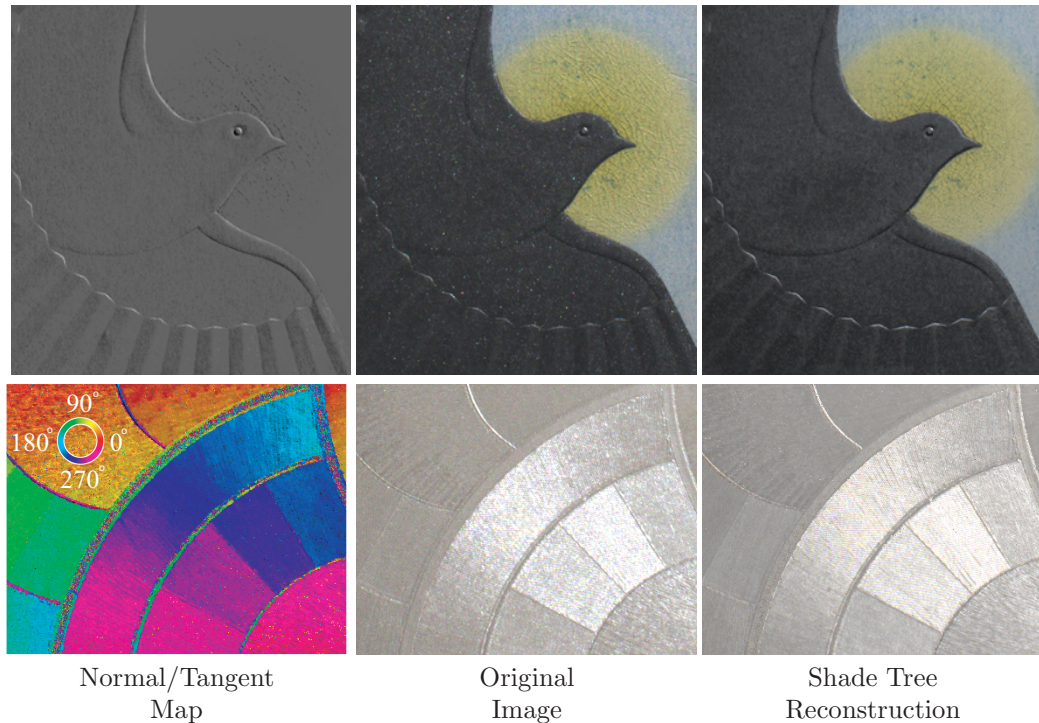


Figure 4.11: Normal and tangent maps. **Left:** We show $N \cdot L$ for the Dove normal map and a near grazing light direction. For tangent maps, we set the hue of each pixel to be the tangent direction. Note that this direction is undefined in regions with isotropic reflectance. **Middle:** original images. **Right:** images rendered using a 3-term shade tree with normal and tangent maps (note: blending weights not shown).

easy with alternative matrix factorization methods, which do not provide a meaningful separation of materials or individual BRDFs.

4.7.1 SVBRDF Editing

Changing Spatial Distribution of Materials: Perhaps the most obvious edit is to change the spatial distribution or “texture” of the basis materials. In Figures 4.1 and 4.13 we have changed the texture by re-painting the blending weight maps. To achieve the edit shown in Figure 4.14, we first define a transparency mask for the tape as the product of its blending weights and a user-set constant. Our resulting shade tree composites the tape *over* the additional layers using this mask. Because our separation was not perfect for this challenging case, we manually repaired some error in the wood blending weights below the tape. We can also interactively re-position the tape (shown in the supplemental video) or remove it altogether. This edit would not be possible

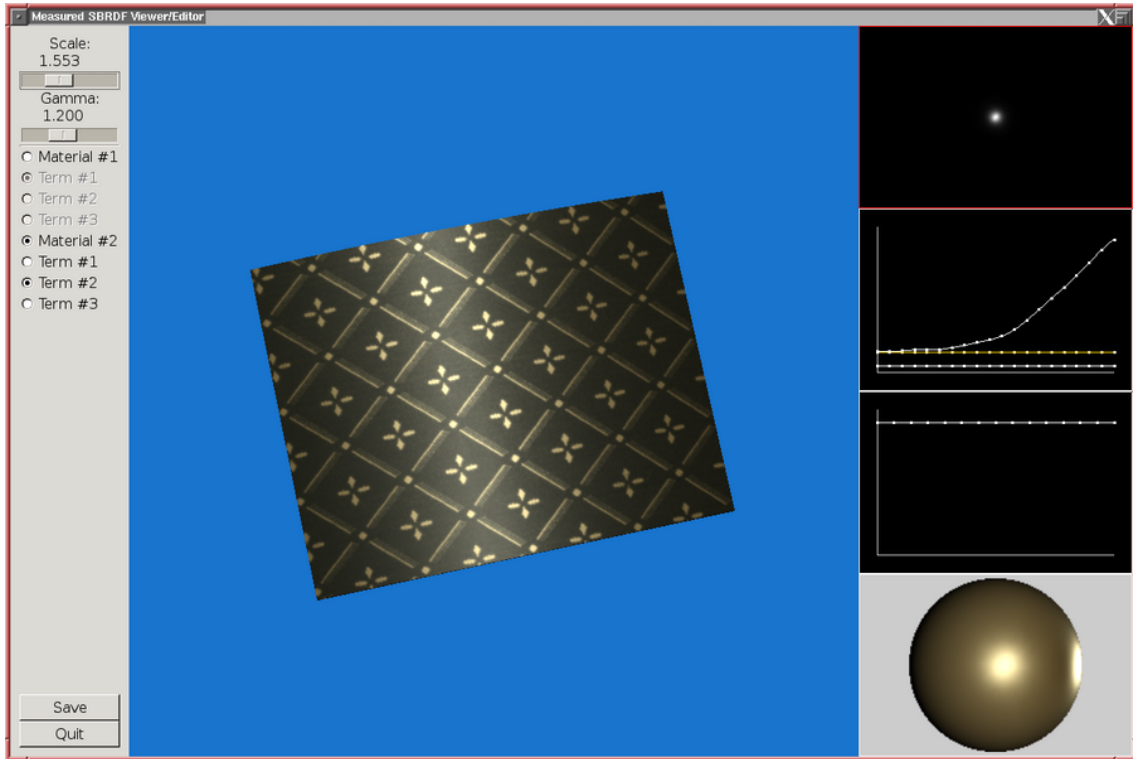


Figure 4.12: We have implemented a prototype system for interactive rendering and manipulation of SVBRDFs represented as different shade trees.

using the decompositions achieved by alternative algorithms (Figures 4.7 and 4.19).

Changing or Combining Basis Materials: Figure 4.1 shows an edit in which one of the component materials was made less shiny (using the BRDF curve editing techniques discussed below), while the hue of the other material was changed. A related edit involves completely replacing basis materials with other measured BRDFs. In Figure 4.15 we replace the metallic-silver BRDF in the *Dove* data with several measured BRDFs from the Matusik [74] database.

4.7.2 BRDF Editing

The BRDF edits available using our 2D anisotropic factors and 1D curves for isotropic materials include many of those that have become familiar to users of parametric models, but have thus far not been easy to perform with non-parametric BRDFs. Several possibilities are shown in Figure 4.16:



Figure 4.13: A key benefit of our Inverse Shade Tree framework is that it supports editing. Here we change the normal map and blending weights in the *Season's Greetings* shade tree: we retain the original BRDFs, but spell a slightly embossed SIGGRAPH 2016 BOSTON.

- a. The diffuse color is changed by editing the $s_d(\theta_d, \lambda)$ curve. Since this is represented in HSV space, it is easy to make changes to the overall color while maintaining any desaturation, color shift, or Fresnel effects present in the original data.
- b. The shape of the highlight is represented by the $f_s(\omega_h)$ map (or the $s_s(\theta_h)$ and $r_s(\phi_h)$ curves if a decomposition into 1D factors has been performed). Warping this maintains the shape of the highlight while making it narrower or wider, or varying the amount of anisotropy.
- c. One drawback of measured data is that it contains noise that may be difficult to remove when using previous non-parametric representations. Although our curve-based model is faithful to the measured data, we can remove noise by smoothing the 1D curves. In the figure, we demonstrate this by smoothing the $s_s(\theta_h)$ curve to remove some noise in the specular highlight.
- d. The Fresnel reflection law predicts that specular highlights will become stronger and less saturated towards grazing incidence. We may introduce such an effect, or exaggerate it, by editing the $s_s(\theta_d, \lambda)$ curve.

Additional effects possible in our framework include changing retroreflective behavior (via the $s_d(\theta_d)$ curve), simulating the color shift of gonio-apparent paints (via the $s_s(\theta_d, \lambda)$ curve), and introducing nonphotorealistic behavior by quantizing the curves.



Figure 4.14: The *Wood-Tape* dataset consists of a piece of oak partially covered by semi-transparent tape and retroreflective bicycle tape. **Left:** Three original images illustrate that the tape disappears for some incident and reflected directions, making this a challenging dataset to separate into its component materials. Also note the significant brightening of the lower image due to retroreflection. **Middle:** Reconstruction provided by a shade tree with five terms, computed using the ACLS algorithm. **Right:** We edit the weight maps to remove the tape. Although our separation was not perfect, the resulting edits display few artifacts.



Figure 4.15: An example of **material transfer**: one of the subtrees of our decomposition is completely replaced with a different one. Here, we replace one of the component BRDFs with several materials from Matusik’s database, while retaining spatial texture and normal maps.

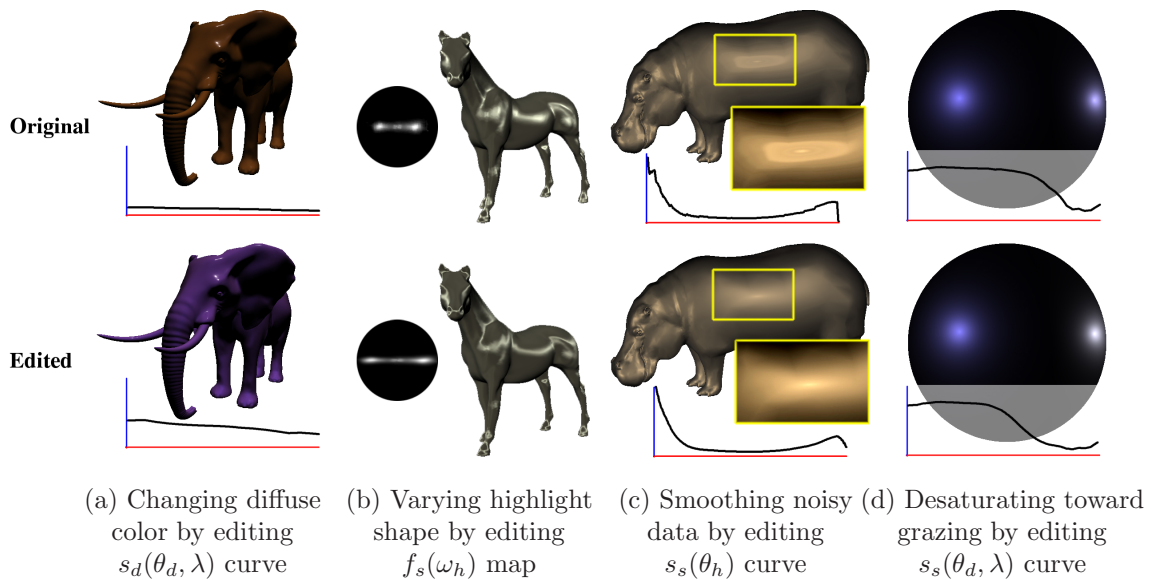


Figure 4.16: Our system allows for BRDF edits similar to those available with parametric representations, implemented by moving or warping the 1D curves and 2D maps defined in tree 2.

4.8 Comparison to Analytic Models

Our ACLS algorithm was designed to create meaningful decompositions into non-parametric shade trees that can be edited. We have compared to earlier matrix factorizations, and it is clear that those methods do not provide separations useful for *editing*—indeed, this was not their design goal. The best previous methods for creating intuitive decompositions are those that fit parametric BRDF models at each point, followed by clustering to give a user control of individual materials everywhere they appear on the surface [34, 64]. In this section, we compare to these methods, showing our higher qualitative and quantitative accuracy.

For these comparisons, we use the Ward BRDF model (as did Goldman et al. [34]). We use

the full anisotropic version of the model, and augment it for retro-reflective materials with a back-scattering lobe consisting of a Gaussian function of θ_d . While other parametric models such as Lafortune lobes can have an arbitrary number of terms, it is difficult to stably fit more than 2-3 lobes, and the form of this model does not represent complex materials and anisotropy well [81].

SVBRDF Accuracy: Figure 4.17 shows a comparison of our algorithm and approximating the SVBRDF as unique parametric fits at each surface position, as well as the result of clustering these fits. Due to the inherent flexibility of our non-parametric representation, our method introduces less error than clustering at any given term count. In fact, with only 2 materials, we are more accurate than fitting an independent Ward model to each position.

Because the RMS error is dominated by large values of the BRDF, arising from either shiny materials or measurements near grazing angles, it is important to also provide a qualitative comparison as done in Figure 4.18. Note that the Ward model is unable to match the irregular shaped anisotropic highlight in the wallpaper (the supplemental video contains additional comparisons). In the bottom of Figure 4.18, it also poorly approximates the shiny materials for positions outside their specular highlights. This is a common problem that occurs when the error of the analytic fit is dominated by the large values of the BRDF near specularities and grazing angles. Because we represent the BRDF as a sampled function, our shade tree is flexible enough to match the measured appearance of these datasets qualitatively better than the analytic model.

SVBRDF Material Separation and Editability: We also evaluate our approach in its ability to provide a final separation of materials that is suitable for editing. We present qualitative comparisons of the separation achieved using our techniques, and parametric clustering, for two particularly challenging cases.

The top rows of Figure 4.19 show the separation of the *Season’s Greetings* dataset into four blending weights computed from clustering Ward fits (top row), and using our ACLS algorithm (second row). Note that clustering the parameters improperly assigns a cluster to the *combination* of the gold foil and white paper (top row, second image) in addition to incorrectly combining the gold and silver foils into a single cluster (top row, left). ACLS correctly associates the four component materials with unique terms, providing a final separation that can be edited (Figure 4.13).

The bottom pair of rows in Figure 4.19 show separation results for the *Wood+Tape* dataset. In

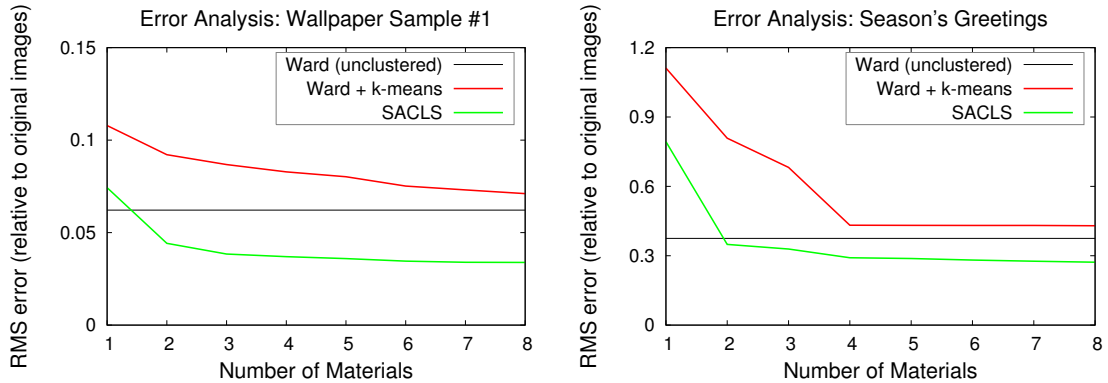
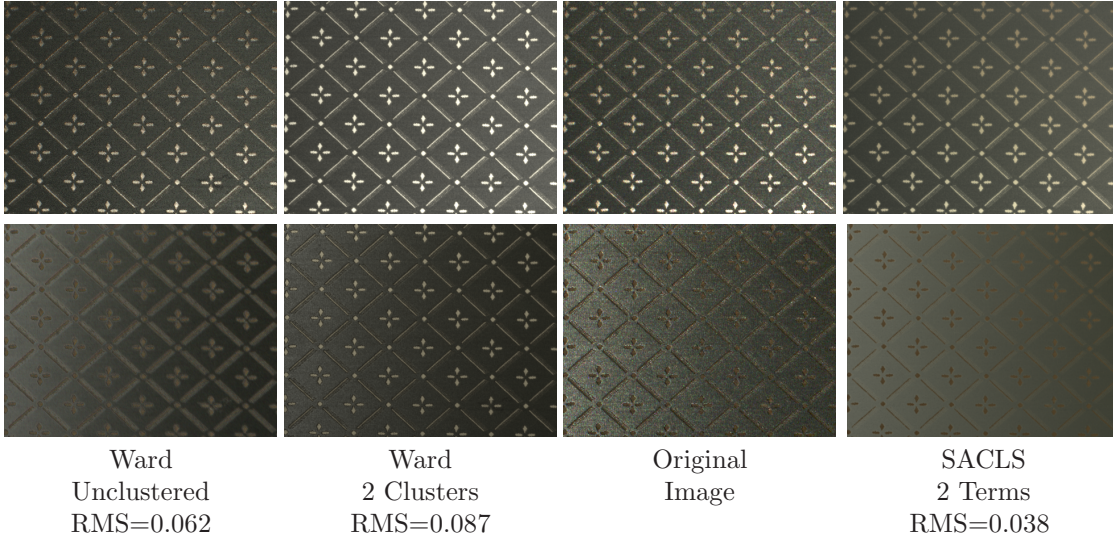


Figure 4.17: Quantitative comparison of representing a measured SVBRDF using the Ward BRDF and k -means clustering vs. our method for increasing number of clusters/terms. For reference, Figure 4.18 shows a visual comparison of the reconstructions of these methods for different numbers of terms.

this case, the parametric approach is unable to recover the transparent tape layer, as its reflectance is always observed in combination with the underlying wood. This results in a separation that incorrectly assigns the same cluster to regions of the wood grain *and* the tape (second to bottom, third column). On the other hand, ACLS (bottom row) automatically separates the SVBRDF into two distinct types of wood grain, a separate layer for the semi-transparent tape, and separate terms for the two colors of the retroreflective bicycle tape. This produces a shade tree with components appropriate for rendering and editing (Figure 4.14).

BRDF Accuracy: We compare the accuracy of our decomposition of the BRDF into 2- and 1-D factors with fitting the anisotropic Ward BRDF model to the original measurements. In Figure 4.20, we show both numerical and qualitative analysis of the error in using these techniques to represent (top) retroreflective bicycle tape and (bottom) green anisotropic wallpaper. Recall that the original measurements are rasterized into a uniformly spaced table of values organized into a matrix. This introduces error into the approximation, which is quantified between the second and third columns in Figure 4.20. Further decomposing this tabular BRDF into 2D factors and 1D curves introduces additional error, as shown in the two rightmost columns. Along the leftmost column, we show qualitative comparisons and error numbers for using the Ward BRDF model to approximate these original measurements. The fixed form of the parametric model leads to higher qualitative and quantitative error for some light sources and views. In particular, the

Visual Comparison: Wallpaper #1



Visual Comparison: Season's Greetings



Figure 4.18: Visual comparison between unclustered Ward model (fit independently to each spatial location), clustered Ward fits, original images, and our method. We also list the RMS values shown in Figure 4.17.

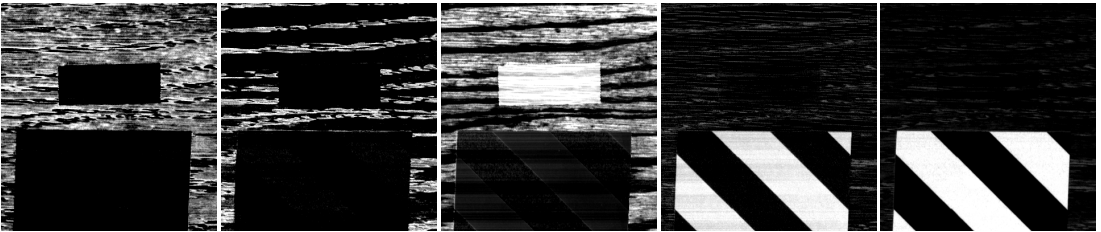
Separation using k -means on Ward Parameters



Separation using ACLS



Separation using k -means on Ward Parameters



Separation using ACLS

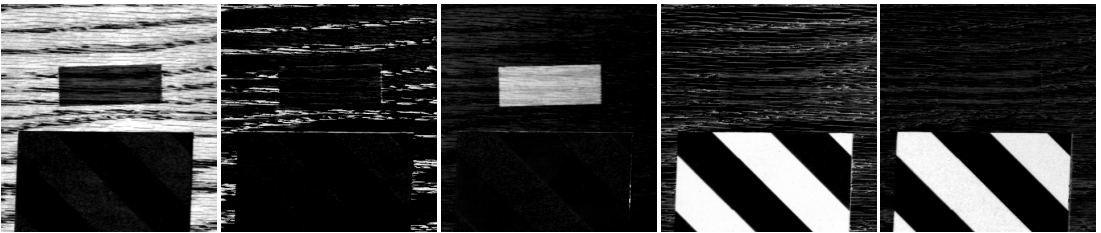


Figure 4.19: Visual comparison of the separation achieved by applying k -means clustering to the fits of a Ward BRDF and that computed by the ACLS algorithm for two different datasets. We computed four terms (resp. clusters) for the (top two rows) “Season’s Greetings” dataset and (bottom two rows) the “Wood-Tape” dataset. For the ACLS algorithm, we weighted the sparsity and L_1 norm constraints with $\lambda = 100.0$ and $\mu = 10.0$ for Season’s Greetings, and $\lambda = 100.0$ and $\mu = 800.0$ for Wood-Tape.

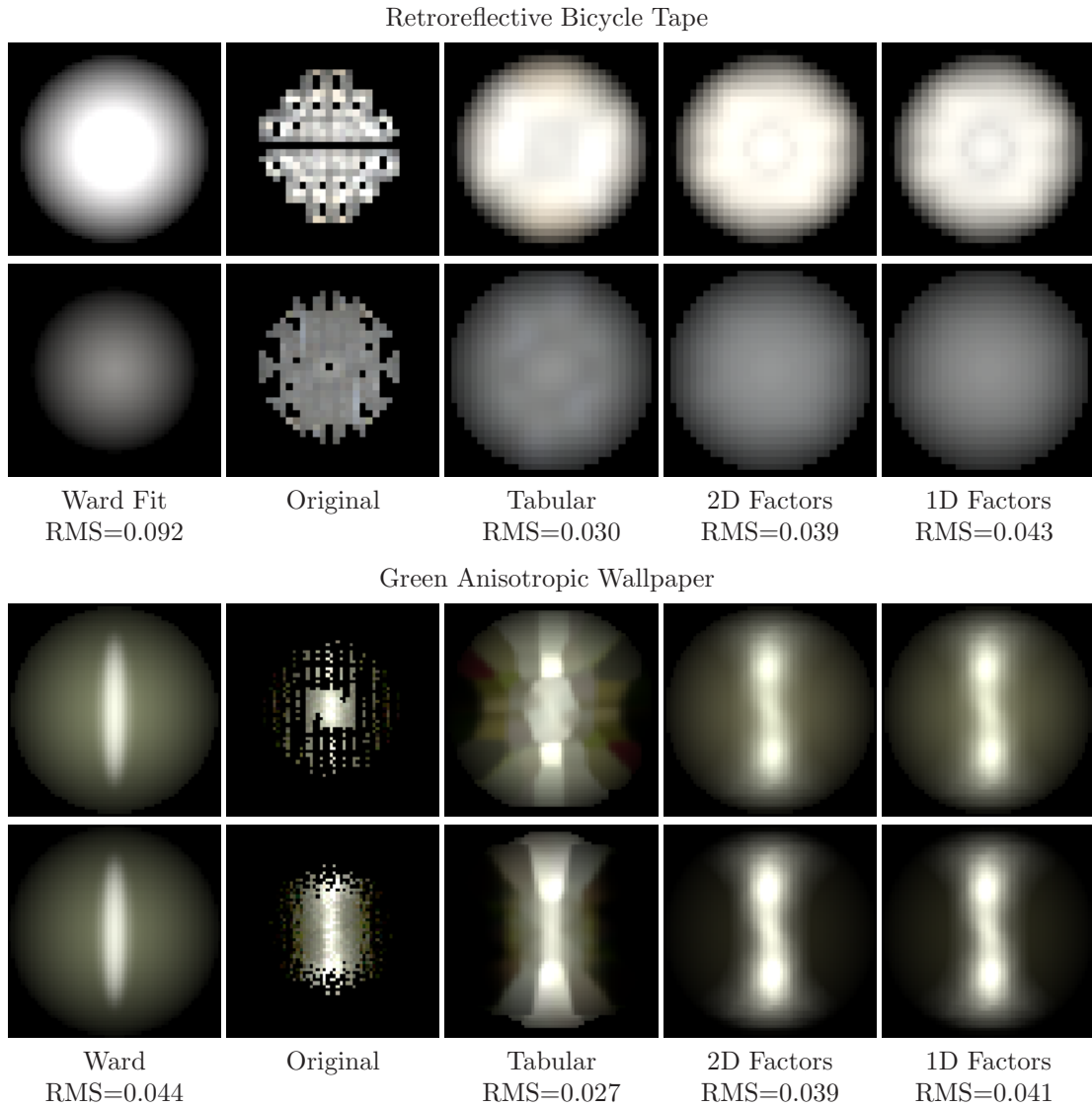


Figure 4.20: Analysis of the error introduced by several levels of our tree-structured decomposition for BRDFs, and comparison with Ward fits. For each material, the top and bottom rows show parabolic maps of ω_h distributions at $(\theta_d = 15^\circ, \phi_d = 90^\circ)$ and $(\theta_d = 45^\circ, \phi_d = 90^\circ)$ respectively.

analytic model overestimates the magnitude of the back-scattering lobe near $\theta_d = 0$. Moreover, the precise shape of the anisotropic highlights (for wallpaper) is not well approximated by an elliptical Gaussian.

4.9 Limitations

Our approach is designed for a variety of real-world spatially-varying materials. An important assumption, however, is that BRDFs are blended linearly over the surface, as in most real materials. It is theoretically possible for the 6D SVBRDF to vary smoothly, but not be easily expressible as a *linear* combination of basis materials or 4D BRDFs. In these cases, alternative representations may be more compact but not editable, since this has not been addressed by previous techniques. Another limitation on this work is that we must build a regularly-sampled data matrix before applying our factorization. By contrast, methods such as nonlinear parameter fitting, homomorphic factorization, or radial basis function interpolation operate directly with scattered input data. In practice, our use of confidence weighting and subsampled reconstruction minimizes the resampling error and additional computational time associated with our use of regularly-sampled matrices.

4.10 Shade Trees for Heterogeneous Subsurface Scattering

We have also investigated how the shade tree framework could be used to represent the spatial component of the Bidirectional Subsurface Scattering Reflectance Distribution Function (BSSRDF) for heterogeneous translucent materials [86]. Unlike homogeneous subsurface scattering, like that observed in milk, these materials cannot be easily fit to simple analytic models such as the dipole approximation [49]. This is because their complex internal structure results in discontinuities that arise from both geometric deficiencies (e.g. veins, cracks, etc.) and the presence of multiple types of materials (see Figure 4.21, top row).

We use a projector-camera pair to record the spatial component of the subsurface scattering of these materials. Specifically, we project a pattern of dots onto a planar material sample and recording its response with a digital camera. This setup allows us to recover a 2D slice of the BSSRDF for a fixed light and view direction: $S' = S(x_{in}, x_{out}) = S(x_{in}, \omega_i, x_{out}, \omega_o)$. This function can be thought of as encoding the amount of light that is transported between *all pairs*

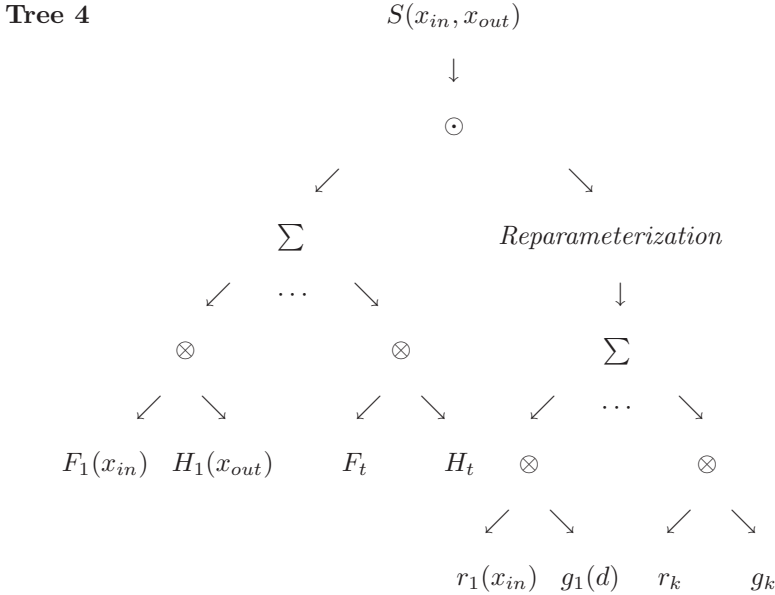
Sample Material	Physical Size (cm^2)	Resolution (pixels)	Kernel Size (pixels)	No of Terms	No of Clusters	Original Data Size	Factored Data Size	Ratio	Min/Avg/Max Relative Error
Candle (Red and Yellow Wax) (a)	19.8×14.3	212×154	49×49	12	100	898MB	13MB	1/69	0.002 / 0.031 / 0.050
White and Green Marble (4×4) (b)	12.6×12.6	277×277	39×39	20	100	1.4GB	39GB	1/37	0.003 / 0.010 / 0.060
White and Green Marble (8×8)	25.1×25.1	222×222	39×39	8	10	859MB	12MB	1/72	0.003 / 0.019 / 0.094
Vertical Lines (White Onyx) (c)	15.2×15.2	229×229	39×39	20	10	914MB	26MB	1/35	0.004 / 0.017 / 0.100
Cracked Material (Crystal Onyx) (d)	18.5×17.6	270×260	45×45	12	5	1.6GB	22MB	1/74	0.011 / 0.040 / 0.560
Marble (close up)	2.6×2.6	128×128	39×39	16	1	286MB	6.6MB	1/43	0.006 / 0.013 / 0.056
Densely Veined Marble	13.0×13.0	213×211	29×29	8	10	433MB	9.9MB	1/44	0.003 / 0.096 / 0.163
Slightly Veined Marble	17.9×17.9	207×207	29×29	20	1	413MB	21.0MB	1/19	0.001 / 0.005 / 0.024

Table 4.2: Statistics and details regarding acquired and factored subsurface scattering materials.

of points on the surface. Acquiring the full 8D BSSRDF is still an open research question.

Unlike the SVBRDF, the structure of S' makes it poorly suited for a linear separation. However, remove its diagonal structure by independently estimating a set of distinct responses that correspond to the *core materials* of the sample through a clustering procedure. The residual heterogeneity is caused by geometric deficiencies in the material such as veins or cracks and is suitable for direct separation using a modified version of the NMF algorithm.

In the end, we have computed a multi-stage decomposition that can be described by the following shade tree:



where \odot represents component-wise multiplication, k are the number of clusters and t are the number of terms in the non-negative factorization.

Table 4.2 lists the statistics for the various datasets we acquired. Figure 4.21 presents a qualitative analysis of the error present in our factored representation. Note that the discontinuities



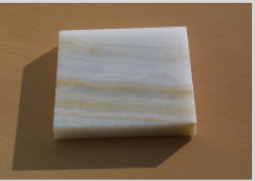

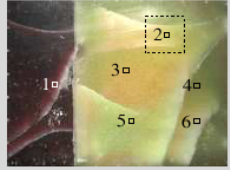
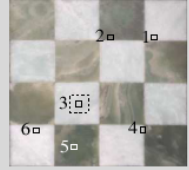
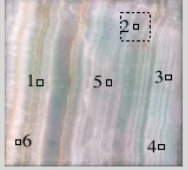

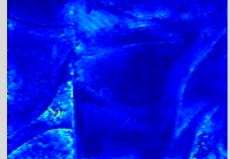
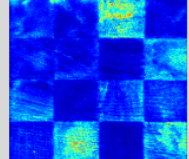
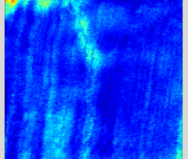
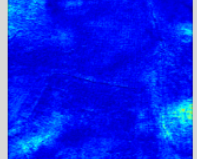
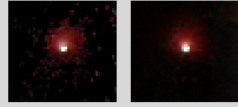
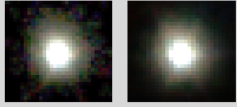
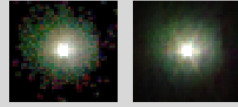
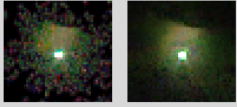

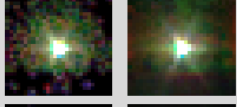

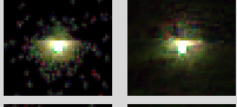
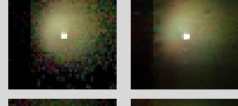
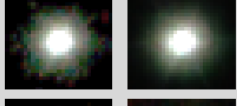
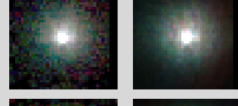
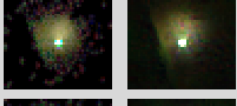
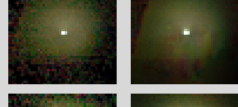
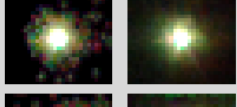

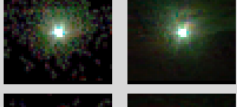

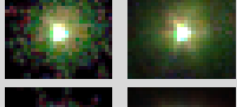

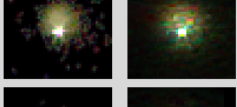
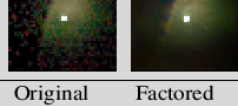
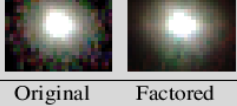
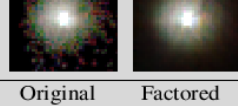
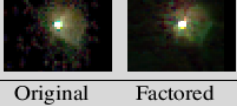
	Candle (a)	Checker board (4 × 4) (b)	White Onyx (c)	Crystal Onyx (d)
Original Object				
Diffuse Albedo				
Relative Error				
1.				
2.				
3.				
4.				
5.				
6.				
	Original Factored	Original Factored	Original Factored	Original Factored

Figure 4.21: A selection of acquired and factored materials. For each material, a photograph of the original object, the diffuse albedo map, a relative error distribution plot, and a selection of measured responses with the corresponding factored approximations are shown. The locations of the responses are marked on the diffuse albedo map. The dashed square illustrates the relative size of the responses. Table 4.2 gives additional information regarding the range of the relative errors.



Figure 4.22: A shade tree computed from a composite wax material applied to the Stanford dragon. The material is composed of two kinds of wax with different scattering properties. Left: illuminated by an area light source from above. Middle: the material’s diffuse albedo (no subsurface scattering). Right: illuminated from above by a textured projection light source.

in the subsurface scattering are preserved by our representation. In Figure 4.22 we show images rendered with a shade tree that was computed from measurements of a composite wax candle material.

4.11 Conclusions and Future Work

We have introduced a nonparametric *Inverse Shade Tree* framework for representing and editing measured spatially- and directionally-dependent surface appearance. The representation is more accurate than parametric models, more intuitive than other non-parametric methods, and well-suited for interactive rendering and editing.

As future work, we would like to investigate algorithms that automatically infer the structure of the tree according to the data, including automatic selection of the number of terms to use at each decomposition. In addition, we may simultaneously decompose the same dataset into multiple trees, any of which may be edited depending on the desired change. Another possible direction for future work is to incorporate additional aspects of reflectance variation such as displacement maps or fine geometric detail typically represented as Bidirectional Texture Functions (BTFs).

Another avenue of future work is related to the ACLS techniques we have proposed. Their flexibility and provable local convergence make them ideal candidates for a broad range of dimensionality reduction applications in data mining and other machine learning contexts. We wish

to evaluate the efficiency and noise-tolerance properties of ACLS, and investigate the impact of various types of additional linear constraints.

Chapter 5

Global Illumination Rendering

5.1 Introduction

The goal of rendering algorithms is to compute the light energy (i.e. radiance) arriving at a virtual camera’s image plane positioned within a 3d scene. The amount of radiance traveling along each ray in space can be expressed with the *rendering equation* [51]:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega_+} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) \cos \theta_i d\omega_i, \quad (5.1)$$

where L_o is the radiance leaving point x along the direction ω_o , L_e is the radiance that is emitted by the material at point x , L_i is the radiance arriving at point x along the direction ω_i and f_r is the BRDF of the material at x . Recall the cosine term is necessary to convert the radiance arriving from direction ω_i into the irradiance incident to the surface at x .

The goal of global illumination algorithms is numerical integration of Equation 5.1. The approach of Monte Carlo algorithms is to evaluate the incoming radiance L_i by recursively casting rays through the scene to simulate light transport paths.

In order to reduce the variance of a Monte Carlo estimator, it is desirable to *importance sample* reflected rays by preferentially considering paths carrying high energy. For scenes with glossy materials and slowly-varying illumination, we prefer to sample according to the product of the BRDF f_r and the incident cosine term $\omega_i \cdot \mathbf{n}$. Section 5.4 describes a factored representation of

the BRDF suitable for importance sampling complex analytic and measured materials. It is also possible to importance sample according to the lighting L_i , as in sampling light sources for direct illumination. In the special case when L_i is described by an environment map, effective illumination sampling methods have recently been demonstrated [1, 55]. In Section 5.5 we introduce a new technique for sampling multi-dimensional measured functions that we apply to sampling incident illumination fields. We also show how this technique allows us to combine strategies for sampling the BRDFs and the illumination.

5.2 Previous BRDF Models

There exist many analytical models for the BRDF that approximate the way specific materials reflect light. Some of these are phenomenological, such as the popular Phong shading model [89]. More sophisticated, physically-based analytical models can capture effects including Fresnel reflection and rough microgeometry [101, 20, 42]. Anisotropic reflection models characterizing the reflective properties of oriented surfaces such as brushed metal have also been developed [50, 109, 90, 5]. Other analytical BRDF models, such as those meant to describe dusty surfaces, exhibit backscattering phenomena [39, 84]. Despite the large amount of research on these BRDFs, most of these models have so far been difficult to sample efficiently. This typically arises because the analytic formula is difficult or impossible to integrate and invert.

The potential benefit of using measurements of a BRDF has also gained recent attention [109, 37, 21, 70]. The measurements of Matusik et al. [74] provide a dense ($90 \times 90 \times 180$) sampling of many isotropic BRDFs. The main drawback of these models is their size, since they typically represent the full 3D isotropic BRDF in tabular form. In his thesis, Matusik [74] also describes one approach for sampling these measured BRDFs, but this representation requires as much storage as the original BRDF, making it difficult to use for scenes containing many materials.

In an effort to reduce the size of measured BRDF models while maintaining an accurate representation of their effects, several researchers have investigated techniques for factoring these large datasets into a more compact, manageable form [53, 77, 98]. In all cases, the 4D BRDF is factored into products of 2-dimensional functions that can be represented as texture maps and used to shade a model in real-time. However, in most cases these factorizations allow only a sin-

gle term approximation. More relevant, there are no techniques for importance sampling these representations.

5.3 Importance Sampling

The benefit of stratified importance sampling within the context of physically-based rendering has certainly been justified by the work of Cook [19]. Since Shirley demonstrated how to efficiently sample the traditional Phong BRDF [96] and Lafortune introduced a generalization of this cosine-lobe model [56], a reasonable approach to importance sampling an arbitrary BRDF has been to sample a best-fit approximation of one of these simpler models. Although this technique marks a clear improvement over random sampling, it has several drawbacks. First, it is not always trivial to approximate the many complex BRDFs that exist in nature with one of these models. Often, a nonlinear optimizer has difficulty fitting more than 2 lobes of a Lafortune model without careful user intervention. Second, since the sampling is only as efficient as the approximation is accurate, it is not always the case that this strategy will optimally reduce the variance for an arbitrarily complex BRDF. Our approach, on the other hand, robustly detects the energy in a BRDF during the factorization step and provides a more efficient sampling strategy.

5.4 A New BRDF Representation for Sampling

In this section, we consider the requirements and design choices in choosing our BRDF representation for sampling. We discuss why our requirements are different from those of previous factored representations, and present a new factorization approach optimized for the needs of importance sampling. Sections 5.4.1 and 5.4.2 go on to discuss implementation details of our representation and sampling algorithms.

We begin with the observation that, in the context of a standard backward ray- or path-tracer, we will generally know the outgoing direction (θ_o, ϕ_o) and will need to sample lighting and visibility over the incident hemisphere. A straightforward approach would be to tabulate (θ_i, ϕ_i) slices of the BRDF for a dense set of directions covering (θ_o, ϕ_o) , and use the appropriate one. This is essentially the approach taken by Matusik [73] and, as far as we know, is the only previous approach for importance sampling of arbitrary measured materials. However, as already noted,

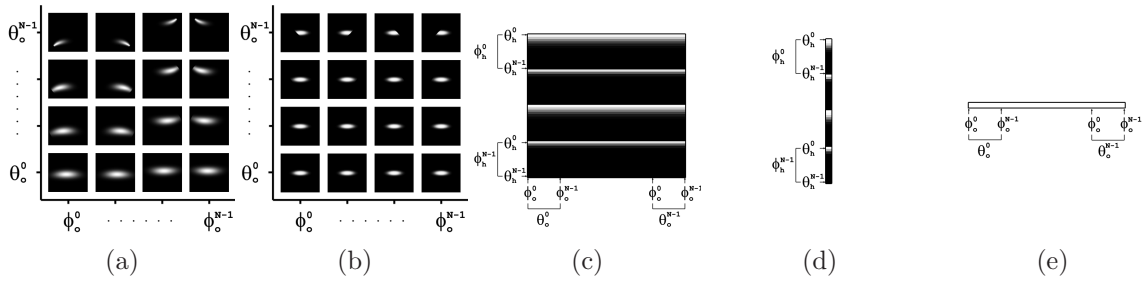


Figure 5.1: Steps in factoring a BRDF into our representation, shown for a Phong-like anisotropic BRDF [5]. (a) We first sample the BRDF at a fixed set of incoming and outgoing directions. Notice that in this case the flattened highlight moves according to the position of perfect specular reflection for each view. (b) We would like to maximize the symmetry in these 2D slices of the BRDF to make the factorization more accurate. We accomplish this by reparameterizing the samples of the BRDF with respect to the half-angle vector, as the energy in this BRDF is symmetric about this direction. (c) We organize these samples into a 2D matrix by unfolding each 2D slice of the BRDF into a separate column in the matrix according to its half-angle parameterization. Notice that our choice of parameterization produces a data matrix that has a rank very close to 1. (d & e) We use non-negative matrix factorization to factor this data matrix into the outer product of two vectors. Because the rank of the original matrix was close to 1, we need only one term in the factorization. (d) In the end, we are left with a column vector dependent only on the incoming direction (G_1 reparameterized with respect to the half-angle vector in this case) and (e) a row vector F_1 dependent only on the outgoing view direction.

this representation requires a large amount of storage space for both analytical and measured materials—as large as or larger than the tabulated BRDF itself.

Instead, we observe that for nearly all common materials there is coherence in the BRDF for different outgoing directions—for instance, the shape of the lobe in a specular BRDF will often remain similar. Our goal is to exploit this coherence to develop a compact representation. First, we **reparameterize** the BRDF, e.g. by using the half-angle [94]. As a number of authors have observed, reparameterization by the half-angle properly aligns BRDF features such as specular reflection, making them simpler to represent. Next, we use **factored forms**, writing the 4D BRDF as a sum of a small number of products of 2D factors, to exploit the coherence in the BRDF and develop a compact representation. Similar approaches have been developed for real-time rendering, and have shown that reparameterization and factorization can be a compact and accurate way to represent most BRDFs [53, 77].

The specific factored decomposition we use is the following:

$$\rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n}) \approx \sum_{j=1}^J F_j(\omega_o) G_j(\omega_p), \quad (5.2)$$

where we have decomposed the original 4D BRDF function (multiplied by the cosine of the incident angle) into a sum of products of 2D functions. One of the functions always depends on the view direction ω_o , and the other function is dependent on some direction ω_p arising from the reparameterization. In the case of a half-angle parameterization, ω_p is taken to be

$$\omega_h = \frac{\omega_i + \omega_o}{|\omega_i + \omega_o|}. \quad (5.3)$$

Unlike previous factorization approaches, this representation satisfies a number of key properties for sampling:

- **One factor dependent on outgoing direction:** When sampling according to our representation, we know the outgoing direction ω_o but not the incident direction (since we are sampling over it). Therefore, we can directly evaluate F and it is critical that it depend *only* on the outgoing direction.
- **Sum of products of two factors:** Each term above is the product of two factors F and G , where F depends only on the outgoing direction. Thus, it is easy to sample according to the second factor G only. On the other hand, approaches such as homomorphic factorization [77] or chained matrix factorization [98] can include multiple factors in a term, making importance sampling difficult. We can also enable multiple terms (with different j) for more accurate sampling—another feature that is difficult to incorporate in homomorphic factorization.
- **Non-negative factors:** As opposed to using a matrix decomposition algorithm such as SVD [53], we use non-negative matrix factorization to ensure that all values are positive. This is necessary for interpreting the resulting factors as probability distributions, according to which we can then sample.

We now observe that most BRDFs can be compactly represented by further factoring each of the G_j into 1D functions $u_{jk}(\theta_p)$ and $v_{jk}(\phi_p)$. By doing this, we can separately treat u and v as 1D distributions, for which importance sampling is straightforward. Our final factored BRDF representation is therefore

$$\rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n}) \approx \sum_{j=1}^J F_j(\omega_o) \sum_{k=1}^K u_{jk}(\theta_p) v_{jk}(\phi_p). \quad (5.4)$$

There are a total of JK terms in the final factorization, each a product of a two-dimensional function (F_j) and two one-dimensional functions (u_{jk} and v_{jk}).

It should be noted that this factorization fits the form of many isotropic and anisotropic BRDFs well. For instance, a Blinn-Phong BRDF [9] can be fit exactly by a single term ($J = K = 1$), with variation appearing only in the $u(\theta_h)$ function. Similarly, only 2 terms ($J = 1, K = 2$) are needed for an anisotropic Phong BRDF [5]. The same holds approximately for many other materials, so the above representation typically gives accurate results with a small number of terms.

Note that, unlike other uses of factored BRDF models, we typically use the representation above only for choosing samples. For actually computing the BRDF value, we use an analytic formula where available. In this case, the representation above is a compact means of sampling these commonly used analytic models, which have hitherto been difficult to sample. Similarly, for compact basis function representations, such as the Zernike polynomial expansions [54] used in the CURET database [21], we can use the BRDF value represented by the basis functions, using our representation only for importance sampling. In other cases, such as the dense measured BRDF representations of Matusik et al. [74], we take advantage of the compactness of our multi-term factorization and use it as the primary representation for both reconstruction and sampling. This provides a 200-fold savings in storage in many cases, while remaining faithful to the original data.

5.4.1 Factorization

We now describe the details of our method to factor a tabular BRDF—Figure 5.1 provides an overview of the process. Unlike some previous methods, we use multiple non-negative terms in equations (5.2) and (5.4). This disallows common techniques such as homomorphic factorization or singular value decomposition. Instead, inspired by Chen et al. [16], we use non-negative matrix factorization (NMF) [62] to decompose the reparameterized BRDF. NMF is an iterative algorithm that allows multi-term factorizations, guaranteeing that all the entries in the factors are non-negative. We have found it to be robust for both single- and multi-term decompositions, and capable of producing accurate approximations for a wide range of both analytical and measured BRDFs.

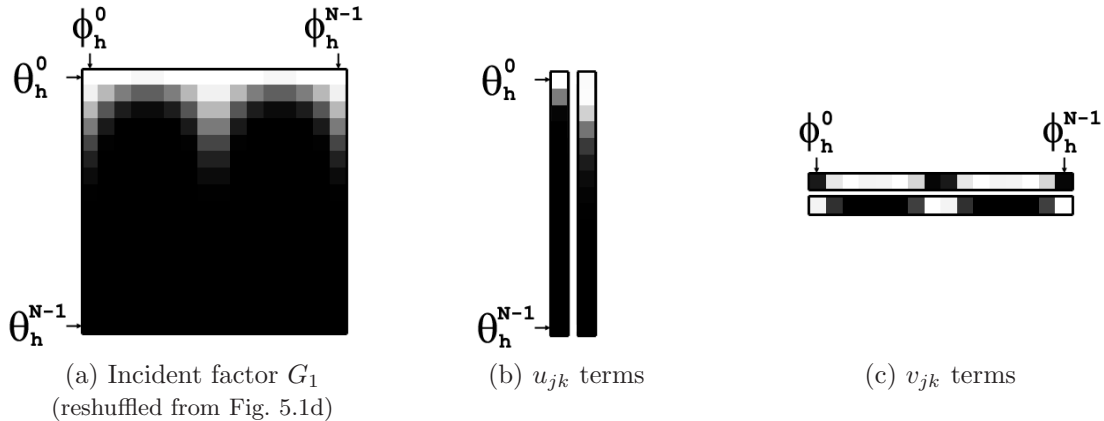


Figure 5.2: In order to optimize our representation for importance sampling, we perform another factorization step on the 2D functions dependent on the incoming direction (a) We first re-organize each column in G into a matrix such that the rows and columns vary with respect to the elevation and azimuthal angles respectively. We again apply NMF to decompose this matrix into an outer product of terms. In this example, we choose to factor this matrix into two terms. In the end, we are left with (b) two column vectors that each depend only on the elevation angle and (c) two row vectors that depend only on the azimuthal angle of the incoming direction. These are the u and v terms, respectively, in our final representation.

Data matrix: We first organize the set of values of the original reparameterized BRDF into a matrix. We consider taking N_{θ_o} regular samples along the outgoing elevation angle and N_{ϕ_o} samples along the outgoing azimuthal angle. For each of these N_{ω_o} view directions, we record N_{ω_p} samples of the BRDF intensity (multiplied by $\cos\theta_i$), spaced equally in azimuthal and elevation angles for the chosen BRDF reparameterization. We organize the initial data samples into an $N_{\omega_p} \times N_{\omega_o}$ matrix Y .

First factorization: Using the appropriate NMF update rules, which are summarized in the appendix, we factor Y into the product of two matrices of lower dimension:

$$\begin{bmatrix} Y \end{bmatrix} = \begin{bmatrix} G \end{bmatrix} \begin{bmatrix} F \end{bmatrix} \quad (5.5)$$

As shown in Figure 5.1, G is an $N_{\omega_p} \times J$ matrix, with each column corresponding to a factor G_j in equation (5.2), while F is a $J \times N_{\omega_o}$ matrix, with each row corresponding to a factor F_j in equation (5.4). In practice, we rarely need more than 3 or 4 terms to achieve an accurate approximation,

and we always reduce the size of the original BRDF by at least an order of magnitude since our factored representation involves 2D functions rather than a 3D or 4D BRDF.

If we were interested only in reducing the size of the BRDF, or in using our representation for real-time rendering, it might be reasonable to use these 2D functions directly. Notice, however, that we have approximated the intensity of the BRDF, $\rho_{int}(\omega_i, \omega_o)$ times the cosine term, which is appropriate for importance sampling. Therefore, we would need to update this representation to account for the wavelength dependence in the original BRDF. We accomplish this by using NMF to compute a single-term approximation of the BRDF at a particular wavelength (e.g. red, green or blue) divided by the intensity. For the red color channel, we would factor a data matrix composed of samples of the function: $\rho_{red}(\omega_i, \omega_o)/\rho_{int}(\omega_i, \omega_o)$ and reconstruct the red value of the BRDF by scaling our approximation of the intensity by the approximation of this function.

There also remains a challenge in sampling according to the 2D distribution G_j . It is possible to use explicit tabular approaches, by storing a Cumulative Distribution Function on θ_p for each ϕ_p , but such representations are not compact. Furthermore, effectively generating stratified samples given these 2D tabulated CDFs proves to be a difficult problem¹ Therefore, we perform a second factorization of G_j into 1D functions dependent on θ_p and ϕ_p , which not only matches the form of most common BRDFs, but also makes the representation easy to sample and further reduces storage requirements.

Second factorization: As shown in Figure 5.2, we separately factor each column of the matrix G , corresponding to a 2D function that depends on the reparameterized incoming direction (θ_p, ϕ_p) :

$$\begin{bmatrix} G_j \end{bmatrix} = \begin{bmatrix} u_j \end{bmatrix} \begin{bmatrix} v_j \end{bmatrix}, \quad (5.6)$$

where $u_j(\theta_p)$ is an $N_{\theta_p} \times K$ matrix, with each column corresponding to a factor u_{jk} in equation (5.4), and $v_j(\phi_p)$ is a $K \times N_{\phi_p}$ matrix, with each row corresponding to a factor v_{jk} .

¹Effective 2D stratification in the context of environment map sampling has been proposed by Agarwal et al. [1] using Hochbaum-Shmoys clustering, but this approach requires fixing the number of samples a priori, while in our case the number of samples for each G_j depends on the view.

Normalization: For the purposes of sampling, it is desirable to treat u_{jk} and v_{jk} as normalized 1D probability distribution functions. To do this, we first define

$$\bar{u}_{jk} = \int_0^\pi u_{jk}(\theta_p) \sin \theta_p d\theta_p, \quad \bar{v}_{jk} = \int_0^{2\pi} v_{jk}(\phi_p) d\phi_p. \quad (5.7)$$

Then we normalize each term T_{jk} as

$$\begin{aligned} T_{jk} &= F_j(\omega_o) u_{jk}(\theta_p) v_{jk}(\phi_p) \\ &= (\bar{u}_{jk} \bar{v}_{jk} F_j(\omega_o)) \left(\frac{u_{jk}(\theta_p)}{\bar{u}_{jk}} \right) \left(\frac{v_{jk}(\phi_p)}{\bar{v}_{jk}} \right) \\ &= F'_{jk}(\omega_o) u'_{jk}(\theta_p) v'_{jk}(\phi_p). \end{aligned} \quad (5.8)$$

Finally, dropping the primes and using a single index l , we obtain the final form of our factored representation, where u_l and v_l are proper 1D probability distribution functions,

$$\rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n}) \approx \sum_{l=1}^L F_l(\omega_o) u_l(\theta_p) v_l(\phi_p), \quad L = JK. \quad (5.9)$$

Discussion: Our representation is designed with a view to developing a sampling algorithm, and lacks two properties that are sometimes theoretically desirable. First, the terms in equation (5.2), whose form is essential for sampling, do not explicitly enforce reciprocity. (Of course, since we factor the product of the BRDF and the cosine term, the input function is not reciprocal to begin with.) Second, the representation is not guaranteed to be continuous—there can be a discontinuity at the pole $\theta_p = 0$ in the second factorization in equation (5.4). In either case, we have not observed any drawbacks in practice because of these properties and it can be seen from our results that our multiple-term fits are accurate.

5.4.2 Sampling

We now describe how to use our representation in equation (5.9) for importance sampling. Intuitively, each term in the approximation corresponds to a specific “lobe” of the original BRDF, and the factorization algorithm works to find the best set of lobes to approximate its overall structure. We first randomly select one of these lobes according to the energy it contributes to the overall

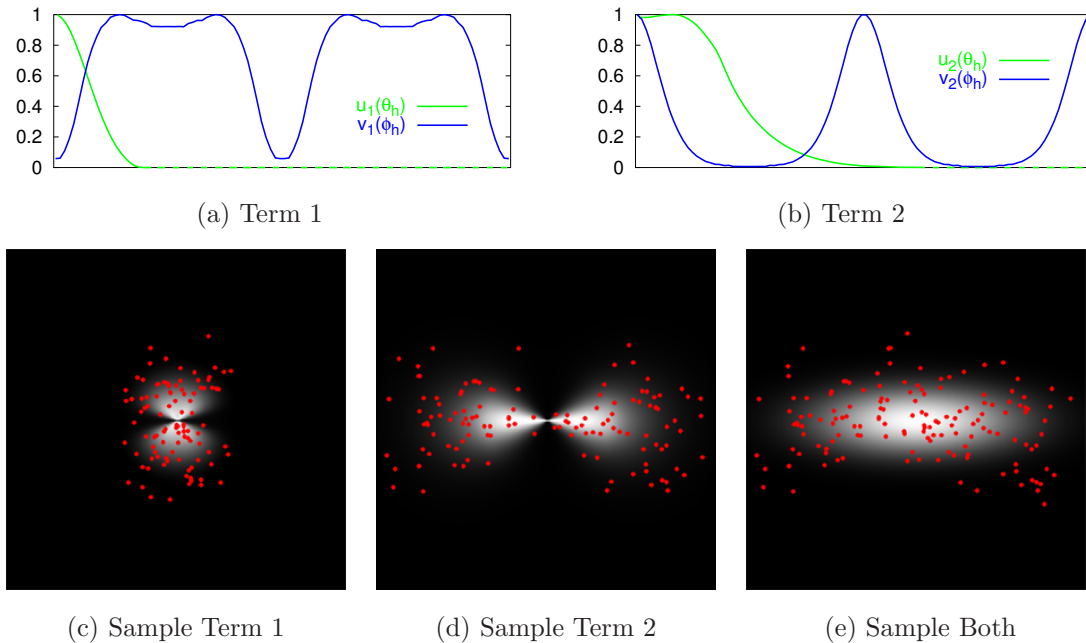


Figure 5.3: The main benefit of our factored representation is that it can be used to efficiently sample a BRDF. (a & b) In the top row, we graph the values of the 1D terms in our factorization of the BRDF considered in Figures 5.1 and 5.2. The green lines show the values of $u_1(\theta_h)$ and $u_2(\theta_h)$, whereas the blue lines represent $v_1(\phi_h)$ and $v_2(\phi_h)$. Using the strategy detailed in Section 5.4.2, we select either the first or second term to generate an incoming direction. (c) Using only the first term to generate samples, we notice that the directions accumulate around a pair of lobes along the y axis, centered within the highlight. (d) Using only the second term to generate samples, the directions accumulate around two lobes centered at $\phi_h = 0$ and $\phi_h = \pi$. (e) When we select between these two terms with equal probability, we produce a sampling pattern that matches the energy in the original BRDF.

BRDF for the current view. Next, we sample the hemisphere according to the shape of this lobe by sequentially generating an elevation and azimuthal angle according to the 1D factors u_l and v_l .

To further demonstrate this idea, consider the pair of factors that we computed to approximate the anisotropic BRDF in Figures 5.1 and 5.2. The first term creates a pair of lobes that extend along the y-axis, centered about the specular direction (Figure 5.3a), and the second term creates a pair of flattened lobes that extend along the x-axis (Figure 5.3b). We could imagine sampling the hemisphere according to just one of these terms (Figure 5.3c,d): using each term alone generates samples in a different region of the BRDF. If we generate samples according to both terms with equal probability, however, the aggregate effect is that we distribute samples along the anisotropic highlight (Figure 5.3e).

Original BRDF	Resolution $\theta_o \times \phi_o \times \theta_p \times \phi_p$	Terms $J \times K$	Param	Normalized MAE	
				Factored	Lafortune
Cook-Torrance	$16 \times 16 \times 32 \times 16$	4×1	ω_h	0.192	0.632
Ward	$16 \times 16 \times 100 \times 100$	2×4	ω_h	0.094	1.092
Poulin-Fournier	$16 \times 16 \times 32 \times 16$	3×1	ω_i	0.142	0.348
Hapke-Lommel	$16 \times 16 \times 32 \times 16$	3×1	ω_i	0.186	0.464
Nickel	$16 \times 16 \times 128 \times 16$	2×1	ω_h	0.201	0.643
Plastic	$16 \times 16 \times 128 \times 16$	3×1	ω_h	0.266	0.874
Metallic-Blue	$16 \times 16 \times 128 \times 16$	4×1	ω_h	0.118	0.464

Table 5.1: Accuracy of the factored BRDF representation. We factored 4 analytical BRDFs: Cook-Torrance ($d = 0.1, R_d = [0.12, 0.22, 0.48], s = 0.9, F_0 = R_d, m = 0.2$), Ward ($\rho_d = 0.1, \rho_s = 1.2, \alpha_x = 0.2, \alpha_y = 0.02$), Poulin-Fournier ($d = 2.0, h = 0.0, n = 20.0, R_s = 0.8, R_d = 0.2$), and Hapke-Lommel ($g = 0.6, f = 0.1, r = 1.0$), along with 3 measured BRDFs from Matusik et. al. [2003]: nickel, plastic, and metallic-blue. For each BRDF we list the resolution of the original data matrix Y , the number of “outer” and “inner” terms (J and K , respectively) of the factorization, and the parameterization of the incoming hemisphere. We report the mean absolute error (MAE) of the final factorization, normalized by the mean BRDF value. This is compared to the error resulting from fitting a multi-lobe Lafortune model to the original BRDF using a standard non-linear optimizer.

Importance Sampling

We now describe the mathematics of sampling more formally. We will be interested in evaluating the integral of the incident illumination for a fixed outgoing direction ω_o at a given pixel with location x and surface normal \mathbf{n} ,

$$\begin{aligned}
 & \int_{\Omega_{2\pi}} L_i(x, \omega_i) \rho(x, \omega_i, \omega_o) (\omega_i \cdot \mathbf{n}) d\omega_i \\
 \approx & \frac{1}{n} \sum_{s=1}^n L_i(x, \omega_s) \left[\frac{\rho(x, \omega_s, \omega_o) (\omega_s \cdot \mathbf{n})}{\gamma_i(\omega_s | \omega_o)} \right].
 \end{aligned} \tag{5.10}$$

The first line is simply the reflection equation—the incident lighting L_i may be evaluated iteratively or recursively for global illumination. The second line is a Monte Carlo estimator that describes the standard approach to importance sampling. It represents a weighted average of each of the samples, each divided by the probability γ_i of generating sample direction ω_s assuming that ω_o is fixed. The subscript in γ_i denotes that the probability distribution is over incident directions. γ_i should be non-negative and normalized, i.e. $\int_{\Omega} \gamma_i(\omega_i | \omega_o) d\omega_i = 1$. Our representation is used to generate samples ω_s , according to the probabilities γ_i . For analytic models, the actual BRDF can be used to evaluate ρ . The more accurate our representation is, the lower the variance, but

equation (5.10) is always accurate and unbiased.

If our factored representation exactly represents the BRDF multiplied by the cosine term, the numerator in the bracketed term in equation (5.10) will be exactly proportional to the denominator, and that term will simply be a constant. The estimator will then represent the ideal importance sampling method based on the BRDF, and will have low variance. In fact, in the limiting case of a constant environment (L_i is constant), there will be zero variance. In practice our representation is not exact, but it is a good approximation and importance sampling with it significantly reduces variance.

Sampling Algorithm

We now describe how to choose directions ω_s and evaluate γ_i in equation (5.10). Our method chooses the lobe l , azimuthal angle ϕ_p , and elevation angle θ_p in turn, with each step involving computing one random number and inverting a 1D Cumulative Distribution Function.

Choosing a term l : The probability of choosing a term l , for a given outgoing direction ω_o is given by

$$\gamma(l|\omega_o) = \frac{F_l(\omega_o)}{\sum_{j=1}^L F_j(\omega_o)}. \quad (5.11)$$

From these probabilities, we calculate a 1D CDF over l . To select a term, we generate a uniform random variable in $[0, 1]$ and perform a binary search on the CDF to transform the random variable into a value of l . Notice that the probabilities depend on the view direction, so we must recompute this CDF each time the outgoing direction changes. However, L is typically very small, and the same CDF can be used for all samples through a given pixel (since ω_o is fixed), so the computation is inexpensive.

Choosing azimuthal and elevation angles ϕ_p and θ_p : Having chosen the term l to sample, we must now choose ϕ_p based on the probability distribution $v_l(\phi_p)$. As before, we generate a uniformly distributed random variable in $[0, 1]$, and numerically invert the CDF at that value to determine ϕ_p . Choosing θ_p follows the same methodology, but because of the $\sin \theta_p$ area measure, we find it simpler to define $z = \cos \theta_p$, and use $u_l(z)$ as the probability distribution. Inverting the CDF then yields z , from which we find $\theta_p = \cos^{-1} z$. Note that we can precompute these CDFs

because the probabilities do not depend on ω_o —a significant benefit of our factorization.

Computing probability: Given θ_p and ϕ_p , it is straightforward to generate a direction ω_s . Due to reparameterization, it is possible for the sample directions ω_s to occasionally take values below the horizon, but we can simply set the estimator to 0 for those directions without introducing inaccuracy or bias. Otherwise, we calculate the probability for equation (5.10) as the sum of the marginal probabilities for each term:

$$\gamma_p(z, \phi_p | \omega_o) = \sum_{l=1}^L \frac{F_l(\omega_o) u_l(z) v_l(\phi_p)}{\sum_{j=1}^L F_j(\omega_o)}. \quad (5.12)$$

One issue we must address is reparameterization, since equation (5.10) is in terms of the incident direction ω_i while our factors are reparameterized using the half-angle or, in general, some alternative parameterization ω_p . Since it is easy to convert between them, there is no difficulty in evaluating equation (5.10). However, our probability distributions γ_p are in terms of the new parameterization, and must be modified to conform to equation (5.10). In particular,

$$\gamma_i(\omega_i | \omega_o) = \gamma_p(\omega_p | \omega_o) \left| \frac{\partial \omega_p}{\partial \omega_i} \right|, \quad (5.13)$$

where the last term is equivalent to the Jacobian for changing variables in multidimensional integration, and converts differential areas in ω_p to those in ω_i . For the half-angle, this function has been computed in many derivations, such as for calculating the Torrance-Sparrow BRDF [101], and is given by

$$\left| \frac{\partial \omega_h}{\partial \omega_i} \right| = \frac{1}{4(\omega_i \cdot \omega_h)}. \quad (5.14)$$

Stratification: The preceding algorithm generates single samples independently, but it can easily be extended to generate stratified samples: we simply stratify each of the individual stages. Because these stages depend only on 1D probability distribution functions, this is accomplished by stratifying the domain of the uniform random variables used in those stages. We have found this to be an effective method of further reducing variance in the generated images.

5.4.3 Results

We now present the results of factoring both analytical and measured BRDFs, describing the accuracy and compactness of our representation. In 5.4.3, we analyze the efficiency of sampling according to this representation.

Factorization

We factored four analytic BRDF models of varied behavior: the Cook-Torrance [20] rough-surface model, an anisotropic Ward model [109], Poulin-Fournier [90] anisotropic reflection from cylinders, and the Hapke-Lommel BRDF [39] with strong back-scattering effects. We also tested three measured BRDFs acquired by Matusik et al. [73]: nickel, plastic and metallic blue. Table 5.1 lists the resolution and parameterization of each factorization along with the normalized mean absolute error (MAE) in the approximation. These errors were computed over a dense set of samples of the 4D domain, independent of the resolution of the factorization. We compare this with a best-fit 2-lobe Lafortune model, except for the Cook-Torrance BRDF, to which we fit a 3-lobe model. For most materials, we chose a half-angle parameterization $\omega_p = \omega_h$, while for the more diffuse models (Poulin-Fournier and Hapke-Lommel), we used the standard parameterization by incident angle $\omega_p = \omega_i$.

We see that in all cases factorization produces an accurate result, in many cases significantly more accurate than fitting an analytic model such as Lafortune. This accuracy in the representation explains the high quality of our sampling algorithm. We further note that fitting a 3-lobe Lafortune model can be unstable, often taking minutes to hours to converge in a nonlinear minimizer, and can require manual tuning to find a good fit. By contrast, our method is automatic, robust, and fast (taking only a few minutes to factor the BRDFs considered in these experiments).

We observe, as previous authors have, that MAE or RMS errors are imperfect measures of the accuracy and visual quality of a BRDF approximation: in practice, the numerical error is dominated by regions such as the specular highlight and grazing angles. To this end, Figure 5.4 shows the appearance of some factored models, as compared to the originals, under point illumination. We see that throughout most of the BRDF the representation accuracy is, in fact, better than the numbers in Table 5.1 would suggest, and the error of our approximation decreases rapidly as more terms are added (Figure 5.5). For the case of measured nickel, note that our representation

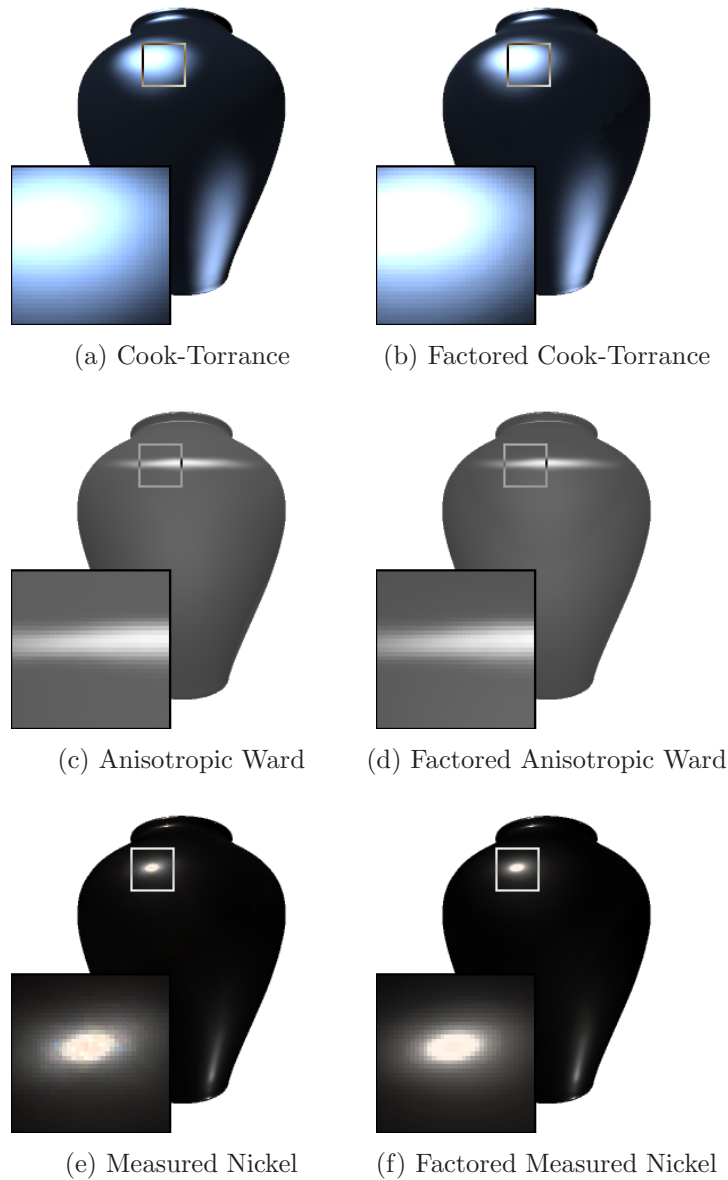


Figure 5.4: Accuracy of the BRDF factorization. The left column shows a vase rendered with (a) a Cook-Torrance BRDF, (c) a Ward anisotropic BRDF and (e) a measured nickel BRDF under direct illumination. (b, d & f) The right column shows the same vase rendered with a factored approximation of the original BRDF. (b) Notice the slight banding effects that appear in the factored highlight of the Cook-Torrance BRDF, which result from the finite sampling resolution along θ_h . (d & f) The factorization actually regularizes some of the measurement noise that appears in the highlight of the measured nickel BRDF.

regularizes some of the measurement noise around the highlight, relative to the original data. We conclude that, for measured data, our representation appears to produce results comparable with measurement error (Matusik et al. observe errors, such as deviation from reciprocity, of 10 – 15% at normal angles, ranging to 60 – 70% at grazing angles [personal communication]).

Selecting the appropriate resolution for the factorization, and the parameterization of the incoming hemisphere is a manual process. In most cases, the analytical formula (for parametric BRDFs) or general appearance (for measured BRDFs) provides enough information for an accurate estimate of how many samples are sufficient and what parameterization is optimal. Theoretically, the number of terms should be proportional to the ranks of the matrices Y and G_j (or, at least, the number of significant eigenvalues of these matrices). In practice, however, we simply increase the number of terms (J and K) until the error in the approximation plateaus. Figure 5.5 shows this convergence process for factorizations of the anisotropic Ward BRDF listed in Table 5.1.

Since our goal is to develop a representation suitable for efficient sampling, rather than a factorization method more accurate than previous approaches, we did not directly compare with previous factorization approaches that cannot be easily sampled. However, we did factor a Poulin-Fournier model with qualitatively comparable parameters to the one listed in [77], and produced a factorization with RMS error comparable with that approach (although not given in Table 5.1, the RMS error for that factorization is 0.094). While this is not the focus of our paper, these results indicate that the benefits of a multi-term nonnegative factorization may be applicable in other areas such as real-time rendering.

Sampling

We next consider the efficiency of importance sampling using our factored representation. For a controlled quantitative comparison, we conducted tests involving images of a sphere (so visibility is not considered), lit by a constant environment map (so complex illumination is not considered). The comparison methods are uniform sampling of a cosine-weighted hemisphere, analytic sampling of either a best-fit multi-lobe Lafortune model [56] or a generalized Blinn-Phong model developed by Ashikhmin and Shirley [5]², and an approach based on explicit tabulation of the BRDF [73]. All methods were stratified.

²Because we are interested in sampling the BRDF, we actually fit the probability distribution they present, $p_h(\mathbf{h})$, plus a diffuse term to the original BRDF.

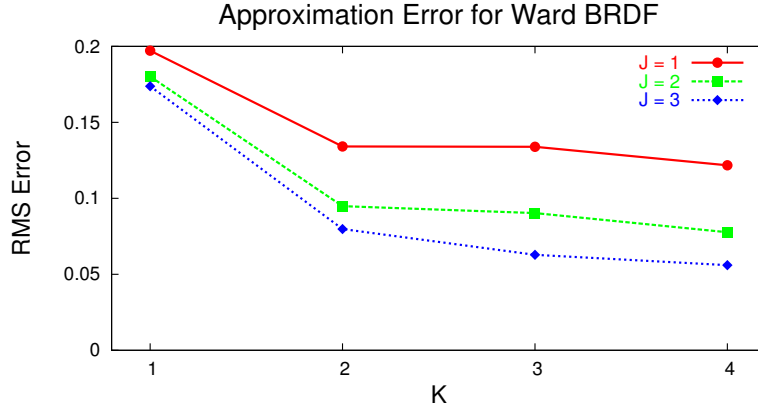


Figure 5.5: Comparison of the RMS approximation error of an anisotropic Ward BRDF, as a function of the number of terms in the factorization. Each line shows a different number of “outer” terms (J) while the number of “inner” terms (K) increases along the x axis. Note the drop in error at $K = 2$: this shows that at least two inner terms are necessary to capture the anisotropic shape of this BRDF.

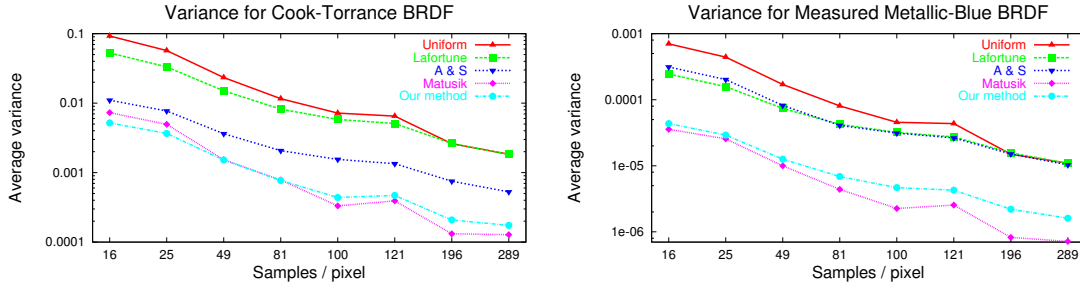


Figure 5.6: Image variance as a function of the number of samples. These plots show the relationship between the average image variance and the number of samples/pixel for all 5 sampling strategies considered in this paper. **Top:** the variance in the image of a sphere with the Cook-Torrance BRDF from Table 1 under constant illumination. **Bottom:** variance in the image of a sphere with a measured metallic-blue BRDF under constant illumination. As expected, the variance converges to 0 as the sample counts increase, confirming that each strategy produces an unbiased estimate. At 100 paths/pixel we see the values for which the factor of improvement is listed in Table 2.

We compared variance (averaged over 50 trials) as a function of the number of samples used (ground truth was taken as the limit with a very large number of samples) for the BRDFs considered in Table 5.1. We verified for all sampling techniques that they were unbiased, and that the image variance decayed approximately as the inverse of the number of samples (Figure 5.6). Table 5.2 reports the ratio of the variance of the comparison methods to our approach with 100 samples—the relative performance with a different sample count would be essentially the same. This is an

Efficiency of BRDF Sampling in Constant Environment

Original BRDF	Improvement relative to:			
	Unif.	Laf.	A&S	Mat.
Cook-Torrance	16.38	13.27	3.53	0.75
Poulin-Fournier	5.86	1.85	6.11	n/a
Hapke-Lommel	3.32	2.14	11.61	1.99
Measured Nickel	306.17	11.52	2.17	1.66
Measured Plastic	157.12	14.40	1.34	18.53
Measured Metallic-Blue	8.88	6.73	6.75	0.44

Table 5.2: Efficiency of importance sampling the BRDF. This table lists the factor of improvement in variance resulting from sampling the BRDF according to our factored representation, compared to four alternative approaches: uniformly sampling a cosine-weighted hemisphere, sampling a best-fit multi-lobe Lafortune model, sampling a best-fit generalized Blinn-Phong model described by Ashikhmin and Shirley, and sampling from a dense set of tabulated CDFs, as described by Matusik. Because variance is linearly proportional to running time, these values can be interpreted as the factor of time, or number of paths, that would be required for the other sampling approaches to reach the same noise level as our representation. All results are for an image of a sphere in a constant environment, and use 100 stratified samples.

appropriate metric, since it directly corresponds to how much longer the alternative approaches would need to run (i.e., how many more samples they would require) to produce the same quality results as our method. The image RMS error corresponds roughly to the standard deviation, which is the square root of the variance.

We see that compared to uniform random sampling, BRDF importance sampling always does at least 5 to 10 times better, and significantly better for shiny materials such as measured nickel. Relative to analytic models, the degree of improvement depends on how closely the analytic model is able to match the BRDF. Lafortune’s model, for instance, is a good fit of the Poulin-Fournier and Hapke-Lommel BRDFs (as seen in Table 5.1). Note that these materials are more diffuse and random sampling also does fairly well on them. However, we always do at least twice as well as sampling based on a Lafortune fit, and for measured materials, and even for the widely known Cook-Torrance model, we do an order of magnitude better. On the other hand, the Ashikhmin-Shirley model represents lobes depending on the half-angle well, and therefore does better than Lafortune at sampling metals and plastics such as the Cook-Torrance, nickel and plastic BRDFs. However, our method is still at least a factor of 2 better, and for many of the materials, we see an improvement by a factor of 5-10.

We also measured the effectiveness of importance sampling the BRDF using the factored representation under complex illumination (Table 5.3). The experimental setup is identical to that

for Table 5.2, except that the sphere was placed in the beach environment³. Because the shape of the integrand of the rendering equation is affected by varying illumination, sampling the BRDF alone will not be as efficient as for constant illumination. As expected, for more diffuse BRDFs (Hapke-Lommel and Poulin-Fournier) we notice that the illumination becomes the dominant factor in the integrand and uniform random sampling is a reasonable strategy. For the more specular BRDFs, however, we still see the benefits of importance sampling the BRDF, and our method decreases the variance by a factor of 2-20 over best-fits of either parametric formula. One example of this is the measured metallic-blue BRDF. The specular peak of this BRDF deviates substantially from the ideal specular direction, and is also not well approximated by a function of θ_h , particularly as the view approaches the horizon. As a result, the best-fit Lafortune and Ashikhmin-Shirley parametric models fail to match the BRDF well in these regions. Although our factored representation parameterizes the incoming hemisphere with respect to the half-angle as well, it can handle small deviations from this direction through the inherent flexibility a numerical factored approximation provides. As a result, our technique samples this BRDF more efficiently than a parametric fit (Figure 5.7). Together, these results indicate the generality and efficacy of our approach for importance sampling compared to fitting a specific analytic model.

The only method competitive with ours is that of Matusik [73]. For a set of fixed view directions, this method computes a 2D CDF over incident directions according to the spherical parameterization of the hemisphere. For accurate results, this approach requires dense sampling along all variables, and does not provide the compactness of our factored representation. In fact, Matusik reports using resolutions of $90 \times 90 \times 180$ for isotropic materials, and acknowledges the infeasibility of this approach for anisotropic materials. Even with these resolutions, however, there are still situations when the closest CDF (i.e. the closest view for which the CDF is tabulated) differs significantly from the actual shape of the BRDF. This is apparent with measured nickel and measured plastic, for which the BRDF has a sharp specular peak. For views near normal incidence, sampling according to the spherical coordinates of the incident direction is sufficient to accurately capture the shape of the BRDF. Near grazing angles, however, the 2D CDF for the nearest view often varies significantly from the actual shape of the BRDF, degrading the sampling efficiency in these regions (Figure 5.8). Our factored representation, on the other hand, avoids this situation

³<http://www.debevec.org/Probes/>

Efficiency of BRDF Sampling in Beach Environment

Original BRDF	Improvement relative to:			
	Unif.	Laf.	A&S	Mat.
Cook-Torrance	25.79	10.28	2.23	0.75
Poulin-Fournier	1.40	1.09	1.53	n/a
Hapke-Lommel	0.89	1.61	1.87	1.00
Measured Nickel	572.76	3.45	2.17	4.80
Measured Plastic	381.94	21.60	1.67	55.64
Measured Metallic-Blue	9.17	5.96	5.71	0.55

Table 5.3: Importance sampling the BRDF under complex illumination. This table presents the factor of improvement of our sampling strategy compared to alternative approaches when rendering a particular BRDF in the beach environment. Because the illumination contributes to the shape of the integrand in the rendering equation, sampling according to the BRDF alone will be less efficient than when the illumination is constant. Although our factored representation still outperforms the alternative sampling strategies by a factor of 2-20, these results suggest the potential desirability of combining environment and BRDF sampling.

through a better parameterization of the hemisphere and a more continuous approximation of the BRDF over all views. Moreover, our representation supports anisotropic reflection and is more compact. For the BRDFs presented in this paper, the complete factored representation requires roughly 200KB as compared to the 60MB required to store the samples of the 3D BRDF along with the pre-computed 2D CDFs required for the approach of Matusik.

In generating a sample using our approach, the dominant cost is that of inverting three 1D CDFs using a binary search. This makes our approach reasonably fast, comparable with analytically drawing a sample according to the Lafortune and Phong sampling algorithms. It is somewhat slower than the simpler random sampling, and almost identical to the approach of Matusik, which also inverts a pair of 1D CDFs. In practice, all of these times are small compared to the cost of propagating a sample or tracing a ray for global illumination, and hence the number of samples (and the results in Table 5.2) corresponds closely to actual running time.

Global Illumination

We also rendered a complex scene with global illumination using a path tracer (Figure 5.9). In this case, the incident illumination and visibility are unknown and, consequently, importance sampling the BRDF is the only reasonable strategy (i.e., environment sampling is not possible). We used our factored representation to sample all five BRDFs in the scene and to represent the three measured BRDFs. We compare our results with those of a system using best-fit Lafortune models

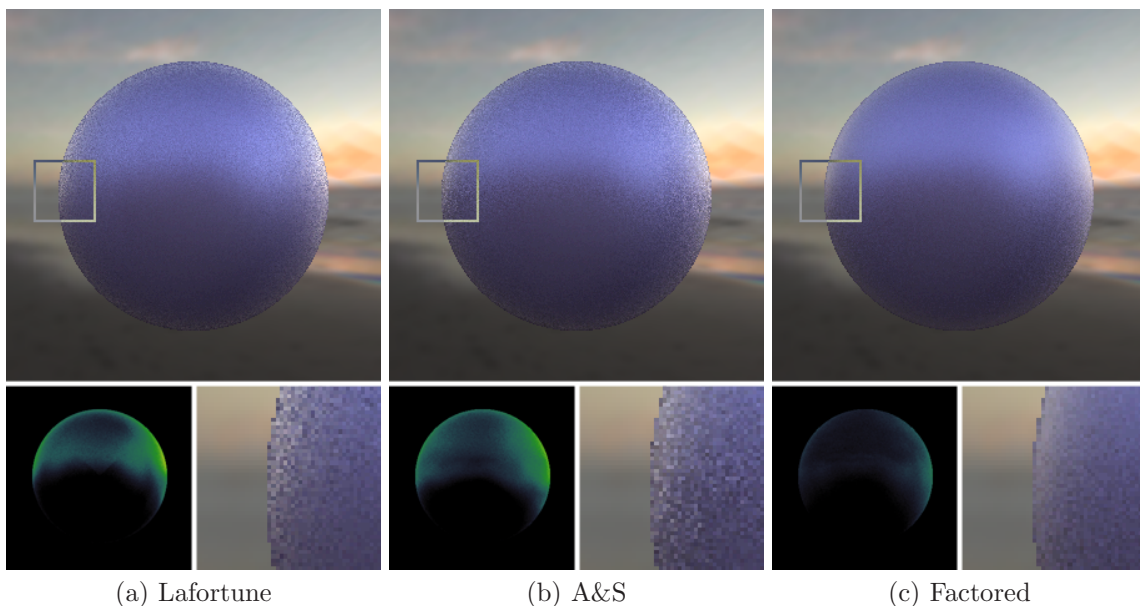


Figure 5.7: Importance sampling a BRDF according to best-fit parametric models and our factored representation (cf. last row of Table 5.3). These images show a metallic-blue sphere in the beach environment, rendered with 100 samples generated according to (a) a best-fit 2-lobe Lafortune model, (b) a best-fit Ashikhmin-Shirley model, and (c) our factored representation. We show both a variance plot on a logarithmic scale and a closeup at a region where the view approaches the horizon. In this part of its domain, the BRDF has a shape that is difficult to fit with either of the parametric models, and our factored representation allows more efficient sampling.

to sample the different BRDFs. We present rendered images at equal time (300 paths/pixel for both sampling strategies) and equal quality (1200 paths/pixel for Lafortune sampling) along with false-color visualizations of the variance in the scene on a logarithmic scale and several magnified views showing different BRDFs in the scene. Clearly, different regions of the scene converge at different rates, but our method is roughly 4-5 times more efficient overall and an order of magnitude more efficient on difficult BRDFs such as the plastic handle. This example highlights the usefulness of a general approach to both representing and importance sampling BRDFs.

5.4.4 Limitations

One limitation of our representation is that it does not obey reciprocity. In practice, however, we factor the product of the BRDF and the cosine term, $\rho(\omega_i, \omega_o)(\omega_i \cdot n)$, so the original function is not reciprocal either.

As mentioned previously, another limitation of our system is that we cannot handle mixed

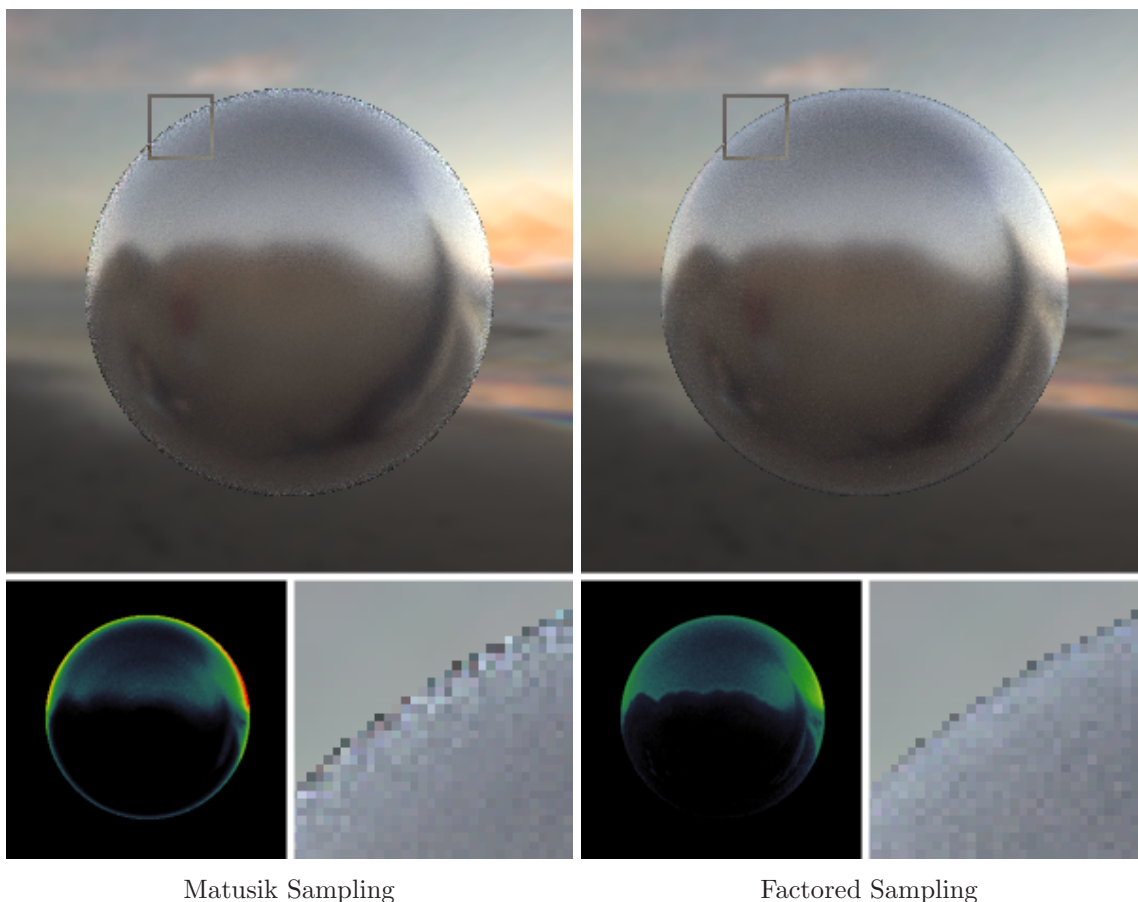


Figure 5.8: Sampling measured nickel with a dense set of 2D CDFs, as described by Matusik et. al. and using our factored representation (cf. fourth row of Table 5.3). For such shiny BRDFs, computing a fixed set of 2D CDFs can still cause problems for regions of the domain for which the nearest pre-computed CDF of a particular view poorly matches the actual BRDF. Our factored representation, on the other hand, gains better continuity through an appropriate parameterization and approximation, resulting in more efficient importance sampling throughout the domain.

parameterizations of the BRDF. We would like to extend our representation to allow each term in the approximation to use a different parameterization of the incoming hemisphere. This would greatly improve the accuracy of our approximation in cases where the BRDF exhibits several different types of scattering (e.g. side, backward and forward) at the same time. Another limitation of our system is that we cannot use measured BRDF data directly. Instead, we rely on a simple reconstruction of the scattered data (e.g. Zernike polynomials, bi-cubic polynomials, McCool’s factorization method [77]) and evaluate this reconstruction at regular intervals over the 4D domain in order to seed the initial data matrix F .

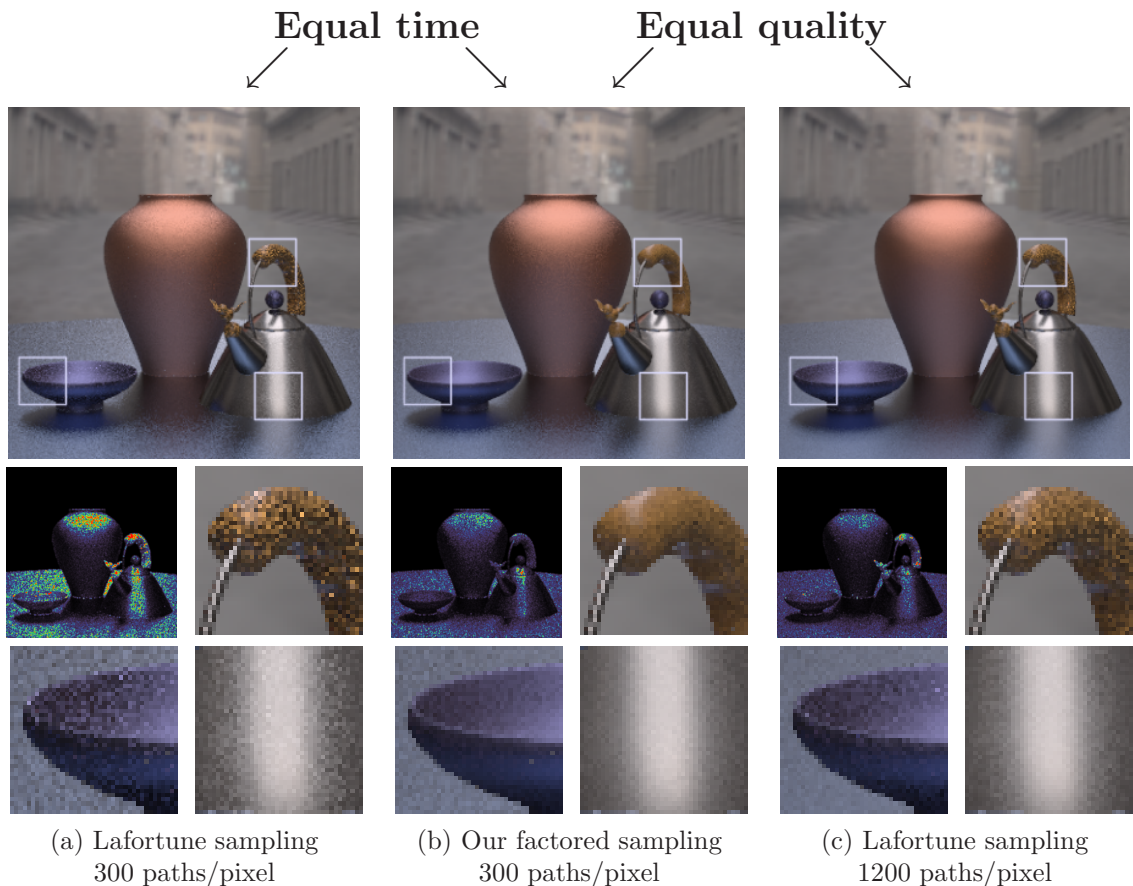


Figure 5.9: This scene was rendered using a path tracer for global illumination. (a) We generated 300 stratified importance samples of the local hemisphere using a best-fit Lafortune model for each of the 5 BRDFs in the scene. (b) We sampled the incoming hemisphere using our factored representation of each BRDF. (c) Lafortune sampling with 1200 samples. The bottom row shows a false-color variance plot and closeups of some regions. On the whole, we see that our method is approximately four times more efficient than Lafortune sampling, and substantially better for difficult BRDFs such as the plastic teapot handle.

5.5 Sampling n-Dimensional Measured Functions

While several specific representations of measured environment maps and BRDFs do allow direct sampling, there is still no single representation that is appropriate for general multidimensional measured functions. As higher-dimensional datasets find their way into rendered scenes (e.g. light fields, reflectance fields, etc.), a general method for sampling them will become more important. Moreover, for the specific case of environment maps, existing representations do not account for the local orientation of the surface (i.e. the *cosine term* in the rendering equation). This property

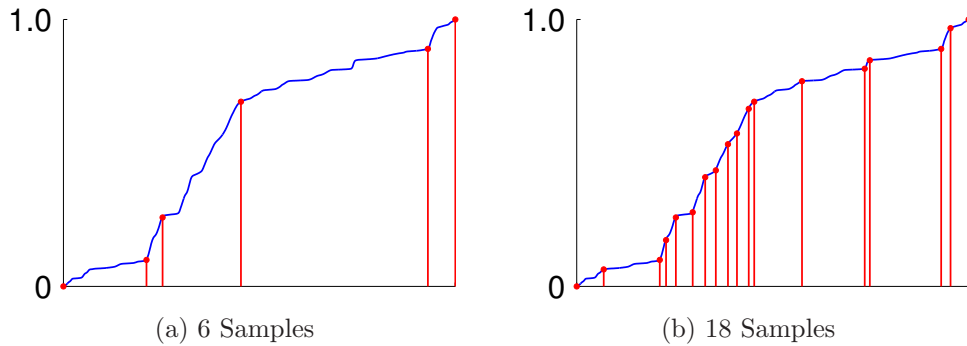


Figure 5.10: We represent a 1D CDF with a set of non-uniformly spaced samples of the original function. This results in a more compact yet accurate approximation of the original function than uniform spacing would allow. In addition, the final CDF maintains many key properties necessary for unbiased multiple importance sampling.

can limit the effectiveness of environment map sampling in reducing variance for many scenes. Another important drawback of some existing environment map representations is that they are not readily incorporated into a multiple importance sampling framework [108].

In this section, we apply a curve approximation algorithm to the task of compressing multi-dimensional tabular Cumulative Distribution Functions (CDFs) derived from measured datasets. Assume we have a 1D CDF, $P(x)$, sampled uniformly in x . In order to compress this function, we lift the restriction that the samples must be uniformly spaced, as shown in Figure 5.10. We use the Douglas-Peucker greedy algorithm for polygonal approximation of 2D curves [27, 45] to compute the location of these *adaptive* samples. We further extend this algorithm to represent multidimensional CDFs. To accomplish this, we compute marginal 1D CDFs in each dimension by summing the energy contained across the orthogonal dimensions. Each of these 1D CDFs is represented by non-uniformly spaced samples and the resulting set of these “cascading CDFs” approximates the original high-dimensional distribution. There are several benefits of using this adaptive numerical representation:

- Allowing placement of non-uniformly spaced samples reduces the number that must be stored to accurately represent the original CDF. This is especially true for multidimensional distributions because the storage grows exponentially with the number of dimensions. Significant reduction is also achieved for common “peaky” distributions, for which many methods require $O(n)$ storage.

- Generating directions according to the distribution, accomplished using numerical inversion of the CDF, simply requires a binary search over the sorted samples of $P(x)$. This is essentially the same algorithm as is used for uniformly sampled CDFs, but with the position of each sample along the domain stored explicitly.
- Storing a “cascading set” of conditional 1D CDFs, each represented by non-uniformly spaced samples of the original functions, promotes a direct implementation of unbiased stratified importance sampling. This results from the fact that each dimension can be sampled independently.
- The probability of a sample not drawn from the CDF itself can be efficiently computed from the final representation. This property is critical for combining distributions with standard multiple importance sampling algorithms.

To demonstrate the benefit of our adaptive representation, we present a novel algorithm for sampling measured environment maps in an orientation-dependent manner. This is accomplished by sampling the 4D function that results from modulating an environment map with the horizon-clipped *cosine term* in the rendering equation. This algorithm is more efficient than existing techniques that sample only a single spherical distribution. Lastly, we show how our adaptive representation can be used within a multiple importance sampling framework.

5.5.1 Related Work

Monte Carlo importance sampling has a long history in Computer Graphics [107]. For stratified sampling from 2D CDFs on a manifold (in the practical examples of this paper, the manifold is a sphere or hemisphere), Arvo [3] describes a particular recipe when an analytic description of the function is available, with analytic sampling strategies available in some cases [2]. When dealing with measured illumination or reflectance data, as in this paper, nonparametric or “numerical” CDFs are unavoidable, and it is important to compress them.

One possible approach is to use general function compression methods, such as wavelets or Gaussian mixture models. Wavelets have been previously used for importance sampling of BRDFs [17, 57]. However, the computational cost for generating a sample can be significant, especially if non-Haar wavelets are used (as is necessary to avoid many kinds of blocking artifacts). Additionally,

the implementation for multidimensional functions such as measured illumination and BRDFs can be difficult, requiring sparse wavelet data structures and a hexa-decary search.

A second approach to compact CDF representation that has been applied for BRDFs is factorization [60]. This method takes advantage of the structure of the BRDF to factor it into 1D and 2D pieces, thereby reducing dimensionality while still allowing for accurate representation and efficient importance sampling. The technique proposed here differs in considering compression of general tabulated CDFs, and is not limited to BRDF sampling. Additionally, the CDF compression considered here is independent of dimension and orthogonal to any factorizations of the input data. It could therefore be applied equally well to methods that use a full tabular BRDF representation and sampling scheme (as shown in this paper), or to lower dimensional components.

Another specialized CDF compression approach, which has been applied to environment maps, is to decompose the function into piece-wise constant Voronoi or Penrose regions on the sphere [1, 55, 85]. As compared to our method, these techniques offer more optimal stratification, but do not directly extend to multidimensional distributions. Another drawback of these representations is that they are difficult to use with standard multiple importance sampling algorithms that require computing the probability of a direction generated from a separate distribution. Lastly, these representations ignore the fact that half of the environment is always clipped against the horizon of the surface and that the illumination is scaled by a cosine term (Figure 5.19).

5.5.2 Background

We seek to generate samples according to some Probability Density Function (PDF), p , which by definition is non-negative and normalized (i.e., it integrates to 1). We accomplish this using the *inversion method*, which pre-computes the corresponding Cumulative Distribution Function (CDF)

$$P(x) = \int_{-\infty}^x p(x') dx' \tag{5.15}$$

and evaluates its inverse $P^{-1}(\zeta)$ at locations given by a uniformly distributed random variable $\zeta \in [0, 1]$.

We are interested in the case of a PDF specified numerically: in 1D, we assume that we are given probabilities p_i at locations x_i . We precompute the corresponding CDF values P_i and, at

run-time, invert the CDF by performing a binary search for the interval $[P_i, P_{i+1}]$ that contains the random value ζ . Note that this search is required whether or not the x_i are spaced uniformly. This will be the key property used by our representation: we can represent many functions more efficiently by having non-uniformly spaced x_i without increasing the run-time cost of importance sampling.

In 2D, the situation is more complex. We must first decompose the 2D PDF $p(x, y)$ into two pieces, one dependent only on x and the other on y :

$$\tilde{p}(x) = \int_{-\infty}^{\infty} p(x, y) dy \quad (5.16)$$

$$p(y|x) = \frac{p(x, y)}{\tilde{p}(x)} \quad (5.17)$$

The numerical representation then consists of a discretized version of \tilde{p} , given as samples \tilde{p}_i at locations x_i , together with a *collection* of discretized conditional probability functions $p_i(y|x_i)$. This technique generalizes naturally to any number of dimensions, producing a “cascading set” of CDFs where a value in each dimension is generated sequentially using the appropriate 1D marginal CDF at each step [95]. As an important special case, we note that functions on a sphere may be represented using the parameterization $p(z, \phi)$, where the usual change of variables $z = \cos \theta$ is used to normalize for the area measure $d\omega = \sin \theta d\theta d\phi$.

If the CDFs are uniformly sampled along their domain, the total size of this set of CDFs will be slightly larger than the size of the original function. In Computer Graphics, it is often the case that these functions can be both high-dimensional and measured at high resolutions. Consequently, the combined size of the resulting 1D CDFs can quickly become prohibitively large. This motivates our investigation into efficient techniques for compressing these sampled functions without compromising their accuracy or utility.

5.5.3 Numerical CDF Compression

We use polygonal curve approximation algorithms to compress a densely sampled CDF by representing it with a reduced set of non-uniformly spaced samples selected to minimize the reconstruction error.

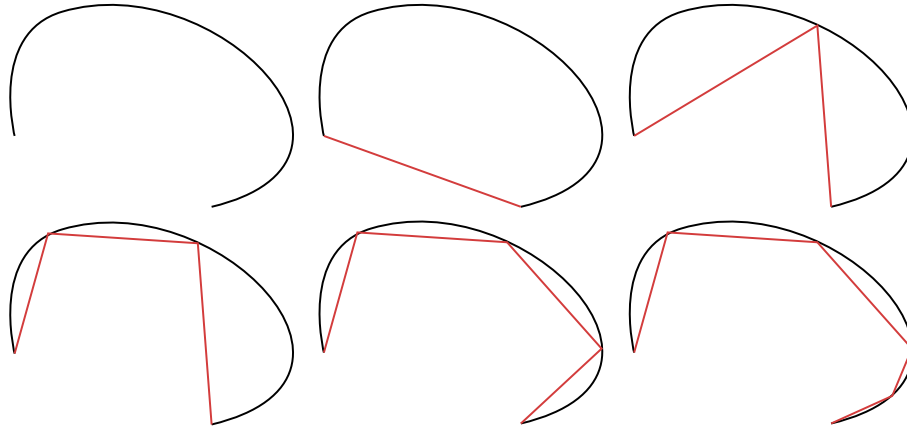


Figure 5.11: The Douglas-Peucker algorithm greedily computes a polyline approximation of a smooth 2D curve. It works by inserting the next sample in the approximation at the point of maximum deviation between the (black) original curve and the (red) current polyline approximation.

5.5.4 Polygonal Curve Approximation

With early roots in cartography, several efficient algorithms have been developed for computing polygonal approximations of digitized curves. Polygonal approximation algorithms take as input a curve represented by an N -segment polyline and produce an M -segment polyline with vertices so as to minimize the error between the two (typically $M \ll N$).

Although algorithms exist that output the optimal solution [14, 35, 15], we instead use the Douglas-Peucker [27, 45] greedy algorithm because of its simplicity and speed. It has also been shown that these greedy algorithms typically produce results within 80% accuracy of the optimal solution [93].

The Douglas-Peucker curve approximation algorithm works by iteratively selecting the vertex furthest from the current polyline as the next vertex to insert into the approximation (Figure 5.11). Initially only the endpoints of the curve are selected, and the algorithm iteratively updates this approximation until either an error threshold is reached or some maximum number of vertices have been used. For curves derived from numerical CDFs, we found this algorithm sufficient for producing near-optimal approximations with few samples.

5.5.5 Applying Curve Approximation to CDFs

There are several ways of applying the above curve approximation algorithms to the task of representing numerical probability functions. First, we can apply them to yield a piecewise linear approximation of the CDF, which is equivalent to a *piecewise constant* approximation of the corresponding PDF. Because the Douglas-Peucker algorithm, when applied to the CDF, is guaranteed to yield a nondecreasing function with a range of $[0..1]$, the resulting approximation may be used directly as a CDF and differentiated to find the corresponding PDF.

A second way of using curve approximation algorithms is to apply them directly to the PDF to obtain a *piecewise linear* approximation (which implies a piecewise quadratic CDF). In this case, the resulting approximation is not guaranteed to integrate to one, and must be normalized before it can be used as a probability function. Figure 5.12, bottom, compares these two strategies on a (relatively smooth) function: note that the two approaches result in samples being placed at different locations in the domain. For comparison, Figure 5.12, top, shows piecewise constant and piecewise linear approximations using uniform sample spacing.

One important difference between uniformly-sampled and adaptively-sampled CDFs is the cost of reconstructing the value of the approximated function (i.e., evaluating the probability) at an arbitrary position. This property is necessary for combining several distributions using multiple importance sampling algorithms [108]. When the samples are uniformly spaced the cost is $O(1)$, whereas adaptively sampled representations require $O(\log N)$ time (here N refers to the number of non-uniform samples). This increased complexity results from having to perform a binary search over the values of the function sorted along the domain to find the desired interval. Because adaptive representations provide such large compression rates, however, N is typically small enough to make this added cost insignificant in practice. In addition, the time complexity of *generating* a sample (as opposed to evaluating the probability) remains the same at $O(\log N)$ in both cases.

In our experiments, we always used a piecewise constant approximation of the PDF (i.e. piecewise linear CDF). Although this results in a slightly larger representation, in our experience this drawback was outweighed by the simpler implementation required for sampling a piecewise constant approximation.

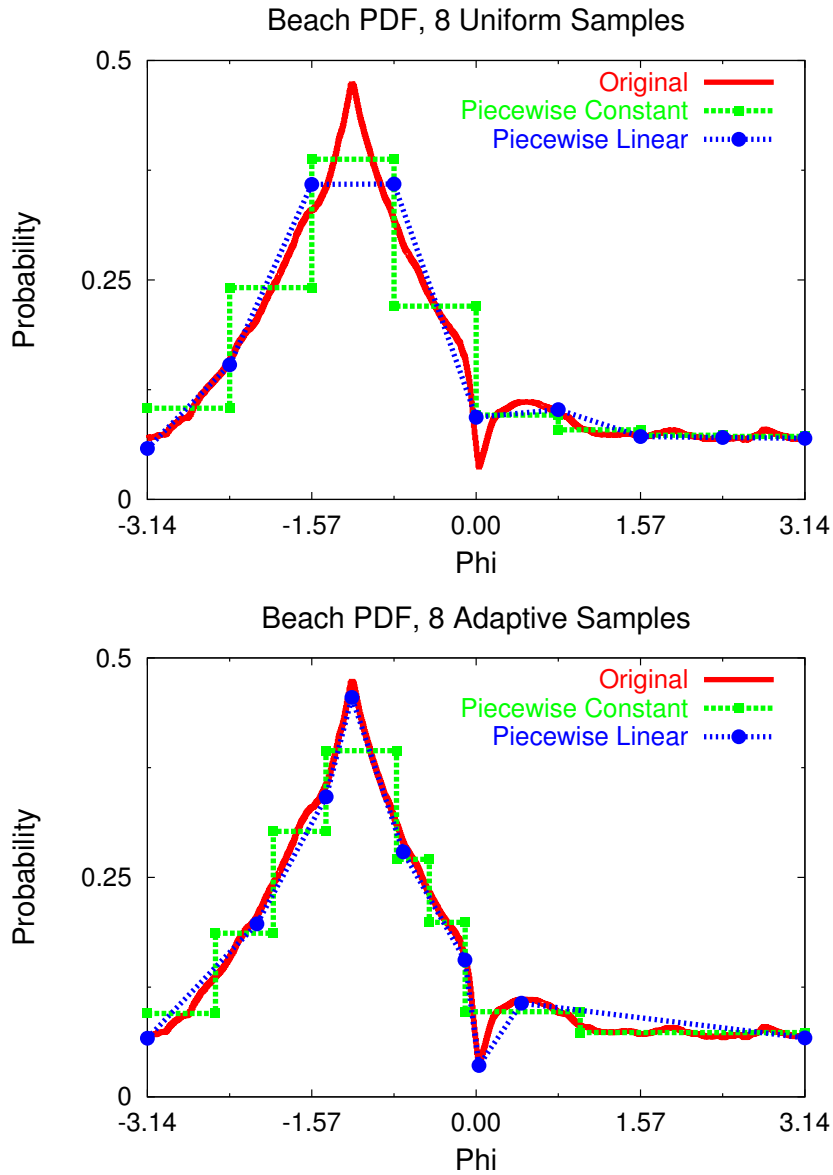


Figure 5.12: A probability density function (corresponding to environment map in Figure 5.16) and its piecewise linear and piecewise constant approximations with 8 samples placed uniformly (top) and computed by the Douglas-Peucker algorithm (bottom). The piecewise constant approximation was computed by running Douglas-Peucker on the integral of the PDF (i.e. the CDF). Note that, for this relatively smooth function, the piecewise linear approximation is closer to the original.

5.5.6 Multidimensional CDFs: The Cascading Douglas-Peucker Algorithm

In the previous section, we discussed how to apply curve approximation algorithms to the task of efficiently representing numerical 1D CDFs. In this section, we extend these ideas to accommodate distributions of higher dimension. For the sake of explanation, we first restrict our discussion to the 2D case and provide an example with synthetic data in Figure 5.13. Extending these techniques to higher dimensions is straightforward and briefly discussed at the end of the section.

Recall that we can convert any 2D distribution (Figure 5.13 top) into a single marginal CDF plus a set of conditional CDFs according to Equations 5.15, 5.16 and 5.17. In order to generate (x, y) pairs with probability proportional to the magnitude of the original function, we first generate a value of x from the marginal CDF $\tilde{P}(x)$ (Figure 5.13 bottom, red curve) and then generate a value of y from the corresponding conditional CDF $P(y|x)$ (not shown in Figure 5.13).

As described previously, we use the Douglas-Peucker algorithm to select a set of non-uniformly spaced samples that accurately represent the marginal 1D CDF, $\tilde{P}(x)$. For the example in Figure 5.13, we can perfectly approximate the marginal CDF with samples at the endpoints A and E and at internal locations B and D . Next, we would compute a set of conditional CDFs, $P(y|x)$; one for each of these regions in x (e.g. in Figure 5.13 these regions are AB , BD and DE). Each conditional CDF is the average across its associated range:

$$p(y|x_i) = \frac{1}{x_i - x_{i-1}} \int_{x_{i-1}}^{x_i} \frac{p(x', y)}{\tilde{p}(x')} dx'. \quad (5.18)$$

For all the examples in this paper on measured data, building a cascading set of CDFs according to Equation 5.18 was sufficient for accurately approximating the original distribution. However, there are potential situations where this approach alone ignores error introduced by approximating the distribution of energy within a region with a single CDF. Figure 5.13 illustrates such a situation. In this case, the distribution of energy within the region BD would be poorly approximated by a single conditional distribution because the two area light sources are at different heights. In order to address this issue, we must also consider the gradient in the x -direction of the original distribution:

$$g(x) = \int_{-\infty}^{\infty} \left| \frac{\partial p(x, y)}{\partial x} \right| dy. \quad (5.19)$$

When the function $g(x)$ is large this indicates locations in x where the conditional CDFs, $P(y|x)$, would *not* be well approximated by a single distribution. Therefore, after our first application of the Douglas-Peucker algorithm to represent $\tilde{P}(x)$, we add additional samples according to this gradient function. Specifically, we can compute a numerical CDF from $g(x)$ and generate a fixed number of stratified samples along the domain (e.g. the x -axis) such that they occur at locations where this function is large. Adding samples according to the gradient guarantees that both $\tilde{P}(x)$ is well represented by the non-uniformly spaced samples and that the conditional CDFs computed for each region, $P(y|x_i)$, well approximate the variation present in the orthogonal dimensions. In the example in Figure 5.13, we additionally sample the marginal CDF at location C , separating the 2D distribution into a total of four regions (AB , BC , CD and DE), where each region is now well approximated by a single CDF.

Lastly, we extend this sampling algorithm to arbitrary dimensions by simply expanding the integrals over the entire range of free variables (as opposed to just y for the 2D example considered above). For an N -dimensional distribution, $p(x_1, x_2, \dots, x_N)$, both the marginal and conditional CDFs are proportional to the integral across the remaining free variables (note: we omit the normalization constant for clarity):

$$p(x_i|x_1 \dots x_{i-1}) \propto \int_{-\infty}^{\infty} dx_{i+1} \dots \int_{-\infty}^{\infty} dx_N p(x_1 \dots x_N),$$

and the gradient function would be computed similarly:

$$g(x_i|x_1 \dots x_{i-1}) = \int_{-\infty}^{\infty} dx_{i+1} \dots \int_{-\infty}^{\infty} dx_N \left| \frac{\partial p(x_1 \dots x_N)}{\partial x_i} \right|.$$

5.5.7 Evaluation of Algorithm

In general, global illumination algorithms perform numerical integration of the rendering equation:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega_{2\pi}} d\omega_i L_i(x, \omega_i) \rho(x, \omega_i, \omega_o) (\omega_i \cdot \mathbf{n}).$$

A common approach to estimating the value of this integral is to perform Monte Carlo integration over the space of incoming directions. Because the entire integrand is usually not known a

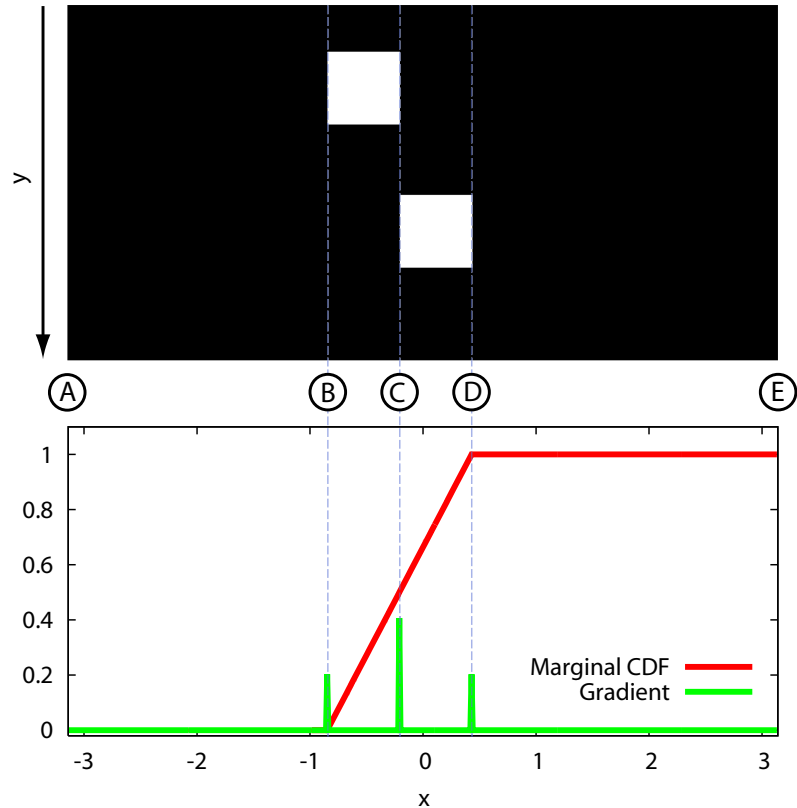


Figure 5.13: Efficiently approximating multi-dimensional distributions requires computing a cascading set of 1D marginal and conditional CDFs. Here we show (top) a synthetic environment map that contains only two equal sized area light sources. We compute (bottom, red curve) a marginal CDF in x by summing the total energy across y . We also consider (bottom, green curve) the average gradient in the x -direction. We place non-uniformly spaced samples according to the Douglas-Peucker algorithm at positions A , B , D and E and any additional points where the gradient function is large (i.e. at position C).

priori, a reasonable strategy is to sample according to the terms that are known. For example, if the incident illumination L_i is represented by an environment map, we may perform environment sampling. BRDF sampling, on the other hand, generates samples according to either ρ itself or $\rho \cdot (\omega_i \cdot \mathbf{n})$. Although algorithms exist for sampling BRDFs and environment maps, these functions provide a convenient platform to evaluate our representation. Moreover, our approach has several desirable properties that these existing techniques lack. These enable novel applications that we present in Section 5.5.10.

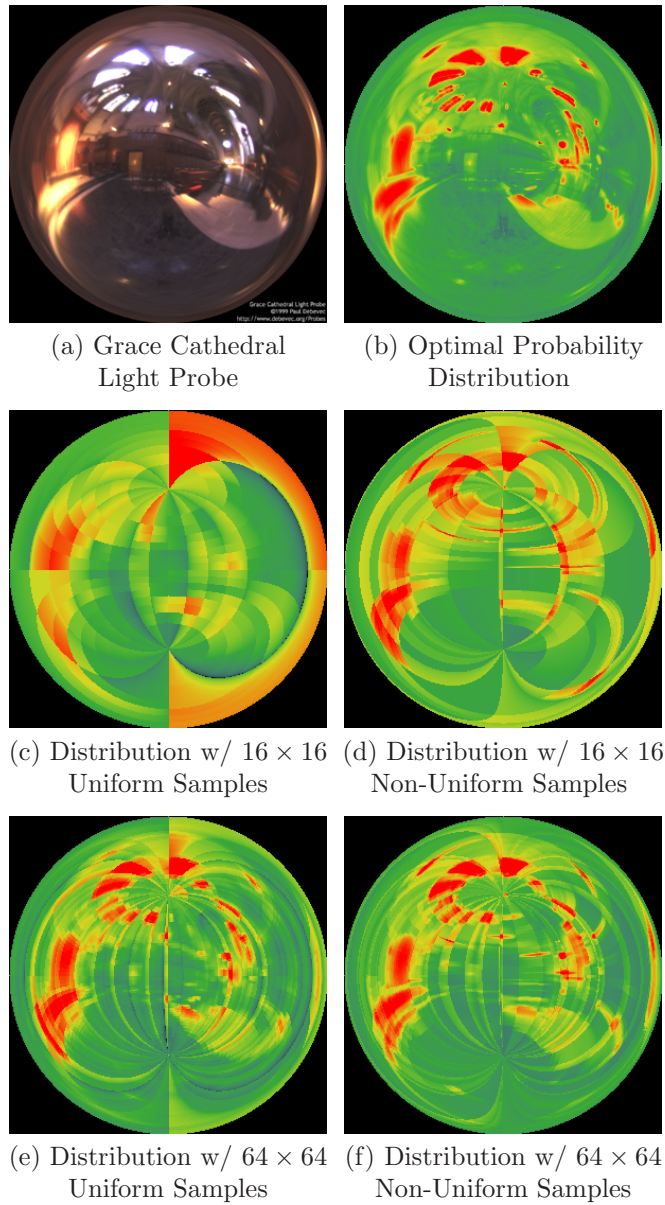


Figure 5.14: False-color visualizations of spherical probability density functions on a logarithmic scale (red = largest probability, green = smallest probability). Directions are mapped to the unit circle according to the parameterization used by Debevec [22]. (a) A measured environment map of the inside of Grace Cathedral. (b) The probability density resulting from using a numerically tabulated CDF sampled uniformly at the same resolution of the original map. The probability distribution of numerical CDFs computed from (c) 16×16 uniform samples (d) 16×16 non-uniform samples (e) 64×64 uniform samples and (f) 64×64 non-uniform samples.

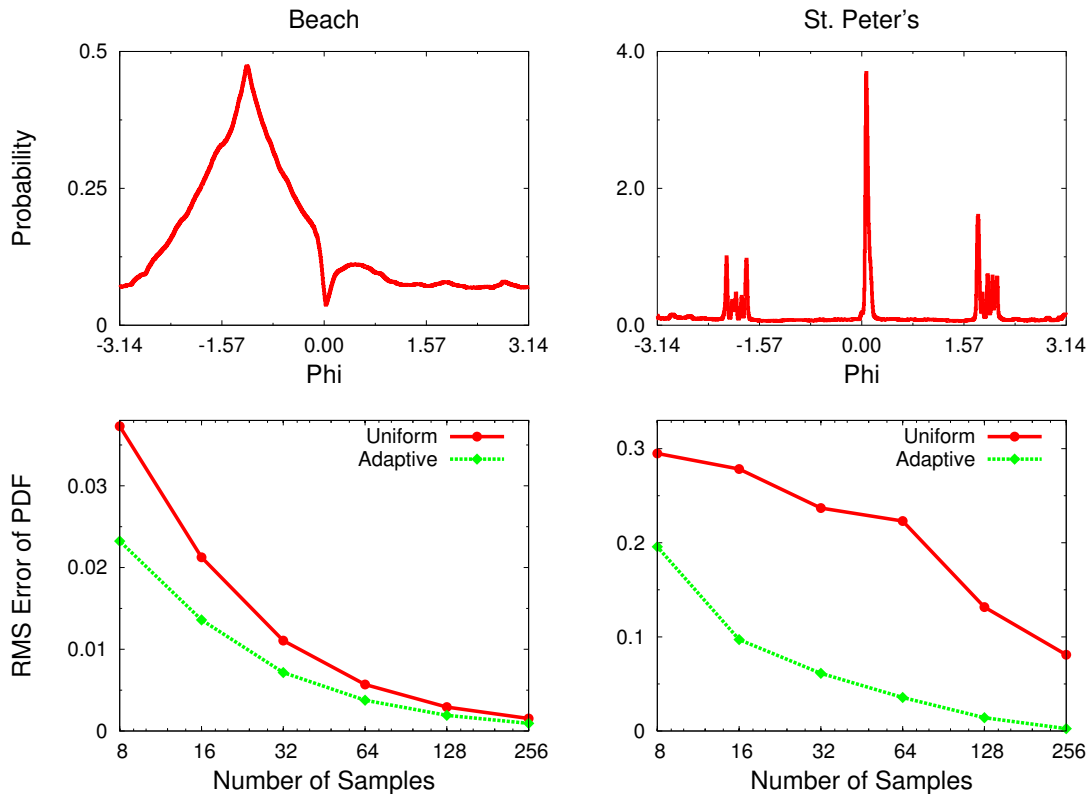


Figure 5.15: Two different probability distribution functions and the RMS error in approximating them using different numbers of points and different sampling strategies. The different sampling algorithms use either uniform or adaptive placement of sample locations.

5.5.8 Environment Map Sampling

One direct approach for generating samples according to a measured environment map [22], is simply to compute a family of numerical 1D CDFs directly from the 2D spherical function [88]. Recall that one CDF will quantify the distribution along ϕ , $\tilde{P}(\phi)$ and a set of 1D CDFs will control the distribution of samples along θ at each sampled location of ϕ , $P_i(\theta|\phi_i)$. These are derived from the *intensity* of each pixel in the environment map (i.e. weighted average of color values) using the method described in Section 5.5.2.

If the resolution of these CDFs is proportional to that of the environment map (as it should be to avoid aliasing) this representation will be slightly larger than the original measured dataset itself. Therefore, there is significant opportunity for compression using our adaptive representation. Figure 5.14 shows false-color visualizations on a logarithmic scale of the full-resolution 1000×1000

$(\theta \times \phi)$ PDF of the Grace Cathedral environment (<http://www.debevec.org/Probes/>), together with 16×16 and 64×64 approximations using both uniform and non-uniform sample selection. As compared to uniform sampling, adaptive sample placement results in a significantly more accurate approximation of the original distribution.

Figure 5.15 compares the error of our adaptive numerical representation with uniform sample placement on two distributions with qualitatively different behaviors. The upper graphs show a single scanline (i.e. varying ϕ for a constant θ) of the environment map, while the graphs at bottom plot the RMS error of the approximation as a function of the number of samples used (note that the horizontal axis is logarithmic). At left, we consider a relatively smooth function. In this case, the gain from nonuniform placement of samples is relatively modest. At right, we show a “peakier” function that is easier to compress with nonuniform sample placement. In this example, our adaptive representation reduces the number of samples required at equal approximation error by a factor of 16 compared to uniform downsampling.

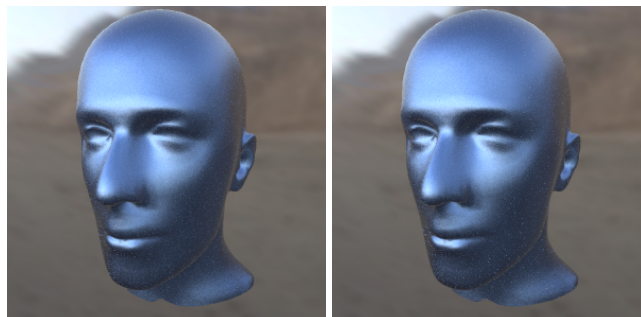
5.5.9 BRDF Sampling

The BRDF gives the ratio of reflected light to incident light for every pair of incoming and outgoing directions: $\rho(\omega_o, \omega_i)$. For glossy materials, it is advantageous to sample the environment according to the distribution of energy in the BRDF. Because this is a 4D function (3D if the BRDF is isotropic), a tabular representation at a modest resolution would still be quite large. Consequently, we apply our adaptive representation to the task of efficiently storing numerical CDFs derived from measured BRDFs.

We compared the size and accuracy of this representation with a standard approach of pre-computing the CDFs at their full resolution [73] for the same set of viewing directions (Figure 5.16). We evaluated the efficiency of generating samples using an adaptive numerical CDF computed from two measured BRDFs [74]: *nickel* and *metallic-blue*.

For these results, we first reparameterized the BRDF into a view/half-angle frame in order to maximize the redundancy among *slices* of the function giving greater opportunity for compression [60]. Each uniformly-sampled CDF had a resolution of $32 \times 16 \times 256 \times 32$ ($\theta_o \times \phi_o \times \theta_h \times \phi_h$) and occupied 65MB. Here, θ_h and ϕ_h are the elevation and azimuthal angles of the half-angle vector respectively. To compute the corresponding adaptive numerical CDFs required, on average, roughly

Cook-Torrance BRDF



65MB

1.5MB

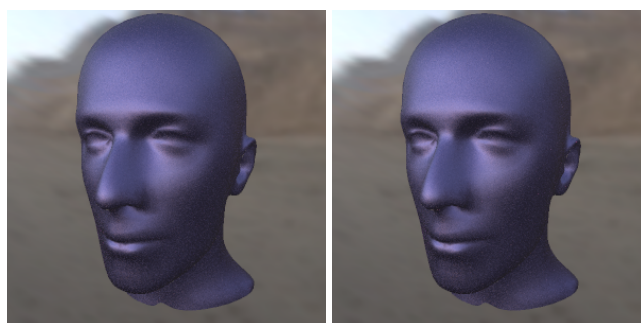
Measured Nickel BRDF



Original (65MB)

Compressed (3.9MB)

Measured Metallic-Blue BRDF



Original (65MB)

Compressed (2.3MB)

Figure 5.16: BRDF importance sampling with adaptive numerical CDFs. We compare the variance in images rendered using a path tracer that generates samples using the fully tabulated CDF and the adaptive CDF. In all cases we estimate the radiance with 80 paths/pixel. We also list the total size of the probability representation below each image.

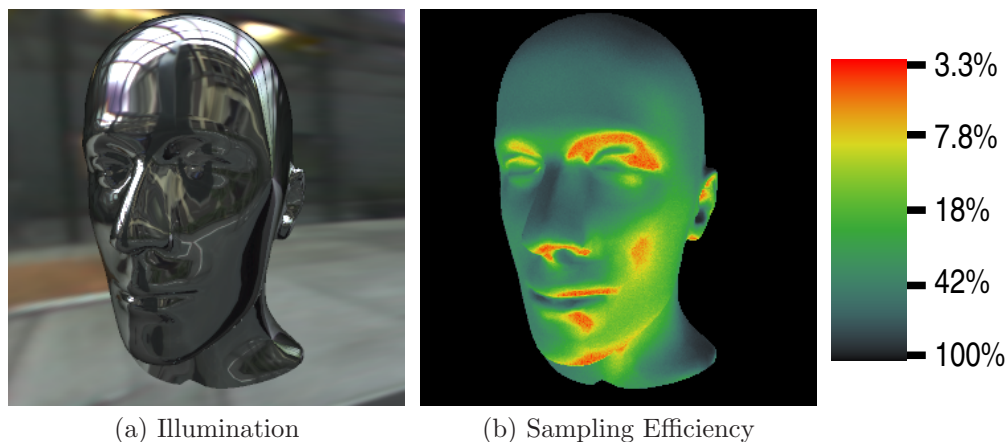


Figure 5.17: For some orientations and lighting, sampling from a single distribution will be inefficient because most of the energy is occluded by the horizon. (a) We examine this inefficiency for an example in which the majority of light is above and slightly behind the object being rendered. (b) A false-color image visualizes the percentage of samples that will be generated above the horizon and, consequently, make a positive contribution to the radiance estimate at that pixel. In many regions of this image only 5% of the samples are generated above the horizon.

30 samples in θ_h and 10 samples in ϕ_h . Using the Douglas-Peucker algorithm, these adaptive samples were selected from an initial set of 2048×1024 ($\theta_h \times \phi_h$) uniformly-spaced samples—a resolution prohibitively expensive for the fully tabulated CDFs. It required 20 minutes of processing time to compute the adaptive representation for each BRDF.

We found that for these BRDFs, sampling the adaptive numerical CDF is nearly as efficient as the full tabular approach. For the measured nickel BRDF, the compact CDF actually produces slightly less variance in the image because the uniform sampling was not sufficiently dense to capture the very sharp highlight.

5.5.10 Novel Applications

In this section we present a new algorithm for sampling illumination from an environment map according to the local orientation of the surface. Additionally, we demonstrate how our representation facilitates multiple importance sampling of both illumination and the BRDF.

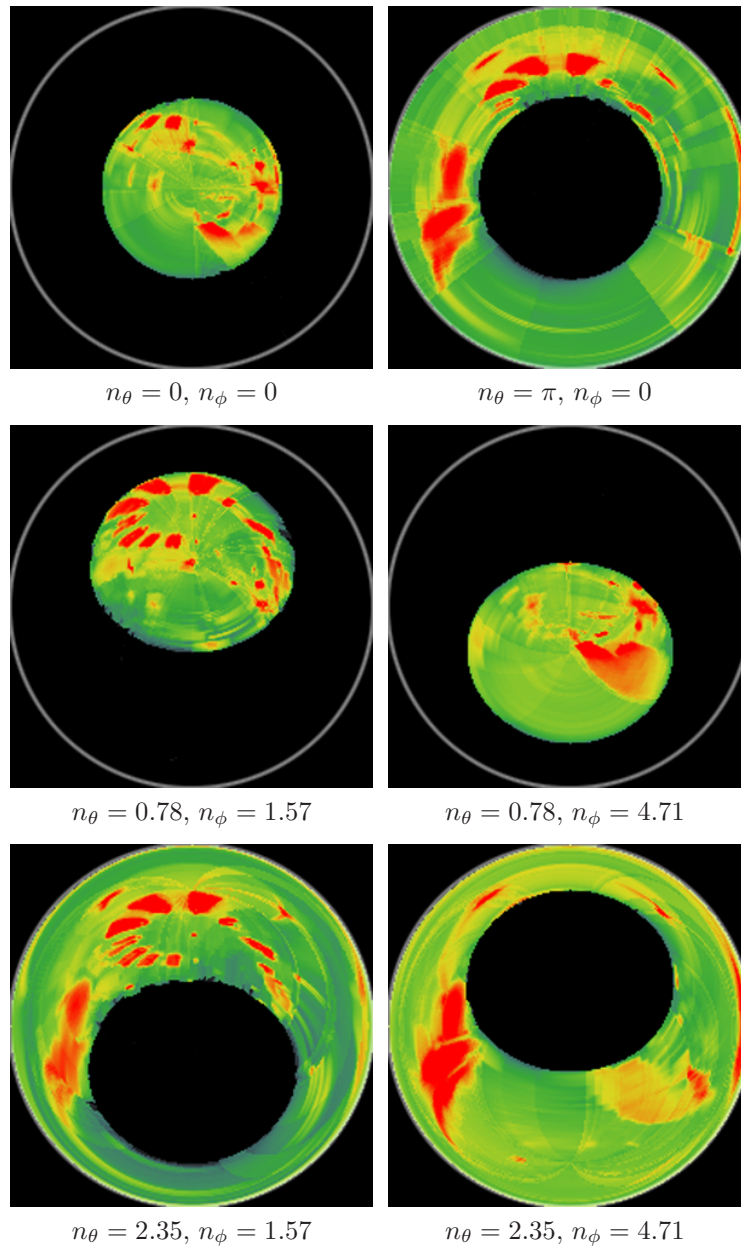


Figure 5.18: False-color visualizations of several CDFs computed at different surface orientations. Each distribution is visualized on a logarithmic scale as in Figure 5.14. For each surface normal considered we clip the environment map to the visible hemisphere and multiply each radiance value by $(\mathbf{n} \cdot \omega_i)$ before computing an adaptive CDF representation of the resulting 4D distribution.

5.5.11 Local Environment Map Sampling

Using adaptive numerical CDFs, we introduce a novel algorithm for sampling an environment map in an orientation-dependent manner. In previous methods of sampling environment maps, incoming directions are drawn from a single spherical distribution [1, 55, 85, 88]. This approach is inefficient when a significant amount of light in the scene happens to fall below the horizon for a large number of pixels. In Figure 5.17, there are many regions of the image where as few as 5% of the samples are generated above the horizon—this also indicates the inefficacy of standard techniques like rejection sampling to address this problem. Furthermore, sampling from a single spherical distribution cannot consider the *cosine term* that appears in the rendering equation (i.e. $\max(0, \mathbf{n} \cdot \omega_i)$). Accounting for this cosine-falloff would require sampling from a 4D function (i.e. there are two degrees of freedom in the incoming direction and two in the normal direction). We show several 2D slices of this function for different normal directions in Figure 5.18. As with BRDFs, representing a 4D distribution even at a modest resolution could require prohibitively large storage.

We can store the 4D distribution that results from modulating an environment map by the cosine term using our adaptive CDF representation. During rendering, each pixel corresponds to a normal direction that becomes an index into the 4D distribution, producing a 2D distribution over incoming directions that we sample from. In our experiments, we evaluated the *local* environment map distribution at 25×10 ($\phi \times \theta$) normal directions and 1000×2000 ($\phi \times \theta$) incoming directions. Storing this tabular CDF directly would require *approx.* 4GB of space. In contrast, our representation requires 10-20MB of storage and 1-2 hours of compute time to provide an acceptable approximation.

We compared *local environment map sampling* with jittered sampling of a stratified representation [1] and sampling from a uniformly-spaced CDF [88] (see Figure 5.19). Jittered sampling (Figure 5.19 left) performed the worst mainly because this technique is ineffective for such low sample counts (note: we are using only 20 samples here). Moreover, there is significant error due to the bias introduced by approximating each strata with a radial disk. Although unbiased jittering is not impossible to achieve, it is not a simple extension to published algorithms and has not been reported in previous work. We also compared our algorithm to sampling from a uniformly-sampled CDF and rejecting those samples that fell below the horizon [88] (Figure 5.19

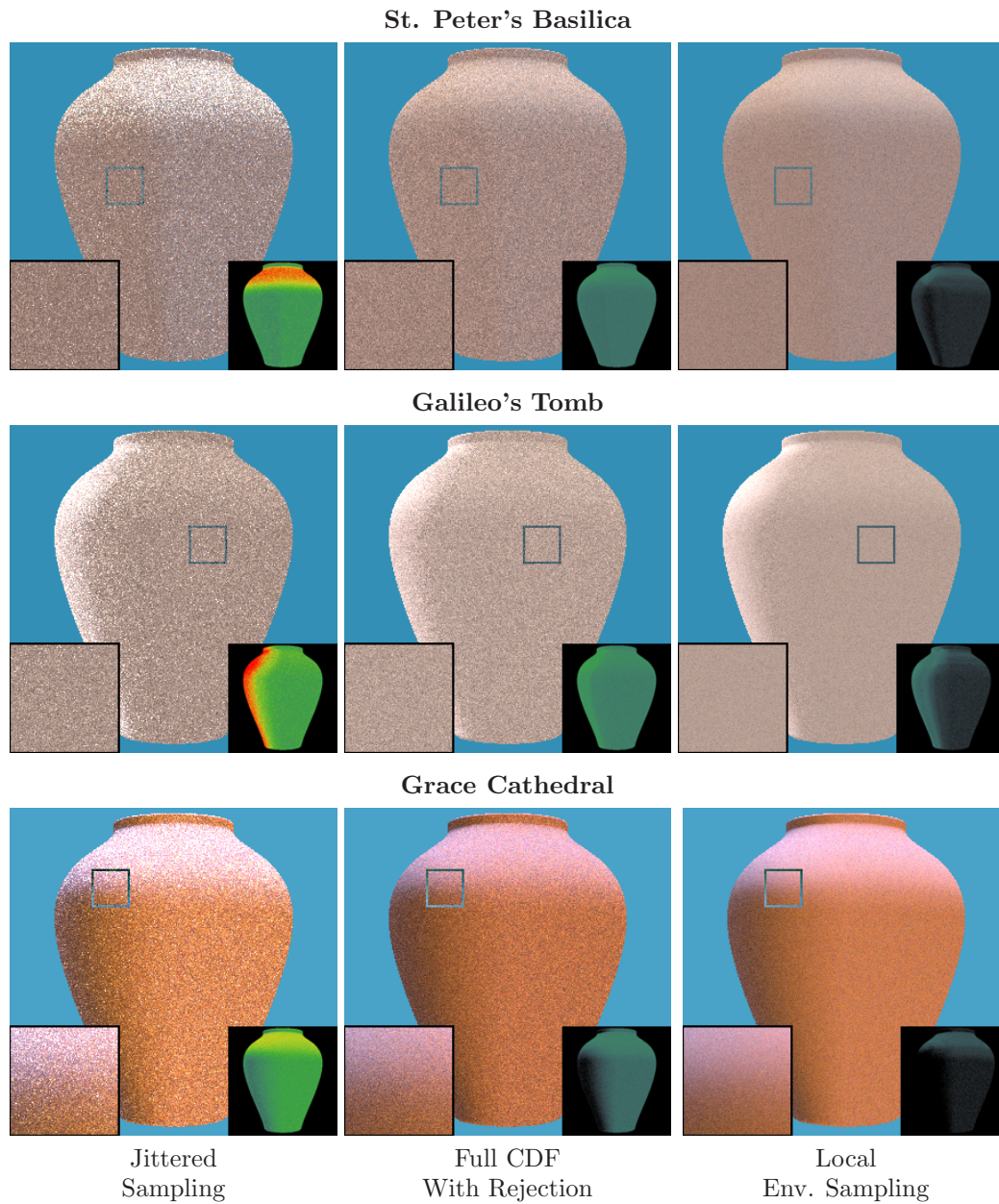


Figure 5.19: We compare the variance of a Monte Carlo estimator computed according to (left column) jittered sampling of a stratified representation [1], (middle column) a uniformly-sampled CDF [88] where we reject samples that fall below the horizon and (right column) using our *local environment map sampling* algorithm. We have rendered a perfectly diffuse object at 20 samples/pixel in three different environments that all exhibit high-frequency lighting. Cutouts include a magnified region of the image and a variance image (note: these are false-color visualizations of the *logarithm* of RMS error in image intensity where black ≤ 0.135 and red ≥ 20.08). All three sampling methods required approximately 15 seconds to render these images.

middle). This strategy is most comparable in quality to our own, but because it does not account for the horizon-clipped cosine term in the rendering equation, it fails to achieve the same rate of convergence. Quantitatively, *local environment map sampling* achieved *approx.* 5 times lower variance than sampling a single CDF computed at full resolution and *approx.* 20 times better than jittered sampling for these test scenes.

5.5.12 Multiple Importance Sampling

In practice, neither the BRDF nor the incident illumination alone determine the final shape of the integrand in the rendering equation. Therefore, it is critical that a CDF representation supports multiple importance sampling [108]. The main criterion this imposes is that the representation must allow efficient computation of the probability of a direction that was not generated from the distribution itself. Algorithms that decompose environment maps into non-overlapping strata [1, 55, 85], for example, do not readily provide this property because determining the probability of an arbitrary direction would require searching over the strata. Although not impossible, making this search efficient has not previously been demonstrated and could be one direction of future work. With our adaptive numerical CDF, however, the probability of an arbitrary direction can be computed in $O(\log N)$ where N is the number of non-uniformly spaced samples. Moreover, because of the compression ratios possible with our representation, N is typically small enough to make this operation inexpensive in practice.

We show several scenes for which multiple importance sampling is critical (Figure 5.20). In these results, we use the *balance heuristic* introduced by [108] to combine 50 samples of the BRDF with 50 samples of the environment. The BRDF samples are generated using our adaptive CDF discussed in Section 5.5.9 and the illumination samples are generated using *local environment map sampling* (see Section 5.5.11). To demonstrate the benefit of a representation that supports multiple importance sampling, we also compare these images to those rendered using 100 samples drawn from either the BRDF or environment alone.

5.5.13 Conclusions and Future Work

This work addresses a long-standing graphics problem of efficiently importance sampling complex analytic and measured BRDFs. We have introduced two methods for importance sampling the

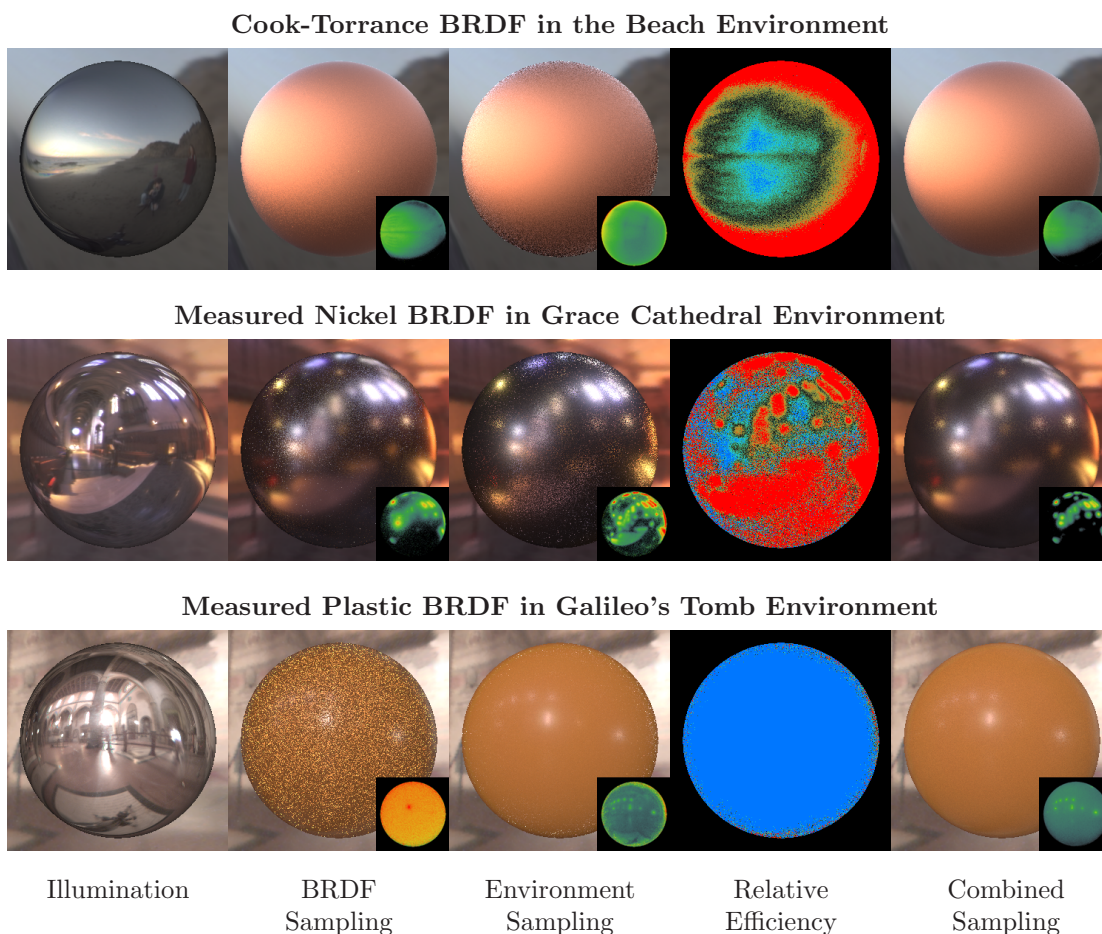


Figure 5.20: Multiple importance sampling using adaptive numerical CDFs computed from both BRDFs and image-based lighting. The 4th column visualizes the relative reduction in variance using environment sampling vs. BRDF sampling: red = BRDF sampling has 8x less variance than environment map sampling, blue = environment sampling has 8x less variance than BRDF sampling. For these scenes, sampling from either the BRDF or environment alone will not effectively reduce variance over the entire image and performing multiple importance sampling is critical.

BRDF that reduces sampling to inverting a handful of 1D cumulative distribution functions. This provides a compact practical representation and a simple algorithm for sampling, which in many cases reduces variance and sampling times by an order of magnitude relative to previous methods. We use our representation and importance sampling method to efficiently render scenes with multiple isotropic and anisotropic materials with global illumination under complex illumination.

In future work, we would like to extend our technique to allow for mixed parameterizations of the factored BRDF, such that each term may have a different parameterization. This would allow us to better approximate BRDFs that exhibit several different types of scattering (e.g. side, backward

and forward) at the same time. Our factorization method might also have applications in sampling bi-directional texture functions (BTFs) and light fields—two examples of high-dimensional functions that, like BRDFs, typically have significant redundancy. More generally, we see our work as a first step towards efficient techniques to sample high-dimensional measured functions. With the increasing importance of measured and image-based data in computer graphics, this problem promises to have growing significance.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Providing computer models that accurately characterize material appearance is central to many problems in computer graphics and computer vision. Conventional techniques based on parametric surface reflectance functions and texture maps cannot capture the complex appearance of many real-world materials. Recent techniques for measuring the way a material reflects light have provided more accurate appearance representations at the cost of expensive acquisition times, delicate calibration procedures and large storage requirements.

A common approach to incorporating measured appearance data into computer generated imagery is to fit the parameters of an analytic scattering function (e.g. the Bidirectional Reflectance Distribution Function (BRDF)) to the measurements. Although this technique provides significant compression ratios, gives a designer the ability to edit the appearance and is already integrated into state-of-the-art rendering algorithms, it can introduce significant approximation error for many materials. On the other hand, nonparametric approaches, including basis function decomposition and standard matrix rank-reduction algorithms such as PCA, can retain high fidelity to the original data. Although these representations provide greater accuracy and generality, they have so far have not incorporated the important design goal of *editability*. Furthermore, previous representations are not optimized for importance sampling; an important technique for improving the efficiency of physically-based rendering algorithms.

We have introduced several new nonparametric representations of surface reflectance functions that address the shortcomings of previous techniques. In particular, the Inverse Shade Tree (IST) framework provides a general approach to estimating the “leaves” of a user-specified shade tree from high-dimensional measured datasets of appearance. These leaves are sampled 1- and 2-dimensional functions that capture both the directional behavior of individual materials and their spatial mixing patterns. In order to compute these shade trees automatically, we map the problem to matrix factorization and introduce a flexible new algorithm that allows for constraints such as non-negativity, sparsity, and energy conservation. Although we cannot infer every type of shade tree, we demonstrate the ability to reduce multi-gigabyte measured datasets of the Spatially-Varying Bidirectional Reflectance Distribution Function (SVBRDF) into a compact representation that may be edited in real time. We also introduce a compact representation for the spatial component of the Bidirectional Subsurface Scattering Reflectance Distribution Function (BSSRDF). Unlike previous techniques that fit a simple analytic model to BSSRDF measurements, our approach can represent materials with heterogeneous subsurface scattering properties.

We also introduce an importance sampling technique for a wide range of BRDFs, including complex analytic models such as Cook-Torrance and measured materials. Our approach is based on a compact factored representation of the BRDF that is optimized for sampling. We show that our algorithm consistently offers better efficiency than alternatives that involve fitting and sampling a Lafortune or Blinn-Phong lobe, and is more compact than sampling strategies based on tabulating the full BRDF. We are able to efficiently create images involving multiple measured and analytic BRDFs, under both complex direct lighting and global illumination.

Lastly, we apply algorithms traditionally used for curve approximation to reduce the size of a multidimensional tabulated Cumulative Distribution Function (CDF) by one to three orders of magnitude. These *adaptive* representations enable new algorithms for sampling image-based illumination and measured BRDFs in the context of global illumination rendering.

6.2 Areas of Future Work

Our work suggests several different areas of future research:

Acquisition Existing techniques for measuring material appearance (such as those discussed in Chapter 3) densely sample the domain of a particular light transport function. These approaches quickly become impractical for high-dimensional functions. Furthermore, the redundant information in these datasets suggests that dense acquisition is not necessary. For example, we are able to generate accurate shade trees of the SVBRDF from a very small portion of the complete input using the subsampling procedures discussed in Section 4.5.8. Future research might consider the question: “How can we optimize our acquisition to avoid measuring redundant or less visually important regions in the input?” It seems that the challenge is two-fold. First, it is unclear what type of device would allow measuring the most useful subset of a particular scattering function. Second, an optimal acquisition strategy will depend on the material properties of the particular sample. Recent work has demonstrated the potential for making use of this redundancy for a variety of appearance functions [63, 113, 65, 41]. Nonetheless, this remains an active area of research and future work might consider applying these ideas to acquiring the appearance of more general light scattering functions.

Another promising approach is to parallelize the acquisition process using multiple sensors. Recent designs [112, 24, 110] have a rigid assembly of many cameras and light sources. Each camera simultaneously records an image of the sample material illuminated by each light-source in rapid succession. This allows acquisition times on the order of a few seconds. However, because the locations of the cameras and lights are fixed, this setup does not allow adaptive acquisition. Investigating setups that combine parallel acquisition with view-planning and adaptive exploration of a function’s domain would be another interesting area of future research.

Representation We believe the Inverse Shade Tree framework could provide novel representations for many other high-dimensional measured appearance functions beyond the BRDF, BSSRDF and SVBRDF previously considered. One possible direction of future work is to incorporate additional aspects of reflectance variation such as displacement maps for fine geometric detail typically represented as Bidirectional Texture Functions (BTFs). Due to the complex shadowing and masking at the surface of these materials, existing algorithms would not be suitable for efficiently estimating their shade trees and future work should consider more general inference.

There are usually multiple ways to factorize a high-dimensional transport function into lower-

dimensional components. For example, the dependency of the SVBRDF on wavelength can be preserved in either the spatial blending weights or the basis BRDFs (we do the latter in Chapter 4). However, different edits would be easier with different decompositions. Future work might investigate alternative decompositions for a particular appearance function. Along these lines, we would like to investigate techniques for automatically inferring the structure of the tree (including the number of terms used at each stage in the decomposition). This could include decomposing the same dataset into multiple trees, either of which could be edited by a designer.

Alternating Constrained Least Squares Another avenue of future work is related to the Alternating Constrained Least Squares (ACLS) techniques we proposed in Section 4.5. Their flexibility and provable local convergence make them ideal candidates for a broad range of dimensionality reduction applications in data mining and other machine learning contexts. Future work could evaluate the efficiency and noise-tolerance properties of ACLS, and investigate the impact of various types of additional linear constraints.

Out-of-core Factorization The massive size of most appearance datasets makes computing their decomposition infeasible. Although subsampling procedures provide one solution, they do so by considering only a small portion of the input. Future work in this area might develop factorization algorithms that work in parallel across a cluster of computers. Another direction would be developing *out-of-core* factorization algorithms that require storing only a small amount of the complete data within main-memory at any one time, but eventually consider the entire input.

Bibliography

- [1] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 22(3):605–612, 2003.
- [2] James Arvo. Stratified sampling of spherical triangles. In *Proceedings of ACM SIGGRAPH 1995*, pages 437–438, 1995.
- [3] James Arvo. Stratified sampling of 2-manifolds. In *SIGGRAPH 2001 Course Notes*, volume 29, 2001.
- [4] Michael Ashikhmin, Simon Premože, and Peter Shirley. A microfacet-based brdf generator. In *SIGGRAPH 2000*, pages 65–74, 2000.
- [5] Michael Ashikhmin and Peter Shirley. An anisotropic phong brdf model. *Journal of Graphics Tools*, 5(2):25–32, 2000.
- [6] David C. Banks. Illumination in diverse codimensions. In *Proceedings of ACM SIGGRAPH 1994*, pages 327–334. ACM Press, 1994.
- [7] P. Beckmann and A. Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. MacMillan, 1963.
- [8] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *Proceedings of European Conference on Computer Vision*, pages 237–252, London, UK, 1992. Springer-Verlag.

- [9] James F. Blinn. Models of light reflection for computer synthesized pictures. In *Computer Graphics (SIGGRAPH '77 Proceedings)*, pages 192–198, New York, NY, USA, 1977. ACM Press.
- [10] James F. Blinn. *Computer Display of Curved Surfaces*. PhD thesis, University of Utah, 1978.
- [11] James F. Blinn. Simulation of wrinkled surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)*, pages 286–292, 1978.
- [12] James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. In *CACM*, pages 542–547, 1976.
- [13] Ed Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, University of Utah, 1974.
- [14] W. S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments. In *Proceedings of the Third International Symposium on Algorithms and Computation*, pages 378–387, London, UK, 1992. Springer-Verlag.
- [15] Danny Z. Chen and Ovidiu Daescu. Space-efficient algorithms for approximating polygonal curves in two-dimensional space. *International Journal of Computational Geometry & Applications*, 13(2):95–111, 2003.
- [16] Wei-Chao Chen, Jean-Yves Bouguet, Michael H. Chu, and Radek Grzeszczuk. Light field mapping: efficient representation and hardware rendering of surface light fields. In *Computer Graphics (SIGGRAPH 2002)*, pages 447–456, 2002.
- [17] Luc Claustres, Mathias Paulin, and Yannick Boucher. Brdf measurement modeling using wavelets for efficient path tracing. *Computer Graphics Forum*, 12(4):1–16, 2003.
- [18] Robert L. Cook. Shade trees. In *SIGGRAPH 1984*, pages 223–231, 1984.
- [19] Robert L. Cook. Stochastic sampling in computer graphics. In *Computer Graphics (ACM SIGGRAPH 1986)*, pages 51–72. ACM Press, 1986.
- [20] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *Computer Graphics (SIGGRAPH 1981)*, pages 7–24, 1981.

- [21] Kristin Dana, Bram van Ginneken, Shree Nayar, and Jan Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.
- [22] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of ACM SIGGRAPH 1998*, pages 189–198, New York, NY, USA, 1998. ACM Press.
- [23] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Computer Graphics (SIGGRAPH 2000)*, pages 145–156, 2000.
- [24] Paul Debevec, Andreas Wenger, Chris Tchou, Andrew Gardner, Jamie Waese, and Tim Hawkins. A lighting reproduction approach to live-action compositing. In *Proceedings of ACM SIGGRAPH 2002*, pages 547–556, 2002.
- [25] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of ACM SIGGRAPH 1997*, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [26] Joel M. DeYoung. Properties of tabulated bidirectional reflectance distribution functions. Master’s thesis, The University of British Columbia, 1996.
- [27] David Douglas and Thomas Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.
- [28] Kai-Tai Fang, Fred J. Hickernell, and Harald Niederreiter. *Monte Carlo and Quasi-Monte Carlo Methods*. Springer, 1st edition, 2002.
- [29] Alain Fournier. Separating reflectance functions for linear radiosity. In *Eurographics Workshop on Rendering*, pages 383–392, 1995.
- [30] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi. Appearance based object modeling using texture database: acquisition, compression and rendering. In *Eurographics Workshop on Rendering*, pages 257–266, 2002.

- [31] Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. Linear light source reflectometry. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 22(3):749–758, 2003.
- [32] P. Gill, W. Murray, M. Saunders, and M. Wright. Procedures for optimization problems with a mixture of bounds and general linear constraints. In *ACM Trans. Mathematical Software*, 1984.
- [33] Michael Goesele, Hendrik P. A. Lensch, Jochen Lang, Christian Fuchs, and Hans-Peter Seidel. Disco: acquisition of translucent objects. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3):835–844, 2004.
- [34] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and spatially-varying brdfs from photometric stereo. In *IEEE International Conference on Computer Vision*, 2005.
- [35] Michael T. Goodrich. Efficient piecewise-linear function approximation using the uniform metric: (preliminary version). In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, pages 322–331. ACM Press, 1994.
- [36] Stephen Gortler, Radek Grzeszczuk, Rick Szeliski, and Michael Cohen. The lumigraph. In *Computer Graphics (ACM SIGGRAPH 96)*, 1996.
- [37] Donald P. Greenberg, Kenneth E. Torrance, Peter Shirley, James Arvo, Eric Lafortune, James A. Ferwerda, Bruce Walter, Ben Trumbore, Sumanta Pattanaik, and Sing-Choong Foo. A framework for realistic image synthesis. In *Proceedings of ACM SIGGRAPH 1997*, pages 477–494, 1997.
- [38] Jefferson Y. Han and Ken Perlin. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 22(3):741–748, 2003.
- [39] Bruce Hapke. A theoretical photometric function for the lunar surface. *Journal of Geophysical Research*, 68(15), 1963.
- [40] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

- [41] Tim Hawkins, Per Einarsson, and Paul Debevec. Acquisition of time-varying participating media. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3):812–815, 2005.
- [42] Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, and Donald P. Greenberg. A comprehensive physical model for light reflection. In *Computer Graphics (SIGGRAPH 1991)*, pages 175–186, 1991.
- [43] Paul Heckbert. Survey of texture mapping. In *IEEE Computer Graphics and Applications*, 1986.
- [44] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, hardware-accelerated shading and lighting. In *Computer Graphics (SIGGRAPH 1999)*, pages 171–178, 1999.
- [45] John Hershberger and Jack Snoeyink. Speeding up the douglas-peucker line-simplification algorithm. In *Proceedings of the 5th International Symposium on Spatial Data Handling*, volume 1, pages 134–143, Charleston, South Carolina, 1992.
- [46] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence*, 1999.
- [47] Patrick O. Hoyer. Non-negative sparse coding. In *IEEE Workshop on Neural Networks for Signal Processing*, pages 557–565, 2002.
- [48] Rafal Jaroszkiwicz and Michael D. McCool. Fast extraction of brdfs and material maps from images. In *Graphics Interface*, 2003.
- [49] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of ACM SIGGRAPH 2001*, pages 511–518, New York, NY, USA, 2001. ACM Press.
- [50] James T. Kajiya. Anisotropic reflection models. In *Computer Graphics (SIGGRAPH 1985)*, pages 15–21, 1985.
- [51] James T. Kajiya. The rendering equation. In *Computer Graphics (ACM SIGGRAPH 1986)*, pages 143–150, 1986.

- [52] James T. Kajiya and Timothy L. Kay. Rendering fur with three dimensional textures. In *Computer Graphics (SIGGRAPH 1989)*, pages 271–280, 1989.
- [53] Jan Kautz and Michael McCool. Interactive rendering with arbitrary brdfs using separable approximations. In *Eurographics Workshop on Rendering*, pages 247–260, 1999.
- [54] J. Koenderink and A. van Doorn. Phenomenological description of bidirectional surface reflection. *Journal of the Optical Society of America A*, 15(11):2903–2912, 1998.
- [55] Thomas Kollig and Alexander Keller. Efficient illumination by high dynamic range images. In *Proceedings of the Eurographics Symposium on Rendering*, pages 45–51, 2003.
- [56] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *Computer Graphics (SIGGRAPH 1997)*, pages 117–126, 1997.
- [57] Paul Lalonde and Alain Fournier. Generating reflected directions from brdf data. *Computer Graphics Forum*, 16(3):293–300, 1997.
- [58] Johann H. Lambert. *Photometria sive de mensura de gratibus luminis, colorum umbrae*. Eberhard Klett, 1760.
- [59] Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3), 2006.
- [60] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3), 2004.
- [61] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Adaptive numerical cumulative distribution functions for efficient importance sampling. In *Proceedings of the Eurographics Symposium on Rendering*, pages 11–20. Eurographics Association, 2005.
- [62] Daniel Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

- [63] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatially varying materials. In *Eurographics Workshop on Rendering*, pages 63–70, 2001.
- [64] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics*, 22(2):234–257, 2003.
- [65] Hendrik P. A. Lensch, Jochen Lang, Asla M. Sá, and Hans-Peter Seidel. Planned sampling of spatially varying BRDFs. In *Proceedings of Eurographics*, pages 473–482, 2003.
- [66] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
- [67] Robert R. Lewis. Making shaders more physically plausible. In *Fourth Eurographics Workshop on Rendering*, pages 47–62, 1993.
- [68] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [69] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial texture maps. In *SIGGRAPH 2001*, pages 519–528, 2001.
- [70] S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg. Image-Based BRDF measurement including human skin. In *Eurographics Rendering Workshop 99*, pages 139–152, 1999.
- [71] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM Transactions on Graphics (ACM SIGGRAPH 2003)*, 22(3):780–791, 2003.
- [72] Stephen R. Marschner, Stephen H. Westin, Adam Arbree, and Jonathan T. Moon. Measuring and modeling the appearance of finished wood. *ACM Transactions on Graphics (ACM SIGGRAPH 2005)*, 24(3):727–734, 2005.

- [73] Wojciech Matusik. *A Data-Driven Reflectance Model*. PhD thesis, Massachusetts Institute of Technology, 2003.
- [74] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 22(3):759–769, 2003.
- [75] Wojciech Matusik, Hanspeter Pfister, Matthew Brand, and Leonard McMillan. Efficient isotropic brdf measurement. In *Eurographics Workshop on Rendering*, pages 241–247, 2003.
- [76] David McAllister. *A Generalized Surface Appearance Representation for Computer Graphics*. PhD thesis, UNC Chapel Hill, 2002.
- [77] Michael D. McCool, Jason Ang, and Anis Ahmad. Homomorphic factorization of brdfs for high-performance rendering. In *Computer Graphics (SIGGRAPH 2001)*, pages 185–194, 2001.
- [78] NAG. Numerical algorithms group c library, 2005.
- [79] John Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–311, 1965.
- [80] L. Neumann and A. Neumann. Photosimulation interreflection with arbitrary reflection models and illumination. *Computer Graphics Forum*, 8(1):21–34, 1989.
- [81] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of brdf models. In *Proceedings of the Eurographics Symposium on Rendering*, pages 117–226. Eurographics Association, 2005.
- [82] F. E. Nicodemus, J. C. Richmond, and J. J. Hsia. Geometrical considerations and reflectance. *National Bureau of Standards*, October 1977.
- [83] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 2002.
- [84] Michael Oren and Shree K. Nayar. Generalization of lambert’s reflectance model. In *Computer Graphics (SIGGRAPH 1994)*, pages 239–246, 1994.

- [85] Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3):488–495, 2004.
- [86] Pieter Peers, Karl vom Berge, Wojciech Matusik, Ravi Ramamoorthi, Jason Lawrence, Szymon Rusinkiewicz, and Philip Dutré. A compact factored representation of heterogeneous subsurface scattering. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3), 2006.
- [87] Ken Perlin. An image synthesizer. In *Computer Graphics (ACM SIGGRAPH 1985)*, pages 287–296, 1985.
- [88] Matt Pharr and Greg Humphreys. *Physically Based Rendering : From Theory to Implementation*. Morgan Kaufmann, 2004.
- [89] Bui Thong Phong. Illumination for computer generated images. *Communications of the ACM*, 18:311–317, 1975.
- [90] Pierre Poulin and Alain Fournier. A model for anisotropic reflection. In *Computer Graphics (SIGGRAPH 1990)*, pages 273–282, 1990.
- [91] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.
- [92] Kekoa Proudfoot, William Mark, Svetoslav Tzvetkov, and Pat Hanrahan. A real-time procedural shading system for programmable graphics hardware. In *Computer Graphics (SIGGRAPH 2001)*, pages 159–170, 2001.
- [93] Paul L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):659–666, 1997.
- [94] Szymon Rusinkiewicz. A new change of variables for efficient BRDF representation. In *Eurographics Workshop on Rendering*, pages 11–22, 1998.
- [95] Richard L. Scheaffer. *Introduction to Probability and Its Applications (Statistics)*. Duxbury Press, 2nd edition, 1994.

- [96] Peter Shirley. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, University of Illinois at Urbana Champaign, 1990.
- [97] Jos Stam. Diffraction shaders. In *Computer Graphics (SIGGRAPH 1999)*, pages 101–110, 1999.
- [98] Frank Suykens, Karl vom Berge, Ares Lagae, and Philip Dutré. Interactive rendering with bidirectional texture functions. In *Eurographics*, 2003.
- [99] Richard Szeliski. Image alignment and stitching: A tutorial. Technical Report MSR-TR-2004-92, Microsoft Research, 2004.
- [100] K. E. Torrance and E. M. Sparrow. Off-specular peaks in the directional distribution of reflected thermal radiation. *Transactions of the ASME*, 1966.
- [101] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57, 1967.
- [102] Norimichi Tsumura, Nobutoshi Ojima, Kayoko Sato, Mitsuhiro Shiraishi, Hideto Shimizu, Hirohide Nabeshima, Syuuichi Akazaki, Kimihiko Hori, and Yoichi Miyake. Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. *ACM Transactions on Graphics (SIGGRAPH 2003)*, 22(3):770–779, 2003.
- [103] Purdue University. *Thermophysical Properties of Matter, vol. 7: Thermal Radiative Properties of Metals*. Plenum, 1970.
- [104] Purdue University. *Thermophysical Properties of Matter, vol. 8: Thermal Radiative Properties of Nonmetallic Solids*. Plenum, 1970.
- [105] Purdue University. *Thermophysical Properties of Matter, vol. 9: Thermal Radiative Properties of Coatings*. Plenum, 1970.
- [106] M. Alex Vasilescu and Demetri Terzopoulos. TensorTextures: Multilinear image-based rendering. *ACM Transactions on Graphics (ACM SIGGRAPH 2004)*, 23(3), 2004.
- [107] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.

- [108] Eric Veach and Leonidas Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of ACM SIGGRAPH 1995*, pages 419–428, 1995.
- [109] Gregory J. Ward. Measuring and modeling anisotropic reflection. In *Computer Graphics (SIGGRAPH 1992)*, pages 265–272, 1992.
- [110] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3):756–764, 2005.
- [111] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. In *Computer Graphics (ACM SIGGRAPH 1992)*, pages 255–264, New York, NY, USA, 1992. ACM Press.
- [112] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, and Markus Gross. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics (SIGGRAPH 2006)*, 25(3), 2006.
- [113] Todd Zickler, Sebastian Enrique, Ravi Ramamoorthi, and Peter Belhumeur. Reflectance sharing: Image-based rendering from a sparse set of images. In *Proceedings of the Eurographics Symposium on Rendering*, pages 253–264. Eurographics Association, 2005.