
A VISUALIZATION SYSTEM FOR BIDIRECTIONAL
REFLECTANCE DISTRIBUTION FUNCTIONS

by

PETER ANDREW WALKER

A THESIS

Presented to the Department of Computer
and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 1999

“A Visualization System For Bidirectional Reflectance Distribution Functions,” a thesis prepared by Peter Andrew Walker in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science. This thesis has been approved and accepted by:



Dr. Gary Meyer

6/6/99

Date

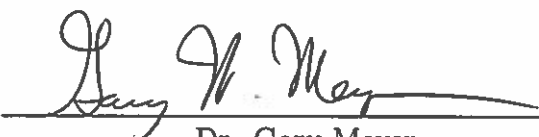
Accepted by:



Dean of the Graduate School

An Abstract of the Thesis of
Peter Andrew Walker for the degree of Master of Science
in the Department of Computer and Information Science
to be taken June 1999

Title: A VISUALIZATION SYSTEM FOR BIDIRECTIONAL
REFLECTANCE DISTRIBUTION FUNCTIONS

Approved: 
Dr. Gary Meyer

Electromagnetic radiation is scattered by the interfaces it encounters. This scattering can be mathematically described using the Bidirectional Reflectance Distribution Function (BRDF). BRDFs are studied in many fields, and the diverse BRDF literature was surveyed with the goal of developing a software library of BRDF representation schemes. The Oregon BRDF Library was subsequently employed in the construction of a visualization tool that is useful for both research and education. This tool, known as BRDFvis, was built using Advanced Visual Systems Inc. Express. BRDFvis permits interactive viewing of the scatter predicted by the models present in the Oregon BRDF Library as well as reflectance distributions stored in a database of measured samples. The Nonconventional Exploitation Factors (NEF) database and the Oregon BRDF Library models were then used to generate images.

CURRICULUM VITA

NAME OF AUTHOR: Peter Andrew Walker

PLACE OF BIRTH: Detroit, Michigan

DATE OF BIRTH: December 22, 1967

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon
Bassist College

DEGREES AWARDED:

Master of Science in Computer and Information Science, 1999,
University of Oregon
Bachelor of Arts in Computer and Information Science and English,
1997, University of Oregon
Associate of Arts in Apparel Design, 1987, Bassist College

AREAS OF SPECIAL INTEREST:

Computer Graphics
Image Science

PROFESSIONAL EXPERIENCE:

Research and Teaching Assistant, Department of Computer and
Information Science, Eugene, Oregon, 1997-1999.

Software Lab Intern, Hewlett-Packard, Camas, Washington, 1997.

Product Developer, Nike, Inc., Beaverton, Oregon, 1987-1994.

ACKNOWLEDGEMENTS

My thanks to my advisor, Dr. Gary Meyer. Although I felt overwhelmed by the complexity of this project more than once, he encouraged me and reiterated that I tend to be my own worst critic. The existence of this thesis is a testimony to his encouragement.

Thanks also to a fellow graduate student and my office-mate, Harold Westlund. His easy-going yet motivated demeanor always encourages me to try harder.

And thanks to my partner, Gary Miller. He always has both feet on the ground. My flights of inspiration (and depression) don't phase him—he is always there for me.

DEDICATION

To my parents, who subconsciously apply the subtle pressure of sibling rivalry—sometimes it produces remarkable results.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
Illumination in 3D Computer Graphics	1
What is the BRDF?	2
Fields Interested in BRDFs	3
BRDF Geometry	4
The BRDF Equation	7
Applications	8
II. A BRDF VISUALIZATION SYSTEM	10
Visualization as a Learning Tool	10
Visualization Tools	11
Designing and Building a BRDF Viewer	12
BRDFvis: An Interactive BRDF Visualization Environment	13
III. A SURVEY OF BRDF MODELS	30
Model Methodology	31
Approaches Used to Derive a Model	33
A Simple Beginning: Diffuse	34
Lambertian	35
Minnaert	37
Introducing Reflections: Specular	38
Phong	38
Back to First Principles: Physics	41
Cook-Torrance	41
He-Torrance	44
Dealing with Asymmetry: Anisotropy	48
Kajiya	48
Poulin-Fournier	49
Ward	52
Not-So-Ideal Diffuse	55
Oren-Nayar	56
Hapke/Lommel-Seeliger	58

	Page
Blinn	60
Measuring the Real Thing	62
Lafortune	63
Beard-Maxwell	65
NEF Database	67
Columbia-Utrecht Database	68
Simulating the Real Thing	69
Cabral	70
Westin	70
Gondek	71
Choosing a Model	72
IV. RENDERING WITH BRDFS	73
Rendering as a Learning Tool	73
Radiance: A Radiometrically Correct Renderer	74
Oregon BRDF Library Renderings	76
IBRDF Renderings	82
 APPENDIX	
A. OREGON BRDF LIBRARY API	84
Library Conventions	84
Overview of Routines	86
Model-Specific APIs	92
B. BRDFVIS NOTES	94
C. EXTENDING BRDFVIS	96
Adding a BRDF to the Oregon BRDF Library	96
Adding a BRDF to BRDFvis	98
D. BRDF MODEL DERIVATIONS	101
Blinn	101
Cook-Torrance	101
He-Torrance	102
Oren-Nayar	104
Poulin-Fournier	104
Ward	105

	Page
BIBLIOGRAPHY	106

LIST OF FIGURES

Figure	Page
1. Specifying BRDF angles.	4
2. BRDFvis AVS Express application.	14
3. BRDFvis Library Workspaces.	15
4. Disconnected sample network (BRDFvis).....	16
5. Connected sample network (BRDFvis).....	18
6. Lighting Controls panel (BRDFvis).....	19
7. Framing Controls panel (BRDFvis).....	21
8. BRDF Selection panel (BRDFvis).....	23
9. General Controls panel (BRDFvis).....	24
10. Color Control (BRDFvis).....	25
11. Axis panel (BRDFvis).....	26
12. Dual BRDF visualization network (BRDFvis).....	27
13. Dual reflectance lobes (BRDFvis).....	28
14. Lambertian reflectance lobe.	35
15. Lambertian spheres.....	36
16. Minnaert reflectance lobes.....	37
17. Phong reflectance lobes.....	39
18. Phong model ψ angle.....	40
19. Phong spheres.....	41
20. Microfacet cross-section.....	42

	Page
21. Cook-Torrance reflectance lobes	43
22. Cook-Torrance spheres	43
23. He-Torrance reflectance lobes	46
24. He-Torrance sphere	47
25. Poulin-Fournier microgeometry	49
26. Poulin-Fournier angles	50
27. Poulin-Fournier reflectance lobe	52
28. Half-angle vector, half-angle, and phase angle specification	53
29. Ward reflectance lobes	54
30. Anisotropic Ward spheres	55
31. Oren-Nayar reflectance lobes	57
32. Hapke/Lommel-Seeliger scattering geometry	58
33. Blinn scattering geometry	60
34. Blinn reflectance lobes	61
35. Lafortune's paint reflectance lobe and rendered sphere	64
36. Beard-Maxwell reflectance lobe for white-primed aluminum	66
37. NEF concrete reflectance lobes	67
38. Thingamajig with a vareity of BRDF-described surfaces	77
39. Vases rendered with Lafortune, Cook-Torrance, and Oren-Nayar BRDFs	79
40. Vases rendered with NEF materials	81
41. Textured cubes rendered with NEF materials	82
42. Textured vases rendered with NEF materials	83

CHAPTER I

INTRODUCTION

Illumination in 3D Computer Graphics

When rendering a 3D scene on the computer, the most important issue is a convincing interaction of light with the objects in the modeled environment. Computer artists are versed in a variety of techniques to create just the right visual effect—3D models, textures, bump and displacement maps, and lighting all contribute to realizing the artist's vision.

But all these items are separate, and remain so until an illumination model glues the components together. The realism achieved by any particular illumination model is not only a function of the artist's effort, but also a result of the physical accuracy of the illumination model. Today, with the general availability of massive computing horsepower, physically correct illumination models are becoming a reality. Whereas the artist of the past had to struggle to achieve a particular appearance with a limited library of tools, the artist of the near future will be able to assign the reflectance properties of real-world surfaces to 3D models. The images generated will have the potential to be both strikingly realistic as well as physically accurate.

The Bidirectional Reflectance Distribution Function, or BRDF, is the means by which physically accurate reflectance is achieved.

What is the BRDF?

A BRDF is a mathematical description of the manner in which electromagnetic radiation is scattered by a particular interface [17]; in the context of this thesis, the electromagnetic radiation will be visible light. An interface is the boundary between two media; one of the two media is typically air, and the other medium is denser than air. Such an interface is commonly referred to as a surface, and throughout this discussion the two will be used interchangeably.

The BRDF is a special-case of a more general function, a Bidirectional Reflectance-Transmittance Distribution Function (BRTDF) or Bidirectional Scatter Distribution Function (BSDF). Since the BRDF is only concerned with the reflected radiation, the computation is limited to the hemisphere of directions above the interface. The word *above* implies the side of the interface from which the inbound radiation arrives. In the case of a BRTDF or BSDF, the full sphere of scatter directions is of interest.

The scattering that results from impinging radiation is a result of a variety of factors, including such elements as the direction of incident illumination, geometry of the interface, indexes of refraction of the two media, spectral distribution of the incident radiation, power of the radiation, and a host of other possibilities. Depending on the particular BRDF model being studied, there are a wide variety of other parameters that are introduced to improve the accuracy or efficiency of a BRDF evaluation.

The BRDF literature does not use incident electromagnetic radiation; rather such impinging energy is referred to as incident luminance, or irradiance. The outbound

energy is reflected luminance, or radiance. A further discussion of these two terms and their units will follow.

Fields Interested in BRDFs

The study of BRDF models is relevant to a wide range of fields, and the cross-pollination of ideas from disparate groups has produced a variety of models. The BRDF originated in the world of physics: the subject most concerned with the interaction of energy and matter. Specifically, the science of thermal radiation and heat transfer is interested in the transportation of energy through a system [23]. The field of measurement science also studies the BRDF; the scattering of light from a surface is used to determine its microgeometry. The scattering event engendered by a machined surface is studied to gauge the accuracy of a machining process. Military applications of the BRDF abound, from determining local terrain composition from satellite reconnaissance, to designing aircraft that foil radar detection through careful management of the scattering caused by the fuselage. The computer graphics field employs the BRDF to render 3D scenes of objects that exhibit real-world scattering properties. And finally, a variety of commercial industries are interested in the BRDF, including the automotive and paint industries; both are concerned with rendering accurate approximations of appearance prior to the creation of a new product. Any one of these fields is potentially interested in the improved accuracy and efficiency of BRDF models.

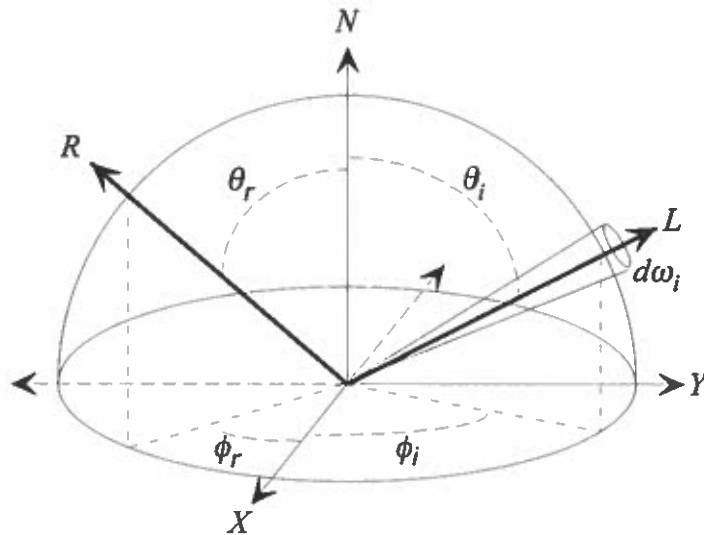


FIGURE 1. Specifying BRDF angles.

BRDF Geometry

As mentioned above, the BRDF is only concerned with the hemisphere above the surface under study. However, the discussion that follows can be easily extended to apply to the full sphere of scatter directions.

Labeling the Parameters

The geometric setting of a BRDF requires the specification of 3D angles relative to a differential patch of the surface instigating the scattering event. The patch is differential in size, therefore it can be reasonably approximated as a planar surface possessing a well-defined surface normal. Together, the normal and patch determine the orientation of the hemisphere of directions the BRDF describes. A light source casts luminance on this patch. For the moment we will assume the light to be a point source. A point source has a single 3D angle associated with it. Specifying a 3D angle requires two 2D angles to be defined on the hemisphere: the

azimuth and polar angles. The azimuth angle is measured by moving around the base of the hemisphere in a counter-clockwise direction, range 0–360°. The polar angle is specified in relation to the surface normal, range 0–90°. The zenith angle can be used in place of the polar angle if desired; zenith is 90° – polar angle. Unless otherwise specified, the polar angle will be used for the vertical component of the 3D angle throughout this document. In order to make the notation compact, the azimuth angle is labeled ϕ , while the polar angle is labeled θ .

To indicate the direction of scattering, we specify a second 3D angle which we temporarily assume to be a point collector. To differentiate between the two 3D angles, they are subscripted i and r , for the *incident* and *reflected* scatter directions respectively. Figure 1 demonstrates the labeling of the incident and reflected directions on the hemisphere centered over the differential surface patch.

Relaxing the constraint that both the light source and collector be mathematical points requires the introduction of a solid angle: the amount of area on the unit circle subtended by a region projected from the unit hemisphere. This is the region through which the illumination arrives or the scatter departs. Solid angles are specified in radians and have units of steradians. A light source impinging from the entire hemisphere would have a solid angle of π . Solid angles are typically labeled ω . Introducing the solid angle permits a variety of BRDF evaluations to be performed. These range from the incident-exitant differential solid angle situation already mentioned, to the dual forms of a differential incident angle scattered over the entire hemisphere vs. a hemispherical incident angle scattered to a differential exitant angle, to a hemispherical incident angle scattered over the entire hemisphere. These types of BRDFs are called in sequence: bidirectional, directional-

hemispherical, hemispherical-directional, and hemispherical [23]. The first form is the focus of this document, though the others are employed in applications such as rendering and measuring.

Though generally not addressed in this thesis, there are two additional parameters that should be specified for physical completeness: polarization and wavelength. Both play a role in determining scatter due to the Fresnel factor. Diffraction, fluorescence, and phosphorescence cannot be modeled unless wavelength is part of the equation.

Isotropic vs. Anisotropic

Up to this point, we have not assigned an axis to the azimuth angle lying on the plane defined by the differential patch and normal. The scattering from the surface was assumed to be radially symmetric. Such a material is termed *isotropic*; the specific ϕ_i and ϕ_r are not important. The relative angle, $|\phi_r - \phi_i|$, is important. Note that an isotropic BRDF is symmetric about the plane defined by the incident and normal vectors, hence the sign of the relative angle is irrelevant.

Many real surfaces are not isotropic; they are *anisotropic*. An anisotropic surface requires the use of both ϕ_i and ϕ_r , since the BRDF is not necessarily symmetric in either of the two ways just described. In order to use an anisotropic BRDF it must be aligned with a local coordinate system at the point of intersection with the interface. In computer graphics, the normal is available at any surface location, but to orient the BRDF a tangent vector is also required. The cross product of the normal and tangent defines the binormal. Together, these three vectors define the coordinate system. In Figure 1, the normal aligns with the Z-axis, the tangent aligns

with the X-axis, and the binormal follows the Y-axis. Using an anisotropic BRDF in a rendering system requires that the incident and reflected rays be transformed between object- and BRDF-space as determined by the basis vectors just described. The transformation is not required for isotropic BRDFs, since the relative azimuth angle does not need to be locally oriented.

Materials that exhibit anisotropic scattering behavior often possess some type of surface microgeometry that interferes with the symmetric scattering of light. The prototypical examples of anisotropic materials are rolled metal and varnished wood, although sand, textiles, and even paint exhibit anisotropic scattering. Each of these materials might be characterized by directional ridges and valleys that prevent light from scattering independent of the azimuth angle [12, 21, 25].

The BRDF Equation

The BRDF equation is the ratio of radiance to irradiance, adjusted by the cosine of the incident polar angle and the solid angle of the incident illumination. The radiance and irradiance are written in a variety of ways, but here will be represented as L_r and E_i , respectively. Note that the irradiance is only a function of the incident 3D angle, whereas the radiance is a function of both the incident and exitant angles.

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{L_r(\theta_r, \phi_r, \theta_i, \phi_i)}{E_i(\theta_i, \phi_i) \cos \theta_i d\omega_i} \quad (\text{I.1})$$

If the BRDF is isotropic, the dependence of radiance on ϕ_r and ϕ_i becomes a dependence on $\phi_r - \phi_i$:

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{L_r(\theta_r, \theta_i, \phi_r - \phi_i)}{E_i(\theta_i, \phi_i) \cos \theta_i d\omega_i} \quad (\text{I.2})$$

The $\cos \theta_i d\omega_i$ term in the denominator indicates the angle-dependent nature of the BRDF. The cosine-term introduces the reduction in flux intercepted by the differential surface patch as the angle between the normal and the polar component of the light direction increases. The solid angle indicates the amount of area on the surface of the hemisphere through which the inbound flux is intercepted.

For physical plausibility, the BRDF cannot scatter more light than is incident. Another way of saying this is that for any incident angle, the total volume encompassed by the BRDF must be less than or equal to one. Many BRDFs are normalized, thereby ensuring that the volume is identically one.

Units

The units of intercepted flux for both radiance and irradiance are watts/steradian/meter². Therefore, the units of $\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i)$ are inverse steradians (steradian⁻¹) due to the solid angle in the denominator.

Applications

Computer graphics will be the focus of the applications presented in this thesis, with the archetypal use occurring in a 3D-renderer. Rendering requires the use of the different angular formulations of the BRDF: the hemispherical-directional and directional-hemispherical formulae. To achieve a realistic approximation of the reflectance from any differential surface patch, all light impinging on a patch must be gathered. The gathered radiance is then scaled by the magnitude of the BRDF in the viewing direction (the 3D angle leading back to either the eye or the image plane). Of course it would be impossible to sample all incident directions, so a reasonable

approximation is achieved by stochastic sampling. A discrete number of rays is cast, each of which is scaled by the magnitude of the BRDF in its particular direction. Then the results are integrated. Increasing the sampling of the BRDF improves the accuracy of the approximation. The technique of stochastically sampling the BRDF is known as Monte Carlo sampling.

If the BRDF is anisotropic, the technique is a bit more complex since the BRDF must be oriented in relation to the normal and tangent at the point of intersection. Prior to casting any sampling rays, each must be transformed from BRDF- to world-space. Only then can it acquire its individual contribution to the overall reflectance.

The preceding discussion uses the BRDF to sample both incoming and outgoing directions from the surface patch; that is, the incoming direction may be either a light source or a viewing position.

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \rho_{bd}(\theta_i, \phi_i, \theta_r, \phi_r) \quad (\text{I.3})$$

This is valid because of the reciprocal nature of the BRDF; reciprocity underlies the theoretical validity of ray tracing, as presented by Whitted [30].

CHAPTER II

A BRDF VISUALIZATION SYSTEM

Visualization as a Learning Tool

“A picture is worth a thousand words.”

The original source of that quote is lost to history—yet it is an aphorism that constantly proves itself true. Just look at the GUI that is part of every modern operating system; little images abound in the form of icons announcing applications that will launch at the click of a mouse.

Likewise, in the scientific computing community pictures have become the most intuitive means to display, analyze, and study huge volumes of data. Computers can now evaluate fluid dynamic simulations that are useful in the design and testing of complex aerodynamic and hydraulic systems; the resulting numerical data from such a simulation can be of multi-gigabyte extent. By running the data through a program that generates an animated 3D representation, gigabytes of data become easily accessible. Patterns and problems can be observed and interpreted; all without building a costly prototype that might subsequently be scrapped due to unforeseen technical difficulties.

This process of building images and animations from data is known as visualization.

Visualization Tools

Both the computer graphics and scientific visualization communities have fostered the growth of tools that permit the visualization of data. These tools vary significantly in the amount of help they offer the developer in building data visualization applications. Also, the portability of such applications tend to vary dramatically depending on the tool employed.

A visualization system could be built using only a graphics library such as OpenGL. As long as a platform-dependent implementation is available, OpenGL is portable. OpenGL typically relies on hardware acceleration to provide most of its functionality, though components of the specification not available on particular hardware can be implemented in software (with a considerable performance penalty).

Using OpenGL is straightforward. However, it requires the developer to build *all* the components of the application: the user interface (UI), the geometry managers, the cameras, the rendering techniques, etc.

In contrast, visualization packages provide much of the core functionality required for building any visualization application. The Visualization Toolkit (VTK) is a package that is available as shareware. It sits on top of an underlying graphics API (such as OpenGL). VTK streamlines the generation of common primitives, transformations, color and texture application, etc. It also provides standard interaction controls for the mouse, thereby simplifying the process of revolving, scaling, and translating a particular visualization.

Commercial visualization packages provide an entire UI in which to build applications and interact with them. Advanced Visual Systems Inc. Express incorporates an application-building environment (the network editor) with a substantial

library of common components and program controls that streamline many of the routine tasks required to develop a data visualization application. The developer instantiates the library modules to build a network; the connecting wires indicate the dataflow between modules. Logical, complex networks can be built, run, and debugged interactively. The applications built in Express make use of standardized GUI controls, fostering a fast learning curve for new users of applications built in the environment; however no guarantees of understanding of the application's subject-matter are offered.

Designing and Building a BRDF Viewer

To provide easy access to the variety of BRDF models included in the Oregon BRDF Library, a visualization application has been conceived. It must be a tool that permits interactive evaluation of the models, and it must meet the following specifications:

- Provide a visualization of the scatter predicted by a BRDF by sampling the hemisphere of scatter directions (data generation and visualization).
- Be an interactive environment that permits all the models in the Oregon BRDF Library to be studied in a consistent manner.
- Provide a straightforward means to study multiple BRDF models simultaneously (employ the network dataflow paradigm).
- Package the functionality into a minimal number of network modules to hide the complexity of the implementation—the goal of the application is to explore BRDF models, not Express.

- Provide controls that allow the user to adjust the interactivity level of the application—if speed is the main requirement, the user can decrease the accuracy of the computed geometry; if accuracy is paramount, the user can sacrifice the frame rate to a more detailed visualization.
- Give the user a means to change the color of the light reflected from the surface by either changing the color of the incident light, or by changing the color of the surface itself.
- Provide hooks and guidelines for extending the visualization tool to render preview surfaces (probably in Radiance), acquire colors from GamVis (a color gamut visualization tool), and add new BRDF representation schemes to the selection of models.

The visualization system developed satisfies these goals by presenting a small number of network modules, providing standardized panels for the distinct models, adjusting the BRDF-specific panels to reflect the current parametric settings, and providing a variety of ways of accomplish the same tasks (via mouse or keyboard). Some of the goals have not been realized, including preview surface rendering and color gamut querying.

Refer to Appendix A for a complete description of the Oregon BRDF Library API. Appendix C is a reference for adding new BRDF models to both the library API and the visualization tool.

BRDFvis: An Interactive BRDF Visualization Environment

A tool called BRDFvis has been developed that permits professionals interested in the reflective properties of materials to visualize the resultant light scatter.

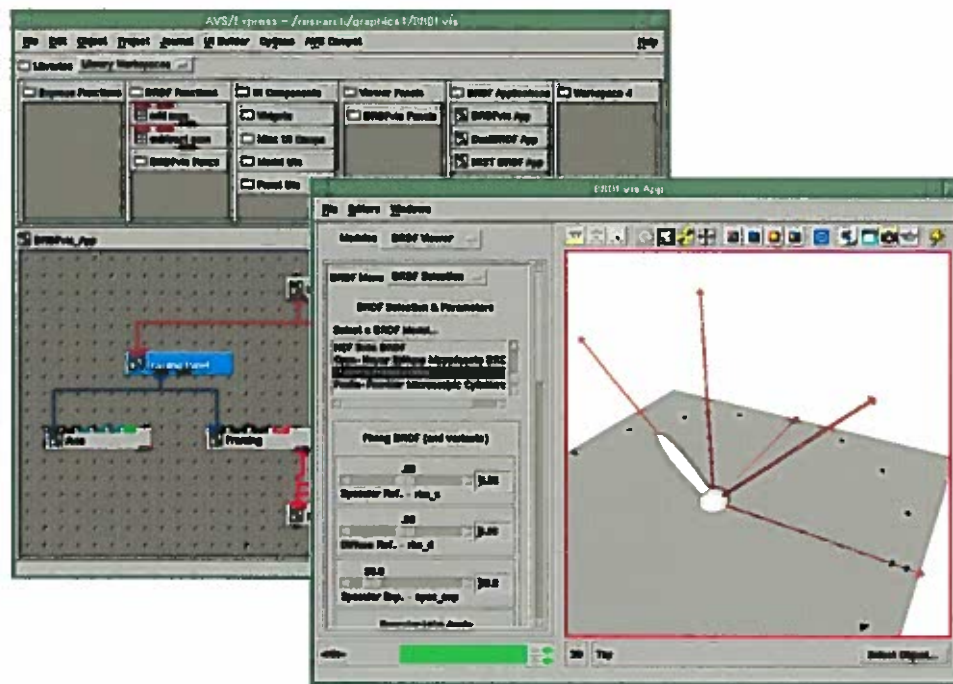


FIGURE 2. BRDFvis AVS Express application.

The software was developed using the commercial software package Advanced Visual Systems Inc. Express. This package, the Oregon Color Software, and the Oregon BRDF Library are the building blocks of BRDFvis.

Since learning Express is a subject of considerable complexity, it will be assumed that the user of BRDFvis is already versed in using Express, or has access to manuals from which to learn the standard Express controls. For the most part, this discussion will not address the common controls available in Express.

BRDF Visualization System Tutorial

Figure 2 provides an overview of what an example BRDFvis application might look like. It is comprised of two windows, the network editor and the application view. The network editor (the background window in Figure 2) is where

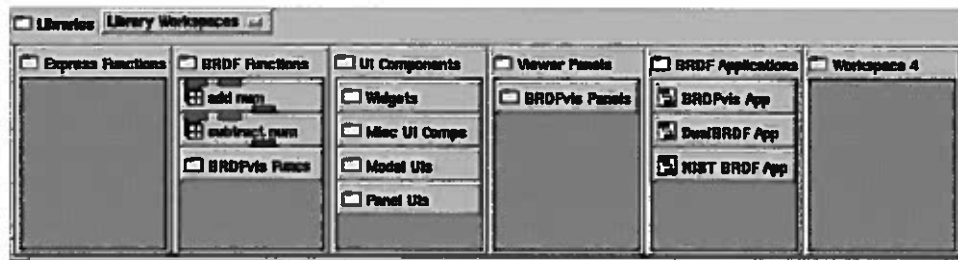


FIGURE 3. BRDFvis Library Workspaces.

applications are built and modified; adding or removing a connection will have immediate results (subject to processor speed). The application view (the foreground window in Figure 2) is the window in which the user interacts with the application; various panels can be selected, each providing access to controls that alter the visualization appearing in the 3D view. In this case, a visualization of a single Phong reflectance lobe is running.

Throughout this chapter, screen captures like Figure 2 will be used to illustrate interaction with BRDFvis. In Chapter III reflectance lobe visualizations will be used to clarify the scatter caused by a particular model.

There are two stages to using any application built in AVS Express; building the network and interacting with the application viewer. To streamline the process of building an application network, pre-built networks have been provided. Applications can also be built from scratch (this will be briefly addressed later in this chapter).

Instantiating a Network

Once Express is running the BRDFvis project, the simplest way to visualize a BRDF is by instantiating the BRDFvis App sample application located in the BRDF Applications sub-library (refer to Figure 3). The user simply pulls the

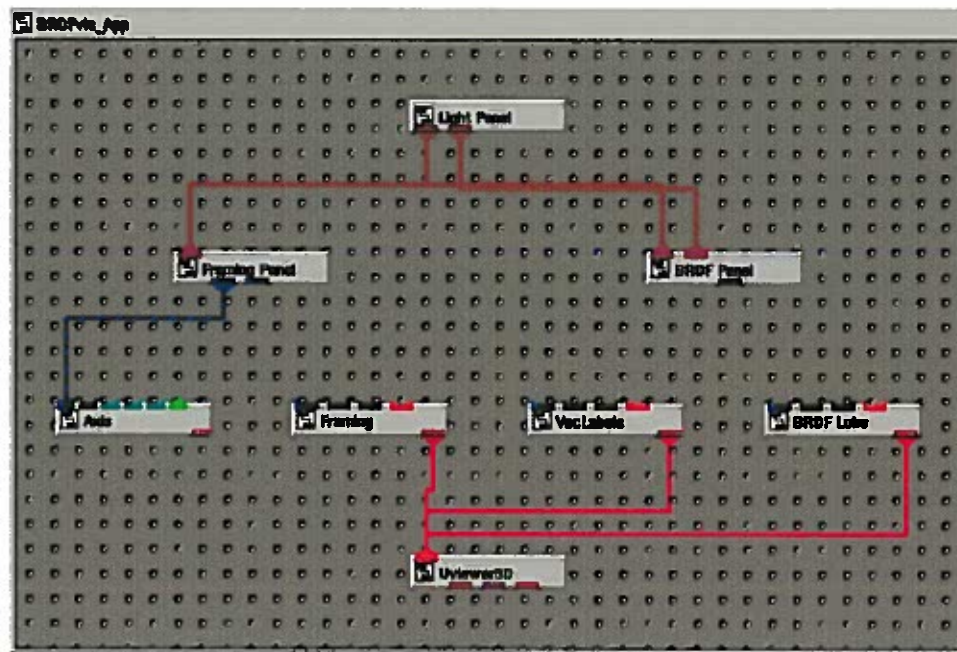


FIGURE 4. Disconnected sample network.

application down into the application workspace as would be done with any Express module.

Figure 4 shows the network that will appear once the `BRDFvis` App starts up (it will take a while to start all the supporting modules from which the application is built). This network is not fully connected, as is apparent from Figure 4. Also, the 3D viewer will be uniformly dark. Three connections will make the `BRDFvis` App fully operational. (These connections are not already present due to the sequence in which Express instantiates modules; pre-connecting them can cause Express to lock up.)

The first of the required connections joins the first output of the `Framing Panel` module to the first input of the `Framing` module. Connections are numbered left-to-right, with outputs on the bottom and inputs on the top. The first connection provides the framing geometry for the viewer: normal, tangent, binormal, light, and

mirror reflection vectors, surface plane, shadow vector, and a few other optional framing components. (Note that this output is already connected to the first input of `Axis`; leave the `Axis` connection alone.)

The second connection joins the second output of the `Framing Panel` module to the first input of the `VecLabels` module. As the name implies, this new connection provides the labels for the vectors. The labels are not strictly necessary, but they are useful as a point of reference once an adjustment has been made to the azimuth angle of incident illumination.

The final connection joins the single output of the `BRDF Panel` module to the first input of the `BRDF Lobe` module. This connection passes the reflectance lobe sample points to the module that manages the lobe geometry.

There is a fourth, optional, connection. Like the vector labels, it elaborates the reference framing provided in the 3D viewer. The single output of the `Axis` module can be joined to the single input of the `Uviewer3D` module (this particular input port will then have four inbound connections: one from the geometry manager of each of the scene's "objects"). This connection adds a wire-frame axis against which to reference the lobes. With all four connections made, the network should resemble Figure 5.

Instantiating either of the other two sample applications in the `BRDF Applications` sub-library is comparable to the instructions just presented. The main difference will be the addition of a second `BRDF Panel/BRDF Lobe` module pair. These should be connected just like the single module pair was connected above.

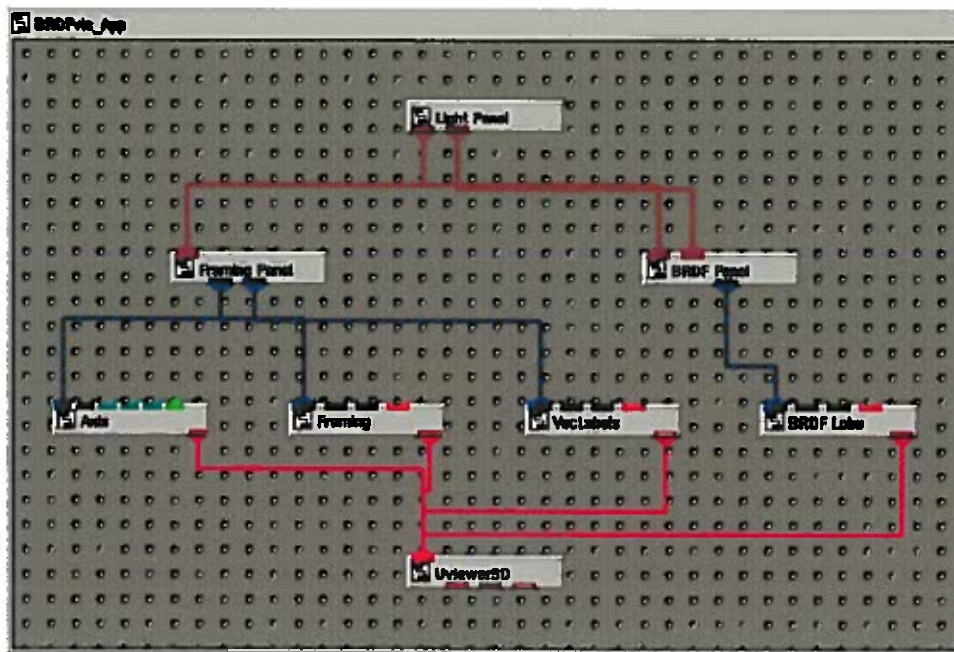


FIGURE 5. Connected sample network.

Interacting with the BRDFvis App Application

Any application built in Express has many features that are common to all Express applications. The following discussion will bypass such features to concentrate on the functionality added by the BRDFvis project. Consult the Express documentation for controls that adjust surface opacity, rendering technique, wire-frame visibility, geometry cutaways, and other effects.

There is one exception—in the Editors menu across the application menu bar, selecting the Camera option gives access to a very useful control: Auto Normalize. When updating the lobe geometry, often the 3D viewer will “jump” to re-center on the modified geometry. This can be disorienting. Turning off Auto Normalize prevents the view from “jumping.”

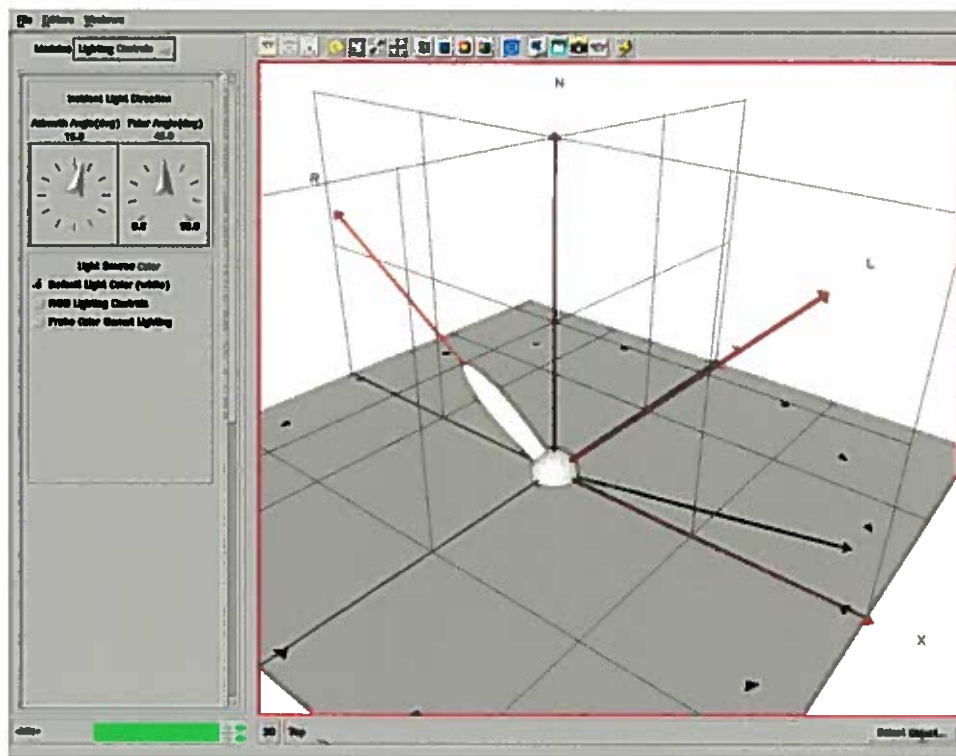


FIGURE 6. Lighting Controls panel (with Phong reflectance lobe).

Lighting Controls

Figure 6 is an example application view window (note that the reference axis is visible in this example). This particular lobe is a result of evaluating the Phong model with the current illumination direction: 15° azimuth, 45° polar. Note the presence of the labels differentiating the light (L) and mirror reflection (R) directions. The light source vector casts a shadow; this is to provide an intuitive representation of the azimuth angle in the 3D viewer.

Aside from the 3D viewer, the application window is primarily composed of standard Express controls. The features that are specific to BRDFvis are located in the Modules drop down list located near the upper left corner of the application

viewer (Figure 6). The currently selected module's panel is called **Lighting Controls**, and it occupies the left-hand side of the viewer.

The **Lighting Controls** panel provides dials and entry fields for adjusting the azimuth and polar angles of the incident light. Angles are measured in degrees. The azimuth angle is measured relative to the X-axis, with positive angles moving counter-clockwise around the normal (N); for example, a light source oriented along the Y-axis will have an azimuth angle of 90° . The polar angle is measured relative to the surface normal; hence light at normal incidence will have a polar angle of 0° . The angles can be adjusted either by using the mouse to move the dial arm to the desired angle, or by selecting the text field immediately above a particular dial with the mouse and then entering a new value from the keyboard.

The color of the incident light can be adjusted from this panel. Typically the selection should remain on **Default Light Color (white)**.

RGB Lighting Controls are also available. Upon selecting **RGB**, a new frame will appear below the light source color options. This frame, which looks a bit like Figure 10, will hold a color bar and three additional dials for adjusting the red, green, and blue components of the incident illumination (on a percentage of full intensity scale). Operation of these dials is analogous to the incident angle controls. The color bar immediately above the RGB dials previews the current mix of red, green, and blue intensity percentages—the current color of the light source.

Probe Color Gamut Lighting is not enabled.

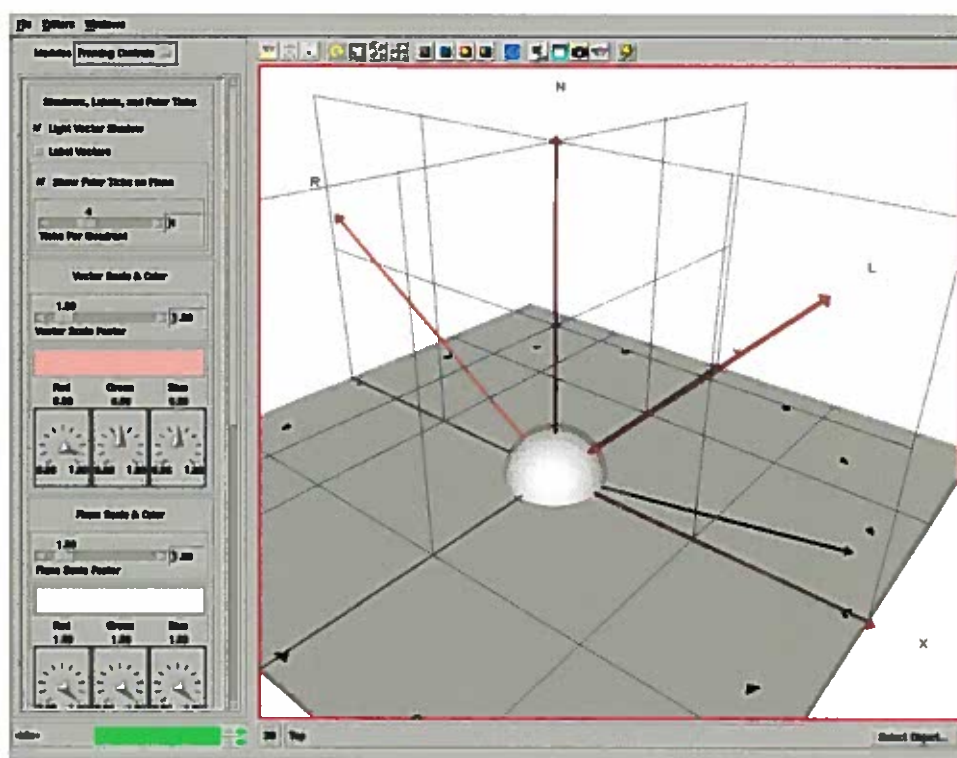


FIGURE 7. Framing Controls panel (with Lambertian reflectance lobe).

Framing Controls

The Framing Controls panel in Figure 7 is the second option available in the Modules drop-down list. This panel provides controls for adjusting the reference components of the 3D viewer; color and scale of the vectors and surface plane, presence of the incident light shadow vector, and number of azimuth angle “ticks” can all be adjusted from this panel.

The Light Vector Shadow toggle simply turns the light direction vector shadow on or off. It is on by default.

The Label Vectors toggle is not currently enabled. Labels can be turned on or off by choosing to add or remove the connection from the second output of the Framing Panel module to the first input of the VecLabels module (in the Network Editor). Refer back to the section on instantiating a sample network for details.

Show Polar Ticks on Plane toggles the tick marks that indicate equal angular divisions of the quadrants. If set, the toggle enables a slider labeled Ticks per Quadrant.

Vector Scale & Color and Plane Scale & Color (the second is available if the panel is scrolled down) can be used to scale the plane and vectors to the size of the reflectance lobe. By default both scale factors are set to one. The color adjustment component for each framing element is comparable to the control used to adjust the light source color—separate dials determine the percentage of full intensity for red, green, and blue (similar to Figure 10). Again, a preview bar demonstrates the current mix (the geometry in the 3D viewer will display the new color once the update is complete).

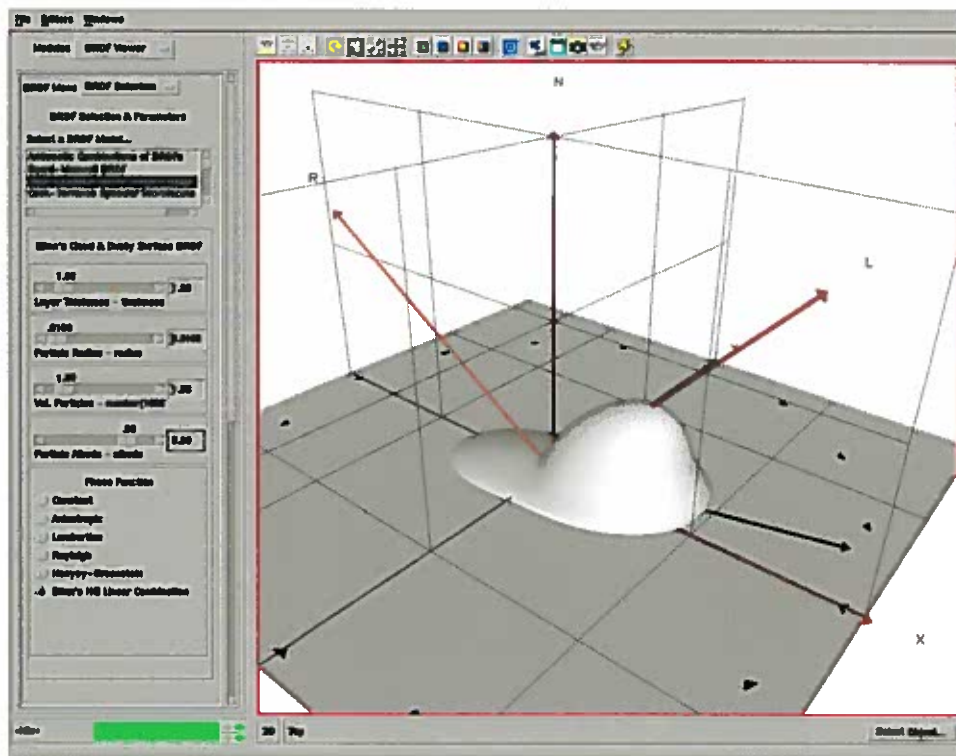


FIGURE 8. BRDF Selection panel (with Blinn reflectance lobe).

BRDF Viewer Panel

BRDF Viewer is the third module available in the Modules drop-down list. Figure 8 shows that this panel is comprised of a hierarchy of sub-panels. A secondary drop-down list appears at the top of the BRDF Viewer panel. The new list has two options: BRDF Selection and General Controls. The BRDF Selection sub-panel is built from model-specific sub-panels.

BRDF Selection

The BRDF Selection panel shown in Figure 8 is the core of BRDFvis. This panel provides the user with the capability to view one or more of the available reflectance functions (multiple BRDF Viewer panels are required to view multiple

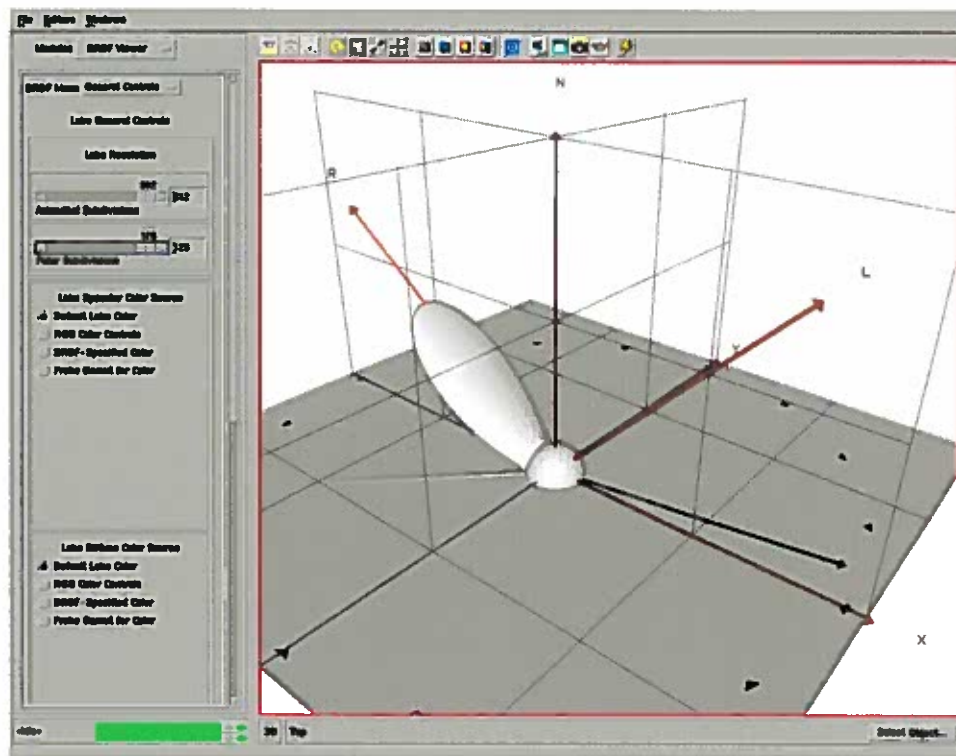


FIGURE 9. General Controls panel (with anisotropic Ward reflectance lobe).

models simultaneously). The type of model for immediate visualization is selected from the scrollable list of alternatives; in Figure 8 we are currently looking at Blinn's Clouds and Dusty Surfaces model [2]. Each BRDF model selection triggers the display of a model-specific sub-panel in the lower portion of the BRDF Selection panel. This model-specific sub-panel permits the user to study the effects of adjusting the parameters provided by each model.

For a discussion of the parameters available for any particular model, refer to the survey of models presented in Chapter III. Additional notes on model implementations in BRDFvis can be located in Appendix B.



FIGURE 10. Color Control for the reflectance lobe’s spectral component.

General BRDF Controls

The second option available from the **BRDF Menu** drop-down list is the **General Controls** sub-panel. This sub-panel allows the resolution of the lobe, as well as the lobe color to be adjusted (refer to Figure 9). The default values for the **Lobe Resolution** are 128 azimuth and 64 polar subdivisions. For lobes with larger gradients (or discontinuities—typically not physically plausible), a higher resolution may be required to generate a sufficiently smooth geometry. The Ward model shown in Figure 9 is rendered at 512 by 128. There is a tradeoff between smoothness and interactivity; more polygons are required to generate a smoother BRDF, and more polygons take longer to render. The slow-down due to higher lobe resolution is especially noticeable when multiple BRDFs are visualized simultaneously.

Consistent with the incident light and framing elements, the lobe can have associated color properties. Figure 10 shows the panel available for adjusting the reflectance lobe’s spectral color. There are separate UIs for specifying both the specular and diffuse colors, however for some reflectance functions such a distinction is inappropriate (such as with measured BRDFs). In such instances, the specular color will be ignored. If a color other than white is specified for the light source, the lobe color will be the product of the light and the computed reflectance lobe colors.

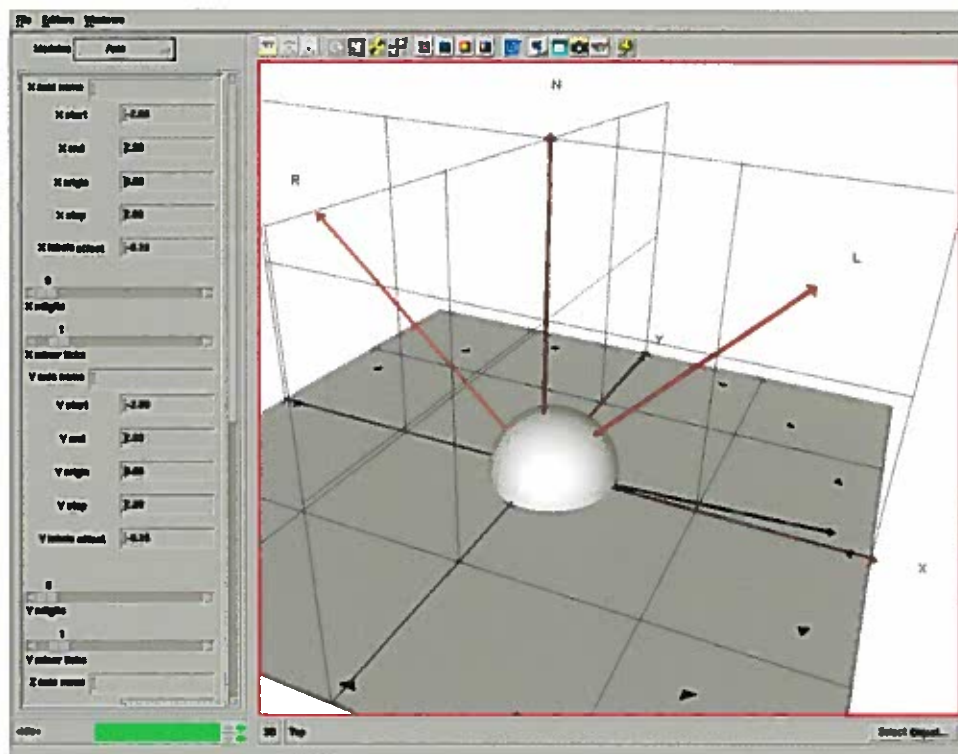


FIGURE 11. Axis panel (with Oren-Nayar reflectance lobe).

Adjusting the lobe color is straightforward, using controls comparable to those used for adjusting the light color. The one difference is the presence of additional options in the list of potential color sources. Probe Gamut for Color and BRDF-Specified Color are currently non-functional.

Axis Panel

The Axis panel shown in Figure 11 is the last of the options in the Modules drop-down list. The Axis panel simply permits the modification of the optional axis displayed in the 3D viewer. Since it is just a slightly customized version of a standard Express module, refer to the Express documentation for further details.

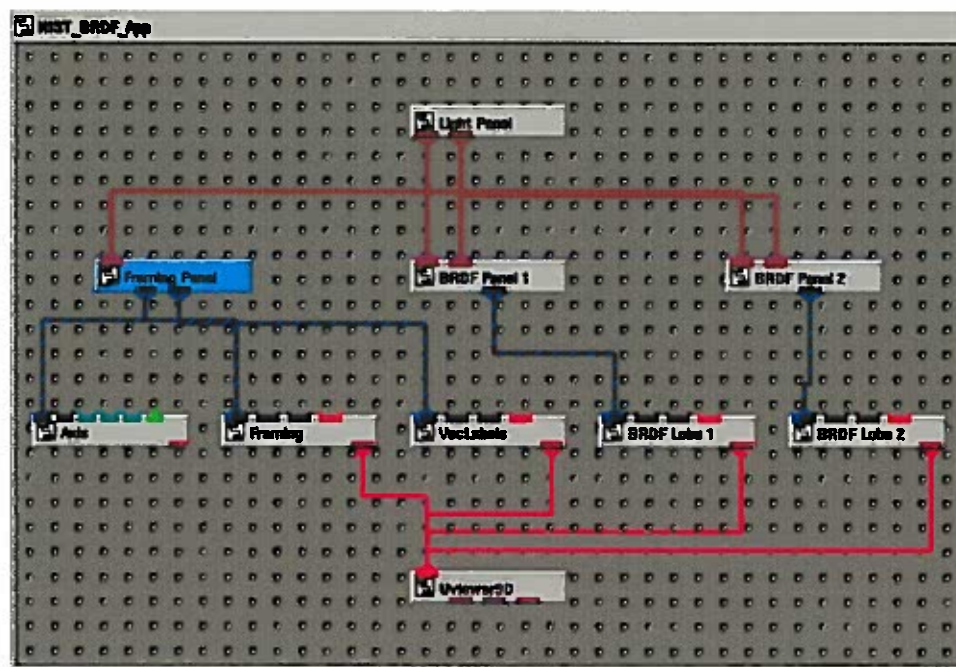


FIGURE 12. Dual BRDF visualization network (without the axis connection).

Visualizing Two BRDFs

To visualize two BRDFs simultaneously, instantiate either of the other two applications in the BRDF Applications sub-library (refer back to Figure 3). Make the connections as described previously, with the addition of the connections required between the second BRDF Panel/BRDF Lobe module pair. Once all the connections are made the network will resemble Figure 12.

In the application viewer, the main difference between this application and the single BRDF application is the introduction of an additional choice to the Modules drop-down list. There are now two BRDF panels, one labeled BRDF Viewer #1, and a second labeled BRDF Viewer #2. Each functions in an identical manner to the panel described in the context of the single-lobed BRDF application.

Figure 13 presents an example view of two BRDFs. Note that one BRDF has

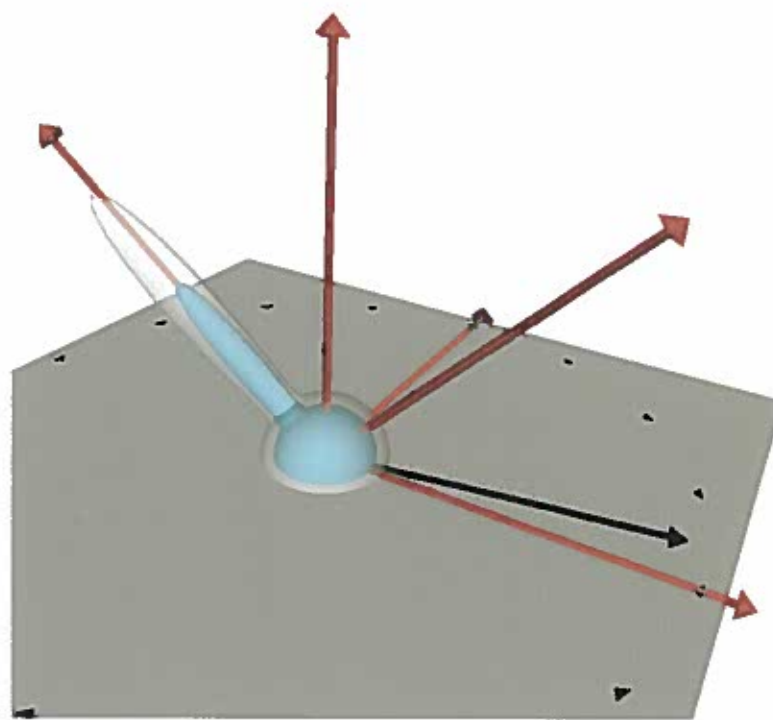


FIGURE 13. Dual reflectance lobes (opaque Phong and transparent Cook-Torrance).

been made transparent to permit the second BRDF to be visible at the same time (the opaque BRDF has also been colored light blue to differentiate it a little better).

Building Applications from Scratch

Although a discussion has not been presented heretofore on how to build networks from scratch (by instantiating modules into an application), the user can do so if desired. The panels are the components that do the work in a BRDFvis application. They acquire parametric settings from the user, and pass them along to the functions that update the geometry as required. These panels are located in the `Viewer Panels` sub-library in the `Library Workspaces` library. The other two sub-libraries, `BRDF Functions` and `UI Components`, provide all the elements from which the panels are built. There is little reason to go into these last two sub-libraries, unless the functionality of BRDFvis is to be extended or modified.

The balance of the modules in each sample application network are standard Express modules available in the standard libraries. The specific modules are: `Axis3D`, `DataObject`, and `Uviewer3D`. The last of these is provided automatically if a new application is generated with a 3D viewer.

CHAPTER III

A SURVEY OF BRDF MODELS

Modeling the scattering of light from a surface is facilitated by adopting a classification scheme. The research group developing a particular BRDF model chooses the characteristics it wishes to consider, and thereby makes the project tractable by constraining the selection of relevant scattering surfaces. Since each group makes its own choice, a wide range of BRDF models have resulted. In combination, most surfaces can be represented, even though an individual model may be severely limited.

In the course of surveying the reflectance model literature, a large assortment of models were encountered; together the BRDF models are capable of addressing nearly any material-centric or application-centric need (both impose constraints). In general, the BRDF models discovered fall into three categories, analytic, empirical, or simulated. These are determined by the methodology used by the group developing the model.

Beyond a choice of methodology, it was also observed that common approaches were often employed to develop a model: analytic models tend either to derive from physics, or are purely appearance-driven; empirical models are either data-driven or focus on creating an efficient data structure to store data; while simulated models generally follow the same approaches used for the empirical models.

Model Methodology

The goal of the research group determines the methodology employed; the three categories are analytic, empirical, and simulated.

Analytic

The analytic BRDF models are those that are evaluated by inserting parametric values into mathematical formulae. The parameters are dependent on the application as well as the particular model, but require at least the incident and exitant directions (specified by 3D angles). Solid angle, wavelength, polarization, and surface tangent may also be required. In addition, individual models employ parameters that allow the model to fit a variety of surfaces. The model-specific parameters might be material absorptance, surface microfacet slope distribution, index of refraction, specular exponent, or a range of other possibilities. The magnitude of the BRDF is the result computed using the appropriate formula.

Accuracy of the model at describing the scatter from a particular surface is dependent on the model's ability to plausibly describe a particular scattering event. The analytic models are of varying degrees of complexity, typically exhibiting a larger number of parameters and a longer evaluation time when greater physical accuracy is required. The simplest of these models, such as the Lambertian diffuse or Phong models, tend to be wildly inaccurate for most real surfaces. However, both of these models possess the desirable property of being quick to evaluate.

All the analytic models employ one or more parameters that fit the model to real surfaces. Because of this, they all suffer from a common problem: fitting the parameter(s). In the case of the Phong model, such a fit is almost impossible for most

real surfaces. For more complex models, such as the Cook-Torrance or Ward models, a reasonable fit for certain types of reflectance distributions may be achievable. However, tuning the parameters is either a black art, achieved through extensive trial and error, or the result of expensive physical measurement and approximation. Not all researchers interested in BRDFs have access to the necessary measurement apparatus.

Empirical

The analytic models are all faced with the problem of fitting a set of parameters to a real surface; the fitting problem can be avoided entirely by using an apparatus to measure the actual scattering from a particular surface. If only the spectral distribution of light is important, a spectrophotometer is used. If the geometric microstructure is important, a goniophotometer is used. And if both are being studied, as is the case with most BRDFs, a goniospectrophotometer is employed. The resultant data can be used as a lookup table when evaluating the BRDF.

Ideally, the appropriate device would measure every incident-exitant direction pair. In practice, a subset of discrete samples is measured, and the intermediate positions are reconstructed via interpolation of the measured sample points. Since physical measuring devices tend to produce noisy data, the data must be pre-filtered to remove discontinuities in the BRDF. This pre-filtering is acceptable as long as the BRDF does not contain discontinuities; typically this is a valid assumption.

Simulated

The final methodology incorporates aspects of both the analytical and empirical categories. A simulation is performed that models the scattering from a patch of surface microstructure; the scatter is captured, and the resulting data can be used for an empirical evaluation. This technique is being explored at the University of Oregon, as well as by researchers at NIST and elsewhere.

Approaches Used to Derive a Model

As already noted, the methodology employed by a research group can be further categorized by labeling the approach as physics-derived, data-driven, data-structured, appearance-based, or linear combination. Although each approach might be expected to be methodology-bound, many of the models transcend specific categorization.

Physics-derived

This classification of BRDF representation attempts to model surface reflection by starting from first principles. The result is an analytical model that tries to be as true to the surface physics as possible. Models which use this approach include: Lambertian, Minnaert, Hapke, Cook-Torrance, He-Torrance, Oren-Nayar, and Blinn.

Data-driven

Starting from a set of measured data, these models attempt to find a representation that best fits the values. An analytical expression is again used for

the representation scheme. Models that fit this include: Ward, Lafortune, Beard-Maxwell, and the NEF and Columbia-Utrecht databases.

Data Structure

Using data obtained from measurements or produced by simulation, these models try to find an efficient method to sample the values. These techniques employ some type of data structure to store the BRDF reflectance values. Models that are data structure based include: Cabral, Westin, and Gondek.

Appearance-based

These models begin with the objective of reproducing a certain appearance phenomenon such as plastic or rolled metal. They are typically developed in the context of realistic image generation, and they often trade physical accuracy for computational efficiency. Phong and Poulin-Fournier fit this methodology.

Linear Combination

Many of the models are created by linearly combining a number of different analytical functions together. Generalizing this idea leads to a linear combination model able to employ any other model as a component of a larger formula.

A Simple Beginning: Diffuse

Perhaps a more useful approach than a discussion of methodology and specific approach is to look at the models from an historical or evolutionary perspective. A BRDF is ultimately just a model of reflection, and it had to start somewhere. A

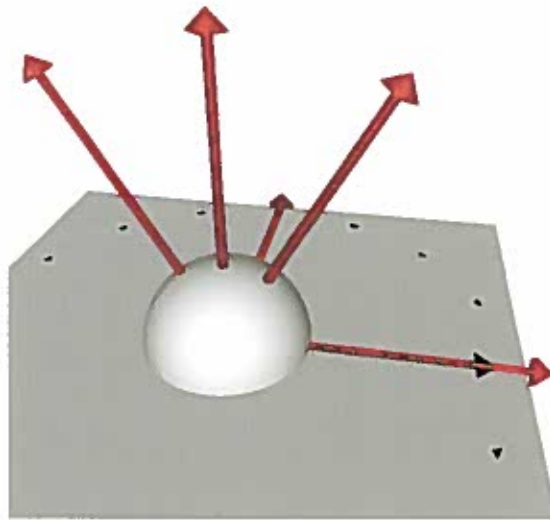


FIGURE 14. Lambertian reflectance lobe.

simple beginning is the best, and as individual components are understood, more complex elements can be addressed. Applications for particular models affect the evolution of new models; a means to evaluate them efficiently drives the effort to produce more comprehensive models.

At the core of the entire discussion of BRDFs lay physics; the interaction of energy with matter. But physics is a multi-layered discipline, just as the various elements of dealing with BRDFs are multiple. In order to deal with the subject, someone had to start somewhere, and the most logical place was to explore objects which appeared to exhibit uniform scattering characteristics. We term such surfaces diffuse reflectors.

Lambertian

Lambert was the first to present a model for the scattering of irradiance from a surface. He assumes the scattering to be constant for all exitant directions. This type of surface is known as an ideal diffuser. The constant scattering lobe shown

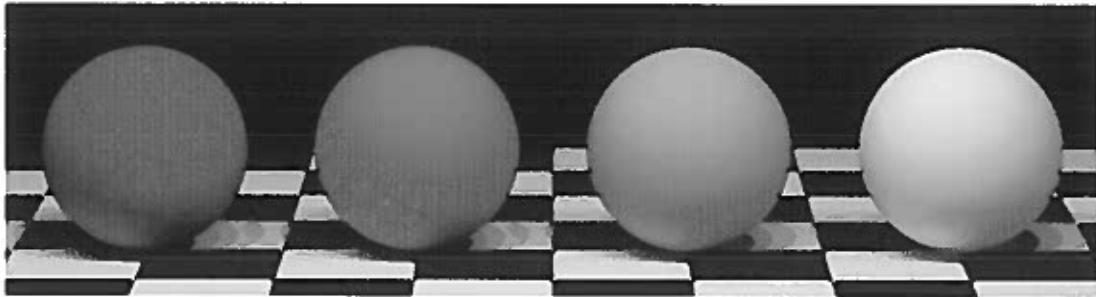


FIGURE 15. Lambertian spheres: $\rho_d = .1, .2, .4, .8$.

in Figure 14 is a reasonable approximation for sufficiently rough surfaces. Such a surface would be comprised of randomly oriented facets on a microscale, theoretically generating a constant scatter.

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{\rho_d}{\pi} \quad (\text{III.4})$$

The total hemispherical reflectance is represented by ρ_d in equation III.4. It is divided by π to normalize the distribution over the hemisphere.

Note that the formula does not demonstrate the cosine-dependent nature of the scatter. The intensity of the incident illumination will be scaled as a function of the orientation of the differential surface patch to the irradiance.

Though Lambert's model is still used as a computationally efficient approximation to the uniform diffuse component of a scattering event, few terrestrial surfaces exhibit the behavior predicted by the model. The spheres in Figure 15 might be visually interpreted as coated in flat paint of varying achromatic intensity.

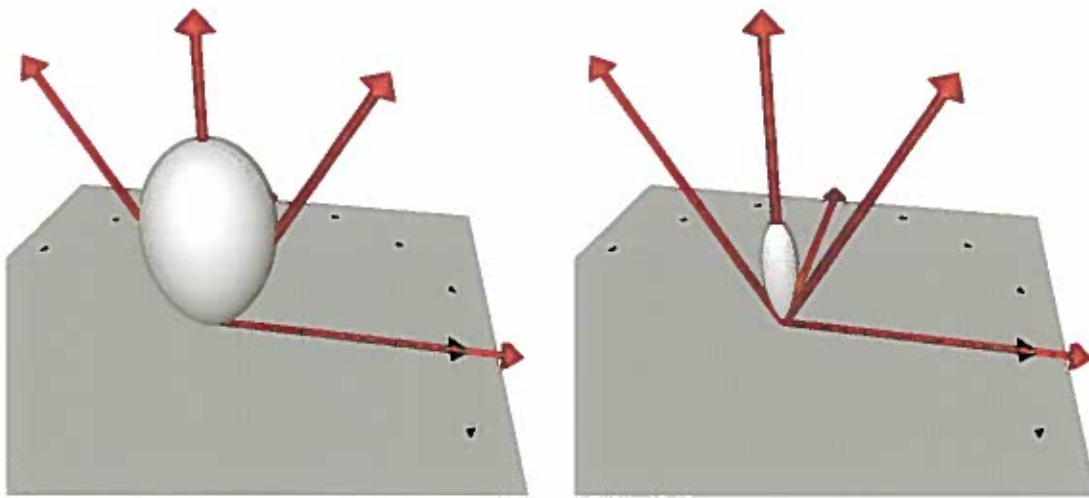


FIGURE 16. Minnaert reflectance lobes: $k = 4, 16$.

Minnaert

Minnaert chose to derive a better model for describing a non-terrestrial surface: the lunar surface. The lunar surface is interesting since it is a diffuser, and yet it behaves in a distinctly non-Lambertian manner (Figure 16). When full, the moon does not appear sufficiently spherical as predicted by Lambert's model. Instead, it appears almost disc-like. Minnaert derived a simple modification to Lambert's model to achieve this effect [16].

He observed that the lunar surface exhibits extra lightening toward the horizons; by introducing a limb-darkening parameter k , he can thereby achieve a flattening of the intensities across the lunar expanse. Though the model does a good job at flattening the intensities, it fails to address the physics of what might be causing the flattened appearance; his technique seems rather arbitrary. In more scientific terminology, he fails to address the types of behavior which have been

identified since: retroreflection, and forward- or back-scattering.

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{\rho_d (\cos \theta_r \cos \theta_i)^{k-1}}{\pi} \quad (\text{III.5})$$

In Minnaert's formula, the total hemispherical reflectance is again ρ_d . The k parameter offers more control over the rate of falloff as the polar angle to the illuminant or viewer increases. Note that the Lambertian model is included as a degenerate case when $k = 1$.

Introducing Reflections: Specular

Few real surfaces come close to matching the behavior of the ideal diffuse BRDF, nor the model developed by Minnaert. These two models assume a uniform distribution of light that is not dependent on how the exitant angle aligns with the mirror direction. But many glossy surfaces do exhibit a directional component—a specular reflection or highlight. Incident illumination is scattered in a more or less concentrated manner in the direction mirrored about the surface normal relative to the incident direction.

Phong

Bui-Tong Phong developed an appearance-based model to simulate this effect [20]. He introduces a cosine lobe which is oriented along the mirror-reflection direction (Figure 17). Scatter which falls within the lobe will be considerably more intense than that which falls outside the cosine lobe. To control the focus of the intense scatter, the cosine lobe's exponent can be adjusted to simulate various materials: a low exponent will generate a larger specular highlight, while a large exponent

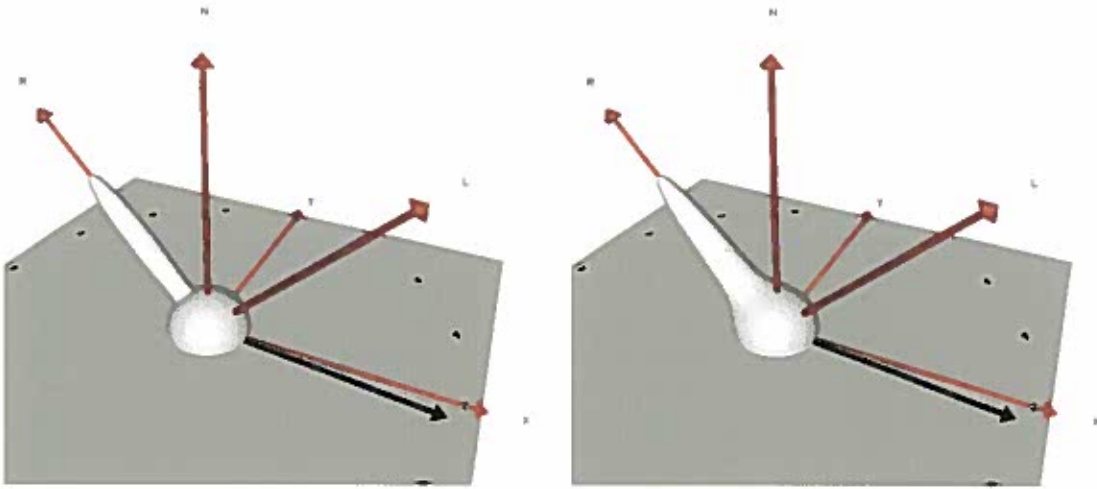


FIGURE 17. Phong reflectance lobe and Schlick rational approximation.

will make the peak more precise. This corresponds to a rough plastic (for the low exponent) and a glossy plastic (for the large exponent). Smooth finish metals will also exhibit a precise highlight.

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \rho_s \cos^n \psi + \frac{\rho_d}{\pi} \quad (\text{III.6})$$

There are now two reflectivities; ρ_s corresponds to the amount of reflectance concentrated in the specular lobe, while ρ_d is still the amount associated with ideal diffuse reflectance. The sum $\rho_s + \rho_d$ should be less than or equal to one.

The first term of equation III.6 calculates the specular lobe, and is dependent on the angle ψ which corresponds to the angle between the mirror-reflection and exitant directions; Figure 18 demonstrates the angle between $(\theta_i, \phi_i \pm 180^\circ)$ and (θ_r, ϕ_r) . By varying the specular exponent, n , the focus of the highlight can be adjusted.

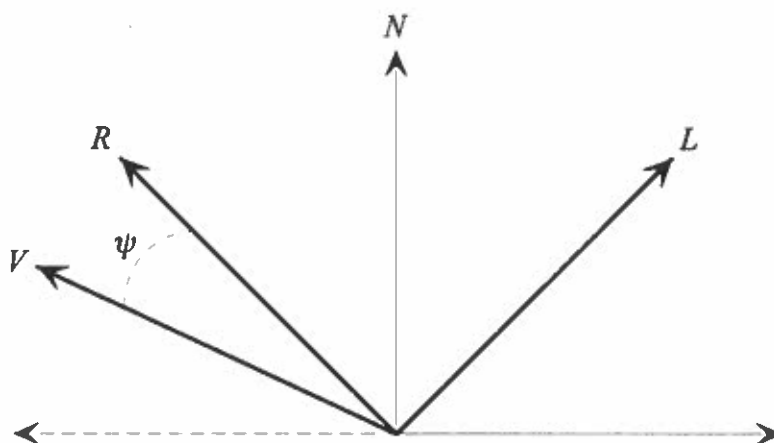


FIGURE 18. In the Phong model, the angle ψ occurs between R and V .

The second component of the equation is an ideal diffuse term; this permits the model to approximate a range of plastic-like surfaces with varying degrees of surface roughness.

Various researchers have proposed optimizations and improvements to Phong's model. One optimization suggests using the cosine of the angle between $\langle \theta_r, \phi_r \rangle$ and the half-angle vector (refer to Figure 28 for the direction half-way between $\langle \theta_i, \phi_i \rangle$ and $\langle \theta_r, \phi_r \rangle$). Another idea, proposed by Schlick [22], is to use a rational approximation of the cosine lobe to shorten the execution time of the model. Figure 17 compares the reflectance lobes using the original formula and Schlick's approximation. Figure 19 demonstrates the change in appearance caused by adjusting the specular exponent; The rightmost two spheres are identical except that one uses Phong's cosine lobe and the other uses Schlick's rational approximation.

Note that the Phong model is not truly a BRDF unless it is normalized, thereby making it physically plausible. Lewis presented a means for achieving this [15].

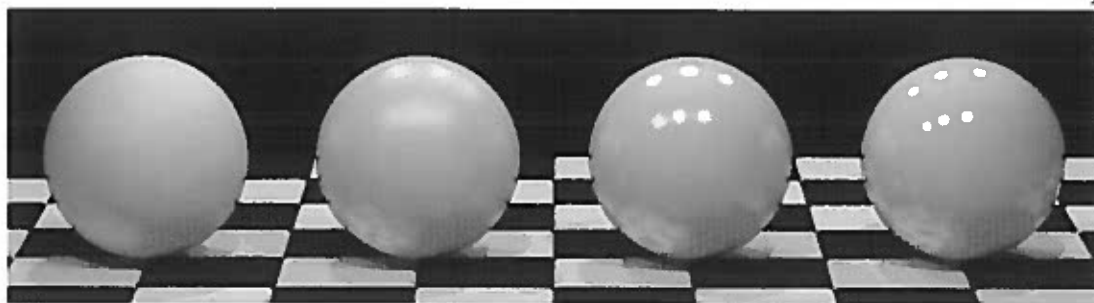


FIGURE 19. Phong spheres. The rightmost sphere uses Phong's cosine lobe: $n = 100$. Left three spheres use Schlick's rational approximation: $n = 4, 20, 100$.

Back to First Principles: Physics

Up to this point, all the models have failed to describe their assumptions about the underlying geometry of the surface generating the BRDF. The Lambertian model requires some unquantified form of isotropic roughness. The Minnaert model adjusts the Lambertian model to fit observed lunar reflectance. And Phong adds an arbitrary mirror-reflection cosine lobe to introduce specular highlights.

They all fail to employ geometric optics in consideration of the surface topology.

Cook-Torrance

Robert Cook and Kenneth Terrence [4] applied the theoretical model derived by Torrance and Sparrow [24] to computer graphics. They view a surface as a statistical distribution of planar microfacets that specularly reflect incident illumination. By considering a patch of these microfacets and integrating the reflectance over the entire patch, a physically-based directional lobe is achieved.

Figure 20 demonstrates the cross-section of a patch of microfacets. Cook and Torrance model the microfacets as a set of symmetric v-shaped grooves; the inherent roughness of these grooves is specified through the standard deviation of the



FIGURE 20. Microfacet cross-section (Cook-Torrance, He-Torrance, Oren-Nayar).

microfacet slopes (σ). The distribution of these grooves can then be computed relative to δ , the angle between the normal and the half-angle vector (refer to Figure 28).

$$D = \frac{1}{4\sigma^2 \cos^4 \delta} \cdot e^{-\tan^2 \delta / \sigma^2} \quad (\text{III.7})$$

Notice that each microfacet will have a local normal which determines the direction of specular reflection for that particular facet. Some of the radiance will be blocked by intervening ridges formed by neighboring microfacets. Also, some of the incident radiance will be blocked by these neighboring ridges. These two effects, known as masking and shadowing, explain the introduction of a geometric attenuation factor G .

$$G = \min(1, G_s, G_m) \quad (\text{III.8})$$

$$G_s = \frac{\cos \delta \cos \theta_i}{\cos \frac{\alpha}{2}} \quad (\text{III.9})$$

$$G_m = \frac{\cos \delta \cos \theta_r}{\cos \frac{\alpha}{2}} \quad (\text{III.10})$$

The factor due to shadowing is G_s , while that due to masking is G_m .

Cook and Torrance did not stop at considering geometric optics, they also wanted to correctly model wavelength-dependent aspects of surface reflection; the

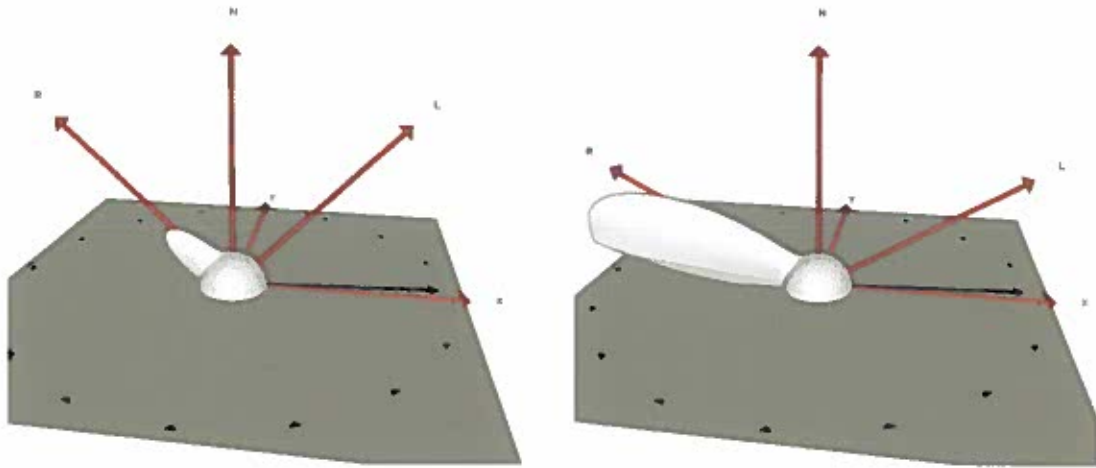


FIGURE 21. Cook-Torrance reflectance lobes: $\sigma = .15$, $n = 1.6 - .2i$, $\theta_i = 45, 60^\circ$.

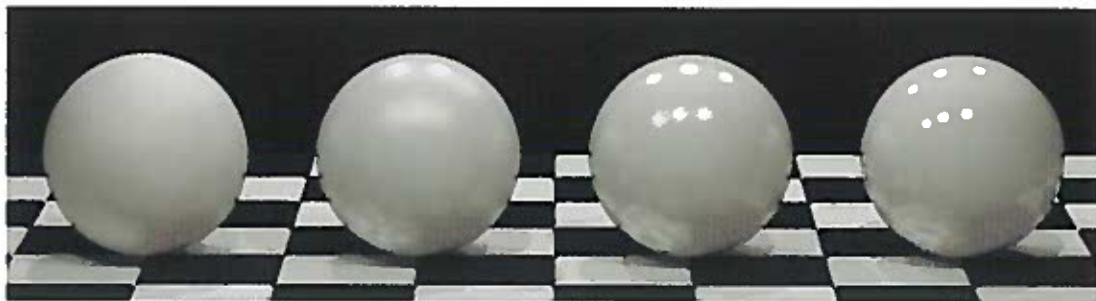


FIGURE 22. Cook-Torrance spheres with decreasing roughness: $\sigma = .3, .15, .08, .04$.

Fresnel term achieves this goal:

$$\rho_\lambda(\lambda, \theta_i) = \frac{1}{2} \cdot \frac{\sin^2(\theta_i - \theta_t)}{\sin^2(\theta_i + \theta_t)} \cdot \left[1 + \frac{\cos^2(\theta_i + \theta_t)}{\cos^2(\theta_i - \theta_t)} \right] \quad (\text{III.11})$$

The angle θ_t is a function of the indexes of refraction of the media defining the interface: $\sin \theta_t = \frac{n_1 \sin \theta_i}{n_2}$. Here n_1 is the index of refraction of the medium through which the radiance arrives, while n_2 is that for the surface material.

Combining these three elements, The Cook-Torrance BRDF equation becomes:

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i, \lambda) = \frac{D \cdot G \cdot \rho_\lambda(\lambda, \theta_i)}{\cos \theta_r} + \frac{\rho_d}{\pi} \cos \theta_i \quad (\text{III.12})$$

The lobe has the capability of appearing almost identical to the Phong model for small polar angles. However, Figure 21 shows that as θ_i increases the specular peak departs from the mirror-reflection direction and masking effects due to the surface microgeometry truncate the underside of the directional lobe. The spheres in Figure 22 are rendered with the Cook-Torrance BRDF. Compare them to those rendered with the phong model in Figure 19.

He-Torrance

Although Cook and Torrance address geometric optics, a complete model should also address physical, or wave optics. A complete model should include light polarization, and should not make the ideal diffuse approximation present in the Cook-Torrance BRDF.

He et al. [10] offer a self-described “comprehensive” physical model that addresses a broad spectrum of deficiencies present in earlier models: polarization and directional Fresnel effects; a more thorough formulation of the statistically-described surface geometry; an “effective” roughness calculation that is dependent upon the incident and reflected angles; and a geometric attenuation factor with better continuity than the one used in Cook-Torrance [4].

Although the model in its entirety addresses polarization, only the equations for unpolarized light will be presented here.

He et al. begin with a different form of the BRDF equation I.1:

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \rho_{bd,sp} + \rho_{bd,dd} + \rho_{bd,ud} \quad (\text{III.13})$$

They introduce their comprehensive components into equation III.13:

$$\rho_{bd,sp} = \frac{\rho_s}{\cos \theta_i d\omega_i} \cdot \Delta \quad (\text{III.14})$$

$$\rho_{bd,dd} = \frac{|F|^2}{\pi} \cdot \frac{G \cdot S \cdot D}{\cos \theta_i \cos \theta_r} \quad (\text{III.15})$$

$$\rho_{bd,ud} = a(\lambda) \quad (\text{III.16})$$

The model relies on several parameters, two of which are wavelength-dependent. Surface roughness σ_0 describes the height of the peaks, while the auto-correlation τ describes the distance between peaks. The index of refraction $\bar{n}(\lambda)$ and the uniform diffuse component $a(\lambda)$, are both wavelength-dependent.

The terms of $\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i)$ are mostly easily explained in comparison to the terms previously presented in the Cook-Torrance model (equation III.12). (Note that any parameter or term not derived in the following paragraphs can be located in Appendix D.)

The first term, $\rho_{bd,sp}$ is new. It introduces a specular peak in the mirror-reflection direction which was absent from Cook-Torrance. The specular reflectance, ρ_s is a function of the Fresnel factor as well as the shadowing/masking factor S , the effective surface roughness σ , and the incident and reflected directions. The presence of the specular term is controlled by a Dirac-delta function Δ . Note that σ , the effective roughness, is a function of both σ_0 and the incident and reflected

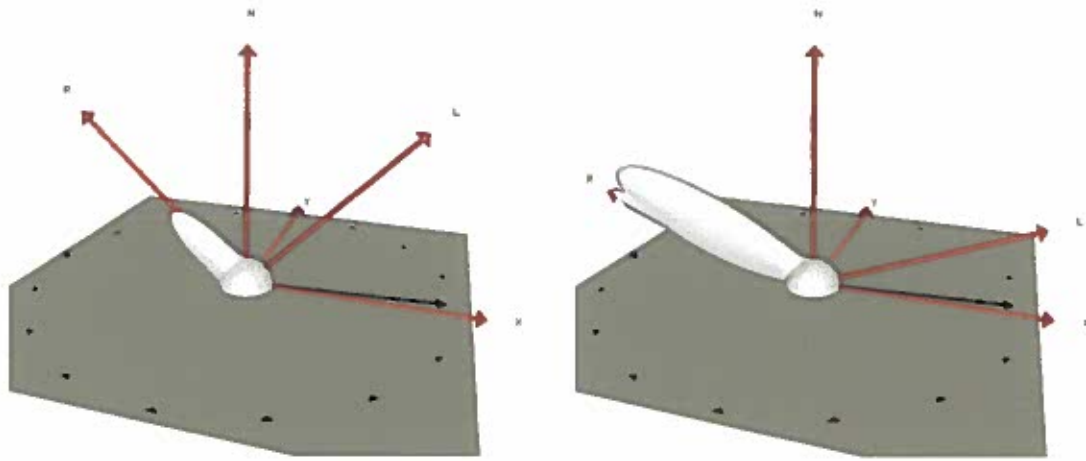


FIGURE 23. He-Torrance reflectance lobes: $\theta_i = 30, 60^\circ$.

directions.

$$\rho_s = |F|^2 \cdot e^{-g} \cdot S \quad (\text{III.17})$$

$$\Delta = \begin{cases} 1 & \text{if in specular cone} \\ 0 & \text{otherwise} \end{cases} \quad (\text{III.18})$$

The second term, $\rho_{bd,dd}$ is a new version of the directional diffuse component introduced by Cook-Torrance. G is the improved geometric attenuation factor; D is the distribution caused by the microfacets; S is the shadowing/masking factor as noted previously. The Fresnel term influences the He-Torrance directional diffuse component, as it did in the Cook-Torrance model.

The final term, $\rho_{bd,ud}$ is a substitute for the uniform diffuse assumption present in Cook-Torrance. The term $a(\lambda)$ is constant, but wavelength dependent; such values can be acquired through experiment or simulation. At run-time $a(\lambda)$ can be efficiently stored in a lookup table. In the absence of either simulated or experimental data a Lambertian model could be substituted for $a(\lambda)$. To assert the accuracy

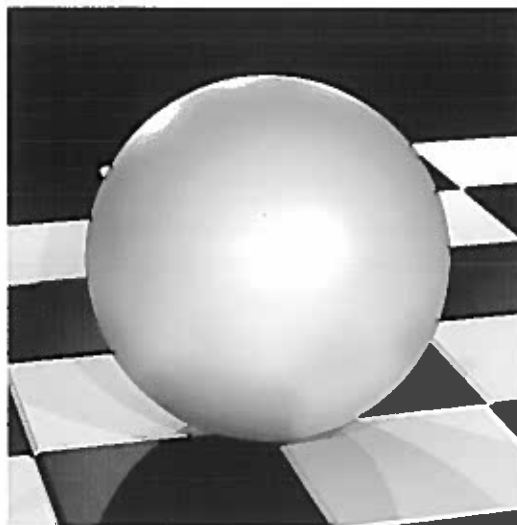


FIGURE 24. He-Torrance sphere: $\sigma_0 = .15\mu m$, $\tau = 1.5\mu m$, $\lambda = 550nm$, $n = 1.6 - .2i$.

and completeness of the model, He et al. measured the surface roughness properties of four materials and fitted the model's parameters to them. They also experimentally measured the BRDF of all four materials. Normalizing and comparing the predicted and measured BRDFs demonstrated a good match between the opposing techniques. Figure 23 demonstrates a potential He-Torrance reflectance lobe for a real surface at two polar incident angles. Figure 24 is a sphere rendered with the settings used to create the reflectance lobes in Figure 23. Note the intense highlights grazing the top of the sphere caused by three lights located behind the sphere. The intense highlight on the face of the ball is due to lights above the camera. The reflection of the tiles in the surface of the ball is subtle, and increases with grazing angle.

Although the physical accuracy of the He-Torrance model was shown to be quite high, the evaluation is computationally expensive; this is due in part to a slowly converging sum. To further complicate the model's usefulness, it is unable

to deal with many common surfaces—surfaces for which reflectance is dependent on azimuth angle. Even though the model addresses polarity, wavelength, and deficiencies present in prior models, it fails to address asymmetry.

Dealing with Asymmetry: Anisotropy

Many interfaces do not behave as consistently as the preceding models imply; the manner in which incident illumination scatters, varies as a function of the azimuth angles: the BRDF is anisotropic.

Kajiya

There are two means of addressing the complication of anisotropy as proposed by James Kajiya [12]. The first is to create a mathematical equation that describes the way light is scattered from a particular bit of microgeometry. The second is to measure the BRDF of a known anisotropic material and use the resultant data either to generate an analytic approximation or to use the data directly (in a lookup table).

Kajiya pursued the first option, with an optimization inspired by the second. He mathematically defines the microgeometry, then pre-computes the scattering from it using Kirchhoff approximation. From the numerical results, he builds a lookup table that is used at run-time to evaluate the BRDF. Although robust in that it can deal with arbitrary surface microstructure, the mathematics are involved and must be solved separately for each particular type of surface microstructure.

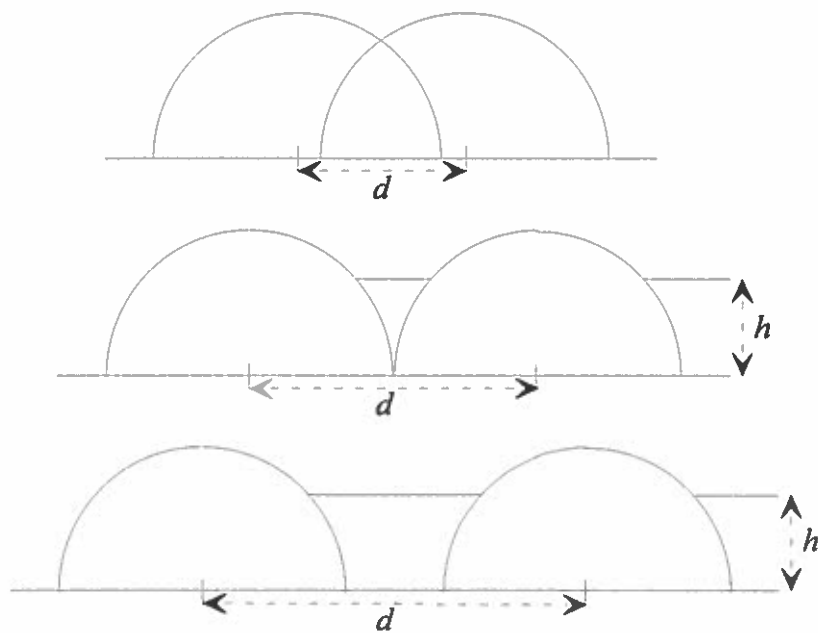


FIGURE 25. Poulin-Fournier microgeometry with d and h parameters.

Poulin-Fournier

Pierre Poulin and Alain Fournier also chose the first approach suggested by Kajiya. They create a mathematical formula to model the scattering from microgeometry, but limit the variety of microgeometry [21]. Because the microgeometry is controlled, they derive an analytical formula with a set of parameters that provide some leeway in describing its specific nature.

The Poulin-Fournier microgeometry begins with a planar surface on which long cylinders of small diameter are arrayed. Figure 25 demonstrates how the surface can be adjusted by modifying the distance between cylinder centers and the height of the floor between the cylinders. Without loss of generality, the cylinders are assumed to have a radius of one unit. The height of the floor is therefore: $h \in [0, 1]$. The distance between cylinder centers has a wider range: $d \in [0, \infty)$. If $h = 1$ or

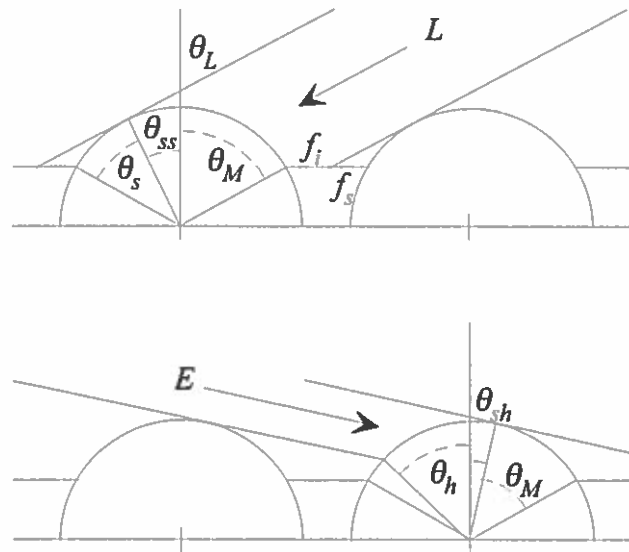


FIGURE 26. Angle specification in the Poulin-Fournier model.

$d = 0$, the model degenerates to a Lambertian distribution. Poulin and Fournier have formulae for both additive and subtractive cylinders, but only the additive version will be presented.

Poulin and Fournier make use of a factor similar to the geometric attenuation factor used in the Cook-Torrance and He-Torrance models. Such a factor addresses the shadowing and masking effects caused by the interaction of reflected rays with the microgeometry. Specifically, by projecting the incident and reflected directions onto the plane determined by the normal and cylinder cross-section, the θ_L and θ_E directions are determined; both are used in calculating the visible, illuminated, shadowed, and hidden extents of both the floor and cylinder arc. Refer to Appendix D for their derivation and Figure 26 for their interpretation.

Once all the angles have been derived, any reflectance model can be used to calculate the scatter from the floor (I_{floor}) and cylinder (I_{arc}). Note that the cylindrical arc will require a sampling strategy to cast rays in a discrete number of the

potential directions. Once the illumination has been evaluated for both components, they must be scaled by the relative amounts of floor and arc that are both visible and illuminated. The arc must also be projected onto the surface plane to ensure floor and arc contribute on an equal scale; l_v is the projected visible arc, while l_{vi} is the visible *and illuminated* projected arc.

$$l_v = \frac{\sin(\theta_h - \theta_E) + \sin(\theta_{sh} + \theta_E)}{\cos \theta_E} \quad (\text{III.19})$$

$$l_{vi} = \begin{cases} \frac{\sin(\theta_l - \theta_E) + \sin(\theta_e + \theta_E)}{\cos \theta_E} & L \text{ and } E \text{ same side of } N \\ \frac{\sin(\theta_e - \theta_E) + \sin(\theta_l + \theta_E)}{\cos \theta_E} & \text{otherwise} \end{cases} \quad (\text{III.20})$$

In the preceding equations, θ_h , θ_{sh} , θ_l , and θ_e correspond to the hiding angle, the self-hiding angle, the illuminated angle of the arc, and the visible angle of the arc respectively.

To finish the illumination calculation, the reflected intensity, I_{ref} is a weighted sum of the arc (I_{arc}) and floor (I_{floor}) intensities, scaled by the relevant visible and illuminated part of both the floor and projected arc.

$$I_{ref} = \frac{(I_{arc} \cdot l_{vi}) + (I_{floor} \cdot f_{vi})}{l_v + f_v} \quad (\text{III.21})$$

Since their model was mainly developed as a means to achieve a specific appearance, it has a few deficiencies. The most important of these is its lack of physical basis. Examining a typical reflectance lobe in BRDF_{vis}, it becomes apparent that the model has continuity problems that are not expected in a real-world scatter distribution, hence only the diffuse component is represented in Figure 27. The ar-

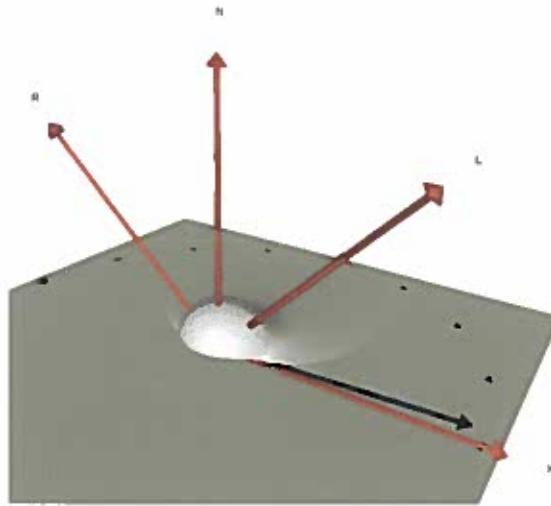


FIGURE 27. Poulin-Fournier reflectance lobe (diffuse only), cylinders are oriented along X-axis.

bitrary nature of the floor-cylinder transition generates a cutoff that complicates the plausibility of the model. In addition, their formula is not strictly normalized (this is dependent on the underlying model used to illuminate the floor and cylindrical arc). Finally, the evaluation is not computationally efficient.

Ward

Greg Ward is the developer of the final model that distinctly addresses anisotropy [25]. Ward's model is of Kajiya's second form—it begins with either a set of measured data, or known physical surface roughness, and fits the model parameters to the given data. Instead of maintaining sample data, a mathematical approximation is achieved.

Ward developed his model in response to the physically questionable results of Phong and others, as well as the computationally prohibitive formulae derived by Poulin and Fournier, He et al., and others. He designed a model that is quick

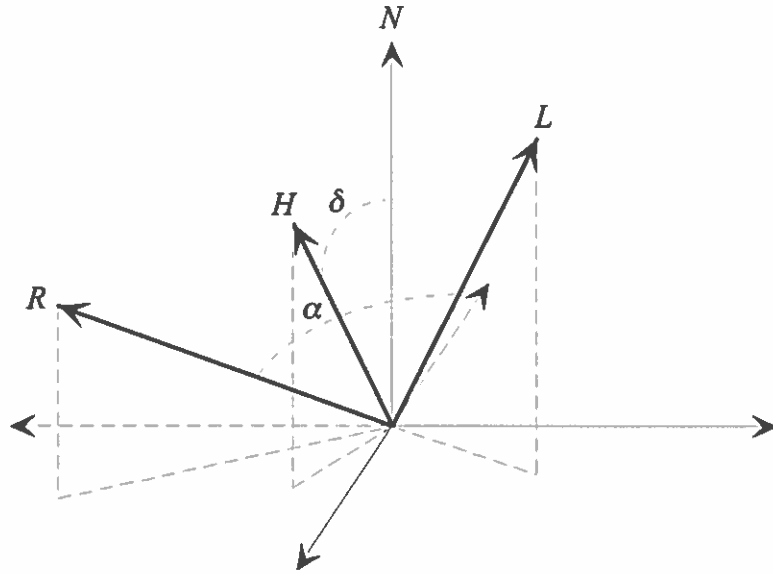


FIGURE 28. Half-angle vector H , phase angle α , and the half-angle δ .

to evaluate, physically plausible (normalized), and could be fit to both isotropic and anisotropic data sets measured with an imaging gonioreflectometer developed at Lawrence Berkeley Laboratory [25].

He achieved all these goals by fitting a gaussian to measured reflectance data. In the case of isotropic materials, the gaussian is circular; its relative size is a function of the standard deviation of the surface roughness (α , which is *not* the phase angle in this context). The angle δ is determined by the surface normal and the half-angle vector (refer to Figure 28).

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \rho_s \cdot \frac{1}{\sqrt{\cos \theta_i \cos \theta_r}} \cdot \frac{e^{-\tan^2 \delta / \alpha^2}}{4\pi \alpha^2} + \frac{\rho_d}{\pi} \quad (\text{III.22})$$

For plausibility, α must not be much greater than .2 and $\rho_s + \rho_d$ should be less than one.

Ward's model comes into its own when describing anisotropic materials. The

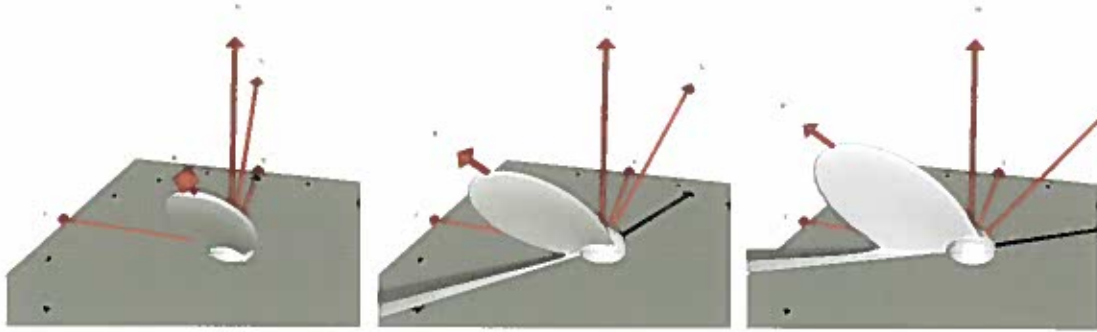


FIGURE 29. Ward reflectance lobes: $\phi_i = 0, -30, -60^\circ$.

same δ angle is used, however the shape of the gaussian is now elliptical, adjusted by the x- and y-oriented surface roughness (α_x and α_y).

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \rho_s \cdot \frac{1}{\sqrt{\cos \theta_i \cos \theta_r}} \cdot \frac{e^{\cos^2 \phi / \alpha_x^2 + \sin^2 \phi / \alpha_y^2}}{4\pi \alpha_x \alpha_y} + \frac{\rho_d}{\pi} \quad (\text{III.23})$$

The product of α_x and α_y should not exceed .2 by a substantial amount. Figure 29 shows the movement of the anisotropic gaussian as ϕ_i changes.

This elliptical gaussian model was fit to such anisotropic materials as varnished wood and rolled metal with spectacular results. As an added benefit, since the highlight is determined by a gaussian, Ward was able to derive an analytic expression for the optimal directions in which to perform Monte Carlo sampling. Refer to Appendix D for the sampling expression.

Figure 30 demonstrates increasing anisotropic roughness from left to right. The leftmost sphere in both rows is isotropic. The top row of spheres has longitudinal anisotropy while the lower row has latitudinal anisotropy.

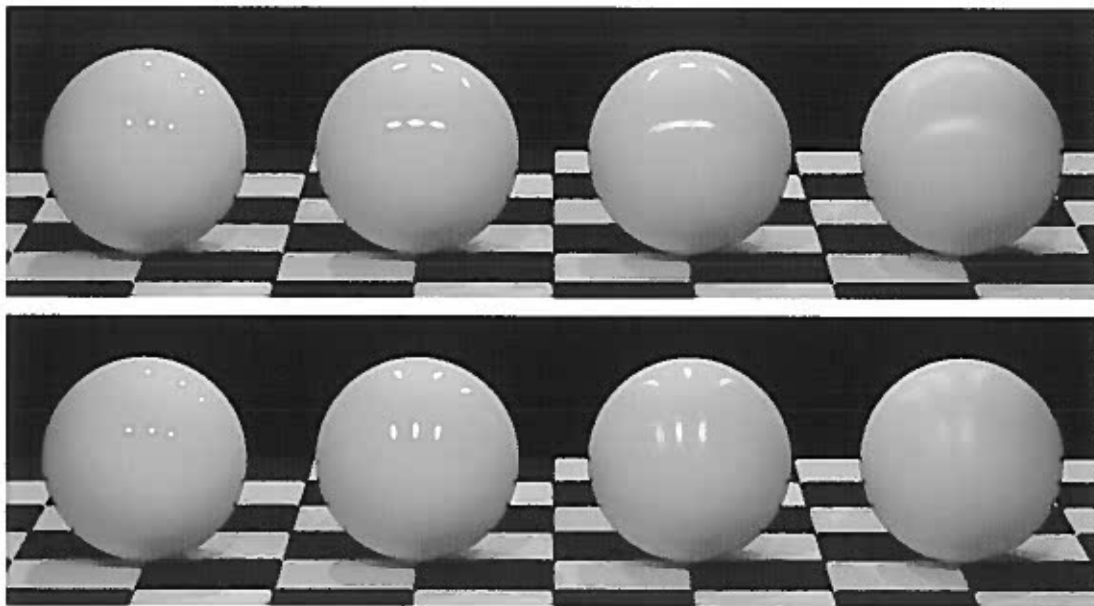


FIGURE 30. Anisotropic Ward spheres: upper row has longitudinal anisotropy; lower row has latitudinal anisotropy.

Not-So-Ideal Diffuse

Even though the preceding models intend to be thorough and physically plausible, they fail to address the underlying diffuse assumption made by Lambert (with the possible exception of He-Torrance). Few surfaces exhibit the ideal diffuse behavior posited by Lambert; surface microstructure should have an effect on the diffuse calculation.

The previously discussed Cook-Torrance and He-Torrance models both may employ the Lambertian assumption. Why should such an assumption be valid, when each BRDF goes to such length to derive an analytical solution to the specular scattering phenomenon, and in the case of He-Torrance for the directional-diffuse component? Is the statistical distribution of microfacets that model the surface microgeometry only relevant to the particular components addressed by those two

models?

Of course not. The same methodology could be applied with equal logic to the uniform-diffuse scatter. However once this jump is made, the uniform-diffuse can depart rather significantly from the Lambertian ideal; it will no longer be uniform.

Oren-Nayar

Michael Oren and Shree Nayar take this approach in building an analytical solution to the diffuse scattering problem [19]. Instead of representing the microfacets as specular reflectors, they employ ideal diffuse reflector microfacets. The resulting diffuse scatter is increasingly non-Lambertian with increasing standard deviation of the microfacet slope (σ) and polar incident angle. (The Lambertian model is included as a limit—when σ is zero.)

Oren and Nayar substitute a sum of two components for the radiance in the numerator of the BRDF equation I.1:

$$L_r(\theta_r, \phi_r, \theta_i, \phi_i) = L_r^1(\theta_r, \phi_r, \theta_i, \phi_i) + L_r^2(\theta_r, \phi_r, \theta_i, \phi_i) \quad (\text{III.24})$$

The L_r^1 term accounts for the diffuse scattering due to direct illumination (first-surface scattering). The second component, L_r^2 corresponds to the radiance due to interreflections between the microfacets. They truncate the summation of interreflections at the second bounce since the falloff will be very quick with Lambertian microfacets (as long as ρ_d is small).

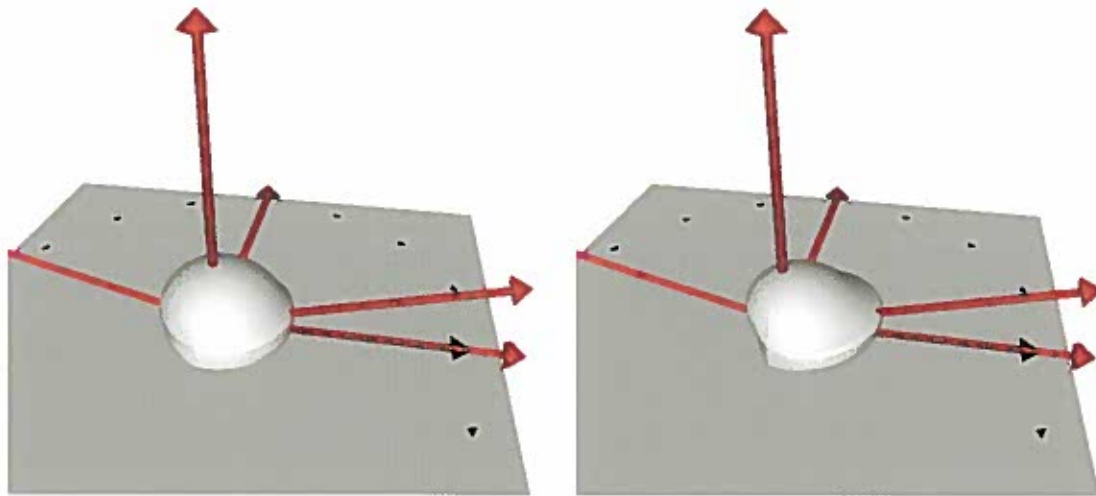


FIGURE 31. Oren-Nayar reflectance lobes: $\sigma = 10, 40^\circ$.

The formulae for the first- and second-surface scattering follow:

$$L_r^1(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{\rho_d}{\pi} E_i \cos \theta_i \left[C_1 + \cos(\phi_r - \phi_i) C_2 \tan \beta + (1 - |\cos(\phi_r - \phi_i)|) C_3 \tan \left(\frac{\alpha + \beta}{2} \right) \right] \quad (\text{III.25})$$

$$L_r^2(\theta_r, \phi_r, \theta_i, \phi_i) = 0.17 \frac{\rho_d^2}{\pi} E_i \cos \theta_i \frac{\sigma^2}{\sigma^2 + 0.13} \left[1 - \cos(\phi_r - \phi_i) \left(\frac{2\beta}{\pi} \right)^2 \right] \quad (\text{III.26})$$

Oren and Nayar define $\alpha = \max(\theta_r, \theta_i)$, and $\beta = \min(\theta_r, \theta_i)$. Note that this is not the same definition of α as employed in either Ward's model (where it is the isotropic surface slope) or in models requiring the phase angle. The formulae for the C_1 , C_2 , and C_3 coefficients can be found in Appendix D.

The BRDF resulting from Oren and Nayar's formula correctly predicts the reduced fall-off of the BRDF as distance from the specular direction increases (refer

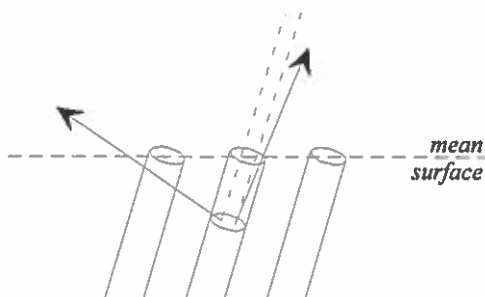


FIGURE 32. Hapke/Lommel-Seeliger scattering geometry.

to Figure 31). On a sphere, the resulting image will appear both less bright overall, and less round in comparison to a Lambertian-shaded surface. Measured surfaces such as sandpaper, clay, and cloth—each having a high surface roughness—are all predicted with good accuracy.

As an addendum, Oren and Nayar proposed a modification to the Cook-Torrance BRDF. They substitute their calculation for the uniform-diffuse assumption present in Cook-Torrance, thereby grounding the physics of the BRDF for both the specular and diffuse components.

Hapke/Lommel-Seeliger

Unlike the preceding diffuse model, the next two models were developed to more accurately model the reflectance of celestial bodies.

Hapke developed a model as a further improvement upon the model proposed by Minnaert for the prediction of the reflectance from the lunar surface. Unlike Minnaert's model, Hapke devises a microgeometric approximation for the surface of the moon, and derives an analytical model which he fits to known reflectance data.

His geometric model, shown in Figure 32, approximates the surface as a volume into which hollow cylinders permit light beams to penetrate; these beams are

scattered and attenuated as they achieve greater depth. He arrived at his model by comparing the reflectance properties of the moon to those of “porus dendritic or reticulated structures” [6, 8]. Note that Hapke uses the Lommel-Seeliger scattering law to derive his model (hence the name).

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{\rho_d S(\alpha) R_f(\alpha, g)}{\pi(\cos \theta_r + \cos \theta_i)} \quad (\text{III.27})$$

Hapke’s formulation introduces both retroreflective and forward- and back-scattered effects through the R_f and S functions. The phase angle α , the angle between the incident and exitant 3D angles (refer to Figure 28), is an argument to both of these functions. The second argument to R_f , g involves the radius of the hollow cylinders y , and the attenuation rate τ : $g = 2y/\tau$.

Note that y and τ are determined by fitting the model to available reflectance data, and the type of scattering observed in the data determines the choice of evaluations in S .

$$R_f(\alpha, g) = \begin{cases} 2 - \frac{\tan \alpha}{2g} (1 - e^{-g/\tan \alpha}) (3 - e^{-g/\tan \alpha}) & \text{for } \alpha \leq \pi/2 \\ 1 & \text{for } \alpha > \pi/2 \end{cases} \quad (\text{III.28})$$

$$S(\alpha) = \begin{cases} \frac{4}{9} \left(1 - \frac{1}{2} \cos \alpha\right)^2 & \text{for forward-scattering} \\ 1 & \text{for isotropic scattering} \\ \frac{\sin \alpha - (\pi - \alpha) \cos \alpha}{\pi} & \text{for back-scattering} \end{cases} \quad (\text{III.29})$$

Although this model is not available in the Oregon BRDF Library, it is employed in the formulation of the Beard-Maxwell model discussed below.

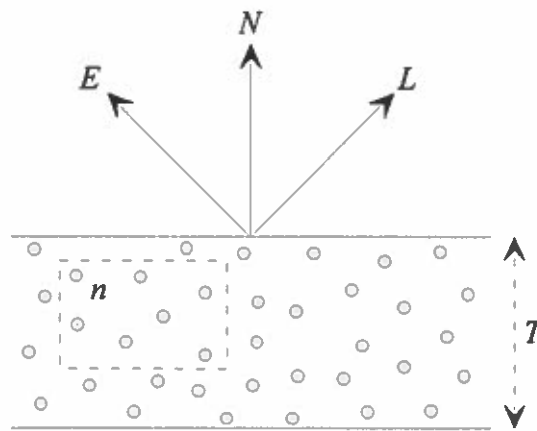


FIGURE 33. Blinn scattering geometry.

Blinn

Quite similar in construction to Hapke/Lommel-Seeliger is a model developed by James Blinn as an improvement to the Lambertian assumption, and perhaps a bit simpler to evaluate than Hapke's. Blinn's goal was to model the reflectance from a volume of particles as would occur in events such as clouds and dusty surfaces. His specific goal was generating images of the rings of Saturn [2].

In consideration of such an event, the microgeometry that must be addressed is a distribution of particles in a volume, onto which luminance falls. In Figure 33 each particle is approximated as a sphere, hence every one has the potential of scattering light in all directions (including back toward the incoming direction). Depending on the phase function used in the equation, his model can correctly predict the measured back-scatter from a variety of surfaces; in Blinn's model the phase functions are chosen to fit known reflectance data, thereby permitting back-scatter to dominate the calculation when appropriate. Interreflections within the volume of particles sometimes allow very little light to escape in the mirror direction, and depending on the thickness of the volume, more light may end up being transmitted (Blinn deals

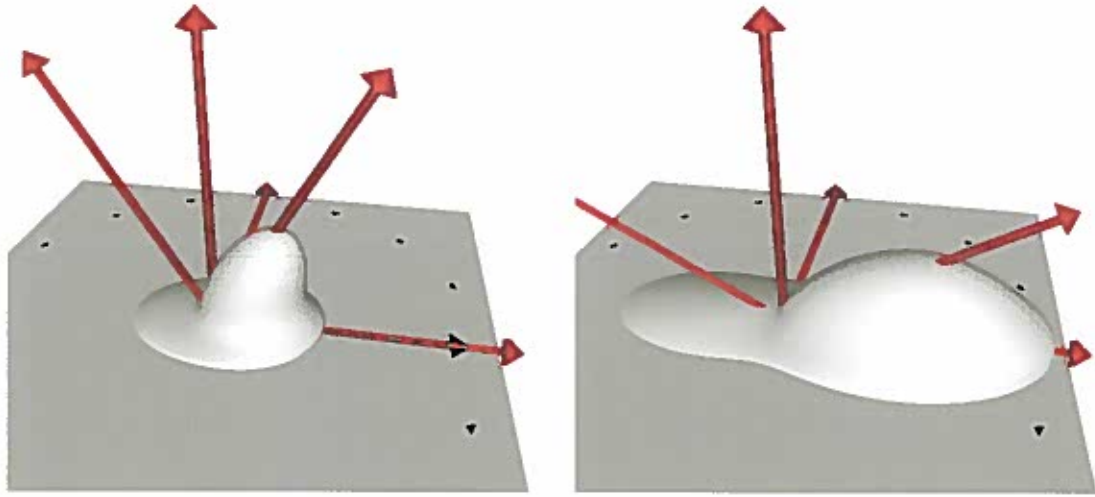


FIGURE 34. Blinn reflectance lobes: $\theta_i = 30, 60^\circ$.

with the transmitted radiance as well).

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \begin{cases} \rho_d \varphi(\alpha) \frac{\mu_0}{\mu_0 + \mu} [1 - e^{-\tau(1/\mu_0 + 1/\mu)}] & \text{top-lit} \\ \rho_d \varphi(\alpha) \frac{\mu_0}{\mu_0 + \mu} [e^{\tau/\mu_0} - e^{-\tau/\mu}] & \text{bottom-lit \& } \mu_0 = -\mu \\ \rho_d \varphi(\alpha) \frac{\tau}{\mu} e^{-\tau/\mu} & \text{bottom-lit \& } \mu_0 = -\mu \end{cases} \quad (\text{III.30})$$

Above, ρ_d is the reflectance of an individual particle (assumed to be relatively low), $\varphi(\alpha)$ is the phase function, α is the phase angle (as previously used in Hapke's model), $\mu_0 = \cos \theta_i$, $\mu = \cos \theta_r$, and $\tau = n\pi\rho T$. The elements of τ are: n = the number of particles per unit volume, ρ = the radius of the particles, and T = the thickness of the layer.

Blinn presents a selection of phase functions $\varphi(\alpha)$, many of which are available in Appendix D. Another option he suggests it to use a linear combination of phase functions, thereby gaining the ability to model multi-lobed scattering events. Blinn takes this approach for the illumination model used for the rings of Saturn; he takes a weighted average of two Henyey-Greenstein phase functions [2]. Figure 34

demonstrates reflectance lobes using Blinn's model fit to the rings of Saturn (These are primarily back-scattering lobes).

Measuring the Real Thing

Now that most of the analytic BRDF models available in the BRDF literature have been discussed, a few deficiencies are obvious; there is a trade-off between the number of surfaces a model can predict and its complexity. And even if the model attempts to be physically exhaustive, as in the case of He-Torrance, it may still fail to address anisotropy. Maybe all the theory is getting too complicated and it would simply be easier to measure the BRDF of the desired material.

This is easier to say than to accomplish; to measure a BRDF for even an isotropic surface requires a high-precision instrument not available to everyone who wants to study reflectance. The instrument must be able to vary the four angles of the BRDF precisely, so that a large number of measurements of the sample can be recorded. The resulting measurements will be noisy, so the data must be pre-filtered before it can be used. And the samples will be discrete, so interpolation will be required.

If the sample is anisotropic, off-plane measurements must be taken as well. This dramatically increases the number of measurements which must be taken, and complicates the interpolation process. If spectral information is also to be accounted for, the number of data values is multiplied by the number of wavelengths. The memory requirement for storing such a deluge of measurements is considerable.

The best alternative at this point would be to employ the idea previously mentioned in both the Ward and He-Torrance models; apply a fitting algorithm to

the data to limit the amount that must be maintained. This reduces the BRDF to either a completely analytic evaluation or a combination of an analytic evaluation and a table lookup.

Lafortune

Lafortune presents a model that takes measured BRDF data and fits it to an entirely analytic model [14]. The model sums a series of arbitrarily oriented cosine lobes to generate the total BRDF. Each particular lobe is defined by an alignment axis and an exponent. Because the formulation operates on vectors, Lafortune et al. recast the bidirectional reflectance in terms of vectors.

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \rho_{bd}(\vec{u}, \vec{v}) \quad (\text{III.31})$$

Here \vec{u} is determined by the incident direction $\langle \theta_i, \phi_i \rangle$, while \vec{v} is determined by the reflection direction $\langle \theta_r, \phi_r \rangle$.

The equation for the BRDF, employing the fitted coefficient vector \vec{C}_i and lobe exponent n_i for the i^{th} lobe is:

$$\rho_{bd}(\vec{u}, \vec{v}) = \sum_i [C_{x,i}u_xv_x + C_{y,i}u_yv_y + C_{z,i}u_zv_z]^{n_i} + \frac{\rho_d}{\pi} \quad (\text{III.32})$$

The summation takes place over the number of cosine lobes used to fit the BRDF data. Though typically Lafortune et al. use three lobes to fit the directional component of the scatter, the model generalizes to an arbitrary number of lobes.

The sign and magnitude of the entries in the coefficient vectors determine the type of lobe. Isotropic lobes will have $C_x = C_y$. If the lobe is isotropic and C_z is less

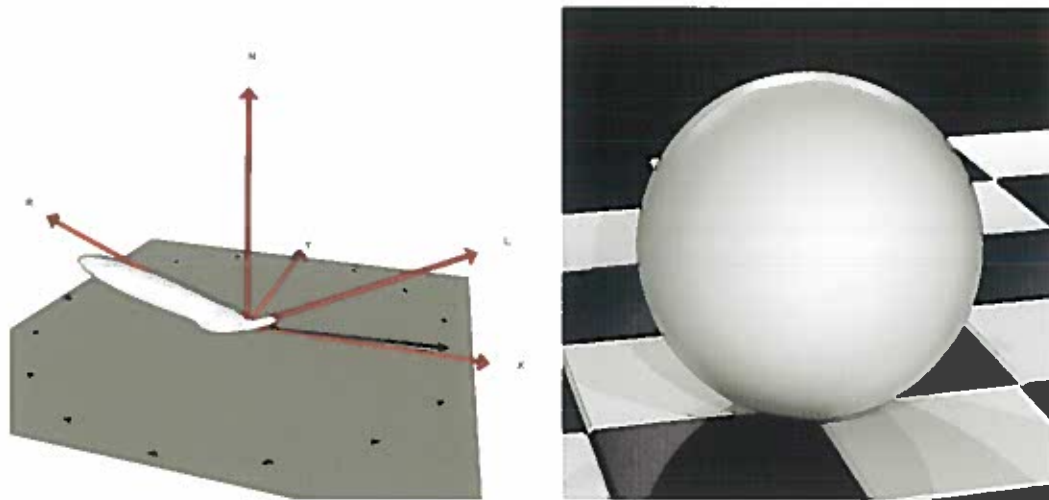


FIGURE 35. Lafortune's paint reflectance lobe and rendered sphere.

than $-C_x$, then the lobe will be an off-specular reflection. And if all three entries are positive, the lobe is retroreflective.

Though Lafortune's three-lobe approximation is quite rough, the pictures generated with it convincingly reproduce complex reflectance properties including increased specular reflection at grazing angles and retroreflection. Figure 35 shows the sum of cosine lobes fit to the measured reflectance of latex paint in both reflectance lobe and rendered sphere contexts. The sphere shows the largely diffuse nature of the BRDF except for polar angles that approach grazing. Both the off-specular directional lobe and the highlights on the top of the sphere demonstrate the increasing specularity for grazing angles.

Beard-Maxwell

Beard and Maxwell employed a combination of fitting and table-lookup in developing their model [1]. They also simplified their model by choosing to exclusively address isotropic materials. Similar to many of the other models, Beard-Maxwell

employs a sum of terms to describe a variety of scatter classifications. One of the terms employs a lookup table recording the first-surface BRDF values at a small number of wavelengths. The remaining wavelengths are extracted by interpolation.

$$\rho_{bd}(\theta_r, \phi_r, \theta_i, \phi_i) = \frac{R(\frac{\alpha}{2})}{R(0)} \frac{\rho_{fs}(\theta_r, \phi_r, \theta_i, \phi_i) \cos^2 \theta_N}{\cos \theta_r \cos \theta_i} \left(\frac{1 + \theta_N}{1 + \frac{\theta_N}{\Omega} e^{-\alpha/\tau}} \right) + \frac{\rho_d}{\pi} + \frac{2\rho_v S(\alpha) R_f(\alpha, g)}{\cos \theta_r + \cos \theta_i} \quad (\text{III.33})$$

Here θ_N comes from permitting the normal to depart from alignment with the Z-axis; hence the normal will have the direction: $\langle \theta_N, \phi_N \rangle$. The phase angle α , and the fitted shadowing/masking parameters τ and Ω are also required.

The first term of the sum employs the recorded first-surface planar BRDFs ($\rho_{fs}(\theta_r, \phi_r, \theta_i, \phi_i)$). The required value is extrapolated, then scaled by the Fresnel and shadowing/masking terms to approximate the polar angle dependency of the directional/specular reflectance.

The second and third terms are a Lambertian and a Hapke/Lommel-Seeliger model, respectively. The Lambertian is fit to account for ideal scattering present in the sample, while the Hapke model can account for retroreflection and forward- and back-scattering. Refer back to equation III.27 for detail on the volumetric Hapke component.

In practice, the fit typically employs either the Lambertian or the Hapke/Lommel-Seeliger component, but not both. To subtract a model, the relevantly subscripted ρ -term is set to zero: ρ_d corresponds to the Lambertian term, while ρ_v is used for the volumetric scattering modeled by Hapke's BRDF.

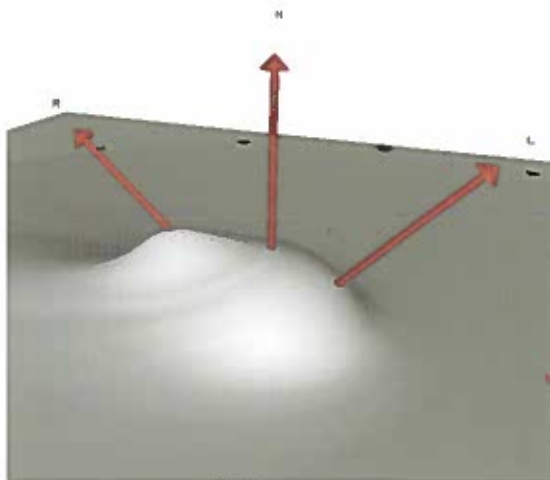


FIGURE 36. Beard-Maxwell reflectance lobe for white-primed aluminum (from NEF database).

Beard and Maxwell developed the model to approximate the BRDFs of paint coatings. The first-surface term corresponds to the interface of the substrate in which pigment particles are suspended. The roughness of this top surface can be compared with the microgeometry employed in the Cook-Torrance, He-Torrance, or Oren-Nayar models. The next two components account for the subsurface scattering from the pigment particles.

The Beard-Maxwell BRDF tends to have high-order continuity which results in images with quite convincing reflective properties (refer to Figure 36).

NEF Database

To be useful, the Beard-Maxwell model requires an accessible database of first-surface BRDF tables and fitted parameters for the remaining components of the model. Fortunately such a database exists and is publicly available; known as the Nonconventional Exploitation Factors (NEF) database, it provides fitted parameters

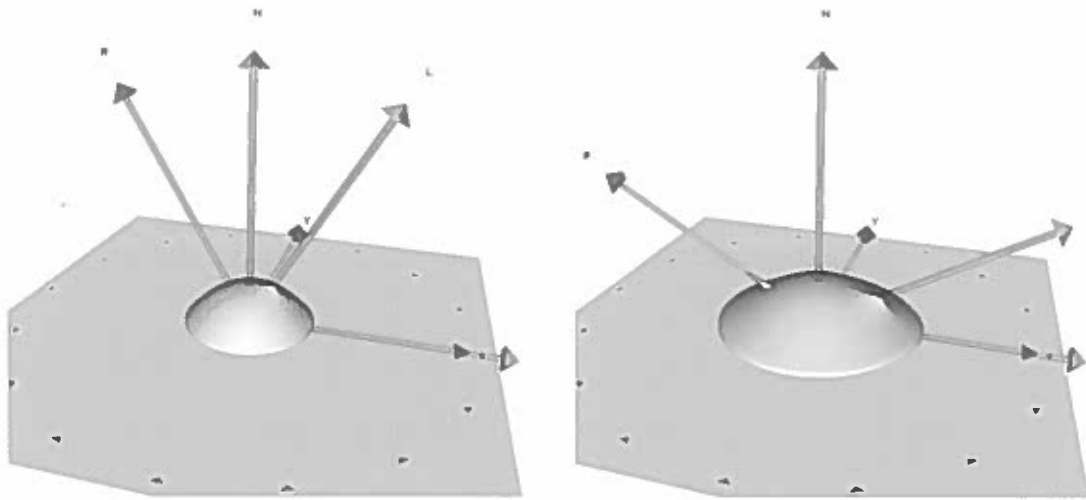


FIGURE 37. Reflectance lobes of NEF weathered concrete sample: $\theta_i = 30, 60^\circ$ (vectors are transparent to make the specular and retroreflective peaks more apparent).

and first-surface BRDFs for a wide range of surfaces [18]. Unfortunately, many of the surfaces included are not directly applicable to computer graphics since the database was funded by the United States military.

NEF includes a database server which can be queried by an application that will evaluate the BRDF for a certain material at a particular set of incident and reflection angles. NEF has the capability to do a full spectral evaluation from visible to infrared wavelengths. By either choosing appropriate wavelengths or by performing a spectral integration, the BRDF evaluation can be used for image rendering.

Figure 37 was constructed by querying the database in a directional-hemispheric manner; the incident angle is held constant while a discrete number of samples on the hemisphere are used for the reflection angle. This particular reflectance lobe sample is from weathered construction concrete. The visualization tells us the material is predominantly diffuse, though non-Lambertian since scattering is more pronounced for grazing angles. Also, the material exhibits a small retroreflective

peak as well as a slight specular highlight for grazing angles.

Figures 40, 41, and 42 in Chapter IV have surfaces that are rendered from NEF materials.

Columbia-Utrecht Database

The Internet has expedited the dispersal of information in nearly every field, computer graphics included. A joint project between Columbia University, New York, and Utrecht University, the Netherlands, produced a set of BRDF-fitting model parameters available online. These parameters fit BRDF data measured for sixty-one rough-surface materials (including anisotropic samples) [5].

The parameters were derived for two specific models. The first of these models is Oren and Nayar's which was previously described. The second model is the decomposition developed by Koenderink et al. [13] that calculates a variable-order vector of coefficients which fit the BRDF data; the higher the order, the larger the number of coefficients stored, and the better the achieved fit. Unlike the Oren-Nayar model, the model by Koenderink et al. is capable of representing both isotropic and anisotropic BRDFs. Both parameter sets are available in achromatic and RGB formats at www.cs.columbia.edu/CAVE/curet.

Dana et al. show that for diffuse samples, the Oren-Nayar model has better performance than a second-order Koenderink decomposition (which requires five coefficients). However, when BRDFs of a non-diffuse nature are fit, an eighth-order Koenderink decomposition (requiring 55 parameters) was sufficient to represent even the anisotropic materials.

Simulating the Real Thing

Capturing the BRDF for a particular surface is not always possible. A researcher may not have access to the required apparatus, or it may prove to be too expensive to perform the number of measurements necessary, especially in the case of anisotropic materials. The surface may even prove to be impossible to measure with the available apparatus. In any of these situations, if the researcher can come up with a suitable geometric model with the correct physical characteristics, a BRDF can be acquired to any level of detail by performing a simulation. These simulations employ a conceptual device known as a virtual goniophotometer.

Many researchers have explored this technique with excellent results; isotropic, anisotropic, and even diffractive effects have been captured using the technique [3, 7, 28]. The issue of greatest debate in running such a simulation is also a problem when empirically measuring the BRDF—what is the most efficient way to store the resulting BRDF for quick evaluation?

The research which follows considers both the simulation and data-storage aspects of BRDFs. Some elegant ideas have been suggested, but the issue of efficient storage and evaluation remains an unsolved problem.

Cabral

Cabral et al. [3] sought a model for the BRDF of rough surfaces. Beginning with a bump-map, they generate a triangularly faceted mesh with which to perform the BRDF calculation. Rays are cast stochastically at the modeled surface, and the reflected rays are captured in bins defined over the hemisphere of scatter directions. The ratio of the number of rays captured per bin to the total number of rays cap-

tured over the hemisphere determines the magnitude of the BRDF in the direction specified for a particular bin.

The simulation just described is straightforward, but because of the discrete nature of the bins the calculated lobe will be discontinuous. Cabral et al. address the two issues of both smoothing the reflectance lobe and reducing the storage requirement by computing a spherical harmonic representation of the simulated BRDF. Spherical harmonics, first used by Kajiya [11], are the three-dimensional analogue of a Fourier decomposition; they make it possible to describe a spherical function of arbitrary topology.

Westin

Westin et al. [28] extended prior work by using spherical harmonics to store anisotropic BRDFs. They also outline a way to avoid the discontinuous bucket technique used by Cabral et al. to simulate a BRDF.

They optimize the spherical harmonic matrix and build it directly from the Monte Carlo samples cast to calculate the BRDF. Although dealing with anisotropy greatly expands the size of the matrix, several optimization techniques bring the size down considerably, permitting them to simulate the reflectance of rolled aluminum, velvet, and woven nylon. The renderings employing the simulated BRDFs have quite subtle and effective reflectance effects.

Gondek

The final simulation-based approach to BRDFs to be addressed, again involves the bucket-capture technique to produce the reflectance lobe for a particular surface

geometry and physics. Instead of using a fixed number of buckets on the hemisphere, Gondek et al. [7] permit the simulation to adaptively subdivide buckets when the relative difference between neighboring buckets warrants. The adaptive subdivision occurs in a data structure modeled on the geodesic sphere.

The difference between neighboring buckets causes a subdivision when it is greater than a specified ε . The simulation tracks all samples, so when subdivision is deemed necessary the samples are redistributed to the appropriate sub-buckets. Subdivision occurs until all neighboring bucket differences are below tolerance. In general, the data structure will be elaborated along areas of rapid change—such as around the mirror-reflection direction. To take advantage of this symmetry, the data structure is rotated to align all mirror-reflection directions, thereby concentrating the subdivisions in a limited region of the data structure.

The simulation technique employed by Gondek et al. extends the illumination model to include phase relationships between light rays and elaborates the microstructure model to include the geometry of subsurface particles. Convincing images highlighting interference and subsurface scattering effects were produced of several real-world materials including thin film coatings, iridescent and pearlescent finishes, and paint.

Choosing a Model

The vast selection of models described is not exhaustive. Current research is focussed on finding more efficient techniques of storing simulated or measured BRDF data. Also, algorithms for choosing the best ray to cast in a Monte Carlo renderer are required to improve the efficiency of rendering from these BRDFs. Pure

stochastic sampling is not guaranteed to sample the hemisphere to best advantage, and may generate an inordinate amount of low-frequency noise until a very large number of samples are taken. Of the models discussed, only Ward [25] directly addresses this common concern, and he is able to do this simply because his BRDF is purely analytic.

For any particular application, the context determines the type of BRDF that would be most appropriate. If evaluation efficiency is of paramount importance, one of the simpler analytic models such as Phong's may suffice. However, if physical accuracy is mandatory a model derived from a simulated or measured scattering event is warranted as long as the application can handle the expense or the performance penalty. If the application falls between the two extremes, an optimized model that employs aspects of both, such as Beard-Maxwell, might be the best.

All the fields interested in BRDFs are heading towards a point where it will be possible to employ models with arbitrary levels of accuracy in computer simulations. Whether these simulations are built in pursuit of improved spacecraft heat shields, non-invasive real-time medical imaging of the human body, or even just more exciting entertainment utilizing computer graphics is merely a function of the creativity directed at improving our ability to correctly predict how electromagnetic radiation is scattered by the matter it encounters.

CHAPTER IV

RENDERING WITH BRDFS

Rendering as a Learning Tool

Analyzing BRDF models with a visualization tool as discussed in chapter II gives the user a visually intuitive representation of the scatter predicted by a particular model. Formulaic evaluations that result in tables of sample points are reinterpreted into a mobile 3D representation. The proportion of scatter in any particular direction is indicated by the magnitude of the reflectance lobe in that direction. The reflectance lobe gives the user a clear indication how the light is distributed by a differential surface patch.

However, to appreciate the surface *appearance* that results from the reflectance distribution, a different type of visualization is necessary. Generally, everyday surfaces are smoothly continuous. They also tend to be illuminated from multiple directions simultaneously. A visualization of the reflectance lobe does little to provide an intuition of what an everyday surface possessing the predicted reflectance will look like.

Photorealistic rendering is a visualization of the BRDF that teaches a user the intuition; the reflectance lobe is translated into a smoothly shaded surface. Studying images of 3D scenes containing multiple objects that exhibit empirical, analytic, or simulated reflectance in concert with reflectance lobe visualizations, permits the user to discover the correlation between the two—lobe topology translates to lighting

effects on “real” objects. For example, place three spheres in close proximity, assign the first Lambertian reflectance, the second Phong reflectance, and the third the reflectance of an NEF material. The user will be able to see what effect the scatter shape has on the shaded surfaces. The Lambertian sphere will appear uniformly smooth, the Phong sphere will have a specular highlight reminiscent of plastic, while the NEF material will most likely have subtle effects that lead it to be the most convincing.

Connecting BRDF model visualization to a rendering system is essential to understand the various models and differentiate their individual strengths.

Radiance: A Radiometrically Correct Renderer

The BRDF equation I.1 is physically-based. Therefore, the logical approach to rendering a BRDF-described surface is to accurately track the computed reflectance values. Most renderers rely on efficient approximation—speed the rendering process at the expense of physical accuracy.

Greg Ward’s Radiance renderer is an exception to this observation. Radiance maintains the full dynamic range of radiometric values[26]; intensities can vary from direct solar glare to shadows on a starlit night. In addition, the ambient values derive from the scene itself rather than being arbitrarily assigned (another common efficiency approximation). Calculated ambient values are stochastically distributed across the scene’s surfaces adaptively. A larger ambient gradient will spawn additional ambient samples. Each ambient sample is a Monte Carlo integration of the light incident at a particular sampling location.

Radiance provides many tools for building a scene and specifying the materials within it. Ward's reflectance model is built into the renderer. As discussed in chapter III, it models both isotropic and anisotropic materials. To render a material with reflectance either determined by a data set or by an analytical formula different from Ward's, a scripting language is available. This language, integral to Radiance, is capable of defining geometry, mapping surfaces, generating textures, and even shading materials.

Radiance's `BRTDFunc` material relies on a user-provided script to define how light scatters from an interface. Using this Radiance material, both lookup table and formulaic BRDFs can be defined. Unfortunately, `BRTDFunc` has a serious shortcoming—it does not participate in the ambient calculation described previously. This makes its utility questionable.

To get around the deficiency of `BRTDFunc`, Radiance has been extended to accept arbitrary BRDFs which *do* participate in the ambient calculation. Harold Westlund [29] has defined a new material type, `IBRDF`, that uses a data file of reflectance measurements to allow Radiance to correctly render BRDF-specified surfaces. This material is general enough that most of the BRDF representation schemes presented in chapter III can be rendered in Radiance (color and anisotropy are not yet available). To use the `IBRDF` material, the BRDF model of interest is sampled into a lookup table. Subsequently, a material referencing the file is defined in Radiance.

When Radiance encounters an `IBRDF` surface, it loads the file, normalizes the data, and begins querying the lookup table. A deterministic and stochastic technique based on lobe magnitude is employed to sample the BRDF—an intense highlight

will be sampled with greater regularity than a direction which contributes little to the result. The lookup table can cause aliasing problems, but increasing the BRDF sample density or the number of Monte Carlo rays improves its continuity. Each IBRDF material specifies a ray count and falloff rate. Specular BRDFs can be sampled sparsely since the rays will all go in roughly the same direction, whereas a diffuse material requires more samples since they will scatter uniformly across the hemisphere. The falloff rate constrains the exponential explosion of rays.

Oregon BRDF Library Renderings

The IBRDF extension to Radiance was used to render surfaces exhibiting the models available in the Oregon BRDF Library. The reflectance lobes of individual models were sampled, and the resulting data sets were used to generate images throughout this document. The current version of the IBRDF material cannot render chromatic or anisotropic materials. Any figures displaying anisotropic reflectance were rendered with Radiance's Ward model.

The figures presented in this section explore some of the reflectance effects that are possible. Simple scenes with multiple copies of interesting objects were built to provide a context in which to study rendered surface reflectance.

Figure 38 presents an object constructed from spheres possessing distinct reflection models. The two models used to shade the top-row spheres are Beard-Maxwell and Minnaert. The bottom-row spheres are shaded with the Ward, Blinn, and He-Torrance models.

Surfaces rendered with the Ward and He-Torrance BRDFs were seen previously in Chapter III. Here, the Ward sphere is highly reflective and slightly anisotropic.

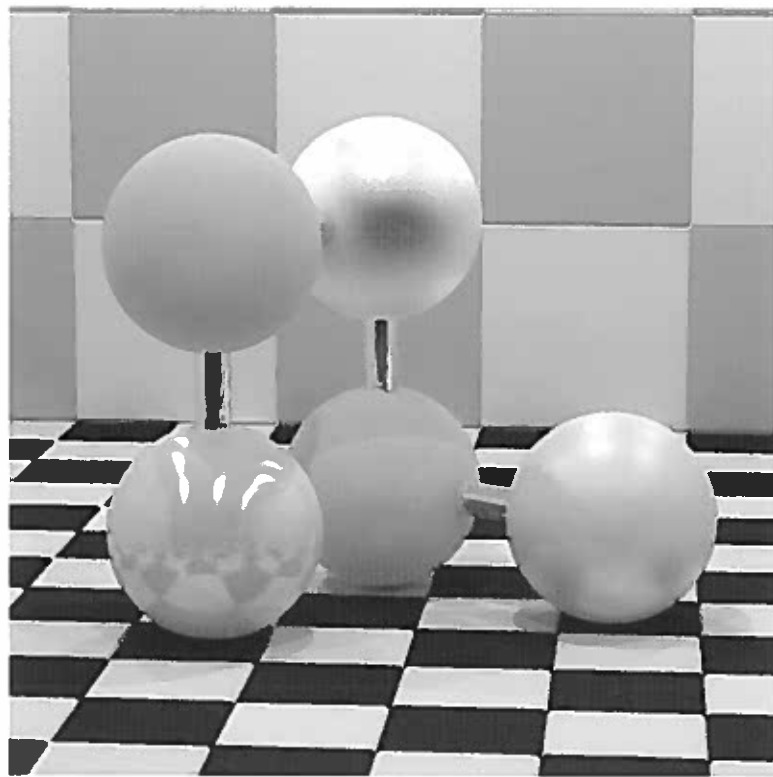


FIGURE 38. Thingamajig rendered with a variety of BRDFs. Top row: Beard-Maxwell, Minnaert. Bottom row: anisotropic Ward, Blinn, and He-Torrance.

The He-Torrance sphere is rendered with the same parameters used for Figure 24; the surface roughness is relatively high, but at grazing angles specular reflections begin to appear (as tends to happen with real-world rough surfaces).

In general, distinguishing the diffuse models is difficult. A sphere shaded with the Oren-Nayar model bears a strong resemblance to one shaded with the Lambertian BRDF. Depending on the parameters, the Minnaert and Blinn models will also be remarkably similar. The differences are in the subtleties: flatter overall appearance, highlights at grazing angles, or highlights when the viewing and illumination directions are approximately equal. The Minnaert and Blinn spheres in this image are those that should by definition be diffuse. Blinn's model as seen in Figure 33 was a predominantly back-scattering diffuser. The Blinn sphere does not appear remarkably different from Lambertian. In contrast, the Minnaert model employs a normal-aligned cosine lobe; the resulting sphere is definitively non-Lambertian—normally aligned illumination is reflected diffusely. This creates a “patchy” appearance.

The final sphere is rendered with the analytic and empirical Beard-Maxwell model. The data for the first-surface BRDF was extracted from the NEF database, along with the appropriate settings for the rest of the model's parameters. The material is white-primed aluminum. As predicted by the reflectance lobe in Figure 36, the sphere demonstrates almost no specularity; it could easily be mistaken for one of the analytic diffuse models.

The next figure, Figure 39, uses three different models to shade an arrangement of vases. The vase is an interesting object because it has both convex and concave curves—smooth curves allow the entire BRDF to be explored.



FIGURE 39. Vases rendered with the following BRDFs (left-to-right): Lafortune's cosine lobes fit to paint, Cook-Torrance gloss paint finish, and Oren-Nayar retro-reflective rough surface ($\sigma = 30^\circ$).

The vase on the left is rendered with Lafortune's cosine lobe approximation of measured paint (also used for Figure 35). Not surprisingly it could be mistaken for a Lambertian surface; however there are a few indications that something more interesting is happening. Along the right side of the vase's neck, a slight highlight appears. This is due to the reflection off the vase in the center. Grazing inbound illumination generates highlights in Lafortune's paint model.

The middle vase employs the Cook-Torrance model. It might be interpreted as gloss paint or plastic, though if it were colored it would also be possible to achieve metallic effects (since Cook-Torrance relies on the index of refraction of the medium).

The final vase on the right side of the figure appears almost flat; the BRDF applied to the surface is Oren-Nayar's. Though it appears almost Lambertian, certain effects distinguish it. The underside of the vase does not diminish in intensity as rapidly as the other two vases; this is caused by the concentration of back-scatter at grazing angles.

And finally, in Figure 40 the three vases we just looked at are rendered with BRDFs taken from the NEF database. From left-to-right the materials are bare construction lumber, gloss paint on metal, and scored aluminum. From left to right they transition from predominantly diffusing surfaces, to focussed specular reflectors. The subtle highlights on the leftmost vase make it especially convincing as smooth-finished wood.



FIGURE 40. Vases rendered with the following materials from the NEF database (left-to-right): bare construction lumber, gloss paint on metal, and scored aluminum.

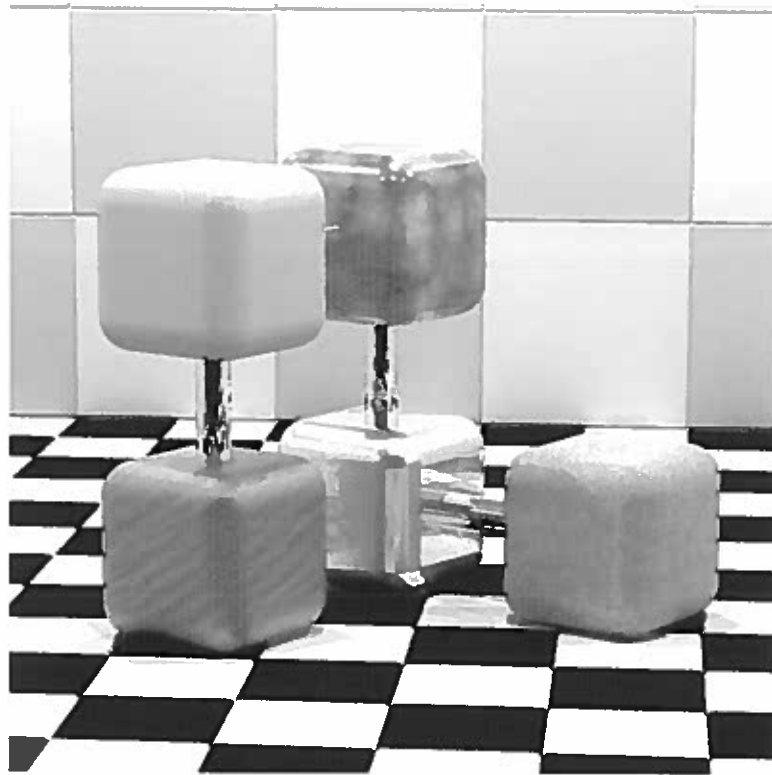


FIGURE 41. Cubes rendered with textured NEF materials. Top row: cement, gloss paint on metal. Bottom row: bare construction lumber, scored aluminum, and weathered concrete.

IBRDF Renderings

Although the images in the preceding section were rendered with the IBRDF material in Radiance, only the BRDF was applied to the surface. Interesting pictures require either texture or additional geometry to enhance the visual impact.

Figures 41 and 42 were embellished with procedural textures to better indicate the materials from which the BRDFs were measured. The data sets come from the NEF database. In particular, the following materials are represented: bare construction lumber, gloss paint on metal, scored aluminum, weathered concrete, and cement.



FIGURE 42. Vases rendered with textured NEF materials (same materials as used in Figure 40).

APPENDIX A

OREGON BRDF LIBRARY API

The Application Programmer Interface (API) will be presented in three phases. First, conventions employed in the construction of the Oregon BRDF Library are presented. Next, a discussion of the library's general routines is offered. Finally, a single model-specific API will be presented for reference; the balance of the APIs are available in a separate document distributed with the software library.

Library Conventions

The Oregon BRDF Library is written in C++. C++ permits the use of object-oriented features in the Library's implementation.

The library is available for incorporation into a wide variety of applications. This goal required two paradigms. The first of these is adherence to a sufficiently entrenched version of the ANSI C++ specification. All compilers are not created equal; the more recent standardized features of the language are not always fully implemented or debugged. Hence newer features of C++ have been avoided in the BRDF library. One of those newer features is the use of namespaces, which are typically used to avoid pollution of the global program scope. The Oregon BRDF Library employs the older C-style of prefixing names with a relevant acronym to avoid collisions in the global scope. In our case the prefix is OBL. Most of the components of the library that make use of this prefix are located in *brdf_defines.h*.

There is another header file that uses this prefixing, and its implementation is meant to avoid the second standardization issue with C++. The most recent standard requires that a host of libraries be distributed with ANSI C++, including a library that provides complex number arithmetic. These standard libraries are not available with all compilers, hence we have included the `OBL_COMPLEX` class with the library. Defined in the *brdf_complex.h* header file, `OBL_COMPLEX` provides all the complex arithmetic likely to be required.

As an aside, all floating-point arithmetic performed in the Oregon BRDF Library is done in double precision. It is possible to quickly change the entire implementation to single precision by redefining the `OBL_FLOAT` type in the *brdf_defines.h* header file and then recompiling the library.

There are two abstract base classes at the core of the library. Both these base classes, as well as the classes derived from them, do not make use of the prefix naming scheme discussed above. Rather, in order to avoid unreasonably long class names (which are unlikely to cause global namespace collisions anyway), the derived classes are suffixed with a reference to their base class. For example:

Abstract Class	Derived Class
BRDF	Lambert_BRDF
	Ward_BRDF
BRDF_EVAL	Radiance_EVAL
	RGB_EVAL

Each derived class first completes the functionality inherited from the superclass; the new class then enriches the core functionality by providing the necessary model-specific methods and optimizations.

Model parameters are for the most part unit-free. However, some parameters, including angles, do have associated units. When angles are specified the units are in radians unless otherwise indicated. Distances are always specified in meters.

Fonts are used to indicate context. Class parameters and method definitions are shown in `typewriter`, as are references to class names and constants. Values assigned to parameters are shown in *italics*.

Note: additional header and source files have been provided as a starting point for users wishing to add their own models. Refer to the *brdf_template.h* and *brdf_template.cxx* files.

Overview of Routines

As mentioned in the preceding section, there are two main abstract classes underlying the Oregon BRDF Library implementation. These two classes are `BRDF` and `BRDF_EVAL`. There is a third abstract base class that can be inherited by classes derived from `BRDF_EVAL`: `Colored_EVAL`. The `Colored_EVAL` class provides RGB and spectral distribution color conversions to XYZ space via the 1931 CIE 2 deg Standard Observer matching functions. The conversion is made possible by the inclusion of the Oregon Color Software with the Oregon BRDF Library.

All BRDF models (analytical, empirical, or simulated) derive from the `BRDF` base class. The `BRDF` class is not instantiated itself, but does provide the functionality common to all the BRDF models. It standardizes the calling convention, as well as dynamically resolving which evaluation function should be called at run-time. The `BRDF_EVAL` class is the second major base class, and it is the return-type of the evaluation of any model. Since `BRDF_EVAL` is abstract, it will never be instanti-

ated either. Instead, depending on the model in question, one of the three derived evaluation classes will be instantiated. A `BRDF_EVAL` has the following functionality:

`OBL_EVAL_TYPE getType()`

Returns the type of the derived evaluation. The return value will be one of `OBL_Radiance`, `OBL_RGB`, or `OBL_Spectral`.

`OBL_FLOAT getMag()`

Returns a "rough-scale" magnitude of any `BRDF_EVAL`. Evaluation of the magnitude varies dependent on the type.

There are three classes of instantiable evaluations. These three permit varying models and data sources to be queried. The `Radiance_EVAL` class is the first, and is used for all the analytic models currently available in the library. This class of evaluation permits the separate querying for specular, diffuse, and transmitted components of the BRDF.

The second type of evaluation is the `Spectral_EVAL` class which is used for data sources that track full spectral distributions. In order to use such data in an interactive context, it is necessary to define some integration of the distribution; since computer graphics is one application area for the Oregon BRDF Library, we have settled upon photopic luminosity as our single-valued magnitude. Previous research at the University of Oregon produced the Oregon Color Software that implements the necessary routines for converting a spectral distribution to the corresponding XYZ tristimulus value. Once converted to XYZ, the Y-component is the photopic luminosity; that is, the gray-scale equivalent of the color to a human observer.

The final class of evaluation is `RGB_EVAL`. It is quite similar to `Spectral_EVAL`, as `RGB_EVAL` also employs the photopic luminosity as its single-valued magnitude. The conversion proceeds by taking the red, green, and blue components, assumed to be in sRGB (standardized monitor RGB) space to XYZ (clamping occurs where

necessary to map the values to photopic luminosity). Conversions from RGB spaces other than sRGB are possible, but require reconfiguration of the Oregon Color Software.

Radiance_EVAL provides the following methods beyond those offered by BRDF_EVAL:

Radiance_EVAL(OBL_FLOAT *spec*, OBL_FLOAT *diff*, OBL_FLOAT *trans*)
 Constructs this type of evaluation. The three parameters specify the relative specular, diffuse, and transmitted radiance.

OBL_FLOAT getSpecMag()

OBL_FLOAT getDiffMag()

OBL_FLOAT getTransMag()

Returns the relative component of the evaluated radiance.

The components of a Radiance_EVAL are separately distinguishable to permit the coloring of the result (among other potential applications). Color specification of a surface might apply a different color to each of the specular, diffuse, and transmitted light. For example, an opaque red plastic might have a specular color of white (the plastic substrate *does not* filter the light), a diffuse color of red (the pigment particles suspended in the substrate *do* filter the light), and no transmitted color (after all, it is opaque).

In contrast to Radiance_EVAL, the other two instantiable evaluations are *already* colored. To achieve this coloring, another abstract base class was defined from which both colored evaluation classes also derive; Colored_EVAL offers the additional functionality required to integrate for photopic luminosity.

OBL_FLOAT getLuminosity()

Returns the photopic luminosity as a percentage of full intensity (0–1).

Full intensity photopic luminosity is defined to be 100%.

OBL_FLOAT getX()

OBL_FLOAT getY()

OBL_FLOAT getZ()

Returns the relative component of the XYZ tristimulus value.

void getRGB(OBL_FLOAT &red, OBL_FLOAT &grn, OBL_FLOAT &blu)

Returns the RGB tristimulus value via the reference parameters. For a RGB_EVAL, it simply returns the tristimulus value the evaluation was instantiated with. For a Spectral_EVAL, a conversion to move from XYZ to RGB tristimulus values is performed.

Spectral_EVAL is the first evaluation derived from both BRDF_EVAL and Colored_EVAL. As previously noted, it integrates spectral distributions to permit their use as either XYZ or RGB tristimulus values.

Spectral_EVAL(OBL_FLOAT *spd, OBL_FLOAT step,
OBL_FLOAT shortest, int samples)

Constructs a Spectral_EVAL dependent upon the four parameters: an array of spectral magnitudes, the *step* between sampled wavelengths, the *shortest* wavelength, and the number of *samples* taken (number of elements in the array). As the default convention suggests, the units of both *step* and *shortest* are in meters. The Trapezoid Rule is used to perform the spectral integration.

The second evaluation derived from both base classes is RGB_EVAL.

RGB_EVAL(OBL_FLOAT red, OBL_FLOAT grn, OBL_FLOAT blu)

Constructs an RGB_EVAL dependent upon three parameters that specify the red, green, and blue magnitudes as percentages of the maximum sRGB tristimulus value. To successfully map to an XYZ value, all three RGB components should be in range 0-1. Values beyond the sRGB gamut could cause XYZ values beyond the acceptable range. These problem-values are possible since not all RGBs map into sRGB color space (they lie outside the sRGB color gamut).

Now that the evaluation classes are fully described, a discussion of the BRDF base class is in order. The BRDF class specifies the following functionality:

BRDF(OBL_BRDF_TYPE *type*)

Constructs a base-class BRDF, requiring a valid OBL_BRDF_Type enumeration (defined in *brdf_defines.h*). This constructor should be called explicitly in any derived model.

OBL_BRDF_TYPE getType()

Returns the type of any BRDF; a value from the OBL_BRDF_Type enumeration.

char* getName()

Returns the name and parameter settings of a particular BRDF. The function allocates a null-terminated character array just large enough to hold the entire model description. The user has to free the memory once finished with the string (by calling `delete []`).

void usage()

Displays instructions for the particular model to standard output (includes parameters and default values).

BRDF_EVAL* eval(OBL_FLOAT θ_i , OBL_FLOAT ϕ_i , OBL_FLOAT θ_r , OBL_FLOAT ϕ_r)

Evaluates any BRDF model. Since it returns a pointer to a BRDF_Eval, any evaluation derived from it can be returned. The model-specific implementation of `eval()` allocates the appropriate type of evaluation. The return will be null if unsuccessful. The user must dispose of the returned object.

Note: The θ_i , ϕ_i , θ_r , and ϕ_r parameters in the `eval()` method are in radians (as expected by convention). The ϕ -angles specify the azimuth angle relative to the positive X-axis (positive angles rotate toward the positive Y-axis). The θ -angles specify the polar angle relative to the Z-axis.

Evaluation of a BRDF occurs in a highly constrained space: BRDF-space. BRDF-space requires the surface normal to be oriented along the Z-axis. In addition, the X- and Y-axes must be available for every point of intersection between a ray and an interface. X- and Y- are overloaded concepts in this context, since they actually refer to the tangent and binormal at the point of intersection. These two vectors

permit the correct orientation of anisotropic BRDFs such as the Poulin-Fournier and Ward models. The transformation to BRDF-space must occur *before* the call to `eval()`, since `eval()` expects all angles to be in BRDF-space.

The target application determines whether the transformation to BRDF-space entails extra work. In the context of a BRDF visualization program, if no surfaces are being rendered, BRDF-space is the natural space to use for construction of the reflectance lobe. In the case of a image rendering system, all geometric primitives must provide a means to reconstruct the tangent and binormal vectors for any point on the surface of the object. The surface normal at the point of intersection, along with the tangent and binormal determine the requisite transformation.

There is one additional function which is implemented in the base class to permit easy display of the BRDF name and parametric settings (accomplished through an internal call to `getName()`):

```
ostream& operator<<(ostream &out, BRDF *brdf)
    prints the name and parameter settings of any model to a C++ output
    stream.
```

Model-Specific APIs

The model-specific APIs follow a consistent interface. To demonstrate this consistent interface, only a single BRDF model will be presented here. To see an API discussion for any other model, refer to the Oregon BRDF Library documentation available with the library distribution (made available December, 1998).

Cook-Torrance Specular Microfacet BRDF

Parameters

roughness: $[0, \infty]$
 height: $[0, \infty]$ meters
 lambda: $[0, \infty]$ meters
 rho_s: $[0, 1]$
 rho_d: $[0, 1]$
 index: $[a + bi]$ where a and b are real numbers
 distribution: [OBL_DST_BECKMANN or OBL_DST_GAUSSIAN]

Constructor

```
CTS_BRDF(OBL_FLOAT roughness, OBL_FLOAT height, OBL_FLOAT lambda,
          OBL_FLOAT n_real, OBL_FLOAT n_imag, OBL_FLOAT rho_s,
          OBL_FLOAT rho_d, OBL_Distribution dist)
```

The constructor clamps its arguments to the ranges defined above. When `distribution` is set to `OBL_DST_BECKMANN`, `height` and `lambda` are not used during evaluation; when `distribution` is `OBL_DST_GAUSSIAN` `roughness` is not used for evaluation.

Model-specific

```

OBL_FLOAT getRoughness()
OBL_FLOAT getHeight()
OBL_FLOAT getLambda()
OBL_FLOAT getIndexReal()
OBL_FLOAT getIndexImag()
OBL_FLOAT getRhoS()
OBL_FLOAT getRhoD()
OBL_Distribution getDist()
OBL_BOOL setRoughness(OBL_FLOAT rough)
OBL_BOOL setHeight(OBL_FLOAT height)
OBL_BOOL setLambda(OBL_FLOAT lambda)
OBL_BOOL setIndexReal(OBL_FLOAT n_real)
OBL_BOOL setIndexImag(OBL_FLOAT n_imag)
OBL_BOOL setRhoS(OBL_FLOAT rho_s)
OBL_BOOL setRhoD(OBL_FLOAT rho_d)
OBL_BOOL setDist(OBL_Distribution dist)

```

Access or modify the parameters of the Cook-Torrance model. All the set methods validate the parameter against the ranges defined above. If validation fails, the function returns OBL_FAILURE; otherwise it returns OBL_SUCCESS.

Discussion

Refer to Chapter III for a discussion of the Cook-Torrance model. Figure 21 presents example Cook-Torrance reflectance lobes, while Figure 22 demonstrates spheres shaded with the Cook-Torrance model. Refer to [4] for a complete discussion of the model, or [27] for a discussion of the distribution function options.

APPENDIX B

BRDFVIS NOTES

There are some limitations to particular models in BRDFvis:

- Arithmetic—only the sum of cosine lobes fit to blue paint as presented in Lafortune’s 1997 paper is available [14].
- Beard-Maxwell—the measured BRDF of material 431 from the NEF database (white primer on aluminum) is hard-coded into the model. Also note that the physical plausibility of moving the sliders is questionable; the Beard-Maxwell model is numerically fit to measured data.
- Cook-Torrance—the gaussian distribution function is non-functional.
- Gondek Geodesic Sphere—displays a Lambertian model.
- NEF Data—the materials available for visualization in the BRDF are limited. The material type is selected from the top menu, while the incident angle is selected from the second. Note that vectors and incident light direction *will not* be automatically updated to reflect the incident angle selected from this panel.
- Poulin-Fournier—the model has an inherent discontinuity caused by the sharp transition between separate cylinders or between the cylinder and

the floor. As a result, the BRDF sampling process cannot capture a clean transition.

- All models—many of the models explode for exitant polar angles approaching 90° (grazing). To deal with this condition, the sampling positions closest to the surface plane are clamped to lie within the bounds of the plane. On some models this clamping can produce strange lobe geometry. Increasing the lobe resolution will improve the continuity of the lobe.

APPENDIX C

EXTENDING BRDFVIS

BRDFvis is a tool; this implies that if it does not have the functionality required for a certain project, you can add it. This appendix will discuss the steps required to add a new BRDF model to the Oregon BRDF Library and the BRDFvis visualization tool.

Adding a new model to the Oregon BRDF Library is the easiest method of introducing a new BRDF into BRDFvis. The BRDF library is designed to treat all models equally whenever possible; this is especially nice when evaluating the entire hemisphere of scatter directions. Unfortunately, since each model may have its own collection of parameters, BRDFvis is not able to treat models so uniformly; passing model-specific parameters from the UI to a particular method in the library generates much of the complexity outlined in this section.

Adding a BRDF to the Oregon BRDF Library

The first step to visualizing your new type of BRDF is adding it to the Oregon BRDF Library. This requires introducing two new files: a header file and a source file that define your new model. Every BRDF in the library derives from the BRDF base class, and your model must also. The core functionality required in every BRDF is discussed in Appendix A. Any model parameters that you wish to adjust interactively in BRDFvis require the introduction of model-specific methods.

Note: any subdirectories mentioned in this section are in relation to the main directory in which the BRDF library is located (typically *BRDFlib*).

To start your header and source files, you will first need to add an enumeration for your BRDF to the `OBL_BRDF_Type` type in the *brdf_defines.h* header file located in the *include* subdirectory. This will provide a unique identifier for your model. Next, build your model, using another model as a starting point or starting with the *brdf_template.h* and *brdf_template.cxx* located in the *include* and *src* subdirectories respectively. Once your model is complete, add its header file to the *brdflib.h* header file and tell the *Makefile* in the *src* subdirectory how to build it.

The core elements of the BRDF library are now complete, however to access the model from BRDFvis an optional component of the library provides the necessary glue: the `Lobe` and `BRDF_Params` classes.

The `BRDF_Params` class is where to add any new parameters which will be acquired from any UI components you will be introducing to BRDFvis. For example if you need a slider for a parameter γ , which happens to take on the values 0–1, then you would add: `OBL_FLOAT gamma`. If a parameter of the correct type and name is already present in `BRDF_Params`, you do not need to introduce a new parameter, assuming the current one is being used by another BRDF model (if the parameter is one determining a general property, such as lobe resolution, *do not use it*).

`Lobe` evaluates the hemisphere of exitant directions for a given BRDF model and incident direction. It performs the evaluation independent of the model type; however, it also takes care of allocating a new BRDF when you change models, and adjusting parametric settings when you change them on the current model. Both these behaviors are managed by `switch` statements.

In the Lobe constructor (the one which takes a `BRDF_Params` argument) and in the `updateBRDF()` method, a switch on the type of BRDF model (which you previously modified in `brdf_defines.h`) calls the constructor of the relevant BRDF. Add a new case statement for your model, calling the appropriate constructor.

There is one more place where a switch on BRDF type occurs—in the `eval()` method that takes a `BRDF_Params` argument. Introduce a case statement and, following the example of neighboring statements, insert the appropriate calls to check and update the settings of your model-specific parameters.

Rebuild the library, including the Lobe support, and you are ready to link your model into `BRDFvis`.

Adding a BRDF to BRDFvis

At this point you have a functional model in the BRDF library, but you want the modules and functions from which `BRDFvis` is built to recognize your model. Adding a new model requires both adding a UI panel and telling `BRDFvis` how to use it.

Note: subdirectory references in this section will be in relation to the `BRDFvis` project directory (typically `BRDFvis`).

To build a UI panel for your model, you will need to modify the `templ.v` file in the `v` subdirectory; please make a backup copy before making any changes.

In the `Model_UIs` library in `templ.v`, there are a selection of BRDF panels that can be used as templates from which to build the panel for your model. Alternatively, you can begin with a copy of the `BRDF_Template_UI` which contains a few of the components common to many of the panels. The basic technique is to introduce

all the sliders, dials, labels, and dialog boxes your model requires for setting its parameters, then gather all the parametric values into the `model_vars` group to make them easy to export.

While building your panel, it is best to test it before incorporating it into the visualization applications; A `Simple_UI` application is available in the `UI.Components.Widgets` sublibrary. Instantiate it, then instantiate your panel into the `Simple_UI` application. Make `SimpleShell` the parent of your panel and then make it visible by connecting `visible` to the other input on your panel. You can now interact with your new panel.

The next step in adding your model to `BRDFvis` is telling the tool how to trigger your model and its associated panel. This is accomplished with a few additional modifications to the `templ.v` file, and some changes to two C++ files in the `src` subdirectory.

In the `templ.v` file, the first module in the `BRDFvis.Funcs` library is `Select_BRDF`. This module activates an output determined by the user selecting a particular model. It relies on the source file `brdf_ui.cxx` in which the function `choose_BRDF()` is defined. A parameter for the new model must be added to `Select_BRDF` in `templ.v`, while a case statement and Express API call must be added to `choose_BRDF()` in `brdf_ui.cxx`.

The changes made so far tell one module about the new model, but now the UI panel needs to be informed. First, in the `Misc.UI.Comps` library, add a `UItext` object for your model, then enter the appropriate reference into the `BRDF_List strings` parameter; if you place your model prior to the reference to the Lambertian model, you will need to increase the value of `selectedItem` by the number of items you

added *before Lambert.text*. Next, in the `Panel.UIs` library, the `Choose_BRDF_UI` module must be given control of your new panel; add a reference to it, following the format of the other panels. You will need to connect its `visible` toggle to the `Select_BRDF` module as well as introduce a reference to the model-specific parameters in the `brdf_vars` group. The UI panel now knows about the panel for your model.

The final step requires you to introduce a model-specific group of parameters to the `Build_BRDF` module in the `BRDFvis_Funcs` library. Follow the example offered by the other BRDFs. This module makes use of a C++ source file *brdf_build.cxx* located in the `src` subdirectory. Add a `case` statement to the `brdf_build_field()` function, and call the appropriate API functions to read values from your new panel. Now, back in *templ.v*, locate the use of `Build_BRDF` in `BRDF_ModPanel` (located in the `BRDFvis_Panels` library), and connect your model's parameters to the ones in `Build_BRDF`.

After all these steps, the next time you start the `BRDFvis` project in `Express`, you will need to recompile. Select the `BRDF_Functions` sublibrary (highlight it), then select `Compile` from the `Project` menu. After a long compilation process, during which the previously built Oregon BRDF Library is linked, the application will be ready to run. Instantiate it as described in Chapter II.

APPENDIX D

BRDF MODEL DERIVATIONS

Throughout chapter III, model formulations were simplified to present the main components of each BRDF computation. This chapter presents the additional formulae necessary to complete specific models.

Blinn

Blinn offers a variety of phase functions as options to be used in achieving a fit to known reflectance from scattering events that might be reasonably approximated by his geometric model in [2].

$$\varphi(\alpha) = \begin{cases} 1 & \text{constant} \\ 1 + x \cos \alpha & \text{anisotropic} \\ \frac{8}{3\pi} (\sin \alpha + (\pi - \alpha) \cos \alpha) & \text{Lambertian} \\ \frac{3}{4}(1 + \cos^2 \alpha) & \text{Rayleigh scattering} \\ \frac{1-g^2}{(1+g^2-2g \cos \alpha)^{3/2}} & \text{Henye-Greenstein} \end{cases}$$

Cook-Torrance

Instead of the Beckmann distribution, Davies suggests a gaussian distribution of microfacets where h is the RMS height of the surface in relation to the average surface [27].

$$D = e^{-\left(\frac{4\pi h}{\lambda} \cos \theta_i\right)^2}$$

He-Torrance

In [10], He et al. present the derivation of the model for both polarized and unpolarized light. For brevity, only the unpolarized calculations are summarized here. Note that even though the formulation is unpolarized, it still requires the polarization vectors \hat{s} and \hat{p} for both the incident and reflected directions, defined by the direction of propagation of the electromagnetic wave. These vectors are derived from the normal (N), and the incident (\hat{k}_i) and reflected (\hat{k}_r) vectors.

$$|F|^2 = \frac{1}{2}(F_s^2 + F_p^2) = f(\theta_i, \theta_r, \bar{n}(\lambda))$$

$$G = \left(\frac{\vec{v} \cdot \vec{v}}{v_z} \right)^2 \cdot \frac{1}{|\hat{k}_r \times \hat{k}_i|^4} \cdot [(\hat{s}_r \cdot \hat{k}_i)^2 + (\hat{p}_r \cdot \hat{k}_i)^2] \cdot [(\hat{s}_i \cdot \hat{k}_r)^2 + (\hat{p}_i \cdot \hat{k}_r)^2]$$

$$S = S(\theta_i, \theta_r, \sigma_0/\tau) = S_i(\theta_i) \cdot S_r(\theta_r)$$

$$S_i(\theta_i) = \left[1 - \frac{1}{2} \operatorname{erfc} \left(\frac{\tau \cot \theta_i}{2\sigma_0} \right) \right] / [\Lambda(\cot \theta_i) + 1]$$

$$S_r(\theta_r) = \left[1 - \frac{1}{2} \operatorname{erfc} \left(\frac{\tau \cot \theta_r}{2\sigma_0} \right) \right] / [\Lambda(\cot \theta_r) + 1]$$

$$\Lambda(\cot \theta) = \frac{1}{2} \left(\frac{2}{\pi^{1/2}} \cdot \frac{\sigma_0}{\tau \cot \theta} - \operatorname{erfc} \left(\frac{\tau \cot \theta}{2\sigma_0} \right) \right)$$

$$D = \frac{\pi^2 \tau^2}{4\lambda^2} \cdot \sum_{m=1}^{\infty} \frac{g^m e^{-g}}{m! \cdot m} \cdot e^{-v_{zv}^2 \tau^2 / 4m}$$

$$g = [(2\pi\sigma)(\cos \theta_i + \cos \theta_r)]^2$$

$$\sigma = \sigma_0 \cdot \left[1 + \left(\frac{z_0}{\sigma_0} \right)^2 \right]^{-1/2}$$

$$\sqrt{\frac{\pi}{2}} z_0 = \frac{\sigma_0}{4} (K_i + K_r) \cdot e^{-z_0^2 / 2\sigma_0^2}$$

$$K_i = \tan \theta_i \cdot \operatorname{erfc} \left(\frac{\tau}{2\sigma_0} \cot \theta_i \right)$$

$$K_r = \tan \theta_r \cdot \operatorname{erfc} \left(\frac{\tau}{2\sigma_0} \cot \theta_r \right)$$

$$\vec{v} = \hat{k}_r - \hat{k}_i$$

$$v_{xy} = \sqrt{v_x^2 + v_y^2}$$

$$\hat{s}_i = \frac{\hat{k}_i \times \hat{n}}{|\hat{k}_i \times \hat{n}|}$$

$$\hat{s}_r = \frac{\hat{k}_r \times \hat{n}}{|\hat{k}_r \times \hat{n}|}$$

$$\hat{p}_i = \hat{s}_i \times \hat{k}_i$$

$$\hat{p}_r = \hat{s}_r \times \hat{k}_r$$

Note that v_x and v_y are the x- and y-components of \vec{v} . Also, z_0 is a place-holder for a value that is derived within the full illumination calculation (it relies on σ_0 , K_i , and K_r).

He et al. subsequently presented an optimization to the model that employs a spline surface to approximate the time-consuming part of the formula [9]. This optimization permits the model to be used in a wider variety of applications by making the computation efficient.

Oren-Nayar

The three coefficients [19] requires in the evaluation of L_r^1 and L_r^2 are functions of ϕ_r , ϕ_i , α , β , and σ :

$$\begin{aligned}
 C_1 &= 1 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.33} \\
 C_2 &= \begin{cases} 0.45 \frac{\sigma^2}{\sigma^2 + 0.09} \sin \alpha & \text{if } \cos(\phi_r - \phi_i) \geq 0 \\ 0.45 \frac{\sigma^2}{\sigma^2 + 0.09} \left(\sin \alpha - \left(\frac{2\beta}{\pi} \right)^3 \right) & \text{otherwise} \end{cases} \\
 C_3 &= 0.125 \left(\frac{\sigma^2}{\sigma^2 + 0.09} \right) \left(\frac{4\alpha\beta}{\pi^2} \right)^2
 \end{aligned}$$

Poulin-Fournier

Poulin and Fournier present their model in pseudo-code in [21]. First they compute ϕ_d and ϕ_h , so they can specify θ_M , the angle between the normal and the beginning of the floor (half the exposed arc length).

$$\begin{aligned}
 \phi_h &= \cos^{-1} h \\
 \phi_d &= \begin{cases} \sin^{-1}(d/2) & \text{if } d < 2 \\ \frac{\pi}{2} & \text{otherwise} \end{cases} \\
 \theta_M &= \min(\phi_d, \phi_h)
 \end{aligned}$$

Once θ_M is available and θ_E and θ_L have been derived by projection, angles, arc lengths, and floor lengths are derived that are variously hiding (h), self-hiding (sh), shadowed (s), self-shadowed (ss), visible (v), illuminated (i), or both visible and illuminated (vi).

$$\begin{aligned}
& \theta_{ss} = \frac{\pi}{2} - \theta_L \\
& f = \max(d - 2\sqrt{1 - h^2}, 0) \\
& \text{if } (\theta_{ss} > \theta_M) \\
& \quad \left\{ \begin{array}{l} \theta_s = \theta_{ss} = \theta_M \\ f_i = f \\ f_s = 0 \end{array} \right. \\
& \quad \left. \vphantom{\text{if}} \right\} \\
& \text{else} \\
& \quad \left\{ \begin{array}{l} \text{if } \left(\frac{1}{\cos \theta_L} < (d - 1)\right) \theta_s = \frac{\pi}{2} \\ \text{else } \theta_s = \theta_L + \sin^{-1}(d \cos \theta_L - 1) \\ \theta_s = \min(\theta_s, \theta_M) \\ f_s = \min\left(\frac{1}{\cos \theta_L} - \sqrt{1 - h^2} - \left(\frac{h \sin \theta_L}{\cos \theta_L}\right), f\right) \\ f_s = \max(f_s, 0) \\ f_i = f - f_s \end{array} \right. \\
& \quad \left. \vphantom{\text{else}} \right\}
\end{aligned}$$

To derive the hiding (θ_h) and self-hiding (θ_{sh}) angles, and the visible (f_v) and hidden (f_h) parts of the floor, swap θ_E in for θ_L in the preceding pseudo-code.

Ward

To perform Monte Carlo sampling of the Ward BRDF, the naive stochastic approach is inefficient. Because the shape of the reflectance lobe is known to be gaussian, the samples can be chosen with a technique that is both deterministic and stochastic, yielding much better results with the same number of samples [25].

A ray should be cast in the direction $\langle \delta, \phi \rangle$, where δ is the polar component and ϕ is the azimuth component of the 3D angle:

$$\begin{aligned}
\delta &= \left[\frac{-\log(u_1)}{\cos^2 \phi / \alpha_x^2 + \sin^2 \phi / \alpha_y^2} \right]^{1/2} \\
\phi &= \tan^{-1} \left[\frac{\alpha_y}{\alpha_x} \tan(2\pi u_2) \right]
\end{aligned}$$

u_1 and u_2 are uniform random variables where $u_i \in (0, 1]$.

BIBLIOGRAPHY

- [1] J. Beard, J.R. Maxwell, S. Weiner, D. Ladd, and S. Ladd. Bidirectional Reflectance Model Validation and Utilization. Technical Report 73-303, AFAL, 1973.
- [2] James F. Blinn. Light Reflection Functions for Simulation of Clouds and Dusty Surfaces. In *ACM Siggraph '82 Conference Proceedings*, volume 16, pages 21–29, July 1982.
- [3] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional Reflection Functions from Surface Bump Maps. In *ACM Siggraph '87 Conference Proceedings*, volume 21, pages 273–282, July 1987.
- [4] Robert L. Cook and Kenneth E. Torrance. A Reflectance Model for Computer Graphics. In *ACM Siggraph '81 Conference Proceedings*, volume 15, pages 307–316, August 1981.
- [5] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink. Reflectance and Texture of Real-World Surfaces. Technical Report CUCS-046-96, Computer Science Department, Columbia University, 1996.
- [6] K. Ellis. Reflectance Phenomenology and Modeling Tutorial. Technical report, Environmental Research Institute of Michigan, 1994.
<http://www.irim.org/on-line-docs/GUIDE/guide.frm.html>.
- [7] J.S. Gondek, Gary W. Meyer, and J.G. Newman. Wavelength Dependent Reflectance Functions. In *ACM Siggraph '94 Conference Proceedings*, pages 213–220, 1994.
- [8] Bruce W. Hapke. A Theoretical Photometric Function for the Lunar Surface. *Journal of Geophysical Research*, 68:4571–4586, August 1963.
- [9] Xiao D. He, Patrick O. Heynen, Richard L. Phillips, Kenneth E. Torrance, David H. Salesin, and Donald P. Greenberg. A Fast and Accurate Light Reflection Model. In *ACM Siggraph '92 Conference Proceedings*, volume 26, pages 253–254, July 1992.

- [10] Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. A Comprehensive Physical Model for Light Reflection. In *ACM Siggraph '91 Conference Proceedings*, volume 25, pages 175–186, July 1991.
- [11] James Kajiya and B. Von Herzen. Ray Tracing Volume Densities. In *ACM Siggraph '84 Conference Proceedings*, volume 18, pages 165–174, July 1984.
- [12] James T. Kajiya. Anisotropic Reflection Models. In *ACM Siggraph '85 Conference Proceedings*, volume 19, pages 15–21, July 1985.
- [13] J.J. Koenderink, A.J. van Doorn, and M. Stavridi. Bidirectional Reflection Distribution Functions Expressed in Terms of Surface Scattering Modes. In *European Conference on Computer Vision*, pages 28–39, 1996.
- [14] Eric Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear Approximation of Reflectance Functions. In *ACM Siggraph '97 Conference Proceedings*, pages 117–126, August 1997.
- [15] R. Lewis. Making Shaders more Physically Plausible. In *Fourth Eurographics Workshop on Rendering*, pages 47–62, June 1993.
- [16] Marcel Minnaert. *Light and Color in the Outdoors*. Springer-Verlag, New York, 1974.
- [17] F.E. Nicodemus, J.C. Richmond, and J.J. Hsia. Geometrical Considerations and Nomenclature for Reflectance. Technical Report MN-160, U.S. Department of Commerce, National Bureau of Standards, October 1977.
- [18] Nonconventional Exploitation Factors (NEF) Modeling. Technical Report 257-96, ORD, August 1996.
- [19] Michael Oren and Shree K. Nayar. Generalization of Lambert's Reflectance Model. In *ACM Siggraph '94 Conference Proceedings*, pages 239–246, July 1994.
- [20] Bui-Thong Phong. Illumination for Computer Generated Images. *Communications of the ACM*, 18:311–317, June 1975.
- [21] Pierre Poulin and Alain Fournier. A Model for Anisotropic Reflection. In *ACM Siggraph '90 Conference Proceedings*, volume 24, pages 273–282, 1990.
- [22] Christophe Schlick. An Inexpensive BRDF Model for Physically-base Rendering. In *Eurographics '94*, volume 13, pages 233–246, 1994.
- [23] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Publishing Corporation, second edition, 1981.

- [24] Kenneth E. Torrance and E.M. Sparrow. Theory for Off-Specular Reflection from Roughened Surfaces. *Journal of the Optical Society of America*, 57, September 1967.
- [25] Gregory J. Ward. Measuring and Modeling Anisotropic Reflection. In *ACM Siggraph '92 Conference Proceedings*, volume 26, pages 265–272, July 1992.
- [26] Greg Ward Larson and Rob Shakespeare. *Rendering with Radiance: The Art and Science of Lighting Visualization*. Morgan Kaufmann Publishers, Inc., 1998.
- [27] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques*. ACM Press, U.K., 1992.
- [28] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting Reflectance Functions from Complex Surfaces. In *ACM Siggraph '92 Conference Proceedings*, volume 26, pages 255–264, July 1992.
- [29] Harold B. Westlund, Gary W. Meyer, and Peter A. Walker. iBRDF. Documentation on the iBRDF extension to Greg Ward's Radiance Renderer, May 1999.
- [30] Turner Whitted. An Improved Illumination Model for Shaded Display. *Communications of the ACM*, 23:343–349, June 1980.

