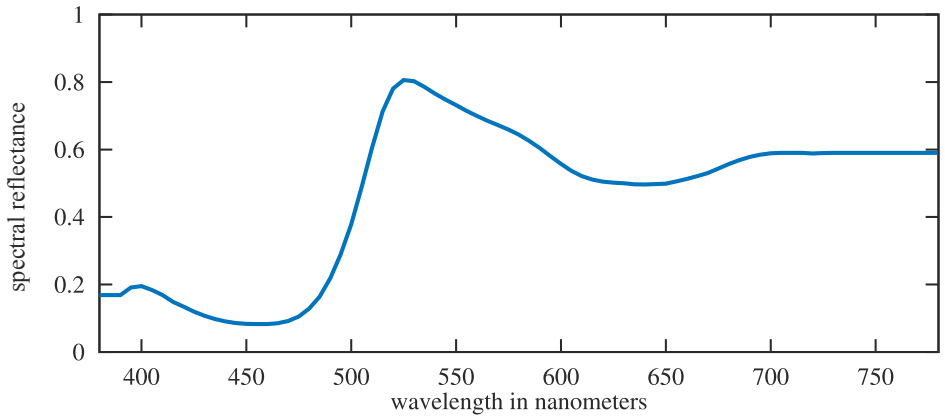

Figure 8.11. The spectral reflectance of a yellow banana [544].

Our example shows a screen material designed for use with laser projectors. It has high reflectance in narrow bands matching laser projector wavelengths and low reflectance for most other wavelengths. This causes it to reflect most of the light from the projector, but absorb most of the light from other light sources. An RGB renderer will produce gross errors in this case.

However, the situation shown in Figure 8.10 is far from typical. The spectral reflectance curves for surfaces encountered in practice are much smoother, such as the one in Figure 8.11. Typical illuminant SPDs resemble the D65 illuminant rather than the laser projector in the example. When both the illuminant SPD and surface spectral reflectance are smooth, the errors introduced by RGB rendering are relatively subtle.

In *predictive rendering* applications, these subtle errors can be important. For example, two spectral reflectance curves may have the same color appearance under one light source, but not another. This problem, called *metameric failure* or *illuminant metamerism*, is of serious concern when painting repaired car body parts, for example. RGB rendering would not be appropriate in an application that attempts to predict this type of effect.

However, for the majority of rendering systems, especially those for interactive applications, that are not aimed at producing predictive simulations, RGB rendering works surprisingly well [169]. Even feature-film offline rendering has only recently started to employ spectral rendering, and it is as yet far from common [660, 1610].

This section has touched on just the basics of color science, primarily to bring an awareness of the relation of spectra to color triplets and to discuss the limitations of devices. A related topic, the transformation of rendered scene colors to display values, will be discussed in the next section.

8.2 Scene to Screen

The next few chapters in this book are focused on the problem of physically based rendering. Given a virtual scene, the goal of physically based rendering is to compute the radiance that would be present in the scene if it were real. However, at that point the work is far from done. The final result—pixel values in the display's framebuffer—still needs to be determined. In this section we will go over some of the considerations involved in this determination.

8.2.1 High Dynamic Range Display Encoding

The material in this section builds upon Section 5.6, which covers display encoding. We decided to defer coverage of high dynamic range (HDR) displays to this section, since it requires background on topics, such as color gamuts, that had not yet been discussed in that part of the book.

Section 5.6 discussed display encoding for standard dynamic range (SDR) monitors, which typically use the sRGB display standard, and SDR televisions, which use the Rec. 709 and Rec. 1886 standards. Both sets of standards have the same RGB gamut and white point (D65), and somewhat similar (but not identical) nonlinear display encoding curves. They also have roughly similar reference white luminance levels (80 $\mathrm{cd/m^2}$ for sRGB, 100 $\mathrm{cd/m^2}$ for Rec. 709/1886). These luminance specifications have not been closely adhered to by monitor and television manufacturers, who in practice tend to manufacture displays with brighter white levels [1081].

HDR displays use the Rec. 2020 and Rec. 2100 standards. Rec. 2020 defines a color space with a significantly wider color gamut, as shown in Figure 8.12, and the same white point (D65) as the Rec. 709 and sRGB color spaces. Rec. 2100 defines two nonlinear display encodings: *perceptual quantizer* (PQ) [1213] and *hybrid log-gamma* (HLG). The HLG encoding is not used much in rendering situations, so we will focus here on PQ, which defines a peak luminance value of $10,000$ $\mathrm{cd/m^2}$.

Although the peak luminance and gamut specifications are important for encoding purposes, they are somewhat aspirational as far as actual displays are concerned. At the time of writing, few consumer-level HDR displays have peak luminance levels that exceed even 1500 $\mathrm{cd/m^2}$. In practice, display gamuts are much closer to that of DCI-P3 (also shown in Figure 8.12) than Rec. 2020. For this reason, HDR displays perform internal tone and gamut mapping from the standard specifications down to the actual display capabilities. This mapping can be affected by metadata passed by the application to indicate the actual dynamic range and gamut of the content [672, 1082].

From the application side, there are three paths for transferring images to an HDR display, though not all three may be available depending on the display and operating system:

1. HDR10—Widely supported on HDR displays as well as PC and console operating systems. The framebuffer format is 32 bits per pixel with 10 unsigned integer bits for each RGB channel and 2 for alpha. It uses PQ nonlinear encoding

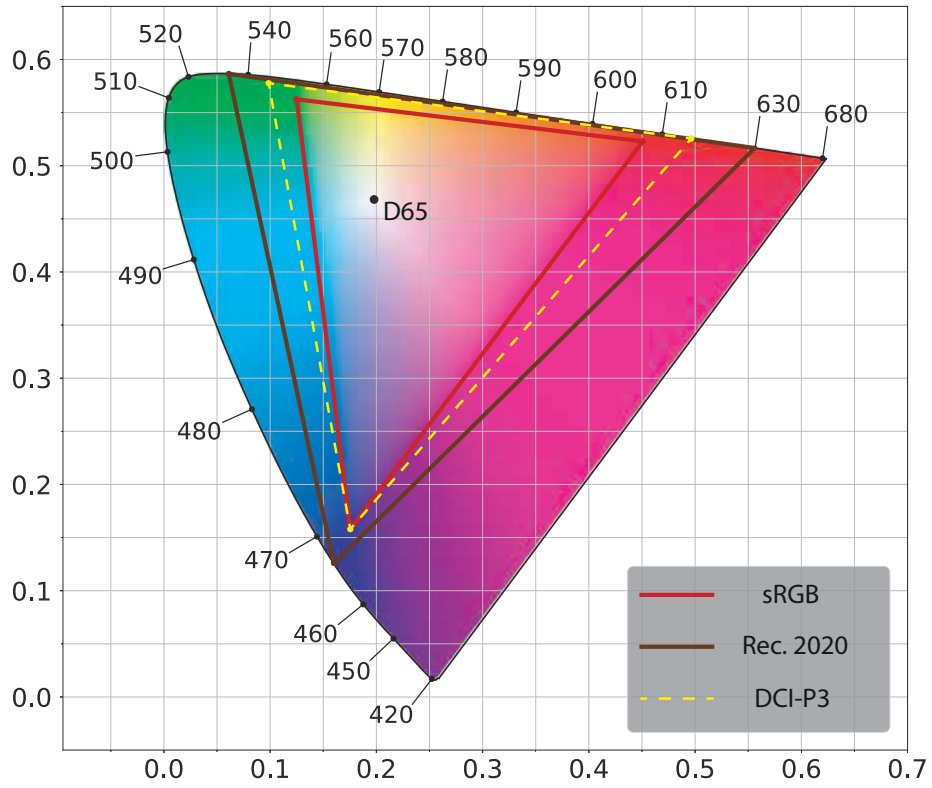

Figure 8.12. A CIE 1976 UCS diagram showing the gamuts and white point (D65) of the Rec. 2020 and sRGB/Rec. 709 color spaces. The gamut of the DCI-P3 color space is also shown for comparison.

and Rec. 2020 color space. Each HDR10 display model performs its own tone mapping, one that is not standardized or documented.

2. scRGB (linear variant)—Only supported on Windows operating systems. Nominally it uses sRGB primaries and white level, though both can be exceeded since the standard supports RGB values less than 0 and greater than 1. The framebuffer format is 16-bit per channel, and stores linear RGB values. It can work with any HDR10 display since the driver converts to HDR10. It is useful primarily for convenience and backward compatibility with sRGB.

3. Dolby Vision—Proprietary format, not yet widely supported in displays or on any consoles (at the time of writing). It uses a custom 12-bit per channel framebuffer format, and uses PQ nonlinear encoding and Rec. 2020 color space. The display internal tone mapping is standardized across models (but not documented).

Lottes [1083] points out that there is actually a fourth option. If the exposure and color are adjusted carefully, then an HDR display can be driven through the regular SDR signal path with good results.

With any option other than scRGB, as part of the display-encoding step the application needs to convert the pixel RGB values from the rendering working space to Rec. 2020—which requires a 3×3 matrix transform—and to apply the PQ encoding, which is somewhat more expensive than the Rec. 709 or sRGB encoding functions [497]. Patry [1360] gives an inexpensive approximation to the PQ curve. Special care is needed when compositing user interface (UI) elements on HDR displays to ensure that the user interface is legible and at a comfortable luminance level [672].

8.2.2 Tone Mapping

In Sections 5.6 and 8.2.1 we discussed display encoding, the process of converting linear radiance values to nonlinear code values for the display hardware. The function applied by display encoding is the inverse of the display's electrical optical transfer function (EOTF), which ensures that the input linear values match the linear radiance emitted by the display. Our earlier discussion glossed over an important step that occurs between rendering and display encoding, one that we are now ready to explore.

Tone mapping or *tone reproduction* is the process of converting scene radiance values to display radiance values. The transform applied during this step is called the *end-to-end transfer function*, or the *scene-to-screen transform*. The concept of *image state* is key to understanding tone mapping [1602]. There are two fundamental image states. *Scene-referred* images are defined in reference to scene radiance values, and *display-referred* images are defined in reference to display radiance values. Image state is unrelated to encoding. Images in either state may be encoded linearly or nonlinearly. Figure 8.13 shows how image state, tone mapping, and display encoding fit together in the *imaging pipeline*, which handles color values from initial rendering to final display.

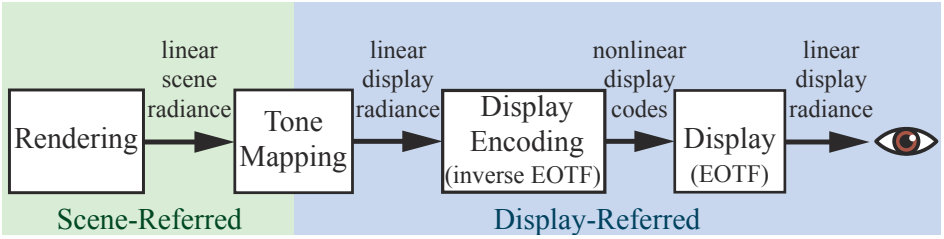

Figure 8.13. The imaging pipeline for synthetic (rendered) images. We render linear scene-referred radiance values, which tone mapping converts to linear display-referred values. Display encoding applies the inverse EOTF to convert the linear display values to nonlinearly encoded values (codes), which are passed to the display. Finally, the display hardware applies the EOTF to convert the nonlinear display values to linear radiance emitted from the screen to the eye.

Figure 8.14. The goal of image reproduction is to ensure that the perceptual impression evoked by the reproduction (right) is as close as possible to that of the original scene (left).

There are several common misconceptions regarding the goal of tone mapping. It is not to ensure that the scene-to-screen transform is an identity transform, perfectly reproducing scene radiance values at the display. It is also not to "squeeze" every bit of information from the high dynamic range of the scene into the lower dynamic range of the display, though accounting for differences between scene and display dynamic range does play an important part.

To understand the goal of tone mapping, it is best to think of it as an instance of image reproduction [757]. The goal of image reproduction is to create a display-referred image that reproduces—as closely as possible, given the display properties and viewing conditions—the perceptual impression that the viewer would have if they were observing the original scene. See Figure 8.14.

There is a type of image reproduction that has a slightly different goal. *Preferred image reproduction* aims at creating a display-referred image that looks better, in some sense, than the original scene. Preferred image reproduction will be discussed later, in Section 8.2.3.

The goal of reproducing a similar perceptual impression as the original scene is a challenging one, considering that the range of luminance in a typical scene exceeds

display capabilities by several orders of magnitude. The saturation (purity) of at least some of the colors in the scene are also likely to far outstrip display capabilities. Nevertheless, photography, television, and cinema do manage to produce convincing perceptual likenesses of original scenes, as did Renaissance painters. This achievement is possible by leveraging certain properties of the human visual system.

The visual system compensates for differences in absolute luminance, an ability called *adaptation*. Due to this ability, a reproduction of an outdoor scene shown on a screen in a dim room can produce a similar perception as the original scene, although the luminance of the reproduction is less than 1% of the original. However, the compensation provided by adaptation is imperfect. At lower luminance levels the perceived contrast is decreased (the *Stevens effect*), as is the perceived "colorfulness" (the *Hunt effect*).

Other factors affect actual or perceived contrast of the reproduction. The *surround* of the display (the luminance level outside the display rectangle, e.g., the brightness of the room lighting) may increase or decrease perceived contrast (the *Bartleson-Breneman effect*). *Display flare*, which is unwanted light added to the displayed image via display imperfections or screen reflections, reduces the actual contrast of the image, often to a considerable degree. These effects mean that if we want to preserve a similar perceptual effect as the original scene, we must boost the contrast and saturation of the display-referred image values [1418].

However, this increase in contrast exacerbates an existing problem. Since the dynamic range of the scene is typically much larger than that of the display, we have to choose a narrow window of luminance values to reproduce, with values above and below that window being clipped to black or white. Boosting the contrast further narrows this window. To partially counteract the clipping of dark and bright values, a soft roll-off is used to bring some shadow and highlight detail back.

All this results in a sigmoid (s-shaped) tone-reproduction curve, similar to the one provided by photochemical film [1418]. This is no accident. The properties of photochemical film emulsion were carefully adjusted by researchers at Kodak and other companies to produce effective and pleasing image reproduction. For these reasons, the adjective "filmic" often comes up in discussions of tone mapping.

The concept of *exposure* is critical for tone mapping. In photography, exposure refers to controlling the amount of light that falls on the film or sensor. However, in rendering, exposure is a linear scaling operation performed on the scene-referred image before the tone reproduction transform is applied. The tricky aspect of exposure is to determine what scaling factor to apply. The tone reproduction transform and exposure are closely tied together. Tone transforms are typically designed with the expectation that they will be applied to scene-referred images that have been exposed a certain way.

The process of scaling by exposure and then applying a tone reproduction transform is a type of *global tone mapping*, in which the same mapping is applied to all pixels. In contrast, a *local tone mapping* process uses different mappings pixel to pixel, based on surrounding pixels and other factors. Real-time applications have almost ex-

clusively used global tone mapping (with a few exceptions [1921]), so we will focus on this type, discussing first tone-reproduction transforms and then exposure.

It is important to remember that scene-referred images and display-referred images are fundamentally different. Physical operations are only valid when performed on scene-referred data. Due to display limitations and the various perceptual effects we have discussed, a nonlinear transform is always needed between the two image states.

Tone Reproduction Transform

Tone reproduction transforms are often expressed as one-dimensional curves mapping scene-referred input values to display-referred output values. These curves can be applied either independently to R, G, and B values or to luminance. In the former case, the result will automatically be in the display gamut, since each of the display-referred RGB channel values will be between 0 and 1. However, performing nonlinear operations (especially clipping) on RGB channels may cause shifts in saturation and hue, besides the desired shift in luminance. Giorgianni and Madden [537] point out that the shift in saturation can be perceptually beneficial. The contrast boost that most reproduction transforms use to counteract the Stevens effect (as well as surround and viewing flare effects) will cause a corresponding boost in saturation, which will counteract the Hunt effect as well. However, hue shifts are generally regarded as undesirable, and modern tone transforms attempt to reduce them by applying additional RGB adjustments after the tone curve.

By applying the tone curve to luminance, hue and saturation shifts can be avoided (or at least reduced). However, the resulting display-referred color may be out of the display's RGB gamut, in which case it will need to be mapped back in.

One potential issue with tone mapping is that applying a nonlinear function to scene-referred pixel colors can cause problems with some antialiasing techniques. The issue (and methods to address it) are discussed in Section 5.4.2.

The Reinhard tone reproduction operator [1478] is one of the earlier tone transforms used in real-time rendering. It leaves darker values mostly unchanged, while brighter values asymptotically go to white. A somewhat-similar tone-mapping operator was proposed by Drago et al. [375] with the ability to adjust for output display luminance, which may make it a better fit for HDR displays. Duiker created an approximation to a Kodak film response curve [391, 392] for use in video games. This curve was later modified by Hable [628] to add more user control, and was used in the game *Uncharted 2*. Hable's presentation on this curve was influential, leading to the "Hable filmic curve" being used in several games. Hable [634] later proposed a new curve with a number of advantages over his earlier work.

Day [330] presents a sigmoid tone curve that was used on titles from Insomniac Games, as well as the game *Call of Duty: Advanced Warfare*. Gotanda [571, 572] created tone transforms that simulate the response of film as well as digital camera sensors. These were used on the game *Star Ocean 4* and others. Lottes [1081] points out that the effect of display flare on the effective dynamic range of the display is

significant and highly dependent on room lighting conditions. For this reason, it is important to provide user adjustments to the tone mapping. He proposes a tone reproduction transform with support for such adjustments that can be used with SDR as well as HDR displays.

The *Academy Color Encoding System* (ACES) was created by the Science and Technology Council of the Academy of Motion Picture Arts and Sciences as a proposed standard for managing color for the motion picture and television industries. The ACES system splits the scene-to-screen transform into two parts. The first is the *reference rendering transform* (RRT), which transforms scene-referred values into display-referred values in a standard, device-neutral output space called the *output color encoding specification* (OCES). The second part is the *output device transform* (ODT), which converts color values from OCES to the final display encoding. There are many different ODTs, each one designed for a specific display device and viewing condition. The concatenation of the RRT and the appropriate ODT creates the overall transform. This modular structure is convenient for addressing a variety of display types and viewing conditions. Hart [672] recommends the ACES tone mapping transforms for applications that need to support both SDR and HDR displays.

Although ACES was designed for use in film and television, its transforms are seeing growing use in real-time applications. ACES tone mapping is enabled by default in the Unreal Engine [1802], and it is supported by the Unity engine as well [1801]. Narkowicz gives inexpensive curves fitted to the ACES RRT with SDR and HDR ODTs [1260, 1261], as does Patry [1359]. Hart [672] presents a parameterized version of the ACES ODTs to support a range of devices.

Tone mapping with HDR displays requires some care, since the displays will also apply some tone mapping of their own. Fry [497] presents a set of tone mapping transforms used in the Frostbite game engine. They apply a relatively aggressive tone reproduction curve for SDR displays, a less-aggressive one for displays using the HDR10 signal path (with some variation based on the peak luminance of the display), and no tone mapping with displays using the Dolby Vision path (in other words, they rely upon the built-in Dolby Vision tone mapping applied by the display). The Frostbite tone reproduction transforms are designed to be neutral, without significant contrast or hue changes. The intent is for any desired contrast or hue modifications to be applied via color grading (Section 8.2.3). To this end, the tone reproduction transform is applied in the IC_TC_P color space [364], which was designed for perceptual uniformity and orthogonality between the chrominance and luminance axes. The Frostbite transform tone-maps the luminance and increasingly desaturates the chromaticity as the luminance rolls off to display white. This provides a clean transform without hue shifts.

Ironically, following issues with assets (such as fire effects) that were authored to leverage the hue shifts in their previous transform, the Frostbite team ended up modifying the transform, enabling users to re-introduce some degree of hue shifting to the display-referred colors. Figure 8.15 shows the Frostbite transform compared with several others mentioned in this section.

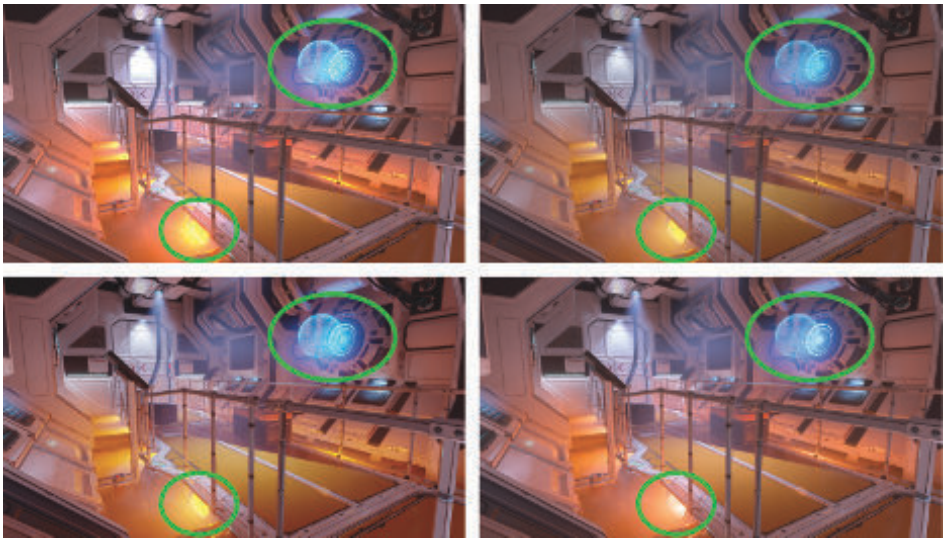

Figure 8.15. A scene with four different tone transforms applied. Differences are primarily seen in the circled areas, where scene pixel values are especially high. Upper left: clipping (plus sRGB OETF); upper right: Reinhard [1478]; lower left: Duiker [392]; lower right: Frostbite (hue-preserving version) [497]. The Reinhard, Duiker, and Frostbite transforms all preserve highlight information lost by clipping. However, the Reinhard curve tends to desaturate the darker parts of the image [628, 629], while the Duiker transform increases saturation in darker regions, which is sometimes regarded as a desirable trait [630]. By design, the Frostbite transform preserves both saturation and hue, avoiding the strong hue shift that can be seen in the lower left circle on the other three images. *(Images courtesy of ©2018 Electronic Arts Inc.)*

Exposure

A commonly used family of techniques for computing exposure relies on analyzing the scene-referred luminance values. To avoid introducing stalls, this analysis is typically done by sampling the previous frame.

Following a recommendation by Reinhard et al. [1478], one metric that was used in earlier implementations is the log-average scene luminance. Typically, the exposure was determined by computing the log-average value for the frame [224, 1674]. This log-average is computed by performing a series of down-sampling post-process passes, until a final, single value for the frame is computed.

Using an average value tends to be too sensitive to outliers, e.g., a small number of bright pixels could affect the exposure for the entire frame. Subsequent implementations ameliorated this problem by instead using a histogram of luminance values. Instead of the average, a histogram allows computing the median, which is more robust. Additional data points in the histogram can be used for improved results. For example, in *The Orange Box* by Valve, heuristics based on the 95th percentile and the median were used to determine exposure [1821]. Mittring describes the use of compute shaders to generate the luminance histogram [1229].

The problem with the techniques discussed so far is that pixel luminance is the wrong metric for driving exposure. If we look at photography practices, such as Ansel Adams' Zone System [10] and how incident light meters are used to set exposure, it becomes clear that it is preferable to use the lighting alone (without the effect of surface albedo) to determine exposure [757]. Doing so works because, to a first approximation, photographic exposure is used to counteract lighting. This results in a print that shows primarily the surface colors of objects, which corresponds to the *color constancy* property of the human visual system. Handling exposure in this way also ensures that correct values are passed to the tone transform. For example, most tone transforms used in the film or television industry are designed to map the exposed scene-referred value 0.18 to the display-referred value 0.1, with the expectation that 0.18 represents an 18% gray card in the dominant scene lighting [1418, 1602].

Although this approach is not yet common in real-time applications, it is starting to see use. For example, the game *Metal Gear Solid V: Ground Zeroes* has an exposure system based on lighting intensity [921]. In many games, static exposure levels are manually set for different parts of the environment based on known scene lighting values. Doing so avoids unexpected dynamic shifts in exposure.

8.2.3 Color Grading

In Section 8.2.2 we mentioned the concept of preferred image reproduction, the idea of producing an image that looks better in some sense than the original scene. Typically this involves creative manipulation of image colors, a process known as *color grading*.

Digital color grading has been used in the movie industry for some time. Early examples include the films *O Brother, Where Art Thou?* (2000) and *Amélie* (2001). Color grading is typically performed by interactively manipulating the colors in an example scene image, until the desired creative "look" is achieved. The same sequence of operations is then re-applied across all the images in a shot or sequence. Color grading spread from movies to games, where it is now widely used [392, 424, 756, 856, 1222].

Selan [1601] shows how to "bake" arbitrary color transformations from a color grading or image editing application into a three-dimensional color lookup table (LUT). Such tables are applied by using the input R, G, and B values as x-, y-, and z-coordinates for looking up a new color in the table, and thus can be used for any mapping from input to output color, up to the limitation of the LUT's resolution. Selan's baking process starts by taking an identity LUT (one that maps every input color to the same color) and "slicing" it to create a two-dimensional image. This sliced LUT image is then loaded into a color grading application, and the operations that define a desired creative look are applied to it. Care is needed to apply only color operations to the LUT, avoiding spatial operations such as blurs. The edited LUT is then saved out, "packed" into a three-dimensional GPU texture, and used in a rendering application to apply the same color transformations on the fly to rendered pixels. Iwanicki [806] presents a clever way to reduce sampling errors when storing a color transform in a LUT, using least-squares minimization.

Figure 8.16. A scene from the game *Uncharted 4*. The screenshot on top has no color grading. The other two screenshots each have a color grading operation applied. An extreme color grading operation (multiplication by a highly saturated cyan color) was chosen for purposes of illustration. In the bottom left screenshot, the color grading was applied to the display-referred (post-tone-mapping) image, and in the bottom right screenshot, it was applied to the scene-referred (pre-tone-mapping) image. (*UNCHARTED 4 A Thief's End* ©/^{TM} *2016 SIE. Created and developed by Naughty Dog LLC.)*

In a later publication, Selan [1602] distinguishes between two ways to perform color grading. In one approach, color grading is performed on display-referred image data. In the other, the color grading operations are performed on scene-referred data that is previewed through a display transform. Although the display-referred color grading approach is easier to set up, grading scene-referred data can produce higher-fidelity results.

When real-time applications first adopted color grading, the display-referred approach was predominant [756, 856]. However, the scene-referred approach has since been gaining traction [198, 497, 672] due to its higher visual quality. See Figure 8.16. Applying color grading to scene-referred data also provides the opportunity to save some computation by baking the tone mapping curve into the grading LUT [672], as done in the game *Uncharted 4* [198].

Before LUT lookup, scene-referred data must be remapped to the range $[0, 1]$ [1601]. In the Frostbite engine [497] the perceptual quantizer OETF is used for this purpose, though simpler curves could be used. Duiker [392] uses a log curve, and Hable [635] recommends using a square root operator applied once or twice.

Hable [635] presents a good overview of common color grading operations and implementation considerations.

Further Reading and Resources

For colorimetry and color science, the "bible" is *Color Science* by Wyszecki and Stiles [1934]. Other good colorimetry references include *Measuring Colour* by Hunt [789] and *Color Appearance Models* by Fairchild [456].

Selan's white paper [1602] gives a good overview of image reproduction and the "scene to screen" problem. Readers who want to learn still more about this topic will find *The Reproduction of Colour* by Hunt [788] and *Digital Color Management* by Giorgianni and Madden [537] to be excellent references. The three books in the *Ansel Adams Photography Series* [9, 10, 11], especially *The Negative*, provide an understanding of how the art and science of film photography has influenced the theory and practice of image reproduction to this day. Finally, the book *Color Imaging: Fundamentals and Applications* by Reinhard and others [1480] gives a thorough overview of the whole area of study.

Chapter 9
Physically Based Shading

"Let the form of an object be what it may,—light, shade, and perspective will always make it beautiful."
—John Constable

In this chapter we cover the various aspects of physically based shading. We start with a description of the physics of light-matter interaction in Section 9.1, and in Sections 9.2 to 9.4 we show how these physics connect to the shading process. Sections 9.5 to 9.7 are dedicated to the building blocks used to construct physically based shading models, and the models themselves—covering a broad variety of material types—are discussed in Sections 9.8 to 9.12. Finally, in Section 9.13 we describe how materials are blended together, and we cover filtering methods for avoiding aliasing and preserving surface appearance.

9.1 Physics of Light

The interactions of light and matter form the foundation of physically based shading. To understand these interactions, it helps to have a basic understanding of the nature of light.

In physical optics, light is modeled as an electromagnetic *transverse wave*, a wave that oscillates the electric and magnetic fields perpendicularly to the direction of its propagation. The oscillations of the two fields are coupled. The magnetic and electric field vectors are perpendicular to each other and the ratio of their lengths is fixed. This ratio is equal to the phase velocity, which we will discuss later.

In Figure 9.1 we see a simple light wave. It is, in fact, the simplest possible—a perfect sine function. This wave has a single *wavelength*, denoted with the Greek letter λ (lambda). As we have seen in Section 8.1, the perceived color of light is strongly related to its wavelength. For this reason, light with a single wavelength is called *monochromatic light*, which means "single-colored." However, most light waves encountered in practice are *polychromatic*, containing many different wavelengths.

The light wave in Figure 9.1 is unusually simple in another respect. It is *linearly polarized*. This means that for a fixed point in space, the electric and magnetic fields

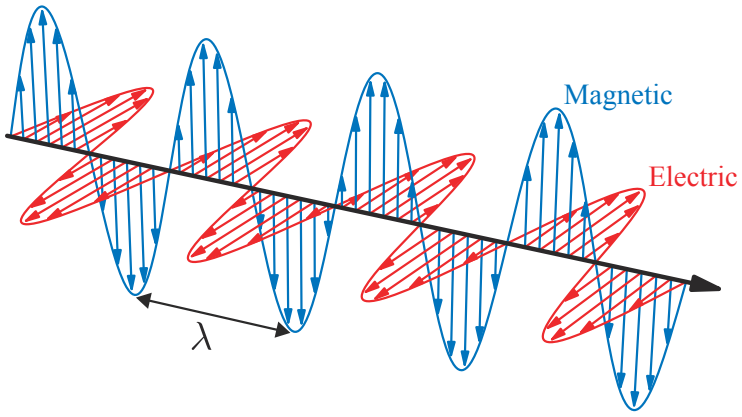

Figure 9.1. Light, an electromagnetic transverse wave. The electric and magnetic field vectors oscillate at 90° to each other and to the direction of propagation. The wave shown in the figure is the simplest possible light wave. It is both monochromatic (has a single wavelength λ) and linearly polarized (the electric and magnetic fields each oscillate along a single line).

each move back and forth along a line. In contrast, in this book we focus on *unpolarized* light, which is much more prevalent. In unpolarized light the field oscillations are spread equally over all directions perpendicular to the propagation axis. Despite their simplicity, it is useful to understand the behavior of monochromatic, linearly polarized waves, since any light wave can be factored into a combination of such waves.

If we track a point on the wave with a given phase (for example, an amplitude peak) over time, we will see it move through space at a constant speed, which is the wave's *phase velocity*. For a light wave traveling through a vacuum, the phase velocity is c, commonly referred to as the speed of light, about 300,000 kilometers per second.

In Section 8.1.1 we discussed the fact that for visible light, the size of a single wavelength is in the range of approximately 400–700 nanometers. To give some intuition for this length, it is about a half to a third of the width of a single thread of spider silk, which is itself less than a fiftieth of the width of a human hair. See Figure 9.2. In optics it is often useful to talk about the size of features relative to light wavelength. In this case we would say that the width of a spider silk thread is about 2λ–3λ (2–3 light wavelengths), and the width of a hair is about 100λ–200λ.

Light waves carry energy. The density of energy flow is equal to the product of the magnitudes of the electric and magnetic fields, which is—since the magnitudes are proportional to each other—proportional to the squared magnitude of the electric field. We focus on the electric field since it affects matter much more strongly than the magnetic field. In rendering, we are concerned with the *average* energy flow over time, which is proportional to the squared wave amplitude. This average energy flow density is the *irradiance*, denoted with the letter E. Irradiance and its relation to other light quantities were discussed in Section 8.1.1.

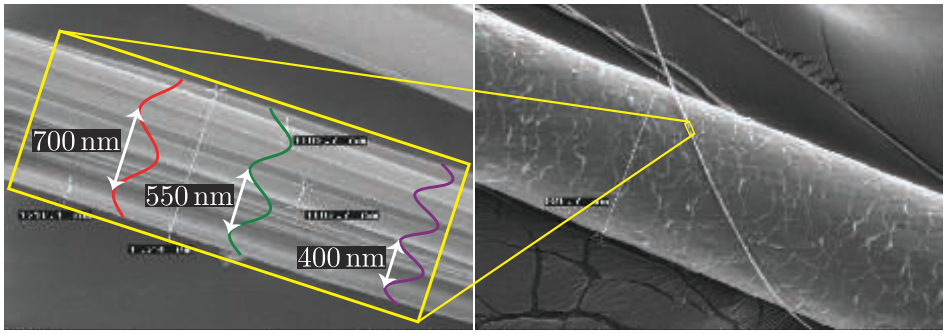

Figure 9.2. On the left visible light wavelengths are shown relative to a single thread of spider silk, which is a bit over 1 micron in width. On the right a similar thread of spider silk is shown next to a human hair, to give some additional context. *(Images courtesy of URnano/University of Rochester.)*

Light waves combine linearly. The total wave is the sum of the component waves. However, since irradiance is proportional to the square of the amplitudes, this would seem to lead to a paradox. For example, would summing two equal waves not lead to a "$1 + 1 = 4$" situation for irradiance? And since irradiance measures energy flow, would this not violate conservation of energy? The answers to these two questions are "sometimes" and "no," respectively.

To illustrate, we will look at a simple case: the addition of n monochromatic waves, identical except for phase. The amplitude of each of the n waves is a. As mentioned earlier, the irradiance E_1 of each wave is proportional to a^2, or in other words $E_1 = ka^2$ for some constant k.

Figure 9.3 shows three example scenarios for this case. On the left, the waves all line up with the same phase and reinforce each other. The combined wave irradiance is

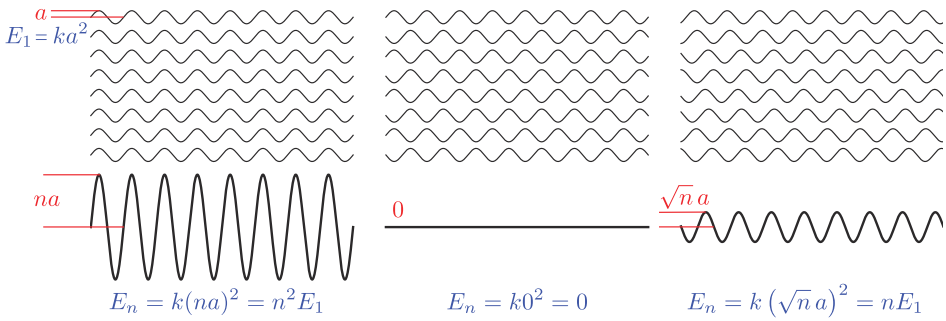

Figure 9.3. Three scenarios where n monochromatic waves with the same frequency, polarization, and amplitude are added together. From left to right: constructive interference, destructive interference, and incoherent addition. In each case the amplitude and irradiance of the combined wave (bottom) is shown relative to the n original waves (top).

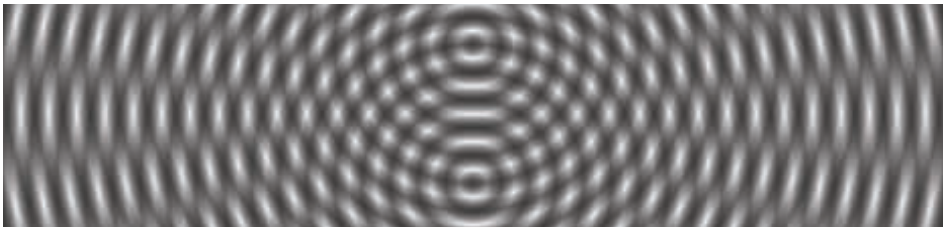

Figure 9.4. Monochromatic waves spreading out from two point sources, with the same frequency. The waves interfere constructively and destructively in different regions of space.

n^2 times that of a single wave, which is n times greater than the sum of the irradiance values of the individual waves. This situation is called *constructive interference*. In the center of the figure, each pair of waves is in opposing phase, canceling each other out. The combined wave has zero amplitude and zero irradiance. This scenario is *destructive interference*.

Constructive and destructive interference are two special cases of *coherent addition*, where the peaks and troughs of the waves line up in some consistent way. Depending on the relative phase relationships, coherent addition of n identical waves can result in a wave with irradiance anywhere between 0 and n^2 times that of an individual wave.

However, most often when waves are added up they are mutually *incoherent*, which means that their phases are relatively random. This is illustrated on the right in Figure 9.3. In this scenario, the amplitude of the combined wave is $\sqrt{n}\,a$, and the irradiance of the individual waves adds up linearly to n times the irradiance of one wave, as one would expect.

It would seem that destructive and constructive interference violate the conservation of energy. But Figure 9.3 does not show the full picture—it shows the wave interaction at only one location. As waves propagate through space, the phase relationship between them changes from one location to another, as shown in Figure 9.4. In some locations the waves interfere constructively, and the irradiance of the combined wave is greater than the sum of the irradiance values of the individual waves. In other locations they interfere destructively, causing the combined irradiance to be less than the sum of the individual wave irradiance values. This does not violate the law of conservation of energy, since the energy gained via constructive interference and the energy lost via destructive interference always cancel out.

Light waves are emitted when the electric charges in an object oscillate. Part of the energy that caused the oscillations—heat, electrical energy, chemical energy—is converted to light energy, which is radiated away from the object. In rendering, such objects are treated as light sources. We first discussed light sources in Section 5.2, and they will be described from a more physically based standpoint in Chapter 10.

After light waves have been emitted, they travel through space until they encounter some bit of matter with which to interact. The core phenomenon underlying the

majority of light-matter interactions is simple, and quite similar to the emission case discussed above. The oscillating electrical field pushes and pulls at the electrical charges in the matter, causing them to oscillate in turn. The oscillating charges emit new light waves, which redirect some of the energy of the incoming light wave in new directions. This reaction, called *scattering*, is the basis of a wide variety of optical phenomena.

A scattered light wave has the same frequency as the original wave. When, as is usually the case, the original wave contains multiple frequencies of light, each one interacts with matter separately. Incoming light energy at one frequency does not contribute to emitted light energy at a different frequency, except for specific—and relatively rare—cases such as fluorescence and phosphorescence, which we will not describe in this book.

An isolated molecule scatters light in all directions, with some directional variation in intensity. More light is scattered in directions close to the original axis of propagation, both forward and backward. The molecule's effectiveness as a scatterer—the chance that a light wave in its vicinity will be scattered at all—varies strongly by wavelength. Short-wavelength light is scattered much more effectively than longer-wavelength light.

In rendering we are concerned with collections of many molecules. Light interactions with such aggregates will not necessarily resemble interactions with isolated molecules. Waves scattered from nearby molecules are often mutually coherent, and thus exhibit interference, since they originate from the same incoming wave. The rest of this section is devoted to several important special cases of light scattering from multiple molecules.

9.1.1 Particles

In an *ideal gas*, molecules do not affect each other and thus their relative positions are completely random and uncorrelated. Although this is an abstraction, it is a reasonably good model for air at normal atmospheric pressure. In this case, the phase differences between waves scattered from different molecules are random and constantly changing. As a result, the scattered waves are incoherent and their energy adds linearly, as in the right part of Figure 9.3. In other words, the aggregate light energy scattered from n molecules is n times the light scattered from a single molecule.

In contrast, if the molecules are tightly packed into clusters much smaller than a light wavelength, the scattered light waves in each cluster are in phase and interfere constructively. This causes the scattered wave energy to add up quadratically, as illustrated in the left part of Figure 9.3. Thus the intensity of light scattered from a small cluster of n molecules is n^2 times the light scattered from an individual molecule, which is n times more light than the same number of molecules would scatter in an ideal gas. This relationship means that for a fixed density of molecules per cubic meter, clumping the molecules into clusters will significantly increase the intensity of scattered light. Making the clusters larger, while still keeping the overall molecular

density constant, will further increase scattered light intensity, until the cluster diameter becomes close to a light wavelength. Beyond that point, additional increases in cluster size will not further increase the scattered light intensity [469].

This process explains why clouds and fog scatter light so strongly. They are both created by condensation, which is the process of water molecules in the air clumping together into increasingly large clusters. This significantly increases light scattering, even though the overall density of water molecules is unchanged. Cloud rendering is discussed in Section 14.4.2.

When discussing light scattering, the term *particles* is used to refer to both isolated molecules and multi-molecule clusters. Since scattering from multi-molecule particles with diameters smaller than a wavelength is an amplified (via constructive interference) version of scattering from isolated molecules, it exhibits the same directional variation and wavelength dependence. This type of scattering is called *Rayleigh scattering* in the case of atmospheric particles and *Tyndall scattering* in the case of particles embedded in solids.

As particle size increases beyond a wavelength, the fact that the scattered waves are no longer in phase over the entire particle changes the characteristics of the scattering. The scattering increasingly favors the forward direction, and the wavelength dependency decreases until light of all visible wavelengths is scattered equally. This type of scattering is called *Mie scattering*. Rayleigh and Mie scattering are covered in more detail in Section 14.1.

9.1.2 Media

Another important case is light propagating through a *homogeneous medium*, which is a volume filled with uniformly spaced identical molecules. The molecular spacing does not have to be perfectly regular, as in a crystal. Liquids and non-crystalline solids can be optically homogeneous if their composition is pure (all molecules are the same) and they have no gaps or bubbles.

In a homogeneous medium, the scattered waves are lined up so that they interfere destructively in all directions except for the original direction of propagation. After the original wave is combined with all the waves scattered from individual molecules, the final result is the same as the original wave, except for its phase velocity and (in some cases) amplitude. The final wave does not exhibit any scattering—it has effectively been suppressed by destructive interference.

The ratio of the phase velocities of the original and new waves defines an optical property of the medium called the *index of refraction* (IOR) or refractive index, denoted by the letter n. Some media are *absorptive*. They convert part of the light energy to heat, causing the wave amplitude to decrease exponentially with distance. The rate of decrease is defined by the *attenuation index*, denoted by the Greek letter κ (kappa). Both n and κ typically vary by wavelength. Together, these two numbers fully define how the medium affects light of a given wavelength, and they are often combined into a single complex number $n + i\kappa$, called the *complex index of refraction*.

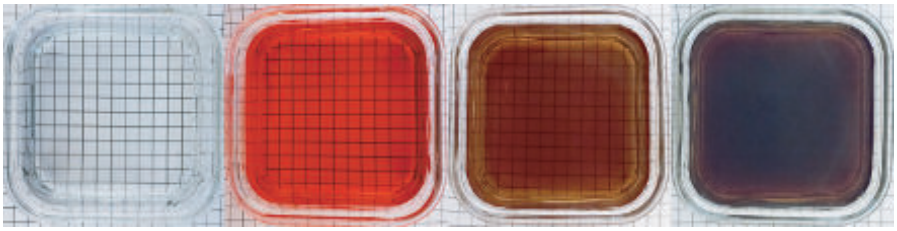

Figure 9.5. Four small containers of liquid with different absorption properties. From left to right: clean water, water with grenadine, tea, and coffee.

The index of refraction abstracts away the molecule-level details of light interaction and enables treating the medium as a continuous volume, which is much simpler.

While the phase velocity of light does not directly affect appearance, *changes* in velocity do, as we will explain later. On the other hand, light absorption has a direct impact on visuals, since it reduces the intensity of light and can (if varying by wavelength) also change its color. Figure 9.5 shows some examples of light absorption.

Nonhomogeneous media can often be modeled as homogeneous media with embedded scattering particles. The destructive interference that suppresses scattering in homogeneous media is caused by the uniform alignment of molecules, and thus of the scattered waves they produce. Any localized change in the distribution of molecules will break this pattern of destructive interference, allowing scattered light waves to propagate. Such a localized change can be a cluster of a different molecule type, an air gap, a bubble, or density variation. In any case, it will scatter light like the particles discussed earlier, with scattering properties similarly dependent on the cluster's size. Even gases can be modeled in this way. For these, the "scattering particles" are transient density fluctuations caused by the constant motion of the molecules. This model enables establishing a meaningful value of n for gases, which is useful for understanding their optical properties. Figure 9.6 shows some examples of light scattering.

Figure 9.6. From left to right: water, water with a few drops of milk, water with about 10% milk, whole milk, and opalescent glass. Most of milk's scattering particles are larger than visible light wavelengths, so its scattering is primarily colorless, with a faint blue tint apparent in the middle image. The scattering particles in the opalescent glass are all smaller than visible light wavelengths and thus scatter blue light more strongly than red light. Due to the split light and dark background, transmitted light is more visible on the left and scattered light is more visible on the right.

Figure 9.7. The left image shows that, over a distance of multiple meters, water absorbs light, especially red light, quite strongly. The right image shows noticeable light scattering over multiple miles of air, even in the absence of heavy pollution or fog.

Scattering and absorption are both scale-dependent. A medium that does not produce any apparent scattering in a small scene may have quite noticeable scattering at larger scales. For example, light scattering in air and absorption in water are not visible when observing a glass of water in a room. However, in extended environments both effects can be significant, as shown in Figure 9.7.

In the general case, a medium's appearance is caused by some combination of scattering and absorption, as shown in Figure 9.8. The degree of scattering determines cloudiness, with high scattering creating an opaque appearance. With somewhat rare exceptions, such as the opalescent glass in Figure 9.6, particles in solid and liquid media tend to be larger than a light wavelength, and tend to scatter light of all visible wavelengths equally. Thus any color tint is usually caused by the wavelength dependence of the absorption. The lightness of the medium is a result of both phenomena. A white color in particular is the result of a combination of high scattering and low absorption. This is discussed in more detail in Section 14.1.

9.1.3 Surfaces

From an optical perspective, an object surface is a two-dimensional interface separating volumes with different index of refraction values. In typical rendering situations, the outer volume contains air, with a refractive index of about 1.003, often assumed to be 1 for simplicity. The refractive index of the inner volume depends on the substance from which the object is made.

When a light wave strikes a surface, two aspects of that surface have important effects on the result: the substances on either side, and the surface geometry. We will start by focusing on the substance aspect, assuming the simplest-possible surface geometry, a perfectly flat plane. We denote the index of refraction on the "outside" (the side where the incoming, or *incident*, wave originates) as n_1 and the index of refraction on the "inside" (where the wave will be transmitted after passing through the surface) as n_2.

We have seen in the previous section that light waves scatter when they encounter a discontinuity in material composition or density, i.e., in the index of refraction. A

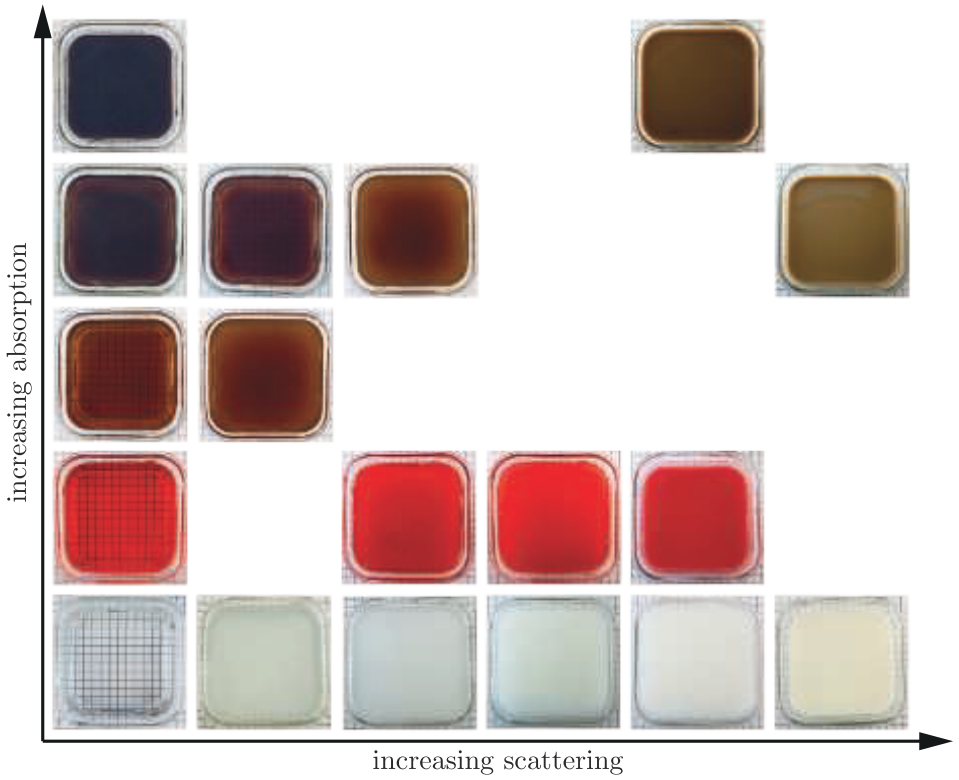

Figure 9.8. Containers of liquids that exhibit varying combinations of absorption and scattering.

planar surface separating different indices of refraction is a special type of discontinuity that scatters light in a specific way. The boundary conditions require that the electrical field component parallel to the surface is continuous. In other words, the projection of the electric field vector to the surface plane must match on either side of the surface. This has several implications:

1. At the surface, any scattered waves must be either in phase, or 180° out of phase, with the incident wave. Thus at the surface, the peaks of the scattered waves must line up either with the peaks or the troughs of the incident wave. This restricts the scattered waves to go in only two possible directions, one continuing forward into the surface and one retreating away from it. The first of these is the *transmitted wave*, and the second is the *reflected wave*.

2. The scattered waves must have the same frequency as the incident wave. We assume a monochromatic wave here, but the principles we discuss can be applied to any general wave by first decomposing it into monochromatic components.

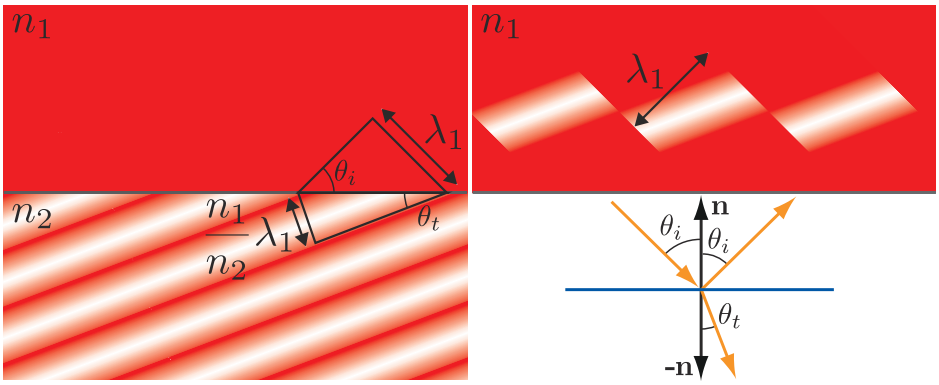

Figure 9.9. A light wave striking a planar surface separating indices of refraction n_1 and n_2. The left side of the figure shows a side view with the incident wave coming in from the upper left. The intensity of the red bands indicate wave phase. The spacing of the waves below the surface is changed proportionally to the ratio (n_1/n_2), which in this case is 0.5. The phases line up along the surface, so the change in spacing bends (refracts) the transmitted wave direction. The triangle construction shows the derivation of Snell's law. For clarity, the upper right of the figure shows the reflected wave separately. It has the same wave spacing as the incident wave, and thus its direction has the same angle with the surface normal. The lower right of the figure shows the wave direction vectors.

3. As a light wave moves from one medium to another, the phase velocity—the speed the wave travels through the medium—changes proportionally to the relative index of refraction (n_1/n_2). Since the frequency is fixed, the wavelength also changes proportionally to (n_1/n_2).

The final result is shown in Figure 9.9. The reflected and incident wave directions have the same angle θ_i with the surface normal. The transmitted wave direction is bent (*refracted*) at an angle θ_t, which has the following relation to θ_i:

$$\sin(\theta_t) = \frac{n_1}{n_2}\sin(\theta_i). \tag{9.1}$$

This equation for refraction is known as *Snell's law*. It is used in global refraction effects, which will be discussed further in Section 14.5.2.

Although refraction is often associated with clear materials such as glass and crystal, it happens at the surface of opaque objects as well. When refraction occurs with opaque objects, the light undergoes scattering and absorption in the object's interior. Light interacts with the object's medium, just as with the various cups of liquid in Figure 9.8. In the case of metals, the interior contains many free electrons (electrons not bound to molecules) that "soak up" the refracted light energy and redirect it into the reflected wave. This is why metals have high absorption as well as high reflectivity.

Figure 9.10. An example of light paths bending due to gradual changes in index of refraction, in this case caused by temperature variations. *("EE Lightnings heat haze," Paul Lucas, used under the CC BY 2.0 license.)*

The surface refraction phenomena we have discussed—reflection and refraction—require an abrupt change in index of refraction, occurring over a distance of less than a single wavelength. A more gradual change in index of refraction does not split the light, but instead causes its path to curve, in a continuous analog of the discontinuous bend that occurs in refraction. This effect commonly can be seen when air density varies due to temperature, such as mirages and heat distortion. See Figure 9.10.

Even an object with a well-defined boundary will have no visible surface if it is immersed in a substance with the same index of refraction. In the absence of an index of refraction change, reflection and refraction cannot occur. An example of this is seen in Figure 9.11.

Until now we have focused on the effect of the substances on either side of a surface. We will now discuss the other important factor affecting surface appearance: geometry. Strictly speaking, a perfectly flat planar surface is impossible. Every surface has irregularities of some kind, even if only the individual atoms comprising the surface. However, surface irregularities much smaller than a wavelength have no effect on light, and surface irregularities much larger than a wavelength effectively tilt the surface without affecting its *local* flatness. Only irregularities with a size in the range of 1–100 wavelengths cause the surface to behave differently than a flat plane, via a phenomenon called *diffraction* that will be discussed further in Section 9.11.

In rendering, we typically use *geometrical optics*, which ignores wave effects such as interference and diffraction. This is equivalent to assuming that all surface irregularities are either smaller than a light wavelength or much larger. In geometrical optics light is modeled as rays instead of waves. At the point a light ray intersects with a surface, the surface is treated locally as a flat plane. The diagram on the bottom right of Figure 9.9 can be seen as a geometrical optics picture of reflection and refraction, in contrast with the wave picture presented in the other parts of that figure. We will

Figure 9.11. The refractive index of these decorative beads is the same as water. Above the water, they have a visible surface due to the difference between their refractive index and that of air. Below the water, the refractive index is the same on both sides of the bead surfaces, so the surfaces are invisible. The beads themselves are visible only due to their colored absorption.

keep to the realm of geometrical optics from this point until Section 9.11, which is dedicated to the topic of shading models based on wave optics.

As we mentioned earlier, surface irregularities much larger than a wavelength change the local orientation of the surface. When these irregularities are too small to be individually rendered—in other words, smaller than a pixel—we refer to them as *microgeometry*. The directions of reflection and refraction depend on the surface normal. The effect of the microgeometry is to change that normal at different points on the surface, thus changing the reflected and refracted light directions.

Even though each specific point on the surface reflects light in only a single direction, each pixel covers many surface points that reflect light in various directions. The appearance is driven by the aggregate result of all the different reflection directions. Figure 9.12 shows an example of two surfaces that have similar shapes on the macroscopic scale but significantly different microgeometry.

For rendering, rather than modeling the microgeometry explicitly, we treat it statistically and view the surface as having a random distribution of microstructure normals. As a result, we model the surface as reflecting (and refracting) light in a continuous spread of directions. The width of this spread, and thus the blurriness of reflected and refracted detail, depends on the statistical variance of the microgeometry normal vectors—in other words, the surface microscale *roughness*. See Figure 9.13.

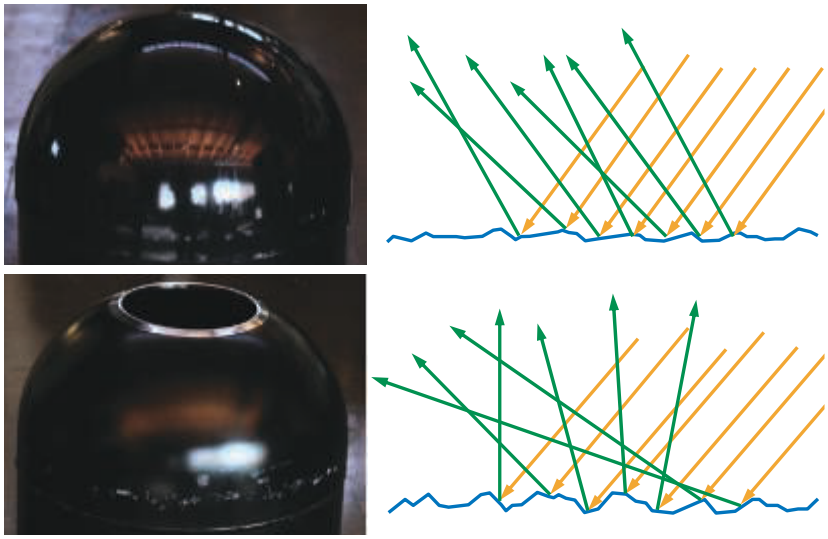

Figure 9.12. On the left we see photographs of two surfaces, with diagrams of their microscopic structures on the right. The top surface has slightly rough microgeometry. Incoming light rays hit surface points that are angled somewhat differently and reflect in a narrow cone of directions. The visible effect is a slight blurring of the reflection. The bottom surface has rougher microgeometry. Surface points hit by incoming light rays are angled in significantly different directions and the reflected light spreads out in a wide cone, causing blurrier reflections.

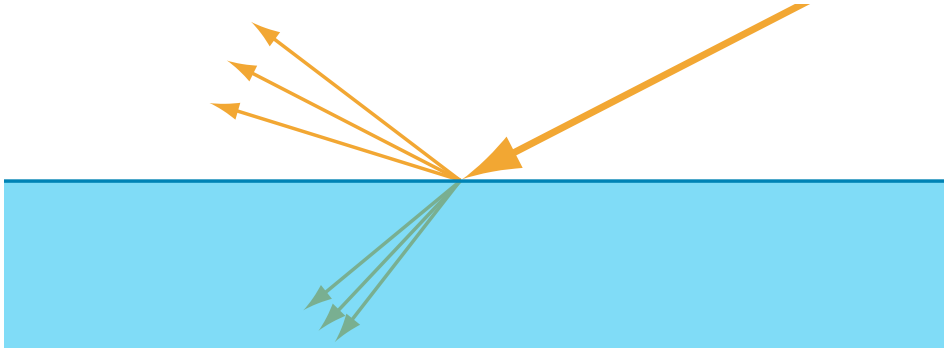

Figure 9.13. When viewed macroscopically, surfaces can be treated as reflecting and refracting light in multiple directions.

9.1.4 Subsurface Scattering

Refracted light continues to interact with the interior volume of the object. As mentioned earlier, metals reflect most incident light and quickly absorb the rest. In contrast, non-metals exhibit a wide variety of scattering and absorption behaviors, which

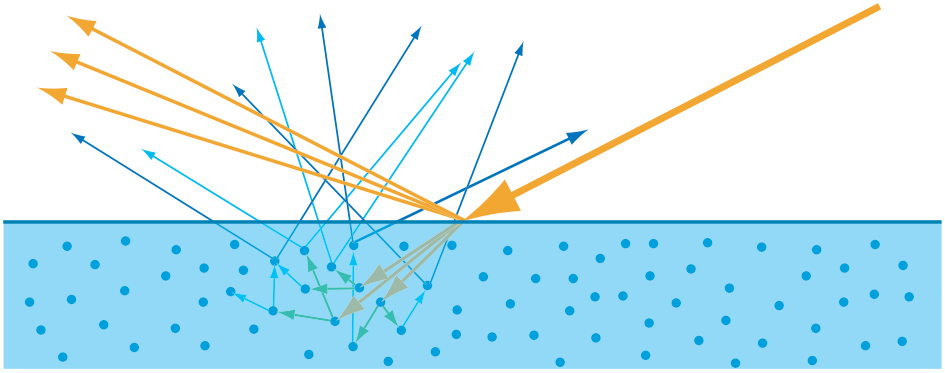

Figure 9.14. The refracted light undergoes absorption as it travels through the material. In this example most of the absorption is at longer wavelengths, leaving primarily short-wavelength blue light. In addition, it scatters from particles inside the material. Eventually some refracted light is scattered back out of the surface, as shown by the blue arrows exiting the surface in various directions.

are similar to those seen in the cups of liquid in Figure 9.8. Materials with low scattering and absorption are transparent, transmitting any refracted light through the entire object. Simple methods for rendering such materials without refraction were discussed in Section 5.5, and refraction will be covered in detail in Section 14.5.2. In this chapter we will focus on opaque objects, in which the transmitted light undergoes multiple scattering and absorption events until finally some of it is re-emitted back from the surface. See Figure 9.14.

This *subsurface-scattered* light exits the surface at varying distances from the entry point. The distribution of entry-exit distances depends on the density and properties of the scattering particles in the material. The relationship between these distances and the shading scale (the size of a pixel, or the distance between shading samples) is important. If the entry-exit distances are small compared to the shading scale, they can be assumed to be effectively zero for shading purposes. This allows subsurface scattering to be combined with surface reflection into a local shading model, with outgoing light at a point depending only on incoming light at the same point. However, since subsurface-scattered light has a significantly different appearance than surface-reflected light, it is convenient to divide them into separate shading terms. The *specular term* models surface reflection, and the *diffuse term* models *local subsurface scattering*.

If the entry-exit distances are large compared to the shading scale, then specialized rendering techniques are needed to capture the visual effect of light entering the surface at one point and leaving it from another. These *global subsurface scattering* techniques are covered in detail in Section 14.6. The difference between local and global subsurface scattering is illustrated in Figure 9.15.

It is important to note that local and global subsurface scattering techniques model exactly the same physical phenomena. The best choice for each situation depends not

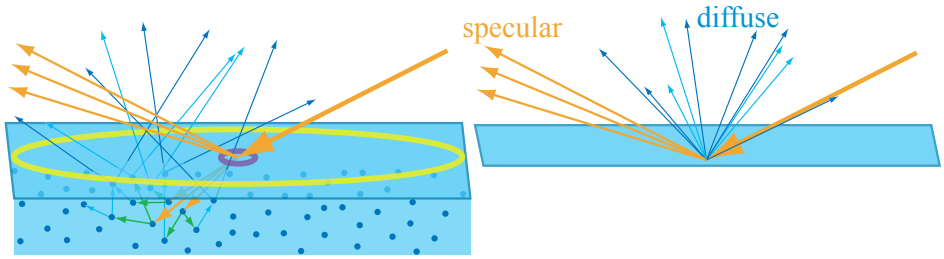

Figure 9.15. On the left, we are rendering a material with subsurface scattering. Two different sampling sizes are shown, in yellow and purple. The large yellow circle represents a single shading sample covering an area larger than the subsurface scattering distances. Thus, those distances can be ignored, enabling subsurface scattering to be treated as the diffuse term in a local shading model, as shown in the separate figure on the right. If we move closer to this surface, the shading sample area becomes smaller, as shown by the small purple circle. The subsurface scattering distances are now large compared to the area covered by a shading sample. Global techniques are needed to produce a realistic image from these samples.

only on the material properties but also on the scale of observation. For example, when rendering a scene of a child playing with a plastic toy, it is likely that global techniques would be needed for an accurate rendering of the child's skin, and that a local diffuse shading model would suffice for the toy. This is because the scattering distances in skin are quite a bit larger than in plastic. However, if the camera is far enough away, the skin scattering distances would be smaller than a pixel and local shading models would be accurate for both the child and the toy. Conversely, in an extreme close-up shot, the plastic would exhibit noticeable non-local subsurface scattering and global techniques would be needed to render the toy accurately.

9.2 The Camera

As mentioned in Section 8.1.1, in rendering we compute the radiance from the shaded surface point to the camera position. This simulates a simplified model of an imaging system such as a film camera, digital camera, or human eye.

Such systems contain a sensor surface composed of many discrete small sensors. Examples include rods and cones in the eye, photodiodes in a digital camera, or dye particles in film. Each of these sensors detects the irradiance value over its surface and produces a color signal. Irradiance sensors themselves cannot produce an image, since they average light rays from all incoming directions. For this reason, a full imaging system includes a light-proof enclosure with a single small *aperture* (opening) that restricts the directions from which light can enter and strike the sensors. A lens placed at the aperture focuses the light so that each sensor receives light from only a small set of incoming directions. The enclosure, aperture, and lens have the combined effect of causing the sensors to be *directionally specific*. They average light over a

small area and a small set of incoming directions. Rather than measuring average irradiance—which as we have seen in Section 8.1.1 quantifies the surface density of light flow from all directions—these sensors measure average radiance, which quantifies the brightness and color of a single ray of light.

Historically, rendering has simulated an especially simple imaging sensor called a *pinhole camera*, shown in the top part of Figure 9.16. A pinhole camera has an extremely small aperture—in the ideal case, a zero-size mathematical point—and no lens. The point aperture restricts each point on the sensor surface to collect a single ray of light, with a discrete sensor collecting a narrow cone of light rays with its base covering the sensor surface and its apex at the aperture. Rendering systems model pinhole cameras in a slightly different (but equivalent) way, shown in the middle part of Figure 9.16. The location of the pinhole aperture is represented by the point $\mathbf{c}$, often referred to as the "camera position" or "eye position." This point is also the center of projection for the perspective transform (Section 4.7.2).

When rendering, each shading sample corresponds to a single ray and thus to a sample point on the sensor surface. The process of antialiasing (Section 5.4) can be interpreted as reconstructing the signal collected over each discrete sensor surface. However, since rendering is not bound by the limitations of physical sensors, we can treat the process more generally, as the reconstruction of a continuous image signal from discrete samples.

Although actual pinhole cameras have been constructed, they are poor models for most cameras used in practice, as well as for the human eye. A model of an imaging system using a lens is shown in the bottom part of Figure 9.16. Including a lens allows for the use of a larger aperture, which greatly increases the amount of light collected by the imaging system. However, it also causes the camera to have a limited depth of field (Section 12.4), blurring objects that are too near or too far.

The lens has an additional effect aside from limiting the depth of field. Each sensor location receives a cone of light rays, even for points that are in perfect focus. The idealized model where each shading sample represents a single viewing ray can sometimes introduce mathematical singularities, numerical instabilities, or visual aliasing. Keeping the physical model in mind when we render images can help us identify and resolve such issues.

9.3 The BRDF

Ultimately, physically based rendering comes down to computing the radiance entering the camera along some set of view rays. Using the notation for incoming radiance introduced in Section 8.1.1, for a given view ray the quantity we need to compute is $L_i(\mathbf{c}, -\mathbf{v})$, where $\mathbf{c}$ is the camera position and $-\mathbf{v}$ is the direction along the view ray. We use $-\mathbf{v}$ due to two notation conventions. First, the direction vector in $L_i()$ always points away from the given point, which in this case is the camera location. Second, the view vector $\mathbf{v}$ always points toward the camera.

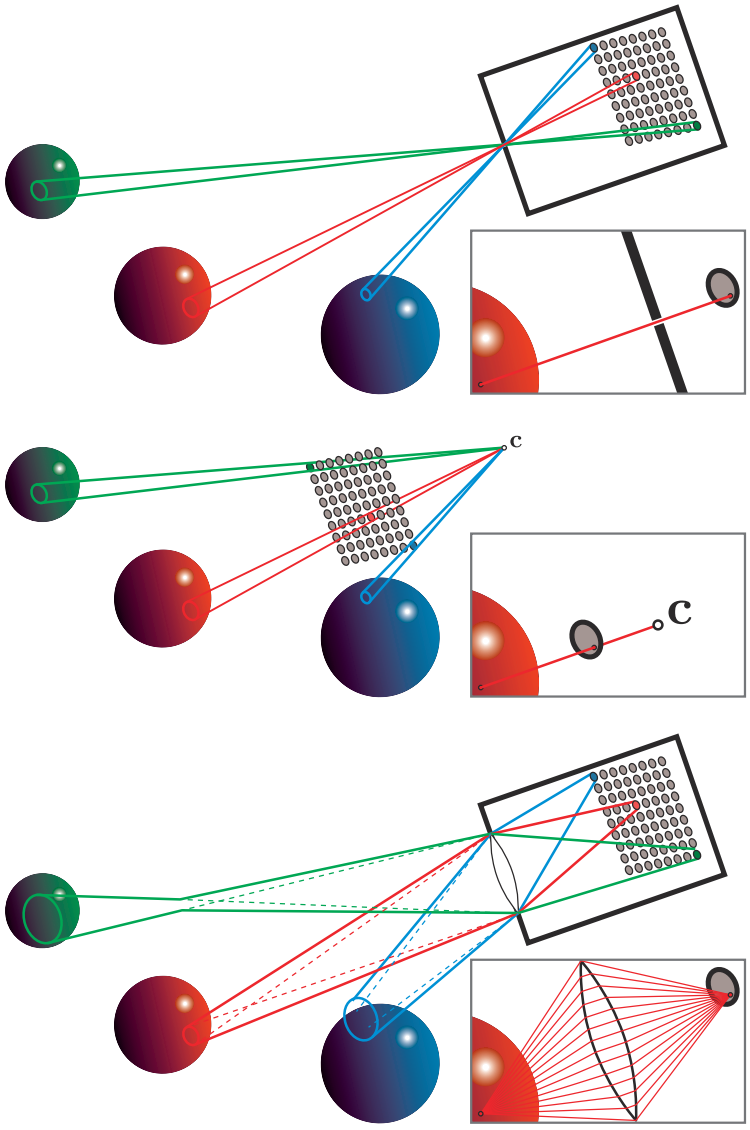

Figure 9.16. Each of these camera model figures contains an array of pixel sensors. The solid lines bound the set of light rays collected from the scene by three of these sensors. The inset images in each figure show the light rays collected by a single point sample on a pixel sensor. The top figure shows a pinhole camera, the middle figure shows a typical rendering system model of the same pinhole camera with the camera point **c**, and the bottom figure shows a more physically correct camera with a lens. The red sphere is in focus, and the other two spheres are out of focus.

In rendering, scenes are typically modeled as collections of objects with media in between them (the word "media" actually comes from the Latin word for "in the middle" or "in between"). Often the medium in question is a moderate amount of relatively clean air, which does not noticeably affect the ray's radiance and can thus be ignored for rendering purposes. Sometimes the ray may travel through a medium that does affect its radiance appreciably via absorption or scattering. Such media are called *participating media* since they participate in the light's transport through the scene. Participating media will be covered in detail in Chapter 14. In this chapter we assume that there are no participating media present, so the radiance entering the camera is equal to the radiance leaving the closest object surface in the direction of the camera:

$$L_i(\mathbf{c}, -\mathbf{v}) = L_o(\mathbf{p}, \mathbf{v}), \tag{9.2}$$

where $\mathbf{p}$ is the intersection of the view ray with the closest object surface.

Following Equation 9.2, our new goal is to calculate $L_o(\mathbf{p}, \mathbf{v})$. This calculation is a physically based version of the shading model evaluation discussed in Section 5.1. Sometimes radiance is directly emitted by the surface. More often, radiance leaving the surface originated from elsewhere and is reflected by the surface into the view ray, via the physical interactions described in Section 9.1. In this chapter we leave aside the cases of transparency (Section 5.5 and Section 14.5.2) and global subsurface scattering (Section 14.6). In other words, we focus on local reflectance phenomena, which redirect light hitting the currently shaded point back outward. These phenomena include surface reflection as well as local subsurface scattering, and depend on only the incoming light direction $\mathbf{l}$ and the outgoing view direction $\mathbf{v}$. Local reflectance is quantified by the *bidirectional reflectance distribution function* (BRDF), denoted as $f(\mathbf{l}, \mathbf{v})$.

In its original derivation [1277] the BRDF was defined for uniform surfaces. That is, the BRDF was assumed to be the same over the surface. However, objects in the real world (and in rendered scenes) rarely have uniform material properties over their surface. Even an object that is composed of a single material, e.g., a statue made of silver, will have scratches, tarnished spots, stains, and other variations that cause its visual properties to change from one surface point to the next. Technically, a function that captures BRDF variation based on spatial location is called a *spatially varying BRDF* (SVBRDF) or *spatial BRDF* (SBRDF). However, this case is so prevalent in practice that the shorter term BRDF is often used and implicitly assumed to depend on surface location.

The incoming and outgoing directions each have two degrees of freedom. A frequently used parameterization involves two angles: elevation θ relative to the surface normal $\mathbf{n}$ and azimuth (horizontal rotation) ϕ about $\mathbf{n}$. In the general case, the BRDF is a function of four scalar variables. *Isotropic* BRDFs are an important special case. Such BRDFs remain the same when the incoming and outgoing directions are rotated around the surface normal, keeping the same relative angles between them. Figure 9.17 shows the variables used in both cases. Isotropic BRDFs are functions of three scalar variables, since only a single angle ϕ between the light's and camera's

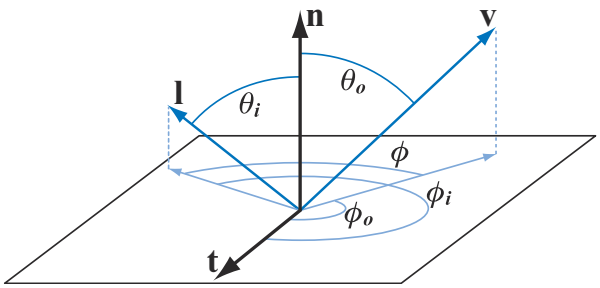

Figure 9.17. The BRDF. Azimuth angles ϕ_i and ϕ_o are given with respect to a given tangent vector $\mathbf{t}$. The relative azimuth angle ϕ, used for isotropic BRDFs instead of ϕ_i and ϕ_o, does not require a reference tangent vector.

rotation is needed. What this means is that if a uniform isotropic material is placed on a turntable and rotated, it will appear the same for all rotation angles, given a fixed light and camera.

Since we ignore phenomena such as fluorescence and phosphorescence, we can assume that incoming light of a given wavelength is reflected at the same wavelength. The amount of light reflected can vary based on the wavelength, which can be modeled in one of two ways. Either the wavelength is treated as an additional input variable to the BRDF, or the BRDF is treated as returning a spectrally distributed value. While the first approach is sometimes used in offline rendering [660], in real-time rendering the second approach is always used. Since real-time renderers represent spectral distributions as RGB triples, this simply means that the BRDF returns an RGB value.

To compute $L_o(\mathbf{p}, \mathbf{v})$, we incorporate the BRDF into the *reflectance equation*:

$$L_o(\mathbf{p}, \mathbf{v}) = \int_{\mathbf{l} \in \Omega} f(\mathbf{l}, \mathbf{v}) L_i(\mathbf{p}, \mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\mathbf{l}. \tag{9.3}$$

The $\mathbf{l} \in \Omega$ subscript on the integral sign means that integration is performed over $\mathbf{l}$ vectors that lie in the unit hemisphere above the surface (centered on the surface normal $\mathbf{n}$). Note that $\mathbf{l}$ is swept continuously over the hemisphere of incoming directions—it is not a specific "light source direction." The idea is that any incoming direction can (and usually will) have some radiance associated with it. We use $d\mathbf{l}$ to denote the differential solid angle around $\mathbf{l}$ (solid angles are discussed in Section 8.1.1).

In summary, the reflectance equation shows that outgoing radiance equals the integral (over $\mathbf{l}$ in Ω) of incoming radiance times the BRDF times the dot product between $\mathbf{n}$ and $\mathbf{l}$.

For brevity, for the rest of the chapter we will omit the surface point $\mathbf{p}$ from $L_i()$, $L_o()$, and the reflectance equation:

$$L_o(\mathbf{v}) = \int_{\mathbf{l} \in \Omega} f(\mathbf{l}, \mathbf{v}) L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\mathbf{l}. \tag{9.4}$$

When computing the reflectance equation, the hemisphere is often parameterized using spherical coordinates ϕ and θ. For this parameterization, the differential solid angle $d\mathbf{l}$ is equal to $\sin\theta_i \, d\theta_i \, d\phi_i$. Using this parameterization, a double-integral form of Equation 9.4 can be derived, which uses spherical coordinates (recall that $(\mathbf{n} \cdot \mathbf{l}) = \cos\theta_i$):

$$L_o(\theta_o, \phi_o) = \int_{\phi_i=0}^{2\pi} \int_{\theta_i=0}^{\pi/2} f(\theta_i, \phi_i, \theta_o, \phi_o) L(\theta_i, \phi_i) \cos\theta_i \sin\theta_i d\theta_i d\phi_i. \tag{9.5}$$

The angles θ_i, ϕ_i, θ_o, and ϕ_o are shown in Figure 9.17.

In some cases it is convenient to use a slightly different parameterization, with the cosines of the elevation angles $\mu_i = \cos\theta_i$ and $\mu_o = \cos\theta_o$ as variables rather than the angles θ_i and θ_o themselves. For this parameterization, the differential solid angle $d\mathbf{l}$ is equal to $d\mu_i \, d\phi_i$. Using the (μ, ϕ) parameterization yields the following integral form:

$$L_o(\mu_o, \phi_o) = \int_{\phi_i=0}^{2\pi} \int_{\mu_i=0}^{1} f(\mu_i, \phi_i, \mu_o, \phi_o) L(\mu_i, \phi_i) \mu_i d\mu_i d\phi_i. \tag{9.6}$$

The BRDF is defined only in cases where both the light and view directions are above the surface. The case where the light direction is under the surface can be avoided by either multiplying the BRDF by zero or not evaluating the BRDF for such directions in the first place. But what about view directions under the surface, in other words where the dot product $\mathbf{n} \cdot \mathbf{v}$ is negative? Theoretically this case should never occur. The surface would be facing away from the camera and would thus be invisible. However, interpolated vertex normals and normal mapping, both common in real-time applications, can create such situations in practice. Evaluation of the BRDF for view directions under the surface can be avoided by clamping $\mathbf{n} \cdot \mathbf{v}$ to 0 or using its absolute value, but both approaches can result in artifacts. The Frostbite engine uses the absolute value of $\mathbf{n} \cdot \mathbf{v}$ plus a small number (0.00001) to avoid divides by zero [960]. Another possible approach is a "soft clamp," which gradually goes to zero as the angle between $\mathbf{n}$ and $\mathbf{v}$ increases past 90°.

The laws of physics impose two constraints on any BRDF. The first constraint is *Helmholtz reciprocity*, which means that the input and output angles can be switched and the function value will be the same:

$$f(\mathbf{l}, \mathbf{v}) = f(\mathbf{v}, \mathbf{l}). \tag{9.7}$$

In practice, BRDFs used in rendering often violate Helmholtz reciprocity without noticeable artifacts, except for offline rendering algorithms that specifically require reciprocity, such as bidirectional path tracing. However, it is a useful tool to use when determining if a BRDF is physically plausible.

The second constraint is conservation of energy—the outgoing energy cannot be greater than the incoming energy (not counting glowing surfaces that emit light, which are handled as a special case). Offline rendering algorithms such as path tracing require

energy conservation to ensure convergence. For real-time rendering, exact energy conservation is not necessary, but approximate energy conservation is important. A surface rendered with a BRDF that significantly violates energy conservation would be too bright, and so may look unrealistic.

The *directional-hemispherical reflectance* $R(\mathbf{l})$ is a function related to the BRDF. It can be used to measure to what degree a BRDF is energy conserving. Despite its somewhat daunting name, the directional-hemispherical reflectance is a simple concept. It measures the amount of light coming from a given direction that is reflected at all, into any outgoing direction in the hemisphere around the surface normal. Essentially, it measures energy loss for a given incoming direction. The input to this function is the incoming direction vector $\mathbf{l}$, and its definition is presented here:

$$R(\mathbf{l}) = \int_{\mathbf{v}\in\Omega} f(\mathbf{l},\mathbf{v})(\mathbf{n}\cdot\mathbf{v})d\mathbf{v}. \tag{9.8}$$

Note that here $\mathbf{v}$, like $\mathbf{l}$ in the reflectance equation, is swept over the entire hemisphere and does not represent a singular viewing direction.

A similar but in some sense opposite function, *hemispherical-directional reflectance* $R(\mathbf{v})$ can be similarly defined:

$$R(\mathbf{v}) = \int_{\mathbf{l}\in\Omega} f(\mathbf{l},\mathbf{v})(\mathbf{n}\cdot\mathbf{l})d\mathbf{l}. \tag{9.9}$$

If the BRDF is reciprocal, then the hemispherical-directional reflectance and the directional-hemispherical reflectance are equal and the same function can be used to compute either one. *Directional albedo* can be used as a blanket term for both reflectances in cases where they are used interchangeably.

The value of the directional-hemispherical reflectance $R(\mathbf{l})$ must always be in the range $[0, 1]$, as a result of energy conservation. A reflectance value of 0 represents a case where all the incoming light is absorbed or otherwise lost. If all the light is reflected, the reflectance will be 1. In most cases it will be somewhere between these two values. Like the BRDF, the values of $R(\mathbf{l})$ vary with wavelength, so it is represented as an RGB vector for rendering purposes. Since each component (red, green, and blue) is restricted to the range $[0, 1]$, a value of $R(\mathbf{l})$ can be thought of as a simple color. Note that this restriction does not apply to the values of the BRDF. As a distribution function, the BRDF can have arbitrarily high values in certain directions (such as the center of a highlight) if the distribution it describes is highly nonuniform. The requirement for a BRDF to be energy conserving is that $R(\mathbf{l})$ be no greater than one for all possible values of $\mathbf{l}$.

The simplest possible BRDF is Lambertian, which corresponds to the Lambertian shading model briefly discussed in Section 5.2. The Lambertian BRDF has a constant value. The well-known $(\mathbf{n}\cdot\mathbf{l})$ factor that distinguishes Lambertian shading is not part of the BRDF but rather part of Equation 9.4. Despite its simplicity, the Lambertian BRDF is often used in real-time rendering to represent local subsurface scattering

(though it is being supplanted by more accurate models, as discussed in Section 9.9). The directional-hemispherical reflectance of a Lambertian surface is also a constant. Evaluating Equation 9.8 for a constant value of $f(\mathbf{l}, \mathbf{v})$ yields the following value for the directional-hemispherical reflectance as a function of the BRDF:

$$R(\mathbf{l}) = \pi f(\mathbf{l}, \mathbf{v}). \tag{9.10}$$

The constant reflectance value of a Lambertian BRDF is typically referred to as the *diffuse color* $\mathbf{c}_{\text{diff}}$ or the *albedo* ρ. In this chapter, to emphasize the connection with subsurface scattering, we will refer to this quantity as the *subsurface albedo* ρ_{ss}. The subsurface albedo is discussed in detail in Section 9.9.1. The BRDF from Equation 9.10 gives the following result:

$$f(\mathbf{l}, \mathbf{v}) = \frac{\rho_{\text{ss}}}{\pi}. \tag{9.11}$$

The $1/\pi$ factor is caused by the fact that integrating a cosine factor over the hemisphere yields a value of π. Such factors are often seen in BRDFs.

One way to understand a BRDF is to visualize it with the input direction held constant. See Figure 9.18. For a given direction of incoming light, the BRDF's values

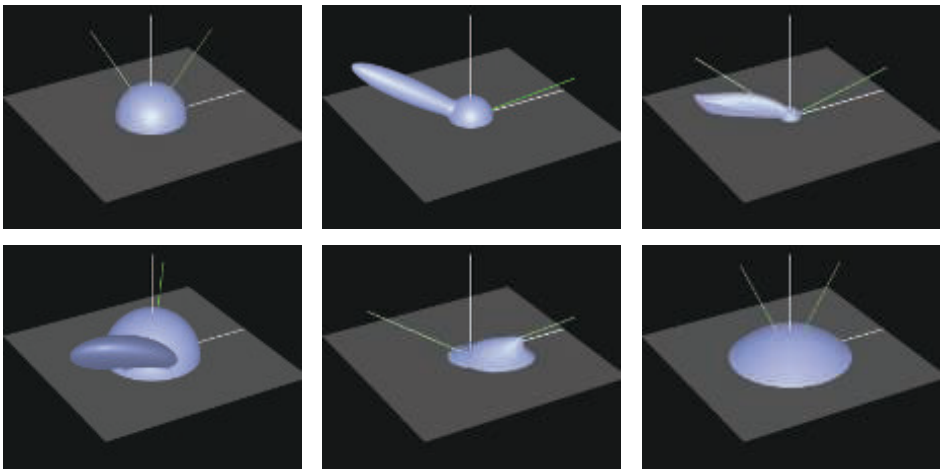

Figure 9.18. Example BRDFs. The solid green line coming from the right of each figure is the incoming light direction, and the dashed green and white line is the ideal reflection direction. In the top row, the left figure shows a Lambertian BRDF (a simple hemisphere). The middle figure shows Blinn-Phong highlighting added to the Lambertian term. The right figure shows the Cook-Torrance BRDF [285, 1779]. Note how the specular highlight is not strongest in the reflection direction. In the bottom row, the left figure shows a close-up of Ward's anisotropic model. In this case, the effect is to tilt the specular lobe. The middle figure shows the Hapke/Lommel-Seeliger "lunar surface" BRDF [664], which has strong retroreflection. The right figure shows Lommel-Seeliger scattering, in which dusty surfaces scatter light toward grazing angles. *(Images courtesy of Szymon Rusinkiewicz, from his "bv" BRDF browser.)*

are displayed for all outgoing directions. The spherical part around the point of intersection is the diffuse component, since outgoing radiance has an equal chance of reflecting in any direction. The ellipsoidal piece is the *specular lobe*. Naturally, such lobes are in the reflection direction from the incoming light, with the thickness of the lobe corresponding to the fuzziness of the reflection. By the principle of reciprocity, these same visualizations can also be thought of as how much each different incoming light direction contributes to a single outgoing direction.

9.4 Illumination

The $L_i(\mathbf{l})$ (incoming radiance) term in the reflectance equation (Equation 9.4) represents light impinging upon the shaded surface point from other parts of the scene. *Global illumination* algorithms calculate $L_i(\mathbf{l})$ by simulating how light propagates and is reflected throughout the scene. These algorithms use the *rendering equation* [846], of which the reflectance equation is a special case. Global illumination is discussed in Chapter 11. In this chapter and the next, we focus on *local illumination*, which uses the reflectance equation to compute shading locally at each surface point. In local illumination algorithms $L_i(\mathbf{l})$ is given and does not need to be computed.

In realistic scenes, $L_i(\mathbf{l})$ includes nonzero radiance from all directions, whether emitted directly from light sources or reflected from other surfaces. Unlike the directional and punctual lights discussed in Section 5.2, real-world light sources are *area lights* that cover a nonzero solid angle. In this chapter, we use a restricted form of $L_i(\mathbf{l})$ comprised of only directional and punctual lights, leaving more general lighting environments to Chapter 10. This restriction allows for a more focused discussion.

Although punctual and directional lights are non-physical abstractions, they can be derived as approximations of physical light sources. Such a derivation is important, because it enables us to incorporate these lights in a physically based rendering framework with confidence that we understand the error involved.

We take a small, distant area light and define $\mathbf{l}_c$ as the vector pointing to its center. We also define the light's color $\mathbf{c}_{\text{light}}$ as the reflected radiance from a white Lambertian surface facing toward the light ($\mathbf{n} = \mathbf{l}_c$). This is an intuitive definition for authoring, since the color of the light corresponds directly to its visual effect.

With these definitions, a directional light can be derived as the limit case of shrinking the size of the area light down to zero while maintaining the value of $\mathbf{c}_{\text{light}}$ [758]. In this case the integral in the reflectance equation (Equation 9.4) simplifies down to a single BRDF evaluation, which is significantly less expensive to compute:

$$L_o(\mathbf{v}) = \pi f(\mathbf{l}_c, \mathbf{v})\mathbf{c}_{\text{light}}(\mathbf{n} \cdot \mathbf{l}_c). \tag{9.12}$$

The dot product $(\mathbf{n} \cdot \mathbf{l})$ is often clamped to zero, as a convenient method of skipping contributions from lights under the surface:

$$L_o(\mathbf{v}) = \pi f(\mathbf{l}_c, \mathbf{v})\mathbf{c}_{\text{light}}(\mathbf{n} \cdot \mathbf{l}_c)^+. \tag{9.13}$$

Note the x^+ notation introduced in Section 1.2, which indicates that negative values are clamped to zero.

Punctual lights can be treated similarly. The only differences are that the area light is not required to be distant, and $\mathbf{c}_{\text{light}}$ falls off as the inverse square of the distance to the light, as in Equation 5.11 (page 111). In the case of more than one light source, Equation 9.12 is computed multiple times and the results are summed:

$$L_o(\mathbf{v}) = \pi \sum_{i=1}^{n} f(\mathbf{l}_{c_i}, \mathbf{v}) \mathbf{c}_{\text{light}_i} (\mathbf{n} \cdot \mathbf{l}_{c_i})^+, \tag{9.14}$$

where $\mathbf{l}_{c_i}$ and $\mathbf{c}_{\text{light}_i}$ are the direction and color, respectively, of the ith light. Note the similarities to Equation 5.6 (page 109).

The π factor in Equation 9.14 cancels out the $1/\pi$ factor that often appears in BRDFs (e.g., Equation 9.11). This cancellation moves the divide operation out of the shader and makes the shading equation simpler to read. However, care must be taken when adapting BRDFs from academic papers for use in real-time shading equations. Typically, the BRDF will need to be multiplied by π before use.

9.5 Fresnel Reflectance

In Section 9.1 we discussed light-matter interaction from a high level. In Section 9.3, we covered the basic machinery for expressing these interactions mathematically: the BRDF and the reflectance equation. Now we are ready to start drilling down to specific phenomena, quantifying them so they can be used in shading models. We will start with reflection from a flat surface, first discussed in Section 9.1.3.

An object's surface is an interface between the surrounding medium (typically air) and the object's substance. The interaction of light with a planar interface between two substances follows the *Fresnel equations* developed by Augustin-Jean Fresnel (1788–1827) (pronounced freh-*nel*). The Fresnel equations require a flat interface following the assumptions of geometrical optics. In other words, the surface is assumed to not have any irregularities between 1 light wavelength and 100 wavelengths in size. Irregularities smaller than this range have no effect on the light, and larger irregularities effectively tilt the surface but do not affect its local flatness.

Light incident on a flat surface splits into a reflected part and a refracted part. The direction of the reflected light (indicated by the vector $\mathbf{r}_i$) forms the same angle (θ_i) with the surface normal $\mathbf{n}$ as the incoming direction $\mathbf{l}$. The reflection vector $\mathbf{r}_i$ can be computed from $\mathbf{n}$ and $\mathbf{l}$:

$$\mathbf{r}_i = 2(\mathbf{n} \cdot \mathbf{l})\mathbf{n} - \mathbf{l}. \tag{9.15}$$

See Figure 9.19. The amount of light reflected (as a fraction of incoming light) is described by the *Fresnel reflectance* F, which depends on the incoming angle θ_i.

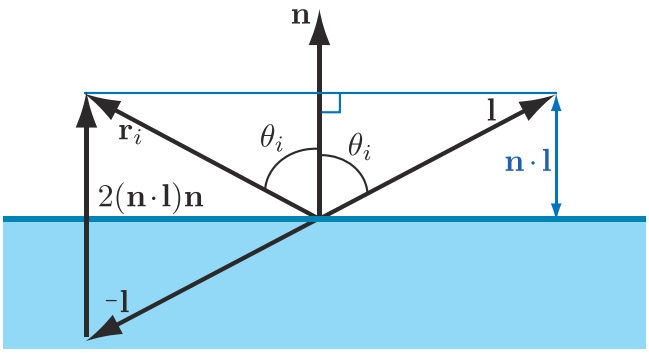

Figure 9.19. Reflection at a planar surface. The light vector $\mathbf{l}$ is reflected around the normal $\mathbf{n}$ in order to generate $\mathbf{r}_i$. First, $\mathbf{l}$ is projected onto $\mathbf{n}$, and we get a scaled version of the normal: $(\mathbf{n}\cdot\mathbf{l})\mathbf{n}$. We then negate $\mathbf{l}$, and add two times the projected vector to obtain the reflection vector.

As discussed in Section 9.1.3, reflection and refraction are affected by the refractive index of the two substances on either side of the plane. We will continue to use the notation from that discussion. The value n_1 is the refractive index of the substance "above" the interface, where incident and reflected light propagate, and n_2 is the refractive index of the substance "below" the interface, where the refracted light propagates.

The Fresnel equations describe the dependence of F on θ_i, n_1, and n_2. Rather than present the equations themselves, which are somewhat complex, we will describe their important characteristics.

9.5.1 External Reflection

External reflection is the case where $n_1 < n_2$. In other words, the light originates on the side of the surface where the refractive index is lower. Most often, this side contains air, with a refractive index of approximately 1.003. We will assume $n_1 = 1$ for simplicity. The opposite transition, from object to air, is called *internal reflection* and is discussed later in Section 9.5.3.

For a given substance, the Fresnel equations can be interpreted as defining a reflectance function $F(\theta_i)$, dependent only on incoming light angle. In principle, the value of $F(\theta_i)$ varies continuously over the visible spectrum. For rendering purposes its value is treated as an RGB vector. The function $F(\theta_i)$ has the following characteristics:

- When $\theta_i = 0°$, with the light perpendicular to the surface ($\mathbf{l} = \mathbf{n}$), $F(\theta_i)$ has a value that is a property of the substance. This value, F_0, can be thought of as the characteristic specular color of the substance. The case when $\theta_i = 0°$ is called *normal incidence*.

- As θ_i increases and the light strikes the surface at increasingly glancing angles, the value of $F(\theta_i)$ will tend to increase, reaching a value of 1 for all frequencies (white) at $\theta_i = 90°$.

Figure 9.20 shows the $F(\theta_i)$ function, visualized in several different ways, for several substances. The curves are highly nonlinear—they barely change until $\theta_i = 75°$ or so, and then quickly go to 1. The increase from F_0 to 1 is mostly monotonic, though some substances (e.g., aluminum in Figure 9.20) have a slight dip just before going to white.

In the case of mirror reflection, the outgoing or view angle is the same as the incidence angle. This means that surfaces that are at a glancing angle to the incoming

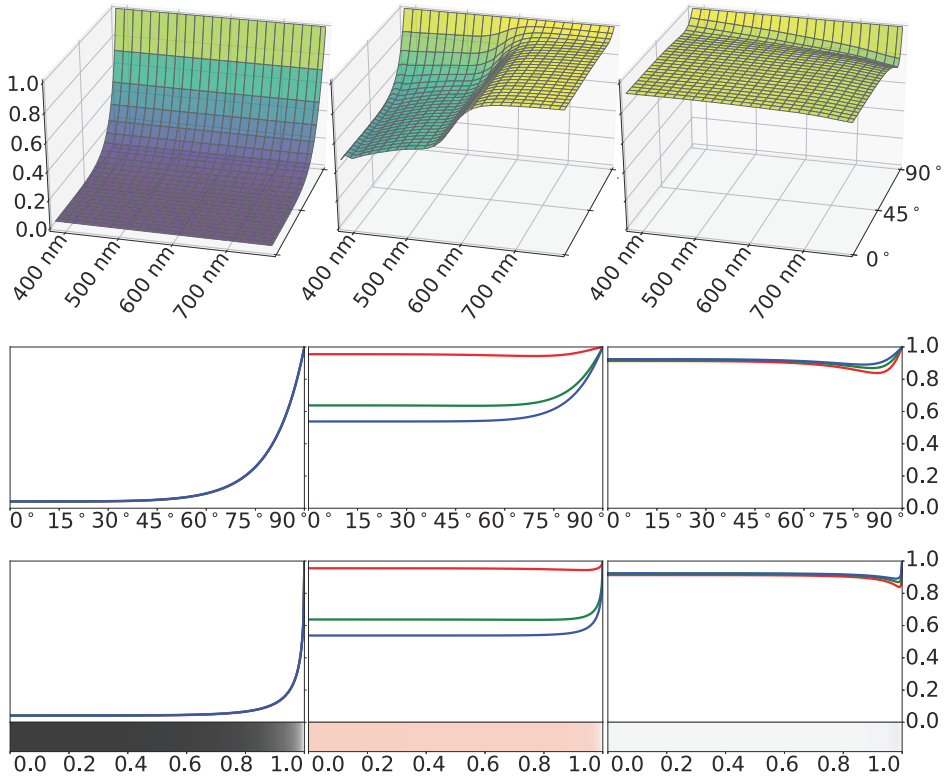

Figure 9.20. Fresnel reflectance F for external reflection from three substances: glass, copper, and aluminum (from left to right). The top row has three-dimensional plots of F as a function of wavelength and incidence angle. The second row shows the spectral value of F for each incidence angle converted to RGB and plotted as separate curves for each color channel. The curves for glass coincide, as its Fresnel reflectance is colorless. In the third row, the R, G, and B curves are plotted against the sine of the incidence angle, to account for the foreshortening shown in Figure 9.21. The same x-axis is used for the strips in the bottom row, which show the RGB values as colors.

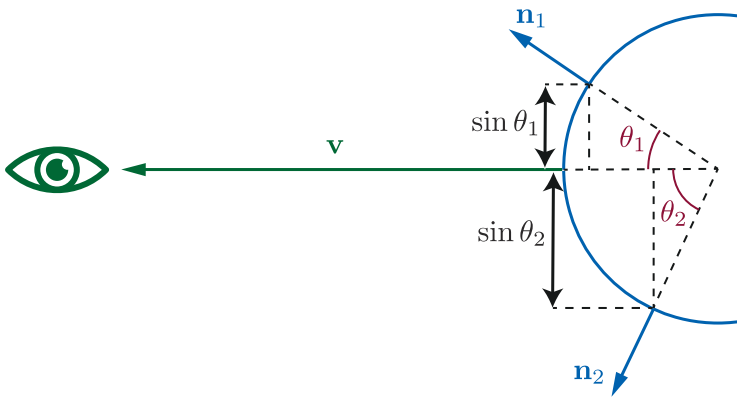

Figure 9.21. Surfaces tilted away from the eye are foreshortened. This foreshortening is consistent with projecting surface points according to the sine of the angle between $\mathbf{v}$ and $\mathbf{n}$ (for mirror reflection this is the same as the incidence angle). For this reason, Fresnel reflectance is plotted against the sine of the incidence angle in Figures 9.20 and 9.22.

light—with values of θ_i close to 90°—are also at a glancing angle to the eye. For this reason, the increase in reflectance is primarily seen at the edges of objects. Furthermore, the parts of the surface that have the strongest increase in reflectance are foreshortened from the camera's perspective, so they occupy a relatively small number of pixels. To show the different parts of the Fresnel curve proportionally to their visual prominence, the Fresnel reflectance graphs and color bars in Figure 9.22 and the lower half of Figure 9.20 are plotted against $\sin(\theta_i)$ instead of directly against θ_i. Figure 9.21 illustrates why $\sin(\theta_i)$ is an appropriate choice of axis for this purpose.

From this point, we will typically use the notation $F(\mathbf{n}, \mathbf{l})$ instead of $F(\theta_i)$ for the Fresnel function, to emphasize the vectors involved. Recall that θ_i is the angle between the vectors $\mathbf{n}$ and $\mathbf{l}$. When the Fresnel function is incorporated as part of a BRDF, a different vector is often substituted for the surface normal $\mathbf{n}$. See Section 9.8 for details.

The increase in reflectance at glancing angles is often called the *Fresnel effect* in rendering publications (in other fields, the term has a different meaning relating to transmission of radio waves). You can see the Fresnel effect for yourself with a short experiment. Take a smartphone and sit in front of a bright area, such as a computer monitor. Without turning it on, first hold the phone close to your chest, look down at it, and angle it slightly so that its screen reflects the monitor. There should be a relatively weak reflection of the monitor on the phone screen. This is because the normal-incidence reflectance of glass is quite low. Now raise the smartphone up so that it is roughly between your eyes and the monitor, and again angle its screen to

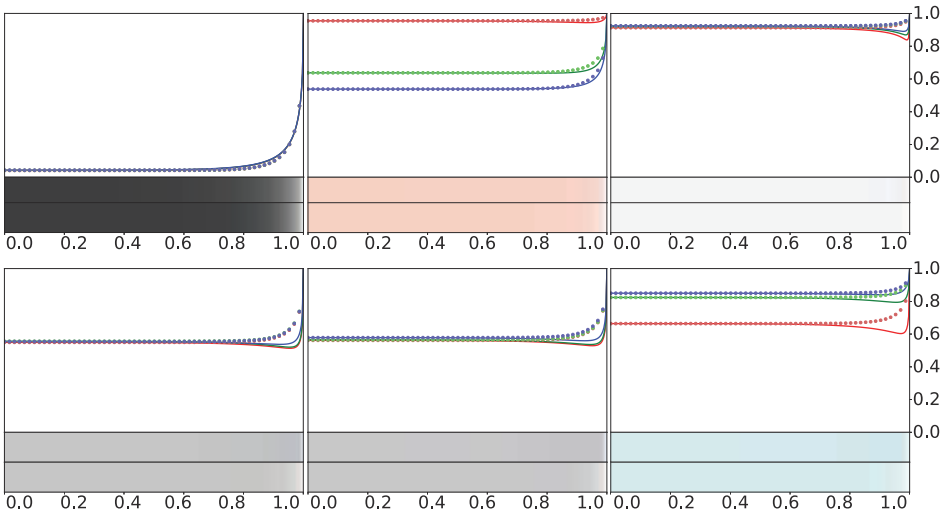

Figure 9.22. Schlick's approximation to Fresnel reflectance compared to the correct values for external reflection from six substances. The top three substances are the same as in Figure 9.20: glass, copper, and aluminum (from left to right). The bottom three substances are chromium, iron, and zinc. Each substance has an RGB curve plot with solid lines showing the full Fresnel equations and dotted lines showing Schlick's approximation. The upper color bar under each curve plot shows the result of the full Fresnel equations, and the lower color bar shows the result of Schlick's approximation.

reflect the monitor. Now the reflection of the monitor on the phone screen should be almost as bright as the monitor itself.

Besides their complexity, the Fresnel equations have other properties that makes their direct use in rendering difficult. They require refractive index values sampled over the visible spectrum, and these values may be complex numbers. The curves in Figure 9.20 suggest a simpler approach based on the characteristic specular color F_0. Schlick [1568] gives an approximation of Fresnel reflectance:

$$F(\mathbf{n}, \mathbf{l}) \approx F_0 + (1 - F_0)(1 - (\mathbf{n} \cdot \mathbf{l})^+)^5. \tag{9.16}$$

This function is an RGB interpolation between white and F_0. Despite this simplicity, the approximation is reasonably accurate.

Figure 9.22 contains several substances that diverge from the Schlick curves, exhibiting noticeable "dips" just before going to white. In fact, the substances in the bottom row were chosen because they diverge from the Schlick approximation to an especially large extent. Even for these substances the resulting errors are quite subtle, as shown by the color bars at the bottom of each plot in the figure. In the rare cases where it is important to precisely capture the behavior of such materials, an alternative approximation given by Gulbrandsen [623] can be used. This approximation can achieve a close match to the full Fresnel equations for metals, though it is more

computationally expensive than Schlick's. A simpler option is to modify Schlick's approximation to allow for raising the final term to powers other than 5 (as in Equation 9.18). This would change the "sharpness" of the transition to white at 90°, which could result in a closer match. Lagarde [959] summarizes the Fresnel equations and several approximations to them.

When using the Schlick approximation, F_0 is the only parameter that controls Fresnel reflectance. This is convenient since F_0 has a well-defined range of valid values in $[0, 1]$, is easy to set with standard color-picking interfaces, and can be textured using texture formats designed for colors. In addition, reference values for F_0 are available for many real-world materials. The refractive index can also be used to compute F_0. It is typical to assume that $n_1 = 1$, a close approximation for the refractive index of air, and use n instead of n_2 to represent the refractive index of the object. This simplification gives the following equation:

$$F_0 = \left(\frac{n-1}{n+1}\right)^2. \tag{9.17}$$

This equation works even with complex-valued refractive indices (such as those of metals) if the magnitude of the (complex) result is used. In cases where the refractive index varies significantly over the visible spectrum, computing an accurate RGB value for F_0 requires first computing F_0 at a dense sampling of wavelengths, and then converting the resulting spectral vector into RGB values using the methods described in Section 8.1.3.

In some applications [732, 947] a more general form of the Schlick approximation is used:

$$F(\mathbf{n}, \mathbf{l}) \approx F_0 + (F_{90} - F_0)(1 - (\mathbf{n} \cdot \mathbf{l})^+)^{\frac{1}{p}}. \tag{9.18}$$

This provides control over the color to which the Fresnel curve transitions at 90°, as well as the "sharpness" of the transition. The use of this more general form is typically motivated by a desire for increased artistic control, but it can also help match physical reality in some cases. As discussed above, modifying the power can lead to closer fits for certain materials. Also, setting F_{90} to a color other than white can help match materials that are not described well by the Fresnel equations, such as surfaces covered in fine dust with grains the size of individual light wavelengths.

9.5.2 Typical Fresnel Reflectance Values

Substances are divided into three main groups with respect to their optical properties. There are *dielectrics*, which are insulators; metals, which are conductors; and semiconductors, which have properties somewhere in between dielectrics and metals.

Fresnel Reflectance Values for Dielectrics

Most materials encountered in daily life are dielectrics—glass, skin, wood, hair, leather, plastic, stone, and concrete, to name a few. Water is also a dielectric. This last may be

Dielectric	Linear	Texture	Color	Notes
Water	0.02	39		
Living tissue	0.02–0.04	39–56		Watery tissues are toward the lower bound, dry ones are higher
Skin	0.028	47		
Eyes	0.025	44		Dry cornea (tears have a similar value to water)
Hair	0.046	61		
Teeth	0.058	68		
Fabric	0.04–0.056	56–67		Polyester highest, most others under 0.05
Stone	0.035–0.056	53–67		Values for the minerals most often found in stone
Plastics, glass	0.04–0.05	56–63		Not including crystal glass
Crystal glass	0.05–0.07	63–75		
Gems	0.05–0.08	63–80		Not including diamonds and diamond simulants
Diamond-like	0.13–0.2	101–124		Diamonds and diamond simulants (e.g., cubic zirconia, moissanite)

Table 9.1. Values of F_0 for external reflection from various dielectrics. Each value is given as a linear number, as a texture value (nonlinearly encoded 8-bit unsigned integer), and as a color swatch. If a range of values is given, then the color swatch is in the middle of the range. Recall that these are specular colors. For example, gems often have vivid colors, but those result from absorption inside the substance and are unrelated to their Fresnel reflectance.

surprising, since in daily life water is known to conduct electricity, but this conductivity is due to various impurities. Dielectrics have fairly low values for F_0—usually 0.06 or lower. This low reflectance at normal incidence makes the Fresnel effect especially visible for dielectrics. The optical properties of dielectrics rarely vary much over the visible spectrum, resulting in colorless reflectance values. The F_0 values for several common dielectrics are shown in Table 9.1. The values are scalar rather than RGB since the RGB channels do not differ significantly for these materials. For convenience, Table 9.1 includes linear values as well as 8-bit values encoded with the sRGB transfer function (the form that would typically be used in a texture-painting application).

The F_0 values for other dielectrics can be inferred by looking at similar substances in the table. For unknown dielectrics, 0.04 is a reasonable default value, not too far off from most common materials.

Metal	Linear	Texture	Color
Titanium	0.542,0.497,0.449	194,187,179	
Chromium	0.549,0.556,0.554	196,197,196	
Iron	0.562,0.565,0.578	198,198,200	
Nickel	0.660,0.609,0.526	212,205,192	
Platinum	0.673,0.637,0.585	214,209,201	
Copper	0.955,0.638,0.538	250,209,194	
Palladium	0.733,0.697,0.652	222,217,211	
Mercury	0.781,0.780,0.778	229,228,228	
Brass (C260)	0.910,0.778,0.423	245,228,174	
Zinc	0.664,0.824,0.850	213,234,237	
Gold	1.000,0.782,0.344	255,229,158	
Aluminum	0.913,0.922,0.924	245,246,246	
Silver	0.972,0.960,0.915	252,250,245	

Table 9.2. Values of F_0 for external reflection from various metals (and one alloy), sorted in order of increasing lightness. The actual red value for gold is slightly outside the sRGB gamut. The value shown is after clamping.

Once the light is transmitted into the dielectric, it may be further scattered or absorbed. Models for this process are discussed in more detail in Section 9.9. If the material is transparent, the light will continue until it hits an object surface "from the inside," which is detailed in Section 9.5.3.

Fresnel Reflectance Values for Metals

Metals have high values of F_0—almost always 0.5 or above. Some metals have optical properties that vary over the visible spectrum, resulting in colored reflectance values. The F_0 values for several metals are shown in Table 9.2.

Similarly to Table 9.1, Table 9.2 has linear values as well as 8-bit sRGB-encoded values for texturing. However, here we give RGB values since many metals have colored Fresnel reflectance. These RGB values are defined using the sRGB (and Rec. 709) primaries and white point. Gold has a somewhat unusual F_0 value. It is the most strongly colored, with a red channel value slightly above 1 (it is just barely outside the sRGB/Rec. 709 gamut) and an especially low blue channel value (the only value in Table 9.2 significantly below 0.5). It is also one of the brightest metals, as can be seen by its position in the table, which is sorted in order of increasing lightness. Gold's bright and strongly colored reflectance probably contributes to its unique cultural and economic significance throughout history.

Recall that metals immediately absorb any transmitted light, so they do not exhibit any subsurface scattering or transparency. All the visible color of a metal comes from F_0.

Substance	Linear	Texture	Color
Diamond	0.171,0.172,0.176	115,115,116	
Silicon	0.345,0.369,0.426	159,164,174	
Titanium	0.542,0.497,0.449	194,187,179	

Table 9.3. The value of F_0 for a representative semiconductor (silicon in crystalline form) compared to a bright dielectric (diamond) and a dark metal (titanium).

Fresnel Reflectance Values for Semiconductors

As one would expect, semiconductors have F_0 values in between the brightest dielectrics and the darkest metals, as shown in Table 9.3. It is rare to need to render such substances in practice, since most rendered scenes are not littered with blocks of crystalline silicon. For practical purposes the range of F_0 values between 0.2 and 0.45 should be avoided unless you are purposely trying to model an exotic or unrealistic material.

Fresnel Reflectance Values in Water

In our discussion of external reflectance, we have assumed that the rendered surface is surrounded by air. If not, the reflectance will change, since it depends on the ratio between the refractive indices on both sides of the interface. If we can no longer assume that $n_1 = 1$, then we need to replace n in Equation 9.17 with the relative index of refraction, n_1/n_2. This yields the following, more general equation:

$$F_0 = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2. \tag{9.19}$$

Likely the most frequently encountered case where $n_1 \neq 1$ is when rendering underwater scenes. Since water's refractive index is about 1.33 times higher than that of air, values of F_0 are different underwater. This effect is stronger for dielectrics than for metals, as can be seen in Table 9.4.

Parameterizing Fresnel Values

An often-used parameterization combines the specular color F_0 and the diffuse color ρ_{ss} (the diffuse color will be discussed further in Section 9.9). This parameterization takes advantage of the observation that metals have no diffuse color and that dielectrics have a restricted set of possible values for F_0, and it includes an RGB surface color $\mathbf{c}_{surf}$ and a scalar parameter m, called "metallic" or "metalness." If $m = 1$, then F_0 is set to $\mathbf{c}_{surf}$ and ρ_{ss} is set to black. If $m = 0$, then F_0 is set to a dielectric value (either constant or controlled by an additional parameter) and ρ_{ss} is set to $\mathbf{c}_{surf}$.

The "metalness" parameter first appeared as part of an early shading model used at Brown University [1713], and the parameterization in its current form was first used by Pixar for the film *Wall-E* [1669]. For the *Disney principled* shading model, used in Disney animation films from *Wreck-It Ralph* onward, Burley added an additional

Substance	Linear	Texture	Color
Skin (in air)	0.028	47	
Skin (in water)	0.0007	2	
Schott K7 glass (in air)	0.042	58	
Schott K7 glass (in water)	0.004	13	
Diamond (in air)	0.172	115	
Diamond (in water)	0.084	82	
Iron (in air)	0.562,0.565,0.578	198,198,200	
Iron (in water)	0.470,0.475,0.492	182,183,186	
Gold (in air)	1.000,0.782,0.344	255,229,158	
Gold (in water)	1.000,0.747,0.261	255,224,140	
Silver (in air)	0.972,0.960,0.915	252,250,245	
Silver (in water)	0.964,0.950,0.899	251,249,243	

Table 9.4. A comparison between values of F_0 in air and in water, for various substances. As one would expect from inspecting Equation 9.19, dielectrics with refractive indices close to that of water are affected the most. In contrast, metals are barely affected.

scalar "specular" parameter to control dielectric F_0 within a limited range [214]. This form of the parameterization is used in the Unreal Engine [861], and the Frostbite engine uses a slightly different form with a larger range of possible F_0 values for dielectrics [960]. The game *Call of Duty: Infinite Warfare* uses a variant that packs these metalness and specular parameters into a single value [384], to save memory.

For those rendering applications that use this metalness parameterization instead of using F_0 and ρ_{ss} directly, the motivations include user convenience and saving texture or G-buffer storage. In the game *Call of Duty: Infinite Warfare*, this parameterization is used in an unusual way. Artists paint textures for F_0 and ρ_{ss}, which are automatically converted to the metalness parameterization as a compression method.

Using metalness has some drawbacks. It cannot express some types of materials, such as coated dielectrics with tinted F_0 values. Artifacts can occur on the boundary between a metal and dielectric [960, 1163].

Another parameterization trick used by some real-time applications takes advantage of the fact that no materials have values of F_0 lower than 0.02, outside of special anti-reflective coatings. The trick is used to suppress specular highlights in surface areas that represent cavities or voids. Instead of using a separate specular occlusion texture, values of F_0 below 0.02 are used to "turn off" Fresnel edge brightening. This technique was first proposed by Schüler [1586] and is used in the Unreal [861] and Frostbite [960] engines.

9.5.3 Internal Reflection

Although external reflection is more frequently encountered in rendering, internal reflection is sometimes important as well. Internal reflection happens when $n_1 > n_2$. In

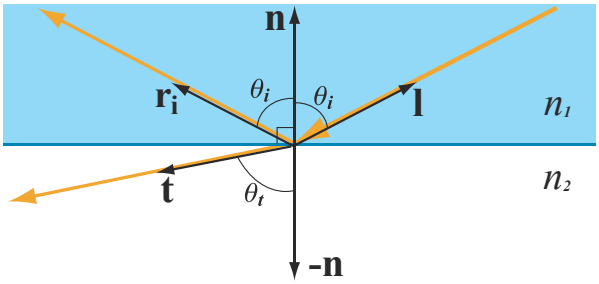

Figure 9.23. Internal reflection at a planar surface, where $n_1 > n_2$.

other words, internal reflection occurs when light is traveling in the interior of a transparent object and encounters the object's surface "from the inside." See Figure 9.23.

Snell's law indicates that, for internal reflection, $\sin\theta_t > \sin\theta_i$. Since these values are both between 0° and 90°, this relationship also implies $\theta_t > \theta_i$, as seen in Figure 9.23. In the case of external reflection the opposite is true—compare this behavior to Figure 9.9 on page 302. This difference is key to understanding how internal and external reflection differ. In external reflection, a valid (smaller) value of $\sin\theta_t$ exists for every possible value of $\sin\theta_i$ between 0 and 1. The same is not true for internal reflection. For values of θ_i greater than a *critical angle* θ_c, Snell's law implies that $\sin\theta_t > 1$, which is impossible. What happens in reality is that there is no θ_t. When $\theta_i > \theta_c$, no transmission occurs, and all the incoming light is reflected. This phenomenon is known as *total internal reflection*.

The Fresnel equations are symmetrical, in the sense that the incoming and transmission vectors can be switched and the reflectance remains the same. In combination with Snell's law, this symmetry implies that the $F(\theta_i)$ curve for internal reflection will resemble a "compressed" version of the curve for external reflection. The value of F_0 is the same for both cases, and the internal reflection curve reaches perfect reflectance at θ_c instead of at 90°. This is shown in Figure 9.24, which also shows that, on average, reflectance is higher in the case of internal reflection. For example, this is why air bubbles seen underwater have a highly reflective, silvery appearance.

Internal reflection occurs only in dielectrics, as metals and semiconductors quickly absorb any light propagating inside them [285, 286]. Since dielectrics have real-valued refractive indices, computation of the critical angle from the refractive indices or from F_0 is straightforward:

$$\sin\theta_c = \frac{n_2}{n_1} = \frac{1-\sqrt{F_0}}{1+\sqrt{F_0}}. \tag{9.20}$$

The Schlick approximation shown in Equation 9.16 is correct for external reflection. It can be used for internal reflection by substituting the transmission angle θ_t for θ_i. If the transmission direction vector $\mathbf{t}$ has been computed (e.g., for rendering refractions—see Section 14.5.2), it can be used for finding θ_t. Otherwise Snell's law could be used

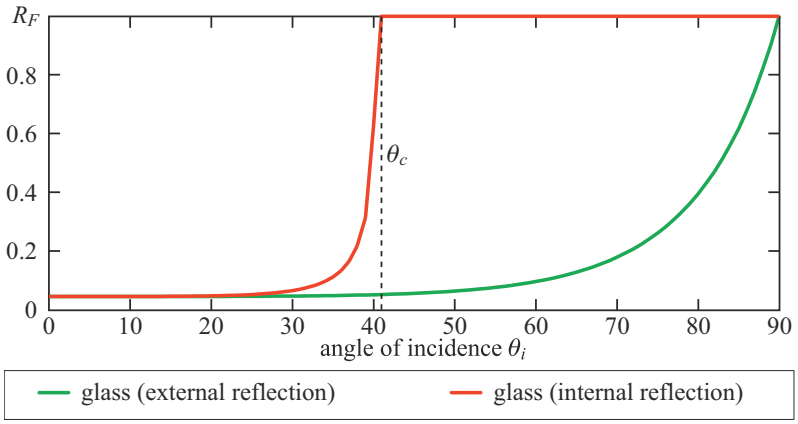

Figure 9.24. Comparison of internal and external reflectance curves at a glass-air interface. The internal reflectance curve goes to 1.0 at the critical angle θ_c.

to compute θ_t from θ_i, but that is expensive and requires the index of refraction, which may not be available.

9.6 Microgeometry

As we discussed earlier in Section 9.1.3, surface irregularities much smaller than a pixel cannot feasibly be modeled explicitly, so the BRDF instead models their aggregate effect statistically. For now we keep to the domain of geometrical optics, which assumes that these irregularities either are smaller than a light wavelength (and so have no effect on the light's behavior) or are much larger. The effects of irregularities that are in the "wave optics domain" (around 1–100 wavelengths in size) will be discussed in Section 9.11.

Each visible surface point contains many microsurface normals that bounce the reflected light in different directions. Since the orientations of individual microfacets are somewhat random, it makes sense to model them as a statistical distribution. For most surfaces, the distribution of microgeometry surface normals is continuous, with a strong peak at the macroscopic surface normal. The "tightness" of this distribution is determined by the surface roughness. The rougher the surface, the more "spread out" the microgeometry normals will be.

The visible effect of increasing microscale roughness is greater blurring of reflected environmental detail. In the case of small, bright light sources, this blurring results in broader and dimmer specular highlights. Those from rougher surfaces are dimmer because the light energy is spread into a wider cone of directions. This phenomenon can be seen in the photographs in Figure 9.12 on page 305.

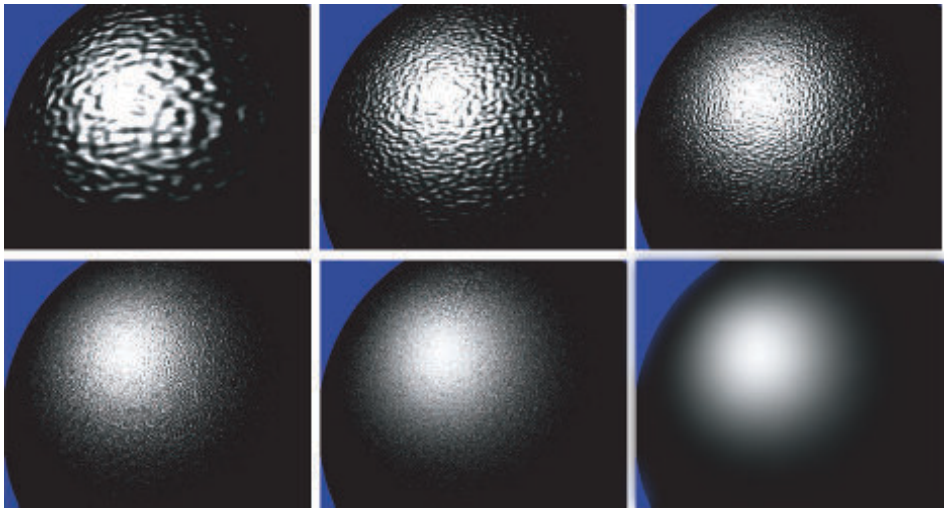

Figure 9.25. Gradual transition from visible detail to microscale. The sequence of images goes top row left to right, then bottom row left to right. The surface shape and lighting are constant. Only the scale of the surface detail changes.

Figure 9.25 shows how visible reflectance results from the aggregate reflections of the individual microscale surface details. The series of images shows a curved surface lit by a single light, with bumps that steadily decrease in scale until in the last image the bumps are much smaller than a single pixel. Statistical patterns in the many small highlights eventually become details in the shape of the resulting aggregate highlight. For example, the relative sparsity of individual bump highlights in the periphery becomes the relative darkness of the aggregate highlight away from its center.

For most surfaces, the distribution of the microscale surface normals is *isotropic*, meaning it is rotationally symmetrical, lacking any inherent directionality. Other surfaces have microscale structure that is *anisotropic*. Such surfaces have anisotropic surface normal distributions, leading to directional blurring of reflections and highlights. See Figure 9.26.

Some surfaces have highly structured microgeometry, resulting in a variety of microscale normal distributions and surface appearances. Fabrics are a commonly encountered example—the unique appearance of velvet and satin is due to the structure of their microgeometry [78]. Fabric models will be discussed in Section 9.10.

Although multiple surface normals are the primary effect of microgeometry on reflectance, other effects can also be important. *Shadowing* refers to occlusion of the light source by microscale surface detail, as shown on the left side of Figure 9.27. *Masking*, where some facets hide others from the camera, is shown in the center of the figure.

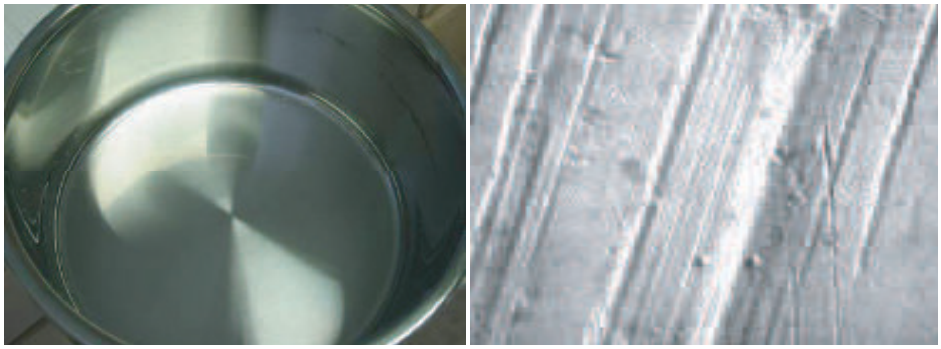

Figure 9.26. On the left, an anisotropic surface (brushed metal). Note the directional blurring of reflections. On the right, a photomicrograph showing a similar surface. Note the directionality of the detail. *(Photomicrograph courtesy of the Program of Computer Graphics, Cornell University.)*

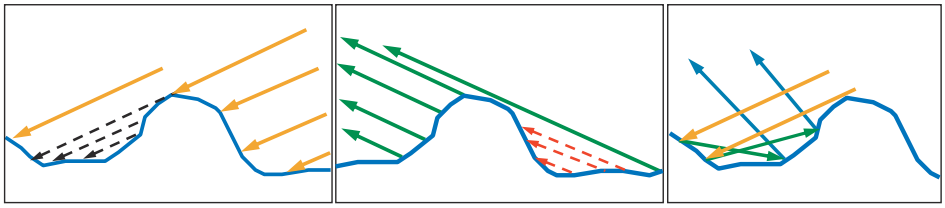

Figure 9.27. Geometrical effects of microscale structure. On the left, the black dashed arrows indicate an area that is shadowed (occluded from the light) by other microgeometry. In the center, the red dashed arrows indicate an area that is masked (occluded from view) by other microgeometry. On the right, interreflection of light between the microscale structures is shown.

If there is a correlation between the microgeometry height and the surface normal, then shadowing and masking can effectively change the normal distribution. For example, imagine a surface where the raised parts have been smoothed by weathering or other processes, and the lower parts remain rough. At glancing angles, the lower parts of the surface will tend to be shadowed or masked, resulting in an effectively smoother surface. See Figure 9.28.

For all surface types, the visible size of the surface irregularities decreases as the incoming angle θ_i to the normal increases. At extremely glancing angles, this effect can decrease the viewed size of the irregularities to be shorter than the light's wavelength, making them "disappear" as far as the light response is concerned. These two effects combine with the Fresnel effect to make surfaces appear highly reflective and mirror-like as the viewing and lighting angles approach 90° [79, 1873, 1874].

Confirm this for yourself. Roll a sheet of non-shiny paper into a long tube. Instead of looking through the hole, move your eye slightly higher, so you are looking down its