Background: Physics and Math of Shading

by Naty Hoffman

In this section of the course notes, we will go over the fundamentals behind physically based shading models, starting with a qualitative description of the underlying physics, followed by a quantitative description of the relevant mathematical models, and finally discussing how these mathematical models can be implemented for shading.

The Physics of Shading

The physical phenomena underlying shading are those related to the interaction of light with matter. To understand these phenomena, it helps to have a basic understanding of the nature of light.

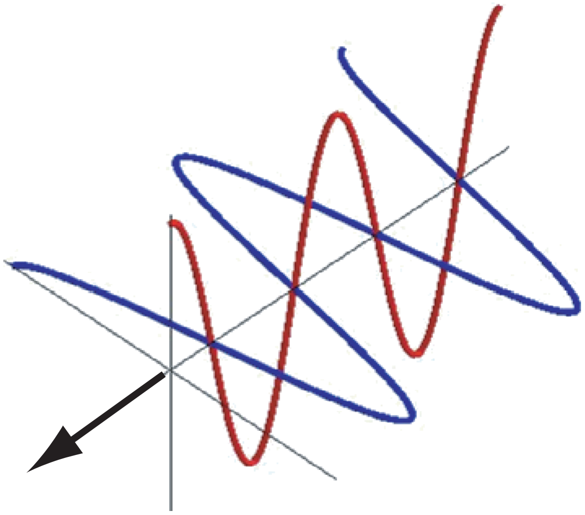

Figure 1: Light is an electromagnetic transverse wave.

Light is an electromagnetic *transverse wave*, which means that it oscillates in directions perpendicular to its propagation (see Figure 1). Since it is a wave, light is characterized by its *wavelength*—the distance from peak to peak. Electromagnetic wavelengths cover a very wide range but only a tiny part of this range (about 400 to 700 nanometers) is visible to humans and thus of interest for shading (see Figure 2).

The effect matter has on light is defined by a property called the *refractive index*. The refractive

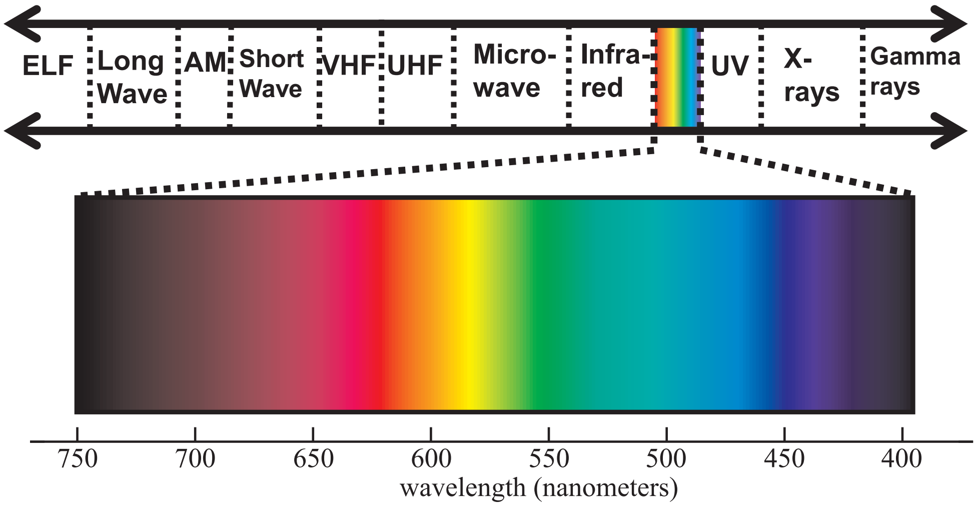

Figure 2: The visible spectrum.

index is a complex number: its real part measures how the matter affects the speed of light (slowing it down relative to its speed in a vacuum) and its imaginary part determines whether the light is *absorbed* (converted to other forms of energy) as it propagates. The refractive index may vary as a function of light wavelength.

Homogeneous Media

The simplest case of light-matter interaction is light propagating through a *homogeneous medium*. This is a region of matter with uniform index of refraction (at the scale of the light wavelength; in the case of visible light this means that any variations much smaller than 100 nanometers or so don't count).

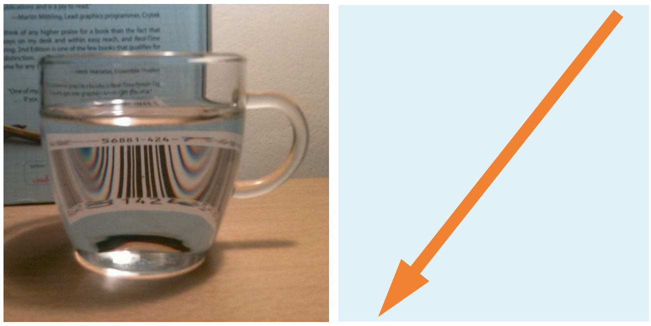

Figure 3: Light in transparent media like water and glass (left) just keeps on propagating in a straight line at the same intensity and color (right).

A *transparent* medium is one in which the complex part of the index of refraction is very low for visible light wavelengths. This means that there is no significant absorption and any light propagating through the medium just keeps on going in a straight line, unchanged. Examples of transparent media include water and glass (see Figure 3).

If a homogeneous medium does have significant absorptivity in the visible spectrum, it will absorb some amount of light passing through it. The farther the distance traveled by the light, the higher the absorption. However, the direction of the light will not change, just its intensity (and, if the absorptivity is selective to certain visible wavelengths, the color)—see Figure 4.

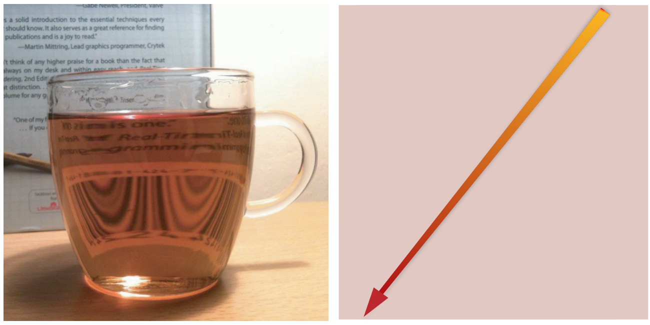

Figure 4: Light propagating through clear, absorbent media (left) continues in a straight line, but loses intensity (and may change color) with distance (right).

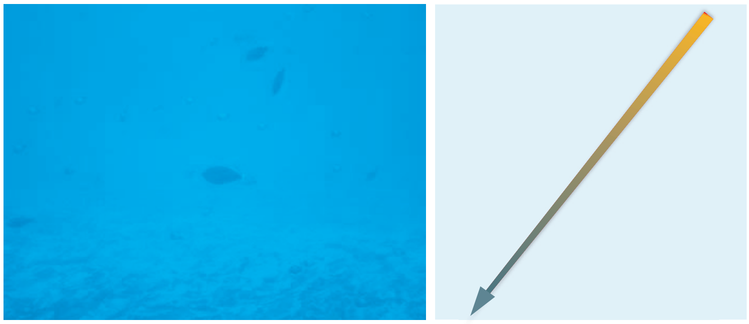

Figure 5: The slight absorptivity of water becomes significant over larger distances.

Note that the scale as well as the absorptivity of the medium matters. for example, water actually absorbs a little bit of visible light, especially on the red end of the spectrum. On a scale of inches this is negligible (as shown in Figure 3) but it is quite significant over many feet of distance—see Figure 5.

Scattering

In homogeneous media, light always travels in a straight line and does not change its direction (although its amount can be reduced by absorption). A *heterogeneous medium* has variations in the index of refraction. If the index of refraction changes slowly and continuously, then the light bends in a curve. However, if the index of refraction changes abruptly (i.e., over a shorter distance than the light wavelength), then the light *scatters*: it splits into multiple directions. Note that scattering does not change the overall amount of light.

Microscopic particles induce an isolated "island" where the refraction index differs from surrounding regions. This causes light to scatter continuously over all possible outgoing directions (see Figure 6). Note that the distribution of scattered light over different directions is typically non-uniform and depends on the type of particle. Some cause forward scattering (more light goes in the forward direction), some cause backscattering (more light goes in the reverse of the original direction), and some have complex distributions with "spikes" in certain directions.

In *cloudy media*, the density of scattering elements is sufficient to somewhat randomize the direction of light propagation (Figure 7). In *translucent* or *opaque media* the density of scattering elements is so high that the light direction is completely randomized (Figure 8).

Like absorption, scattering depends on scale; a medium such as clean air which has negligible

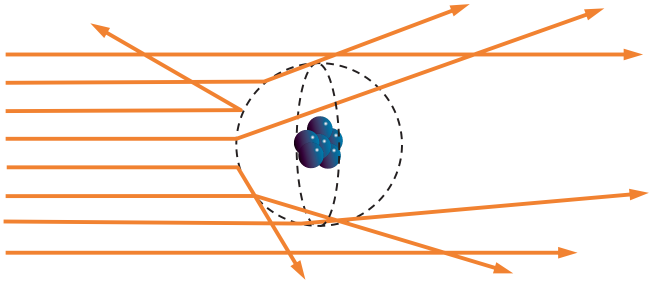

Figure 6: Particles cause light to scatter in all directions.

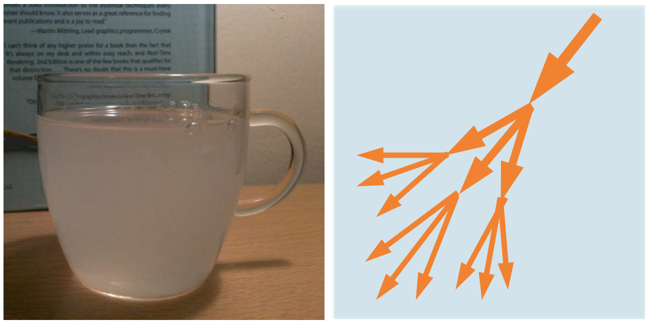

Figure 7: Light in cloudy media (left) has its direction somewhat randomized as it propagates (right).

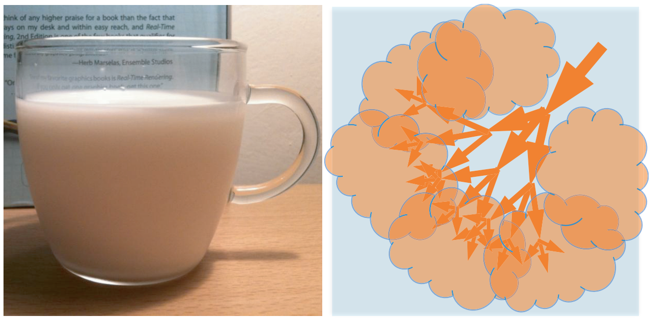

Figure 8: Light in translucent or opaque media (left) has its direction completely randomized as it propagates (right).

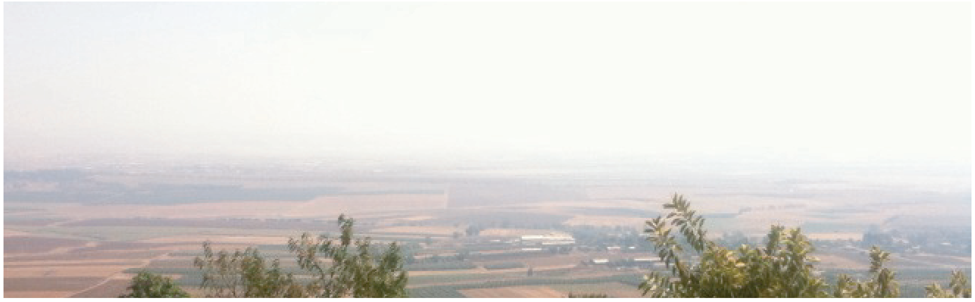

Figure 9: Even clean air causes considerable light scattering over a distance of miles.

scattering over distances of a few feet causes substantial light scattering over many miles (Figure 9).

Media Appearance

Previous sections discussed two different modes of interaction between matter and light. Regions of matter with complex-valued refraction indices cause absorption—the amount of light is lessened over distance (potentially also changing the light color if absorption occurs preferentially at certain wavelengths), but the light's direction does not change. On the other hand, rapid changes in the index of refraction cause scattering—the direction of the light changes (splitting up into multiple directions), but the overall amount or spectral distribution of the light does not change. There is a third mode of interaction: *emission*, where new light is created from other forms of energy (the opposite of absorption). This occurs in light sources, but it doesn't come up often in shading. Figure 10 illustrates the three modes of interaction.

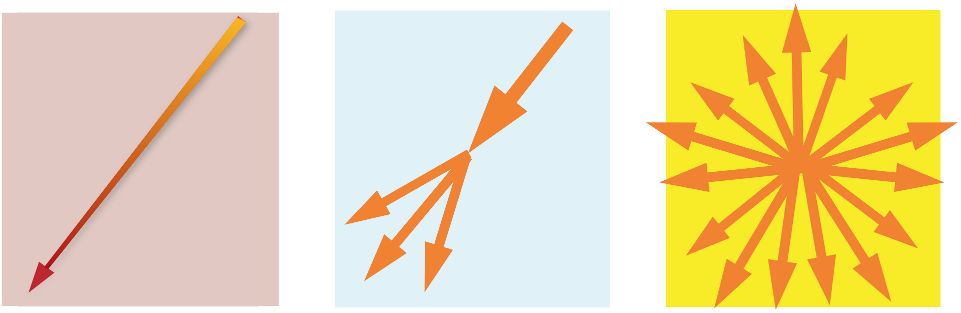

Figure 10: The three modes of interaction between light and matter: absorption (left), scattering (middle), and emission (right).

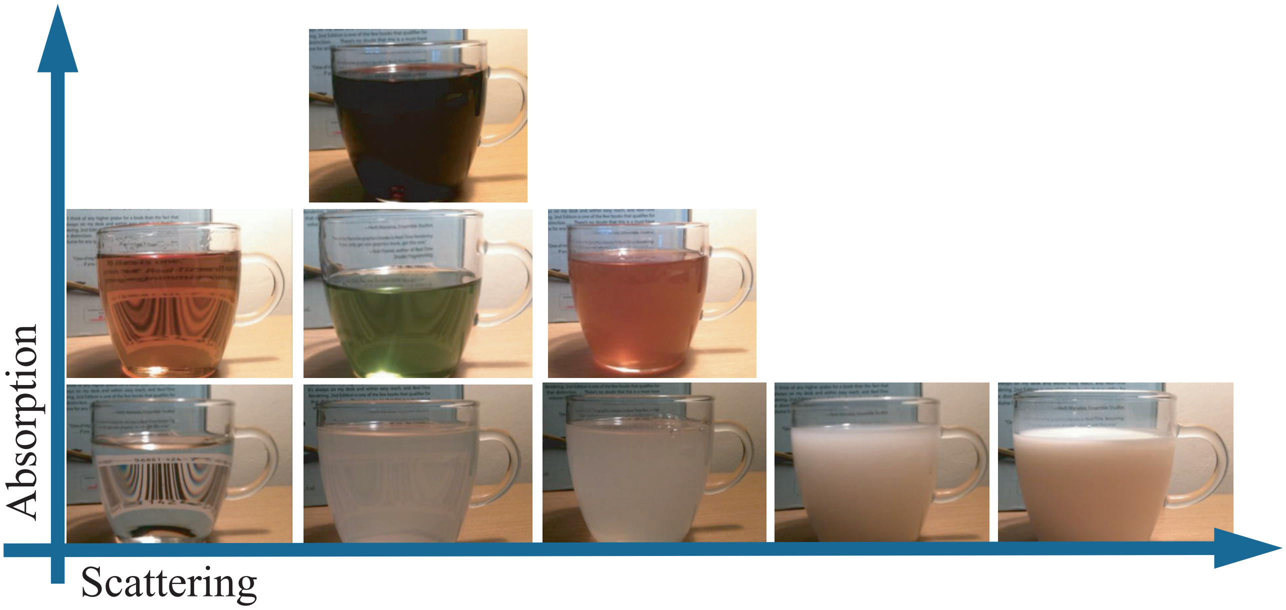

Figure 11: Media with varying amounts of light absorption and scattering. The appearance depends on both properties; for example, a white appearance (lower right) is the result of high scattering combined with low absorption.

Most media both scatter and absorb light to some degree. Each medium's appearance depends on the relative amount of scattering and absorption present. Figure 11 shows media with various combinations of scattering and absorptivity.

Scattering at a Planar Boundary

Maxwell's equations can be used to compute the behavior of light when the index of refraction changes, but in most cases analytical solutions do not exist. There is one special case which does have a solution, and it is especially relevant for surface shading. This is the case of an infinite, perfectly flat planar boundary between two volumes with different refractive indices. This is a good description of an object surface, with the refractive index of air on one side of the boundary, and the refractive index of the object on the other. The solutions to Maxwell's equations in this special case are called the *Fresnel equations*.

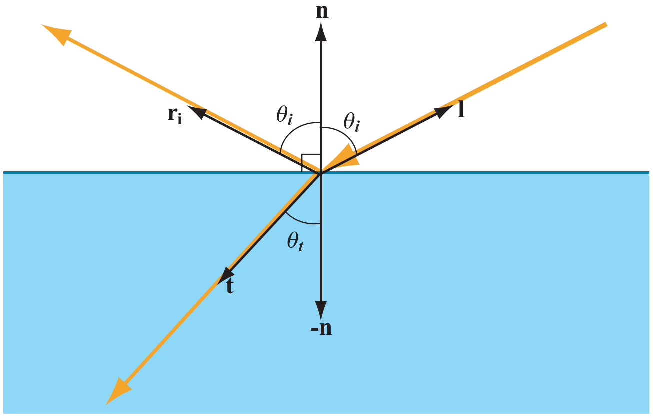

Figure 12: Refractive index changes at planar boundaries cause light to scatter in two directions. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

Although real object surfaces are not infinite, in comparison to the wavelength of visible light they can be treated as such. As for being "perfectly flat", an objection might be raised that no object's surface can truly be flat—if nothing else, individual atoms will form pico-scale "bumps". However, as with everything else, the scale relative to the light wavelength matters. It is indeed possible to make surfaces that are perfectly flat at the scale of hundreds of nanometers—such surfaces are called *optically flat* and are typically used for high-quality optical instruments such as telescopes.

In the special case of a planar refractive index boundary, instead of scattering in a continuous fashion over all possible directions, light splits into exactly two directions: reflection and refraction (Figure 12).

As you can see in Figure 12, the angle of reflection is equal to the incoming angle, but the angle of refraction is different and depends on the refractive index of the medium[1]. The proportions of reflected and refracted light are described by the Fresnel equations, and will be discussed in a later section.

Non-Optically-Flat Surfaces

Of course, most real-world surfaces are not polished to the same tolerances as telescope mirrors. What happens with surfaces that are not optically flat? In most cases, there are indeed irregularities present which are much larger than the light wavelength, but too small to be seen or rendered directly (i.e., they are smaller than the coverage area of a single pixel or shading sample). In this case, the surface behaves like a large collection of tiny optically flat surfaces. The surface appearance is the aggregate

[1] If you are interested in the exact math, look up *Snell's Law*.

result of many points with different surface orientations—each point reflects incoming light in a slightly different direction (Figure 13).

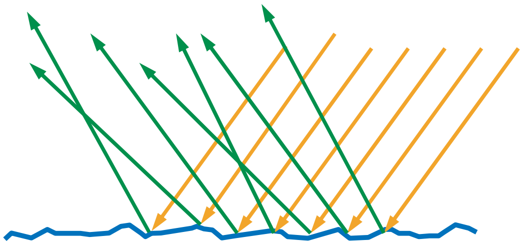

Figure 13: Visible reflections from non-optically flat surfaces are the aggregate result of reflections from many surface points with different orientations. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

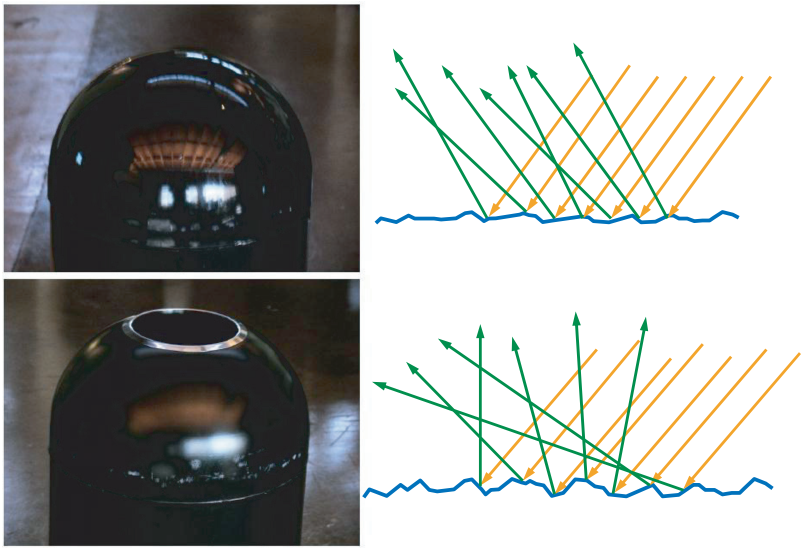

Figure 14: On the top row, the surface is relatively smooth; the surface orientation only varies slightly, resulting in a small variance in reflected light directions and thus sharper reflections. The surface on the bottom row is rougher; different points on the surface have widely varying orientations, resulting in a high variance in reflected light directions and thus blurry reflections. Note that both surfaces appear smooth at the visible scale—the roughness difference is at the microscopic scale. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

The rougher the surface is at this microscopic scale, the blurrier the reflections, as the surface orientations diverge more strongly from the overall macroscopic surface orientation (Figure 14). For shading purposes, it is common to treat this *microgeometry* statistically and view the surface as reflecting (and refracting) light in multiple directions at each point (Figure 15).

Subsurface Scattering

What happens to the refracted light? It depends on the composition of the object. In the case of metals, refracted light is immediately absorbed[2]—the energy is soaked up right away by the free

[2] Metals have very high absorption coefficients (the imaginary part of the refractive index) in the visible spectrum.

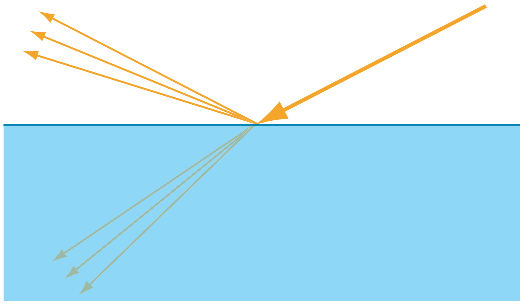

Figure 15: When viewed macroscopically, non-optically flat surfaces can be treated as reflecting (and refracting) light in multiple directions. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

electrons. On the other hand, non-metals (also referred to as dielectrics or insulators) behave as regular participating media once the light is refracted inside of them, exhibiting the range of absorption and scattering behaviors we covered in previous sections. In most cases, some of the refracted light is scattered enough to be re-emitted out of the same surface. Both of these cases are illustrated in Figure 16.

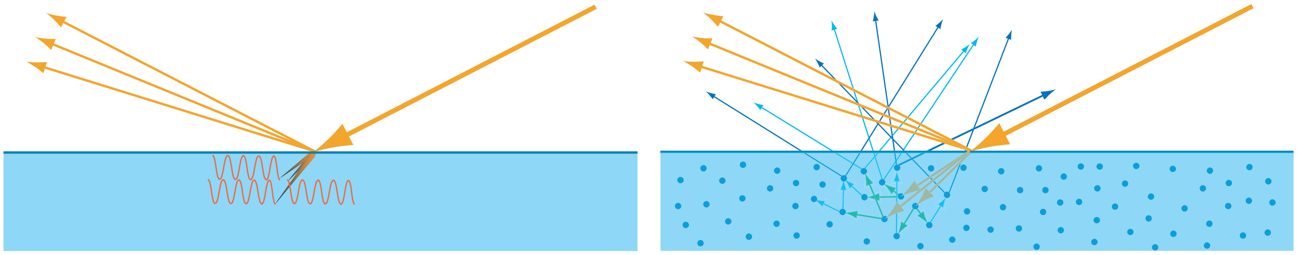

Figure 16: In metals (on the left), all refracted light energy is immediately absorbed by free electrons; in non-metals (on the right) refracted light energy scatters until it re-emerges from the surface, typically after undergoing partial absorption. (*Images from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

On the right side of Figure 16, you can see that the subsurface-scattered light (denoted with blue arrows) is emitted from various points on the surface, at varying distances from the original entrance point of the light. Figure 17 shows the relationship between these distances and the pixel size in two cases. On the upper left, the pixel is larger than the entry-to-exit subsurface scattering distances. In this case, the entry-to-exit distances can be ignored and the subsurface scattered light can be assumed to enter and exit the surface at the same point, as seen on the upper right. This allows shading to be handled as a completely local process: the outgoing light at a point only depends on incoming light at the same point. On the bottom of Figure 17, the pixel is smaller than the entry-to-exit distances. In this case, the shading of each point is affected by light impinging on other points. To capture this effect, local shading will not suffice and specialized rendering techniques need to be used. These are typically referred to as "subsurface scattering" techniques, but it is important to note that "ordinary" diffuse shading is the result of the same physical phenomena (subsurface scattering of refracted light). The only difference is the scattering distance relative to the scale of observation. This insight tells us that materials which are commonly thought of as exhibiting "subsurface scattering" behavior can be handled with regular diffuse shading at larger distances (e.g. the skin of a distant character). On the other hand, materials that are thought of as exhibiting "regular diffuse shading" behavior will have a "subsurface scattering" appearance when viewed very close up (e.g. an extreme close-up of a small

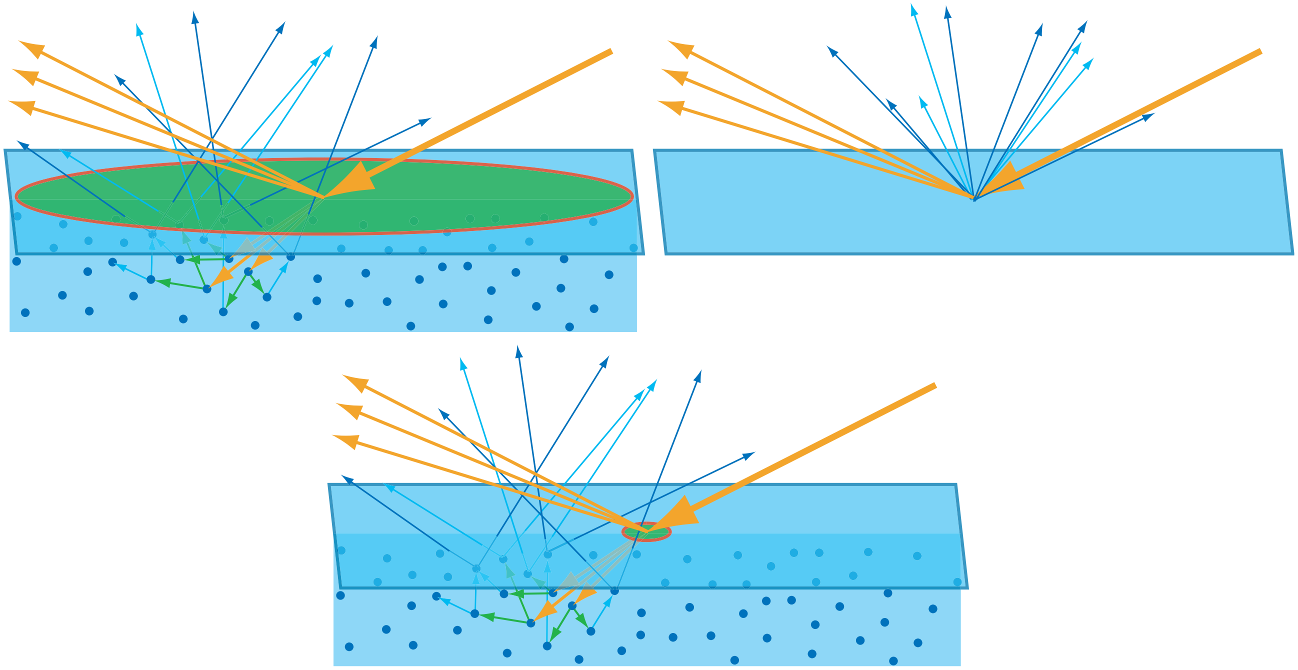

Figure 17: On the upper left, the pixel (green circle with red border) is larger than the distances traveled by the light before it exits the surface. In this case, the outgoing light can be assumed to be emitted from the entry point (upper right). On the bottom, the pixel is smaller than the scattering distances; these distances cannot be ignored if realistic shading is desired. (*Images from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

plastic toy).

The Mathematics of Shading

The measurement of electromagnetic radiation in general (including visible light) is called *radiometry*. There are various radiometric quantities used to measure light over surfaces, over directions, etc.; we will only concern ourselves with *radiance*, which is used to quantify the magnitude of light along a single ray[3]. We will use the common radiometric notation L to denote radiance; when shading a surface point, L_i denotes radiance incoming to the surface and L_o denotes outgoing radiance.

Radiance (like other radiometric quantities) is a *spectral quantity*: the amount varies as a function of wavelength. In theory, to express visible-light radiance, a continuous spectral distribution needs to be stored. Dense spectral samples are indeed used in some specialized rendering applications, but for production (film and game) rendering, RGB triples are used instead. An explanation of how these triples relate to spectral distributions can also be found in many sources, including *Real-Time Rendering* [58].

The BRDF

It is commonly assumed that shading can be handled locally, as illustrated on the upper right of Figure 17. In this case, how a given surface point responds to light only depends on the incoming (light) and outgoing (view) directions. In this document, we will use $\mathbf{v}$ to denote a unit-length vector pointing

[3] An explanation of other radiometric quantities can be found in various texts, including Chapter 7 of the 3rd edition of *Real-Time Rendering* [58] and Dutré's *Global Illumination Compendium* [23].

along the outgoing direction and **l** to denote a unit-length vector pointing opposite to the incoming direction (it is convenient to have all vectors point away from the surface). The surface's response to light is quantified by a function called the BRDF (Bidirectional Reflectance Distribution Function), which we will denote as $f(\mathbf{l}, \mathbf{v})$. Each direction (incoming and outgoing) can be parameterized with two numbers (e.g. polar coordinates), so the overall dimensionality of the BRDF is four. In many cases, rotating the light and view directions around the surface normal does not affect the BRDF. Such *isotropic BRDFs* can be parameterized with three angles, as shown in Figure 18. In practice, the number of angles used to compute a given BRDF varies from one to five; some commonly used angles are illustrated in Figure 19.

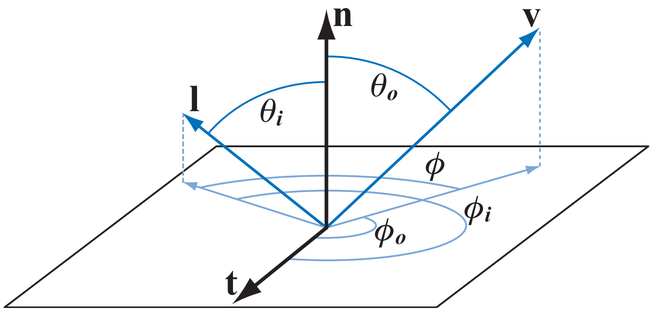

Figure 18: The BRDF depends on incoming and outgoing directions; these can be parameterized with four angles, or three in the case of isotropic BRDFs. Here **n** is the surface normal vector, **l** is the incoming light direction vector, **v** is the outgoing (view) direction vector, and **t** is a tangent vector defining a preferred direction over the surface (this is only used for *anisotropic BRDFs*, where the reflection behavior changes when light and view vector are rotated around **n**). (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

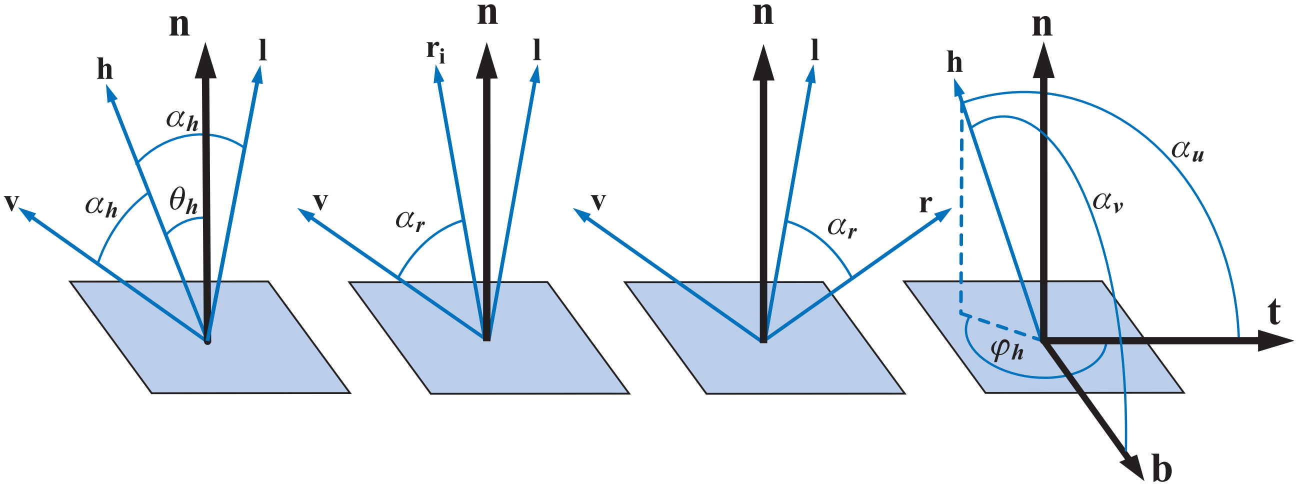

Figure 19: Examples of some angles that are commonly used in BRDF evaluation, in addition to those in Figure 18 (*images from "Real-Time Rendering, 3rd edition" used with permission from A K Peters*).

In principle, the BRDF is only defined for light and view directions above the surface[4]; in other words, the dot products $\mathbf{n} \cdot \mathbf{l}$ and $\mathbf{n} \cdot \mathbf{v}$ (which are commonly found in BRDF expressions) must both

[4] Transmittance through the surface is modeled via a bidirectional transmittance distribution function (BTDF) or the more general bidirectional scattering distribution function (BSDF) which includes both reflection and transmittance.

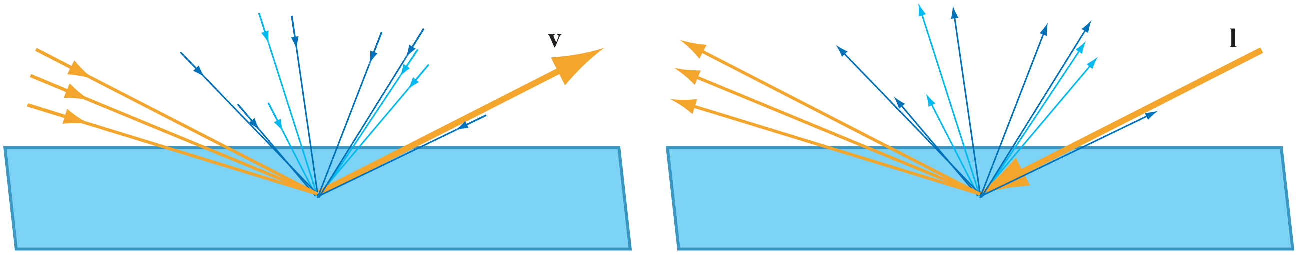

Figure 20: On the left side, we see one interpretation of the BRDF: that for a given outgoing (view) direction, it specifies the relative contributions of incoming light. On the right side we see an alternative interpretation: that for a given incoming light direction, it specifies the distribution of outgoing light. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

be non-negative[5]. Avoiding back-facing light directions is straightforward: either by only gathering incoming light over front-facing directions, or by setting the light contributions from any back-facing direction to zero. Back-facing view directions should in theory never happen, but interpolated vertex normals and normal mapping (both common in games) can create such situations in practice. Evaluation of the BRDF for negative values of $\mathbf{n} \cdot \mathbf{v}$ can be avoided by clamping $\mathbf{n} \cdot \mathbf{v}$ to 0 or using its absolute value. In some cases, simply evaluating the BRDF for negative values may produce better results [59] but then care must be taken to ensure the math is well-behaved.

The BRDF can be intuitively interpreted in two ways, both of which are valid. The first interpretation is that given a ray of light incoming from a certain direction, the BRDF gives the relative distribution of reflected and scattered light over all outgoing directions above the surface. The second interpretation is that for a given view direction, the BRDF gives the relative contribution of light from each incoming direction to the outgoing light. These two interpretations are illustrated in Figure 20.

The BRDF is a spectral quantity. In theory the input and output wavelengths would be additional BRDF inputs, greatly increasing its dimensionality. However, in practice there is no crosstalk between the individual wavelengths[6], so each wavelength of outgoing light is only affected by that same wavelength in the incoming light. This means that instead of treating input and output wavelengths as BRDF inputs, we (more simply) treat the BRDF as a spectral-valued function that is multiplied with a spectral-valued light color. In production shading, this means an RGB-valued BRDF multiplied by RGB-valued light colors.

The BRDF is used in the *reflectance equation*[7]:

$$L_o(\mathbf{v}) = \int_{\Omega} f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l})\, d\omega_i \tag{1}$$

Although this equation may seem a bit daunting, its meaning is straightforward: outgoing radiance equals the integral (over all directions above the surface) of incoming radiance times the BRDF and a cosine factor. If you are not familiar with integrals, you can think of them as a kind of continuous weighted average. The $\otimes$ symbol is used here to denote component-wise vector multiplication; it is used because both BRDF and light color are spectral (RGB) vectors.

Not every function over incoming and outgoing directions can make sense as a BRDF. It is commonly recognized that two properties a BRDF must have to be *physically plausible* are: *reciprocity*

These effects are beyond the scope of these notes—more details on how to model them can be found in a 2007 paper by Walter et al. [73].

[5] Recall that the dot product between two unit-length vectors is equal to the cosine of the angle between them; if this is negative, then the angle exceeds 90°.

[6] There are two physical phenomena involving such crosstalk—fluorescence and phosphorescence—but they rarely occur in production shading.

[7] The reflection equation is a special case of the *rendering equation* [37].

and *energy conservation*. Reciprocity simply means that the BRDF has the same value if $\mathbf{l}$ and $\mathbf{v}$ are swapped:

$$f(\mathbf{l},\mathbf{v}) = f(\mathbf{v},\mathbf{l}) \tag{2}$$

Energy conservation refers to the fact that a surface cannot reflect more than 100% of incoming light energy. Mathematically, it is expressed via the following equation:

$$\forall \mathbf{l}, \int_{\Omega} f(\mathbf{l},\mathbf{v})(\mathbf{n}\cdot\mathbf{v})\, d\omega_o \leq 1 \tag{3}$$

This means that for any possible light direction $\mathbf{l}$, the integral of the BRDF times a cosine factor over outgoing directions $\mathbf{v}$ must not exceed 1.

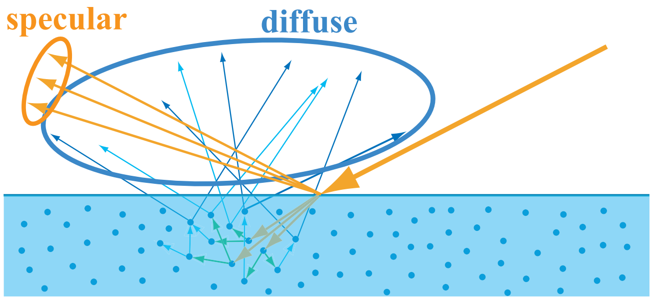

Figure 21: BRDF specular terms are typically used for surface reflection, and BRDF diffuse terms for subsurface scattering. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

The phenomena described by the BRDF includes (at least for non-metals) two distinct physical phenomena: surface reflection and subsurface scattering. Since each of these phenomena has different behavior, BRDFs typically include a separate term for each one. The BRDF term describing surface reflection is usually called the *specular term* and the term describing subsurface scattering is called the *diffuse term*—see Figure 21.

Surface Reflectance (Specular Term)

Physically based specular BRDF terms are typically based on *microfacet theory*. This theory was developed to describe surface reflection from general (non-optically flat) surfaces. The basic assumption underlying microfacet theory is the presence of surface variation (*microgeometry*) at a scale which is smaller than the scale of observation (e.g. the shading resolution) but greater than the scale of visible light wavelengths (so geometric optics apply and wave effects such as diffraction can be ignored)[8]. So far, microfacet theory has only been used to derive expressions for *single-bounce* surface reflection; a comprehensive theory for multiple-bounce surface reflection remains an unsolved problem.

Since the microgeometry is assumed to be significantly larger in scale than visible light wavelengths, it follows that each surface point can be treated as being optically flat. As mentioned in the previous section, an optically flat surface splits light into exactly two directions: reflection and refraction.

Each surface point reflects light from a given incoming direction into a single outgoing direction which depends on the orientation of the *microgeometry normal* $\mathbf{m}$. When evaluating a BRDF term, both the light direction $\mathbf{l}$ and the view direction $\mathbf{v}$ are specified. This means that of all surface points,

[8] The origin of the name "microfacet theory" is the concept that such surfaces are composed of many individually flat microscopic facets. However, the sharp edges of such facets would in fact cause significant diffraction effects. Fortunately, microfacet theory is still valid (if somewhat misnamed) if it is taken instead to refer to a smoothly undulating surface with local radius of curvature significantly larger than visible light wavelengths.

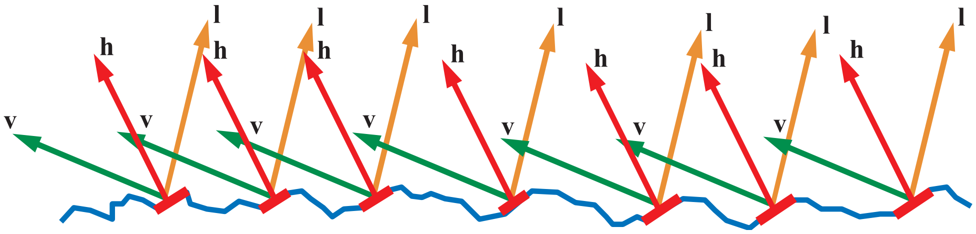

Figure 22: Surface points with $\mathbf{m} = \mathbf{h}$ are oriented to reflect $\mathbf{l}$ into $\mathbf{v}$—other surface points do not contribute to the BRDF. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

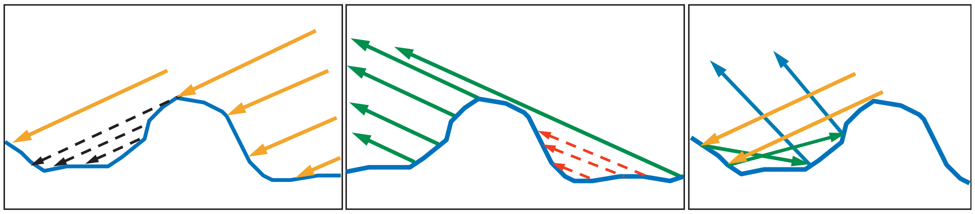

Figure 23: On the left, we see that some surface points are occluded from the direction of $\mathbf{l}$, so they are shadowed and do not receive light (so they cannot reflect any). In the center, we see that some surface points are not visible from the view direction $\mathbf{v}$, so of course any light reflected from them will not be seen. In both cases these surface points do not contribute to the BRDF. In reality, although shadowed regions do not receive any direct light from $\mathbf{l}$ they do receive (and thus reflect) light bounced from other surface regions (as seen on the right). These *interreflections* are ignored by microfacet theory. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

only those that that happen to be oriented just right to reflect $\mathbf{l}$ into $\mathbf{v}$ could potentially contribute to the BRDF value. In Figure 22, we can see that these "correctly oriented" surface points have their surface normal $\mathbf{m}$ oriented exactly halfway between $\mathbf{l}$ and $\mathbf{v}$. The vector halfway between $\mathbf{l}$ and $\mathbf{v}$ is called the *half-vector* or *half-angle vector*; we will denote it as $\mathbf{h}$.

Not all surface points for which $\mathbf{m} = \mathbf{h}$ will actively contribute to the reflection; some are blocked by other surface regions from the direction of $\mathbf{l}$ (*shadowing*), from the direction of $\mathbf{v}$ (*masking*), or from both. Microfacet theory assumes that all shadowed light is lost from the specular term; in reality, due to multiple surface reflections some of it will eventually be visible, but this is not accounted for in microfacet theory. The various types of light-surface interaction are shown in Figure 23.

With these assumptions (locally optically flat surface, no interreflections), a specular BRDF term can be derived from first principles [1, 73]. This specular BRDF term has the following form[9]:

$$f_{\mu\text{facet}}(\mathbf{l}, \mathbf{v}) = \frac{F(\mathbf{l}, \mathbf{h})G(\mathbf{l}, \mathbf{v}, \mathbf{h})D(\mathbf{h})}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})} \tag{4}$$

We will go into each of the terms in more detail, but first a quick summary. $D(\mathbf{h})$ is the microgeometry *normal distribution function* (NDF) evaluated at the half-vector $\mathbf{h}$; in other words, the concentration (relative to surface area) of surface points which are oriented such that they *could* reflect light from $\mathbf{l}$ into $\mathbf{v}$. $G(\mathbf{l}, \mathbf{v}, \mathbf{h})$ is the *geometry function*[10]; it tells us the percentage of surface points

[9] Note that for the dot products in the denominator, it is not sufficient to avoid negative values (as mentioned above for BRDFs in general)—zero values must be avoided as well. In practice, this is typically done by adding a very small positive value after the usual clamp or absolute value operation.

[10] Also known in the literature by various other names, such as: geometry factor, shadowing-masking function, shadowing function, geometry term, etc.

with $\mathbf{m} = \mathbf{h}$ that are *not* shadowed or masked, as a function of the light direction $\mathbf{l}$ and the view direction $\mathbf{v}$. Therefore, the product of $D(\mathbf{h})$ and $G(\mathbf{l}, \mathbf{v}, \mathbf{h})$ gives us the concentration of *active surface points*, the surface points that actively participate in the reflectance by successfully reflecting light from $\mathbf{l}$ into $\mathbf{v}$. $F(\mathbf{l}, \mathbf{h})$ is the Fresnel reflectance of the active surface points as a function of the light direction $\mathbf{l}$ and the active microgeometry normal $\mathbf{m} = \mathbf{h}$. It tells us how much of the incoming light is reflected from each of the active surface points. Finally, the denominator $4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})$ is a correction factor, which accounts for quantities being transformed between the local space of the microgeometry and that of the overall macrosurface.

Fresnel Reflectance

The Fresnel reflectance function computes the fraction of light reflected from an optically flat surface. Its value depends on two things: the *incoming angle* (the angle between the light vector and the surface normal—also referred to as the *incident angle* or *angle of incidence*) and the refractive index of the material. Since the refractive index may vary over the visible spectrum, the Fresnel reflectance is a spectral quantity—for production purposes, an RGB triple. We also know that each of the RGB values have to lie within the 0 to 1 range, since a surface cannot reflect less than 0% or more than 100% of the incoming light.

The full Fresnel equations are somewhat complex, and the required material parameter (complex refractive index sampled densely over the visible spectrum) is not convenient for artists, to say the least. However, a simpler expression with more convenient parametrization can be derived by inspecting the behavior of these equations for real-world materials. With this in mind, let us take a look at the graph in Figure 24.

The materials selected for this graph represent a wide variety. Despite this, some common elements can be seen. Reflectance is almost constant for incoming angles between 0° and about 45°. The reflectance changes more significantly (typically, but not always, increasing somewhat) between 45° and about 75°. Finally, between 75° and 90° reflectance always goes rapidly to 1 (white, if viewed as a color).

It is instructive to see how these values and angle ranges vary over a simple 3D scene. Putting microfacet theory to the side for the moment, we will look at a simpler case: reflection from an optically-smooth metal surface. Such a surface reflects each incoming light ray in exactly one outgoing direction. Representing this behavior as a BRDF is inconvenient (it collapses to a delta function) but there is a much simpler way to model it for rendering: sample incoming radiance from the direction of the reflected view vector and multiply it by the Fresnel reflectance to get outgoing radiance. This works well with illumination models that can represent incoming light from arbitrary directions (illumination models will be further discussed later in these notes). In this case, the "incoming angle" for Fresnel reflectance should be computed using the angle between the surface normal $\mathbf{n}$ and the view vector $\mathbf{v}$ (the incoming vector in this case is the reflection of $\mathbf{v}$ around $\mathbf{n}$ and has the same angle). Figure 25 shows how this angle (and the resulting Fresnel reflectance) varies over a simple 3D scene. We can see that the green and yellow zones predominate—the Fresnel reflectance barely changes over the majority of visible pixels, only significantly increasing at the edges.

In microfacet theory, we are dealing not with a smooth surface but with locally smooth microgeometry. In this case, we are interested in the Fresnel reflectance of the individual surface points. Since the surface points contributing to the BRDF all have microgeometry normals equal to the half vector $\mathbf{h}$ (recall Figure 22), it follows that $\mathbf{h}$ is the vector that should be used to compute Fresnel reflectance. The Fresnel incoming angle is then the angle between $\mathbf{h}$ and $\mathbf{l}$[11].

Figure 26 visualizes the angle between $\mathbf{h}$ and $\mathbf{l}$ in the same way that Figure 25 visualized the angle between $\mathbf{n}$ and $\mathbf{v}$. From the figure it may appear that for certain light angles, the yellow or even red

[11] $\mathbf{v}$ could be used instead of $\mathbf{l}$, since both form the same angle with $\mathbf{h}$ by definition.

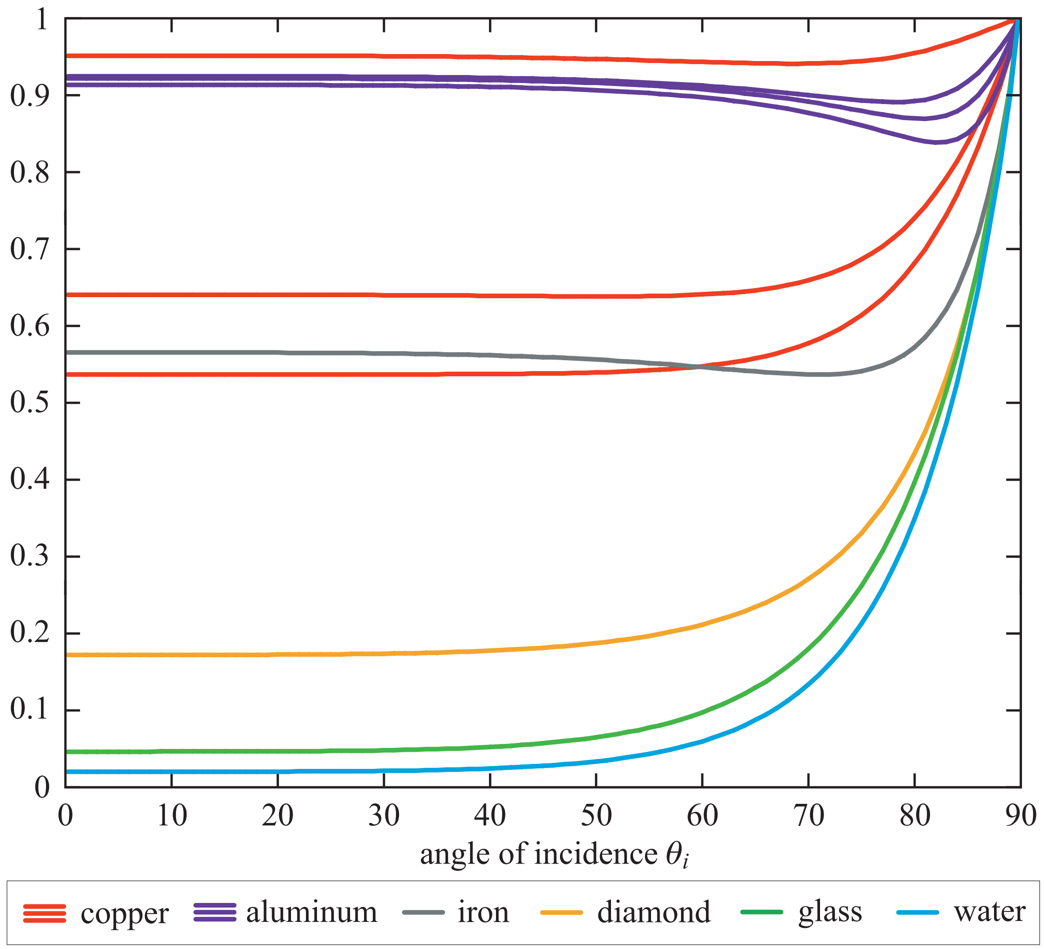

Figure 24: Fresnel reflectance for a variety of substances, plotted (y axis) as functions of incoming angle (x axis). Since copper and aluminum have significant variation in their reflectance over the visible spectrum, their reflectance is shown as three separate curves for R, G, and B. Copper's R curve is highest, followed by G, and finally B (thus its reddish color). Aluminum's B curve is highest, followed by G, and finally R. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

zones can dominate the scene. However, as we can see in Figure 27, this is misleading: when the rest of the BRDF is taken into account, then it is clear that the red zone (where the reflectance increases significantly) is limited to the edges.

Since the Fresnel reflectance stays close to the value for 0° over most of the visible parts of a given 3D scene, we can think of this value (which we will denote F_0) as the *characteristic specular reflectance* of the material. This value has all the properties of what is typically thought of as a "color"—it is composed of RGB values between 0 and 1, and it is a measure of selective reflectance of light. For this reason, we will also refer to this value as the *specular color* of the surface.

F_0 looks like an ideal parameter for a Fresnel reflectance approximation, and indeed Schlick [63] gives a cheap and reasonably accurate approximation that uses it:

$$F_{\text{Schlick}}(F_0, \mathbf{l}, \mathbf{n}) = F_0 + (1 - F_0)(1 - (\mathbf{l} \cdot \mathbf{n}))^5 \tag{5}$$

This approximation is widely used in computer graphics[12]. When used in a microfacet BRDF, the

[12] Games sometimes use a slightly cheaper approximation based on spherical Gaussians [38, 45, 50].

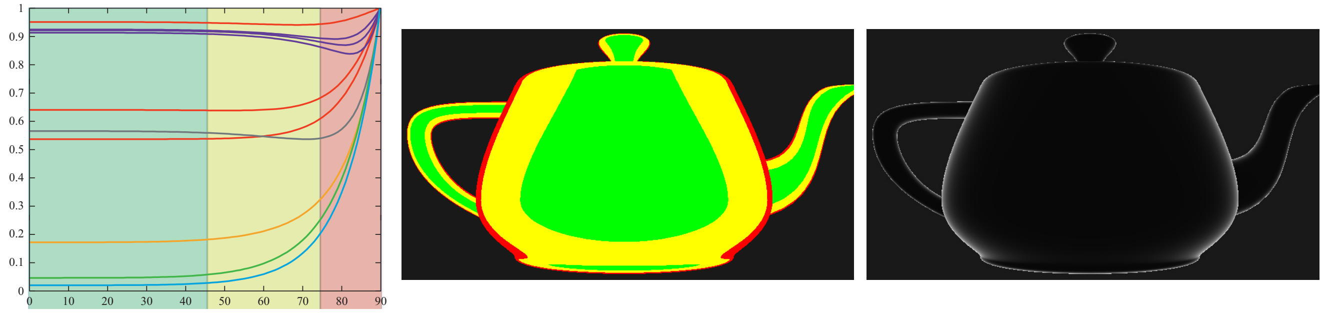

Figure 25: On the left, we can see the Fresnel graph from Figure 24 with overlaid colors for three zones based on the angle between the smooth surface normal and the view or light direction. Green for the zone (0° to 45°) where reflectance is almost constant, yellow for the zone (45° to 75°) where reflectance changes gradually, and red for the zone (75° to 90°) where reflectance goes rapidly to 1. In the center image, these zones are visualized on a 3D object for the angle between **n** and **v** (which is the relevant angle for Fresnel reflectance in the case of an optically flat surface). On the right we can see the Fresnel reflectance of glass (the green curve in Figure 24) visualized as color on the same object.

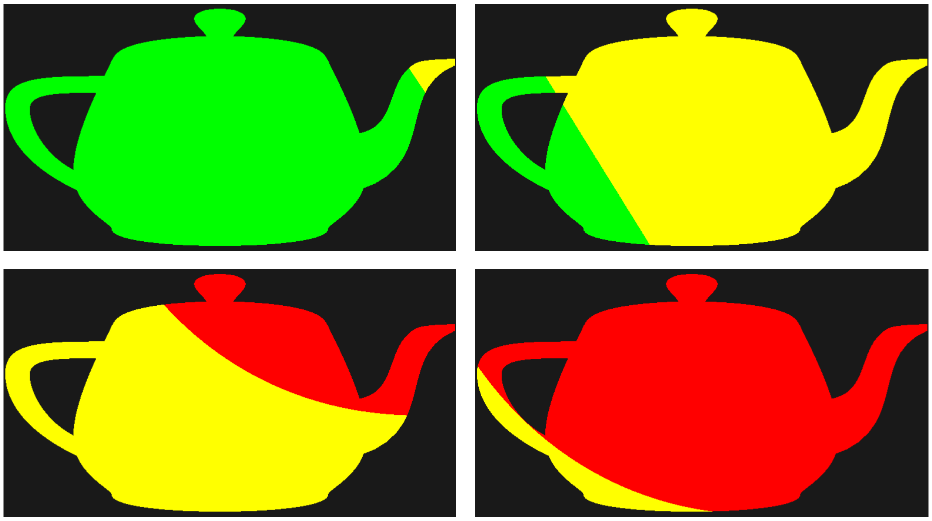

Figure 26: This figure shows the Fresnel zones for the **h** vector, using the same approach as for the **n** vector in Figure 25. Since the **h** vector depends on the light direction **l** as well as the view direction **v**, the figure contains four images—each for a different light direction. We can see that for certain light angles, the entire object may be in the yellow or even the red zone. As we will see in Figure 27, this is somewhat misleading.

active microgeometry normal **h** must be substituted for the surface normal **n**:

$$F_{\text{Schlick}}(F_0, \mathbf{l}, \mathbf{h}) = F_0 + (1 - F_0)(1 - (\mathbf{l} \cdot \mathbf{h}))^5 \tag{6}$$

To know which values are reasonable to assign to F_0, it is useful to look at the values for various real-world materials. These can be found in Table 1. Values are given in both linear and gamma (sRGB) space; we recommend anyone unfamiliar with the importance of shading in linear space (and the issues involved in converting shading inputs from gamma space) to consult some of the articles on the topic [30, 34, 69].

When inspecting Table 1, several things stand out. One is that metals have significantly higher values of F_0 than non-metals. Even iron, a relatively dark metal, reflects more than 50% of incoming

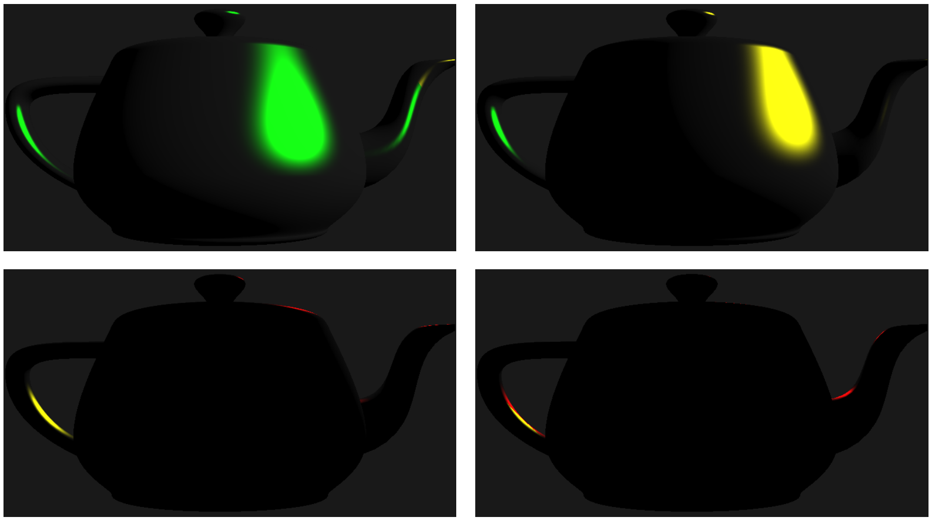

Figure 27: In this figure the Fresnel zone colors (as in Figure 26) are multiplied by the other parts of the microfacet BRDF (normal distribution function, geometry function, etc.). The light directions in the four images match those in Figure 26. We can see that when the complete BRDF is taken into account, the green zone predominates, and the red zone can only be seen at the extreme edges. The surface in this scene has moderate gloss—for glossier surfaces, this effect is even stronger.

Material	F_0 **(Linear)**	F_0 **(sRGB)**	**Color**
Water	0.02,0.02,0.02	0.15,0.15,0.15	⬛
Plastic / Glass (Low)	0.03,0.03,0.03	0.21,0.21,0.21	⬛
Plastic High	0.05,0.05,0.05	0.24,0.24,0.24	⬛
Glass (High) / Ruby	0.08,0.08,0.08	0.31,0.31,0.31	⬛
Diamond	0.17,0.17,0.17	0.45,0.45,0.45	⬛
Iron	0.56,0.57,0.58	0.77,0.78,0.78	⬜
Copper	0.95,0.64,0.54	0.98,0.82,0.76	🟧
Gold	1.00,0.71,0.29	1.00,0.86,0.57	🟨
Aluminum	0.91,0.92,0.92	0.96,0.96,0.97	⬜
Silver	0.95,0.93,0.88	0.98,0.97,0.95	⬜

Table 1: Values of F_0 for various materials. (*table from "Real-Time Rendering, 3rd edition" used with permission from A K Peters*).

light at 0°. Recall that metals have no sub-surface reflectance; a bright specular color and no diffuse color is the distinguishing visual characteristic of metals. On the other hand diamond, one of the brightest non-metals, reflects only 17% of incoming light at 0°; most non-metals reflect significantly less than that. Very few materials have values in the "no man's land" between 20% and 40%; these are typically semiconductors and other exotic materials which are unlikely to appear in production shading situations. The same is true for values lower than 2% (the F_0 value of water). In fact, except for metals, gemstones, and crystals, pretty much any material you are likely to see outside of a laboratory will have a narrow range of F_0 values—between 2% and 5%.

Normal Distribution Function

The microgeometry in most surfaces does not have uniform distributions of surface point orientations. More surface points have normals pointing "up" (towards the macroscopic surface normal $\mathbf{n}$) than "sideways" (away from $\mathbf{n}$). The statistical distribution of surface orientations is defined via the *microgeometry normal distribution function* $D(\mathbf{m})$. Unlike $F()$, the value of $D()$ is not restricted to lie between 0 and 1—although values must be non-negative, they can be arbitrarily large (indicating a very high concentration of surface points with normals pointing in a particular direction). Also, unlike $F()$, the function $D()$ is not spectral nor RGB-valued, but scalar-valued. In microfacet BRDF terms, $D()$ is evaluated for the direction $\mathbf{h}$, to help determine the concentration of potentially active surface points (those for which $\mathbf{m} = \mathbf{h}$). This is why the normal distribution function appears in Equation 4 as $D(\mathbf{h})$.

The function $D()$ determines the size, brightness, and shape of the specular highlight. Several different normal distribution functions appear in the graphics literature. Many are somewhat Gaussian-like, with some kind of "roughness" or variance parameter, whereas anisotropic functions typically have two roughness parameters (as do some recently published isotropic ones [4, 10, 51]). As surface roughness decreases, the concentration of microgeometry normals $\mathbf{m}$ around the overall surface normal $\mathbf{n}$ increases, and the values of $D(\mathbf{m})$ can become very high[13]. Walter et al. [73] discuss the correct normalization of the distribution function, and give several examples; more examples can be found in other papers [2, 3, 4, 10, 44, 51, 74]. Reed [62] gives another good explanation of the normal distribution function.

Geometry Function

The geometry function $G(\mathbf{l}, \mathbf{v}, \mathbf{h})$ represents the probability that surface points with a given microgeometry normal $\mathbf{m}$ will be visible from both the light direction $\mathbf{l}$ and the view direction $\mathbf{v}$. In the microfacet BRDF, $\mathbf{m}$ is replaced with $\mathbf{h}$ (for similar reasons as in the previous two terms). Since the function $G()$ represents a probability, its value is a scalar and constrained to lie between 0 and 1. As in the case of $D()$, there are various analytical expressions for $G()$ in the literature [2, 3, 15, 16, 41, 44, 73], which are usually approximations based on some simplified model of the surface. The geometry function typically does not introduce any new parameters to the BRDF; it either has no parameters, or uses the roughness parameter(s) of the $D()$ function. In many cases, the geometry function partially cancels out the $(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})$ denominator in Equation 4, replacing it with some other expression.

The geometry function is essential for BRDF energy conservation—without such a term the BRDF can reflect arbitrarily more light energy than it receives. A key part of the microfacet BRDF derivation relates to the ratio between the *active surface area* (the area covered by surface regions that reflect light energy from $\mathbf{l}$ to $\mathbf{v}$) and the *total surface area* of the macroscopic surface. If shadowing and masking are not accounted for, then the active area may exceed the total area, an obvious impossibility which can lead to the BRDF not conserving energy, in some cases by a huge amount (see Figure 28).

Limitations of the Microfacet Model

This formulation of microfacet theory is quite powerful and flexible, allowing for a large variety of different appearances by changing the parameter values (specular color, roughness of the normal distribution) or the forms of certain sub-terms (normal distribution function, geometry function). However, there are several phenomena that it does not model. If these are important, then modifications or extensions to the model may be required.

The microfacet model does not take account of pronounced wave optics effects such as diffraction and interference. This is not a large problem in practice since such effects do not occur often in

[13] In the limit, for a perfect mirror, the value is infinity at $\mathbf{m} = \mathbf{n}$.

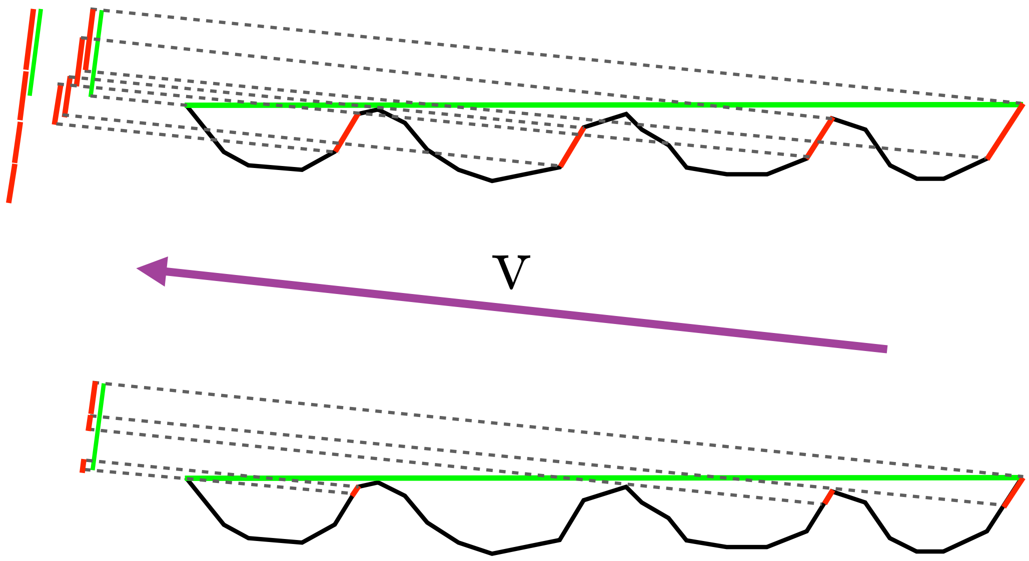

Figure 28: On the top, the flat macroscopic surface is shown in green, and the rugged microscopic surface is shown in black. Surface regions for which $\mathbf{m} = \mathbf{h}$ are marked in red. The projection of the macroscopic surface area (length in this 2D side illustration) onto the view direction (in other words, its foreshortened surface area) is shown as a green line on the upper left (the view direction itself is shown by the purple arrow in the middle of the figure). The projected areas of the individual red surface regions are shown as separate red lines, and on the upper extreme left we see that after adding them up, the projection of the active microgeometry area (long red line) is greater than the projection of the macroscopic area (green line). This is illogical, and more importantly can result in the BRDF reflecting more energy than it receives. On the bottom the red lines only show the visible portions of the active surface regions. The projection of the active microgeometry area is now less than that of the macroscopic area, which is correct. When the viewing angle is lower, then this effect will be even more pronounced—ignoring the effects of masking could lead to the BRDF reflecting thousands of times the amount of energy received or more (the amount of reflected energy would go to infinity in the limit as the angle goes to 90°).

production rendering. When they do, ad hoc techniques are typically used to address them, rather than physically based models. More subtle wave-optics effects can occur as a result of "smooth surface" features with sizes close to visible wavelengths, or larger surface features that become effectively smaller due to foreshortening effects at grazing angles [76]. Published models addressing such effects have typically been full wave-optics models [33, 68] that have not seen much production use due to their complexity. The development of more production-friendly models that address these effects would be welcome. Promising models from the field of optical engineering (e.g., [12]) have already started to influence the graphics literature [51].

In addition, the microfacet model is based on a relatively limited model of the surface microgeometry, with several unstated assumptions. For example, the definition of the normal distribution function assumes that the visible distribution of microgeometry surface orientations does not vary with the direction of observation. This is equivalent to assuming that microgeometry height and normal are uncorrelated [1]. However, this assumption is not always true, which can affect the BRDF. Imagine a surface which was originally uniformly rough but where the raised parts have been polished by friction. At glancing angles, only the raised parts will be visible, causing the surface to be effectively smoother than at other angles. See Figure 29.

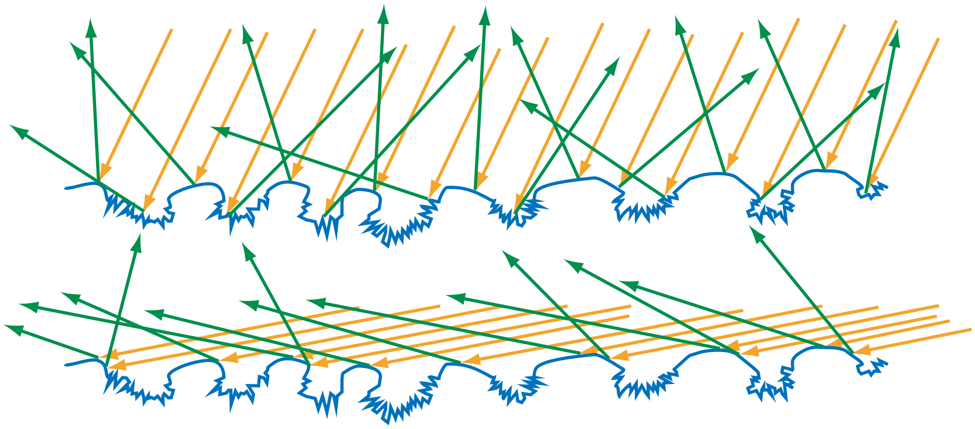

Figure 29: A surface with a strong correlation between height and orientation (raised areas are smooth, lower areas are rough). Top: the light direction is close to the macroscopic surface normal. Many of the light rays reach the rough pits, and are scattered in widely varying directions. Bottom: light comes from a glancing direction. The pits are occluded, so most rays are reflected from the smooth parts of the surface. This causes the effective normal distribution function to vary strongly with the angle of illumination. (*Image from "Real-Time Rendering, 3rd edition" used with permission from A K Peters.*)

Recent research has shown that alternative constructions [51] or modifications of the basic microfacet model [4] may yield better fits to measured data in some cases—most likely due to some combination of the effects discussed above. Further study is needed to understand these results.

Though eventually microfacet theory may need to be replaced (or at least extended) as a theoretical basis for modeling surface reflectance, currently it is the best-understood and most successful tool we have. For this reason, the remainder of these notes will focus on microfacet models.

Subsurface Reflectance (Diffuse Term)

There are several models for subsurface local reflection in the literature; the *Lambertian* model [49] is the simplest and one of the most widely used. The Lambertian BRDF is actually a constant value; the well-known cosine or $(\mathbf{n} \cdot \mathbf{l})$ factor is part of the reflectance equation, not the BRDF (as we saw in Equation 1). The exact value of the Lambertian BRDF is:

$$f_{\mathrm{Lambert}}(\mathbf{l}, \mathbf{v}) = \frac{\mathbf{c}_{\mathrm{diff}}}{\pi} \tag{7}$$

Here, $\mathbf{c}_{\mathrm{diff}}$ is the fraction of light that is diffusely reflected. As in the case of F_0, it is an RGB value with R, G, and B restricted to the $0-1$ range, and corresponds closely to what most people think of as a "surface color". This parameter is typically referred to as the *diffuse color*.

Other diffuse models attempt to address phenomena not modeled by the Lambertian model, such as the trade-off of energy between specular and diffuse terms at glancing angles. The diffuse term models subsurface reflection, which can only utilize incoming energy that was not reflected back at the surface. In a sense, the specular term gets "dibs" on the incoming light energy, and the diffuse term can only use its "leftovers". Since the Fresnel effect causes specular reflectance to increase at glancing angles, it follows that the diffuse term must decrease at those angles. There are various approaches to model this trade-off, from simple (multiplying the diffuse term by one minus the Fresnel factor[14]), to more complex and accurate approaches [2, 3, 41, 65].

[14] This causes the BRDF to have the form of a linear interpolation between diffuse and specular terms using part of the Schlick Fresnel term as the interpolation factor.

Other diffuse models attempt to account for the effect of surface roughness. It is important to understand the role that scale plays in this phenomenon. As we have seen, subsurface scattering causes light to travel some distance under the surface before being re-emitted. Any surface irregularity smaller than this distance will not have an effect on subsurface reflectance, since light being emitted from any surface point will have entered the surface from many points scattered over an area larger than the size of the irregularity. However, certain surfaces are rough on a scale larger than the scattering distance, and these exhibit appearance that notably differs from the Lambertian model. Various models have been developed to address these cases [32, 60]. It's a common misconception that these models are for rendering especially rough surfaces, but that's not true—the key criterion is not the *degree* of roughness but the *size* of the rough surface detail relative to the sub-surface scattering distance.

Other Terms

As mentioned earlier, there are two classes of reflectance phenomena that "fall between the cracks" of the phenomena modeled by the diffuse term (multiple-bounce subsurface scattering) and the microfacet term (single-bounce surface reflectance).

One is subsurface single-scattering, where light is refracted into the surface, undergoes a single scattering event and is then refracted back out of the surface. This phenomenon exhibits some of the characteristics of both diffuse and specular reflectance; it has high angular dependency, but it is affected by the properties of the medium underlying the object surface.

The other phenomenon is multiple-bounce surface reflectance, where light is reflected between multiple surface points before leaving the surface. As discussed earlier, microfacet models ignore multiple surface bounces—effectively assuming that all occluded rays are lost, causing a loss of energy compared to real world behavior.

It may be advisable to introduce additional BRDF terms to cover both of these cases, however there is a lack of good published models for them. It is notable that these phenomena will also affect common methods (such as polarimetry) for separating out surface from subsurface reflectance. For example, single-bounce subsurface reflectance largely preserves polarization, so "surface reflectance" measured via polarimetry will include some contribution from this phenomenon as well.

Implementing Physical Shading Models for Production

In the previous section, we saw the mathematical models that are typically employed to describe surface shading. In this section, we will briefly discuss how such models are used in film and game production.

To implement a shading model, it needs to be combined with an *illumination model*. We will cover the most common types of illumination models and how to combine BRDFs with them in the following sections.

General Lighting

In the most general case, the BRDF must be integrated against incoming light from all directions. This includes skylight and accurate reflections of other objects in the scene. To fully solve this, global illumination algorithms such as Monte Carlo ray tracing are required. Detailed descriptions of these algorithms are outside the domain of this talk, but more details can be found in various references [17, 24, 31, 42, 43].

Image-Based Lighting

Image-based lighting is typically stored in the form of environment maps representing distant lighting. Environment maps can easily represent reflections from very smooth (mirror-like) objects. Fresnel reflectance is modeled via Equation 5, modified by replacing the light vector with the view vector (the angle to the mirror normal remains the same so the two are equivalent). Difficulties may occur when the surface normal is back-facing to the view direction (which can result from interpolated vertex normals and/or bump mapping); this affects both the reflection direction and the Fresnel reflectance value, although incorrect reflection directions are rarely noticeable. As mentioned earlier, either a clamp to 0 or absolute value can be used to avoid negative dot products between the view and normal; in this specific case, taking the absolute value may be preferable. Unlike clamping, taking the absolute value will restrict Fresnel values associated with extreme glancing angles to a narrow band of pixels, which is more visually plausible.

Environment maps can also be used with arbitrary BRDFs, but an accurate result may require many samples to avoid noise. Importance sampling [14] helps to keep the number of samples to a somewhat more manageable number (at least for film rendering). Environment map prefiltering [39, 40] is another approach that can be effective in production, either by itself (i.e., performing a single prefiltered sample—a common technique in games but not typically used in film production), or in combination with importance sampling [13, 14]. While a single prefiltered sample is, in principle, a gross approximation to the reflectance equation, with care it can be quite effective. It is important to ensure that the prefiltering process accounts for the environment map representation used [36] as well as the shape and size of the NDF lobe [38, 46, 50]. Then, when the prefiltered sample is read, it should be multiplied by a factor which reasonably approximates the reflectance integral. This can be seen as factoring the integral into the product of two factors, the prefiltered environment map lookup and a second factor that is applied in the shader [22, 28, 38, 50].

In theory, an environment map can only be used for reflections from an object if it represents the scene (without the reflecting object) as seen from a point close to the object and the reflecting object is convex and distant from the reflected scene. Many of these assumptions are broken in practice. Non-convex objects should self-occlude the environment—this can be ignored (common in games), approximated with some simple occlusion term such as AO [29], or modeled more accurately by tracing rays against some representation of the object [22, 57, 67]. Reflected objects may also be close enough to have noticeable parallax over different surface points—especially common when the reflecting object is large. Finally, the environment map may be sampled from a different location in the scene, or even from a different scene entirely. It turns out that in many cases, the human eye is largely insensitive to the errors caused by parallax or use of environment maps away from their sample location. As long as the overall color and intensity is correct, the shapes being reflected can often be completely wrong without the viewer noticing. It is fairly straightforward to match the overall color and intensity of an environment map to local scene lighting [50], making environment maps an effective tool in many situations. Instances where incorrect reflections are noticeable (e.g. shiny floors, the player's car in a racing game, a shiny metallic hero character in a movie) can often be addressed by warping the reflections as a corrective step [47, 48, 67].

Area Light Sources

Light sources such as the sun and lamps have both intensity and area. In theory, they could be handled with a patch of HDR texels in an environment map, but there are advantages to treating them separately. It is easier to compute shadows from area light sources than for image-based lighting, parallax can be handled more correctly, and it is easier for artists to adjust light location, brightness and size without having to edit an image-based lighting representation [67].

Shading arbitrary BRDFs is also easier with area light sources than with image-based lighting; multiple importance sampling [72] can greatly reduce noise [54] (this technique can be used with image-based lighting as well, but is especially effective with area lights). Real-time approximations exist [22, 38], and are growing in popularity.

Punctual Light Sources

It is common (especially in games, though movies have used this as well) to approximate area light sources with *punctual light sources*. These are the classic computer graphics point, directional, and spot lights (more complex variants are also used [5]). Since they are infinitely small and infinitely bright, punctual lights aren't physically realizable or realistic, but they do produce reasonable results in many cases and are computationally convenient.

The effect of a punctual light source on a given surface point is defined by two quantities: the light color $\mathbf{c}_{\text{light}}$ and the light direction vector $\mathbf{l_c}$. Depending on the type of punctual light, these quantities may be constant over the scene (i.e. in the case of a global directional light) or may vary (e.g. for a point light $\mathbf{c}_{\text{light}}$ will decrease with distance and $\mathbf{l_c}$ will depend on the direction to the light location). For ease of visualization, $\mathbf{c}_{\text{light}}$ is commonly defined as reflected radiance from an idealized surface; i.e., a 100% white Lambertian surface with a normal parallel to the light direction vector ($\mathbf{n} = \mathbf{l_c}$). Like other color quantities we have seen, $\mathbf{c}_{\text{light}}$ is spectral (RGB)-valued, but unlike them its range is unbounded.

The primary advantage of punctual light sources is that they greatly simplify the reflectance equation (Equation 1), as we will show here. We will start with a surface point illuminated solely by a (very small) area light source. The light source is contained within a cone with an angular radius of ε centered on $\mathbf{l_c}$, so no light is incoming from directions outside this cone:

$$\forall \mathbf{l} \,|\, \angle(\mathbf{l}, \mathbf{l_c}) > \varepsilon, L_i(\mathbf{l}) = 0 \tag{8}$$

We measure the illumination of the area light source at the given surface point according to the definition of $\mathbf{c}_{\text{light}}$; in other words, $\mathbf{c}_{\text{light}}$ is equal to the reflected radiance the surface point would have, if its material were 100% white Lambertian and its normal were equal to $\mathbf{l_c}$:

$$\mathbf{c}_{\text{light}} = \frac{1}{\pi} \int_{\Omega} L_i(\mathbf{l})(\mathbf{l_c} \cdot \mathbf{l})\, d\omega_i \tag{9}$$

This is simply the result of Equations 1 and 7 with $\mathbf{c}_{\text{diff}} = 1$ and $\mathbf{n} = \mathbf{l_c}$. Now we will drive ε to 0, keeping the value of $\mathbf{c}_{\text{light}}$ constant. Since we are driving ε (the maximum angle between $\mathbf{l_c}$ and $\mathbf{l}$) to 0, in the limit we can assume $(\mathbf{l_c} \cdot \mathbf{l}) = 1$ which gives us:

$$\mathbf{c}_{\text{light}} = \lim_{\varepsilon \to 0} \left(\frac{1}{\pi} \int_{\Omega} L_i(\mathbf{l})\, d\omega_i \right) \tag{10}$$

Simple rearrangement isolates the value of the integral in the limit:

$$\lim_{\varepsilon \to 0} \left(\int_{\Omega} L_i(\mathbf{l})\, d\omega_i \right) = \pi \mathbf{c}_{\text{light}} \tag{11}$$

Now we shall apply our tiny area light to a general BRDF, and look at its behavior in the limit as ε goes to 0:

$$L_o(\mathbf{v}) = \lim_{\varepsilon \to 0} \left(\int_{\Omega} f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l})\, d\omega_i \right) = f(\mathbf{l_c}, \mathbf{v}) \otimes \lim_{\varepsilon \to 0} \left(\int_{\Omega} L_i(\mathbf{l})\, d\omega_i \right) (\mathbf{n} \cdot \mathbf{l_c}) \tag{12}$$

Note that in Equation 12, by isolating $f(\mathbf{l_c}, \mathbf{v})$ and $(\mathbf{n} \cdot \mathbf{l_c})$ we are effectively switching the order of the limit and integral; as long as the BRDF $f()$ is reasonably well-behaved (which would be true for any BRDF used in practice) this is valid. Substituting Equation 11 into the right part of Equation 12 gives us the final punctual light equation:

$$L_o(\mathbf{v}) = \pi f(\mathbf{l_c}, \mathbf{v}) \otimes \mathbf{c}_{\text{light}}(\mathbf{n} \cdot \mathbf{l_c}) \tag{13}$$

Compared to the original reflectance equation, we have replaced the integral with a single BRDF evaluation, which is *much* cheaper to compute. In games, it is common to clamp the dot product in this equation to 0 as a convenient method of skipping back-facing light contributions.

As mentioned above, in the case of directional light sources (such as the sun) both $\mathbf{l_c}$ and $\mathbf{c}_{\text{light}}$ are constant over the scene. In the case of other punctual light types such as point lights and spotlights, both will vary. In reality, $\mathbf{c}_{\text{light}}$ would fall off (decrease) proportionally to the inverse square of the distance to the light, but in practice other falloff functions are often used[15].

If multiple punctual light sources are illuminating the surface, Equation 13 is computed multiple times and the results summed. Punctual light sources are rarely used by themselves, since the lack of any illumination coming from other directions is noticeable, especially with highly specular surfaces. For this reason, punctual lights are typically combined with some kind of ambient or image-based lighting; the latter has already been discussed, and the former will be discussed in the next section.

Ambient Lighting

Here we define ambient lighting as some numerical representation of low-frequency lighting, ranging from a single constant light color and intensity over all incoming directions to more complex representations such as spherical harmonics (SH). Often this type of lighting environment is only applied to the diffuse BRDF term; higher-frequency, image-based lighting is applied to the specular term. However, it is possible to apply ambient lighting environments to the specular BRDF term as well. Most of the published methods for doing this originate from the games industry [11, 27, 64].

Building a Physically Based Shading Model

In this section, we will discuss building a model from scratch; for a discussion on converting a non-physical model to a physically based one, see our SIGGRAPH 2010 course presentation [35].

When building a physically based shading model according to the principles discussed in previous sections, there are several choices to be made. A diffuse model needs to be selected, as well as $D()$ (normal distribution function) and $G()$ (geometry function) for the microfacet specular model. The remainder of this section will focus on the two specular functions.

The choice of $D()$ and $G()$ functions is somewhat independent[16]. Most papers proposing a new microfacet BRDF model are best understood as introducing a new $D()$ and/or $G()$ function.

Choosing a Normal Distribution Function

The most common NDFs (normal distribution functions) are *isotropic*—they are rotationally symmetrical about the axis defined by the macroscopic surface normal $\mathbf{n}$. This means that these NDFs can be defined as a function of a single variable: the angle between $\mathbf{m}$ (the microgeometry normal) and $\mathbf{n}$. In shaders, it's most convenient to deal with angle cosines, since these can be easily calculated with dot products. For this reason, isotropic NDFs are typically written as functions of $\mathbf{n} \cdot \mathbf{m}$. Various such functions have been proposed in the literature for use as NDFs; however, they all must be properly *normalized* to be used as part of a microfacet BRDF. Several published anisotropic NDFs [2, 3, 74] have also seen use in game and film production, but for space reasons we will confine our discussion here to isotropic NDFs.

Any microgeometry normal distribution needs to fulfill the requirement that from any viewing angle, the visible microgeometry areas add up to the visible macrosurface area. More precisely, the

[15] For reasons of performance (a falloff function that goes to 0 at a finite distance enables culling light computations for distant objects) or artistic preference.

[16] An explanation of the qualifier "somewhat" can be found in the section "Choosing a Geometry Function".

sum of the signed projected areas of the microgeometry needs to equal the signed projected area of the macroscopic surface; this must hold true for any viewing direction [73]. Mathematically, this means that the function must satisfy this equation for any $\mathbf{v}$:

$$(\mathbf{v} \cdot \mathbf{n}) = \int_{\Theta} D(\mathbf{m})(\mathbf{v} \cdot \mathbf{m})\, d\omega_m \tag{14}$$

Note that the integral is over the entire sphere, not just the hemisphere, and the cosine factors are not clamped—back-facing surfaces have a negative contribution. This equation holds for any kind of microsurface, not just heightfields. In the special case, $\mathbf{v} = \mathbf{n}$:

$$1 = \int_{\Theta} D(\mathbf{m})(\mathbf{n} \cdot \mathbf{m})\, d\omega_m \tag{15}$$

For direct BRDF evaluation, it is desirable for the NDF to be cheap to evaluate, especially in games and other real-time rendering applications. For ray-tracing, ease of importance sampling is paramount.

All the microgeometry normal distributions we will review here were designed to be evaluated only for front-facing microgeometry normals (i.e., $\mathbf{n} \cdot \mathbf{m} \geq 0$—this can be interpreted as modeling heightfield surfaces, which never have more than 90° angles between the microgeometry normal and the macroscopic surface normal).

These functions are not intended to be evaluated for negative values of $\mathbf{n} \cdot \mathbf{m}$ and will typically produce undesirable results if thus evaluated. In renderers which enforce front-facing light and view directions, this is not an issue since the half-vector (for which the NDF is evaluated) will then be front-facing as well. Otherwise, normal interpolation and normal mapping may result in back-facing view vectors and thus back-facing half-vectors. This is typically addressed by clamping $\mathbf{n} \cdot \mathbf{m}$ to 0 or (if it appears in a denominator, to avoid dividing by zero) to a small positive epsilon value.

The Phong shading equation [61] is one of the earliest (and definitely the most influential) shading equations proposed in the computer graphics literature. It was modified by Blinn a few years later [7] to better fit the structure of a microfacet BRDF (this modification is commonly referred to as the Blinn-Phong BRDF, but we will refer to the resulting NDF simply as the "Phong NDF"). Although Blinn did not specify a normalization factor, it is easily computed:

$$D_p(\mathbf{m}) = \frac{\alpha_p + 2}{2\pi} (\mathbf{n} \cdot \mathbf{m})^{\alpha_p} \tag{16}$$

The power α_p is the "roughness parameter" of the Phong NDF; high values represent smooth surfaces and low values represent rough ones. Values can go arbitrarily high for very smooth surfaces (a perfect mirror would require $\alpha_p = \infty$) and a maximally random surface (uniform NDF) can be achieved by setting α_p to 0. The α_p parameter is not convenient for artists to manipulate nor paint directly since its visual impact is highly non-uniform. Small numerical changes have huge visual effects for small α_p values, but large values can be changed drastically without much visual impact. For this reason, it is common to have artists manipulate an "interface value" from which α_p is derived via a non-linear function. For example: $\alpha_p = m^s$, where s is an artist-manipulated value between 0 and 1 and m is an upper bound for α_p in a given show or game. Such "interface functions" are generally useful when the behavior of a BRDF native parameter is not convenient for production usage. Figure 30 shows Phong distribution curves for cosine powers evenly spaced according to a logarithmic scale.

In the same paper [7] in which Blinn adapted the Phong shading function into a microfacet NDF, he also proposed two other NDFs. One of these was derived from the Torrance-Sparrow BRDF [70]. When comparing the Torrance-Sparrow NDF to Phong, it appears to have very similar overall behavior at a much higher computational cost (see the Mathematica notebook accompanying these notes for details), so it's a bit of a dead end. Later work by Cook and Torrance [15, 16] proposed replacing it

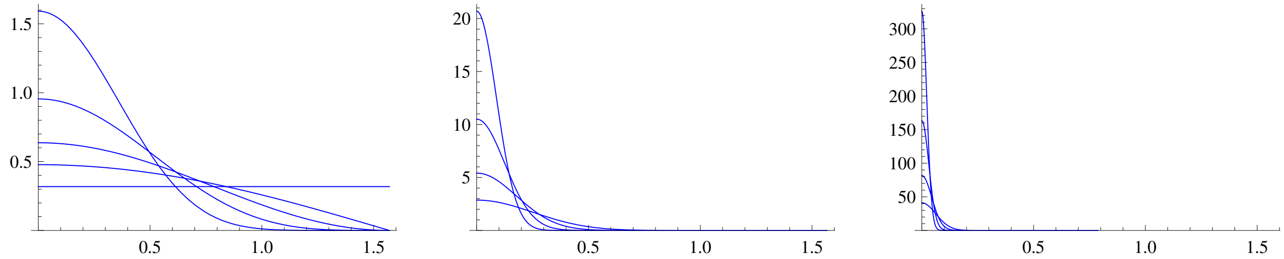

Figure 30: The Phong distribution with logarithmically spaced cosine powers. On the left, α_p (cosine power) values vary from 0 to 8. In the middle, they range from 16 to 128, and on the right they go from 256 to 2048.

with a different NDF, commonly referred to as the Beckmann distribution. When correctly normalized, the Beckmann distribution has the following form:

$$D_b(\mathbf{m}) = \frac{1}{\pi \alpha_b^2 (\mathbf{n} \cdot \mathbf{m})^4} e^{-\left(\dfrac{1-(\mathbf{n} \cdot \mathbf{m})^2}{\alpha_b^2 (\mathbf{n} \cdot \mathbf{m})^2}\right)} \tag{17}$$

The Beckmann distribution is very similar to the Phong distribution in some ways and fundamentally different from it in others. Equivalent values for the two parameters can be found using the relation $\alpha_p = 2\alpha_b^{-2} - 2$ [73], and they match quite well for relatively smooth surfaces ($\alpha_b < 0.5$ or so[17]), as can be seen on the left side of Figure 31. For even smoother surfaces ($\alpha_b < 0.1$ or so) the match is virtually perfect.

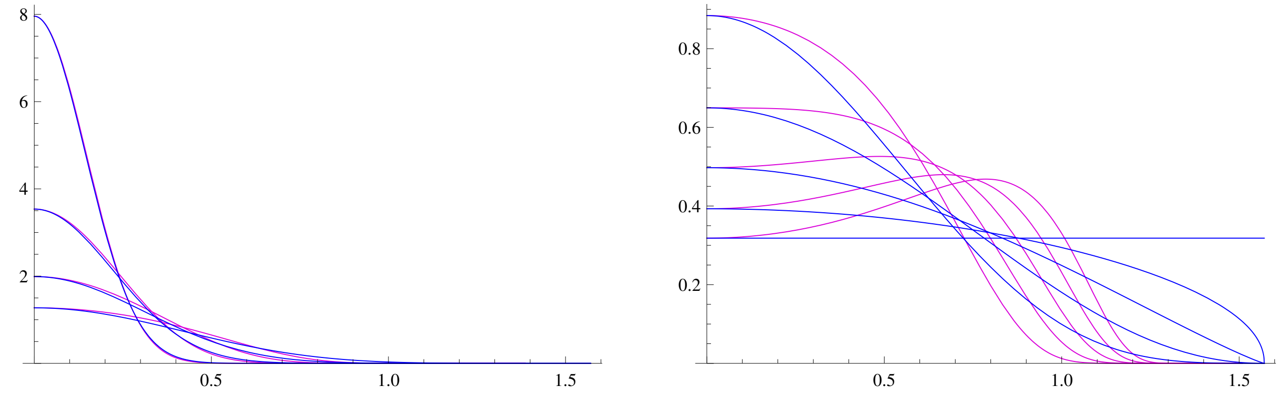

Figure 31: A comparison of Beckmann (magenta) and Phong (blue) distributions. On the left, we see that the two are very similar for smooth surfaces (values of α_b ranging from 0.2 to 0.5). On the right, we see that they diverge somewhat for moderately rough surfaces (values of α_b ranging from 0.6 to 1.0).

Given Beckmann's apparent strong similarity to Phong and its somewhat higher computational cost, it would seem to be a dead end as well. However, there is a fundamental difference between the two resulting from their different parameterizations. The α_b parameter is equal to the root mean square (RMS) slope of the microgeometry surface. So increasing α_b means increasing the average microgeometry slope, which is a different meaning of "roughness" than the "increasing randomness"

[17] Higher values of α_b correspond to *rougher* surfaces, which is the opposite of the α_p parameter.

that results from decreasing α_p. Whereas the Phong distribution has a "maximally rough" parameter value ($\alpha_p = 0$, corresponding to a uniform distribution where microgeometry normals have an equal probability of pointing anywhere in the upper hemisphere), the corresponding value ($\alpha_b = 1$) has no special meaning for Beckmann—it just means that the RMS microgeometry slope is 1, or 45°. For moderately rough values of α_b, we can see Beckmann getting a "dip" in the middle of the distribution instead of flattening out like Phong (see the right side of Figure 31). When α_b increases past 1 you get "super-rough" surfaces with high RMS microgeometry slopes—these are less random than a uniform distribution, but "rougher" in the sense that they are less flat. Looking at the curves (Figure 32), we can see that with increasing roughness the "dip" in the distribution turns into a "reverse peak" (actually a ring) at 90°.

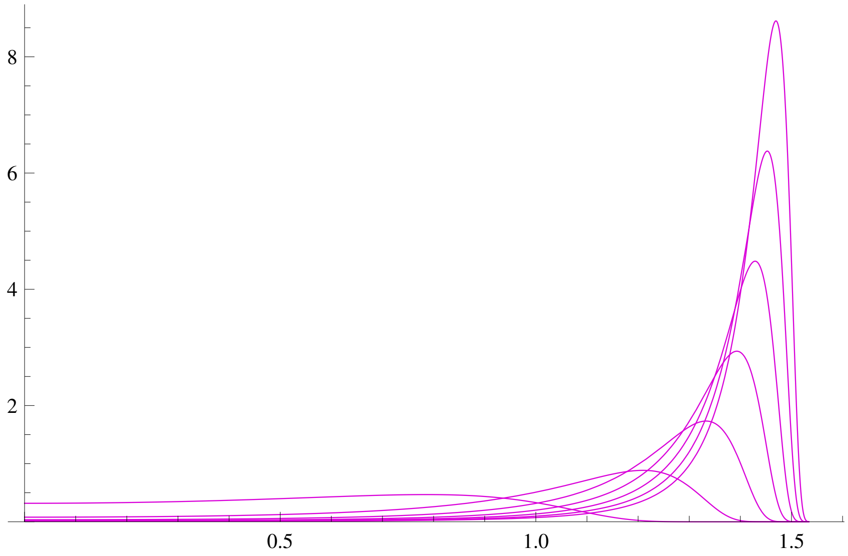

Figure 32: The Beckmann distribution can model "super-rough" surfaces with a majority of steep-sloped microgeometry (values of α_b in this plot range from 1 to 7).

Are such "super-rough" distributions useful for modeling any real-world surfaces? Perhaps, since a surface composed of many sharp upwards-facing fibers would have such a distribution and velvet microstructure appears to resemble this to some degree [1, 75]—other materials may as well. In any case, this behavior of the Beckmann distribution for high values of α_b is good to know, since otherwise it is basically a more expensive version of the Phong NDF.

The last NDF discussed in Blinn's paper [7] (and the one Blinn recommended using) is from Trowbridge and Reitz [71]. Blinn did not specify a normalization factor for the Trowbridge-Reitz NDF either, but a later paper [73]—which refers to it as "the GGX distribution"—does give the correct factor. Since the "GGX" denominator has a slightly more complex (but equivalent) form, we use the original, simpler form here:

$$D_{\mathrm{tr}}(\mathbf{m}) = \frac{\alpha_{\mathrm{tr}}^2}{\pi \left((\mathbf{n} \cdot \mathbf{m})^2 \left(\alpha_{\mathrm{tr}}^2 - 1 \right) + 1 \right)^2} \tag{18}$$

Figure 33 shows the distribution's behavior over a range of α_{tr} parameter values. The parameter control is similar to Beckmann's in that increasing the value makes the surface rougher. Trowbridge-Reitz can model a uniform distribution (like Phong) and also "super-rough" surfaces (like Beckmann).

When comparing the Trowbridge-Reitz distribution to the Phong distribution (Figure 34), it is apparent that the two distributions have fundamentally different shapes. Across the parameter space, Trowbridge-Reitz consistently has narrower peaks than Phong (for parameter values that yield the

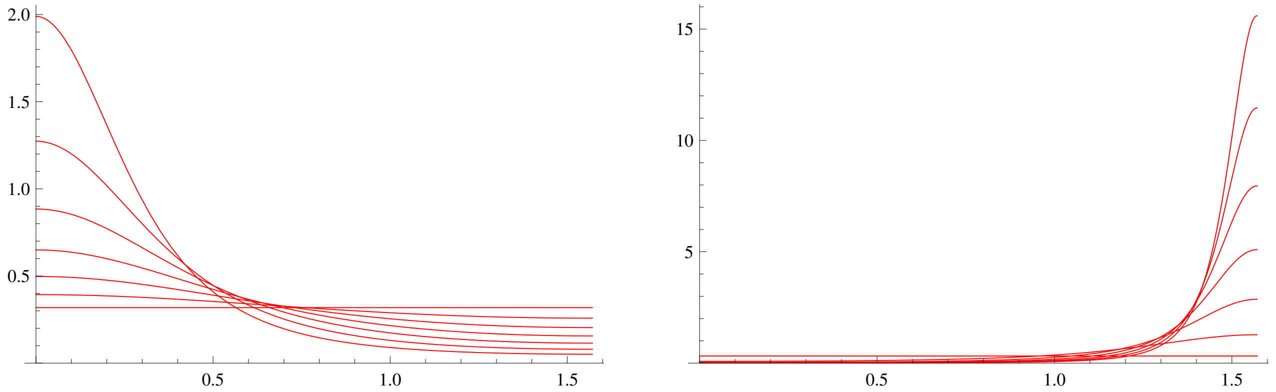

Figure 33: The left side shows the Trowbridge-Reitz distribution for $\alpha_{\rm tr}$ values from 0.4 to 1. This distribution behaves approximately like Beckmann with respect to its parameter: higher parameter values yield rougher surfaces. Unlike Beckmann, a value of 1 gives a uniform distribution. On the right, we see that like Beckmann, Trowbridge-Reitz displays "super-rough" behavior for higher parameter values (values of $\alpha_{\rm tr}$ in this plot range from 1 to 7).

same value at the highlight center[18]), as well as longer "tails" surrounding those peaks.

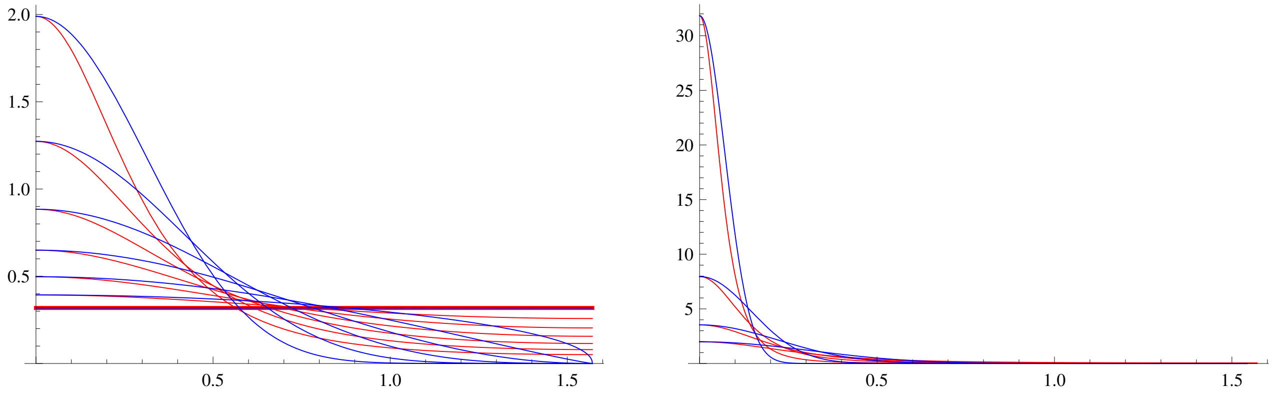

Figure 34: A comparison of Trowbridge-Reitz (red) and Phong (blue) distributions. The left graph shows rough to moderate surfaces ($\alpha_{\rm tr}$ values between 0.4 and 1.0). The right one shows smoother surfaces ($\alpha_{\rm tr}$ values between 0.1 and 0.4). In both plots, Trowbridge-Reitz clearly has a different shape than Phong, with a narrower "peak" and longer "tail".

Burley [10] suggests a simple "interface function" for Trowbridge-Reitz: $\alpha_{\rm tr} = s^2$, where s is an artist-manipulated value between 0 and 1.

Shading models need to perform well when modeling real-world surfaces. Several researchers [20, 53, 55] have published measured BRDF data [18, 19, 56] and others have compared this measured data (or their own measurements) against multiple shading models [10, 52, 76]. These comparisons tend to show that many materials are not well-modeled by any of the existing models. More recently, work has been done to develop new models specifically to provide a better match to measured surfaces [4, 10, 51]. We close our survey of NDFs with descriptions of three of the distributions proposed in this recent work.

[18] This follows the same relationship as the Beckmann distribution: $\alpha_p = 2\alpha_{tr}^{-2} - 2$.

Löw et al. [51] use the ABC distribution first introduced by Church et al. [12] in two different ways. The ABC distribution is used in its original role as the spectral power distribution of the surface heightfield (for smooth surface BRDFs that take wave effects into account) and is also repurposed as a microfacet NDF. The unnormalized distribution function has the following form:

$$D_{\mathrm{uabc}}(\mathbf{m}) = \frac{1}{\left(1 + \alpha_{\mathrm{abc}}\left(1 - (\mathbf{n}\cdot\mathbf{m})\right)\right)^{\gamma_{\mathrm{abc}}}} \tag{19}$$

which has two parameters, α_{abc} (corresponding to the "B" parameter in the original paper) and γ_{abc} (corresponding to the "C" parameter in the original paper—the "A" parameter is a scale factor and thus subsumed into the normalization). The authors did not publish a normalization factor, but I was able to generate one with *Mathematica*:

$$k_{\mathrm{abc}}(\alpha_{\mathrm{abc}}, \gamma_{\mathrm{abc}}) = \frac{\alpha_{\mathrm{abc}}^2 (1+\alpha_{\mathrm{abc}})^{\gamma_{\mathrm{abc}}}(\gamma_{\mathrm{abc}}-2)(\gamma_{\mathrm{abc}}-1)}{2\pi\left((1+\alpha_{\mathrm{abc}})^2 + (1+\alpha_{\mathrm{abc}})^{\gamma_{\mathrm{abc}}}(\alpha_{\mathrm{abc}}(\gamma_{\mathrm{abc}}-2)-1)\right)} \tag{20}$$

This function has singularities at $\gamma_{\mathrm{abc}} = 1$ and $\gamma_{\mathrm{abc}} = 2$. However, we can compute specific normalization factors for these values. These are the same as the limit values of the more general function in the neighborhood of these values, indicating that the singularities are removable:

$$k_{\mathrm{abc}}(\alpha_{\mathrm{abc}}, 1) = \frac{\alpha_{\mathrm{abc}}^2}{2\pi\left((1+\alpha_{\mathrm{abc}})\ln(1+\alpha_{\mathrm{abc}}) - \alpha_{\mathrm{abc}}\right)} \tag{21}$$

$$k_{\mathrm{abc}}(\alpha_{\mathrm{abc}}, 2) = \frac{\alpha_{\mathrm{abc}}^2}{2\pi\left(\alpha_{\mathrm{abc}} - \ln(1+\alpha_{\mathrm{abc}})\right)} \tag{22}$$

The normalized BRDF is equal to $k_{\mathrm{abc}} D_{\mathrm{uabc}}$. The expression for k_{abc} is somewhat complex (especially when the singularity cases are taken into account), but when plotted it yields smooth curves, making it likely that a much cheaper function could be fitted to it.

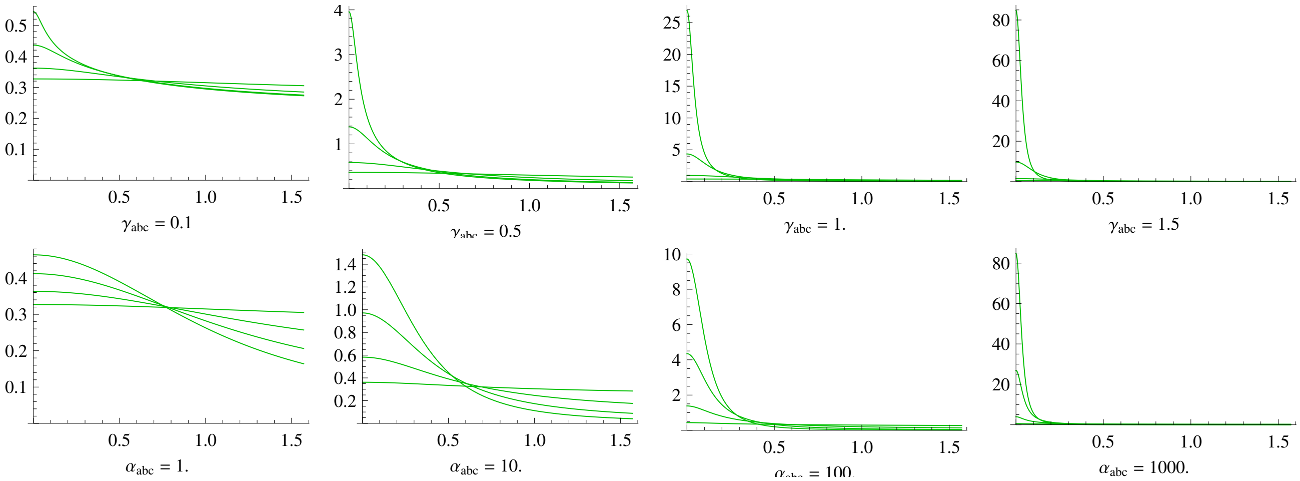

Figure 35: The normalized ABC NDF with various parameter values. Each plot on the top row varies the α_{abc} (B) parameter (from 1 to 1000) while keeping the γ_{abc} (C) value constant. Each plot on the bottom row varies the γ_{abc} (C) parameter (from 0.1 to 1.5) while keeping the α_{abc} (B) value constant. The range of parameter values used in this plot cover most of the Matusik dataset materials fitted by Löw et al. [51].

Figure 35 shows the ABC NDF with various values for both parameters. The NDFs we saw previously only had a single parameter; varying the two parameters of the ABC NDF enables it to form a variety of shapes, allowing for improved BRDF fitting to many different measured materials. Some experimentation shows that by changing the value of the γ_{abc} parameter, the ABC distribution can mimic both Trowbridge-Reitz and Phong (Figure 36). Furthermore, lowering the value of the γ_{abc} parameter results in curves that are even "spikier" (narrower peaks and longer tails) than Trowbridge-Reitz, as can be seen in Figure 37.

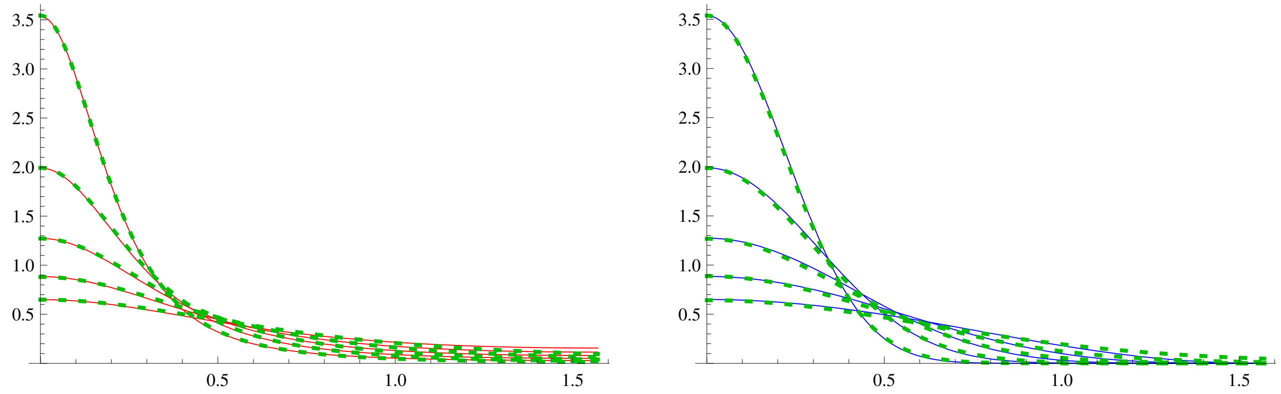

Figure 36: By changing the value of the γ_{abc} parameter, the ABC distribution (dotted green) is shown to match both Trowbridge-Reitz (red, on the left) and Phong (blue, on the right). The left plot shows Trowbridge-Reitz with α_{tr} values between 0.3 and 0.7, and ABC with a γ_{abc} value of 1.75 (values of α_{abc} have been manually adjusted to match the peaks of the corresponding Trowbridge-Reitz curves). The right one shows Phong with α_p values between about 2 and 20, and ABC with a γ_{abc} value of 1000 (ABC appears to asymptotically approach Phong when γ_{abc} goes to infinity; as with the left plot, α_{abc} values were manually adjusted). Although smoother surfaces are not shown, the correspondence holds for them as well. Note that the γ_{abc} values required to get a good fit to Phong are much higher than any used in the Matusik database fitting by Löw et al. [51]—this may indicate that real-world materials tend not to have Gaussian normal distributions.

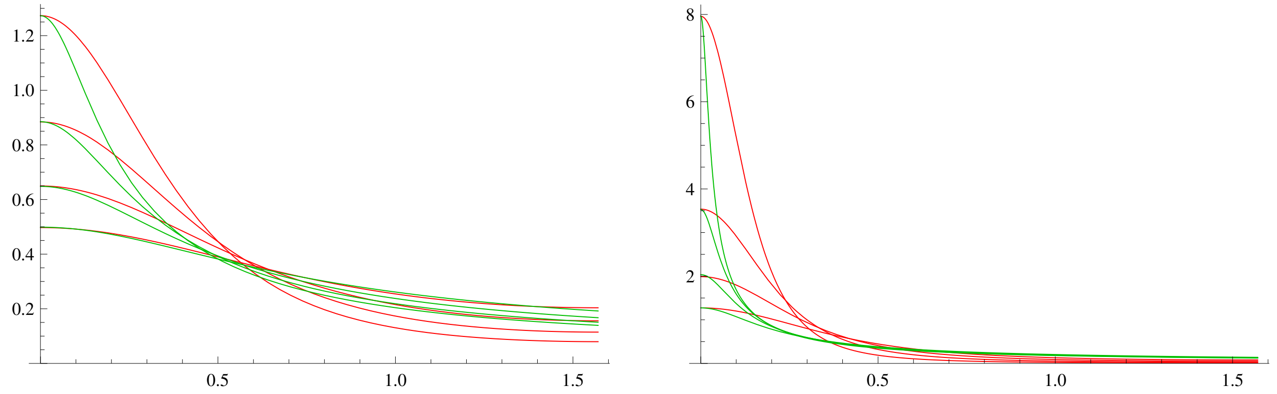

Figure 37: By setting the value of the γ_{abc} parameter to a relatively low value (0.5, which is lower than the values fit by Löw et al. [51] to all but a few matte materials out of the Matusik dataset), we see that the ABC distribution can achieve a "spikier" (narrower peak, longer tails) shape than Trowbridge-Reitz. Both plots compare ABC (green) to Trowbridge-Reitz (red). On the left, we see relatively rough surfaces (values of α_{tr} ranging between 0.5 and 0.8), and on the left, smoother materials (α_{tr} between 0.2 and 0.5). The difference in shape is more pronounced for the smoother surfaces.

A recent paper by Bagher et al. [4] proposed a shifted gamma distribution (SGD) as a microfacet NDF. The NDF has the following form:

$$D_{\text{sgd}}(\mathbf{m}) = \frac{\text{p22}\left[\frac{1-(\mathbf{n}\cdot\mathbf{m})^2}{(\mathbf{n}\cdot\mathbf{m})^2}\right]}{\pi(\mathbf{n}\cdot\mathbf{m})^4} \tag{23}$$

with p22$[x]$ defined thus:

$$\text{p22}[x] = \frac{\alpha_{sgd}^{\gamma_{sgd}-1}}{\Gamma\left(1-\gamma_{sgd},\alpha_{sgd}\right)}\,\frac{e^{-\frac{\alpha_{sgd}^2+x}{\alpha_{sgd}}}}{\left(\alpha_{sgd}^2+x\right)^{\gamma_{sgd}}} \tag{24}$$

where $\Gamma()$ is the incomplete Gamma function: $\Gamma(s,x) = \int_x^\infty t^{s-1}e^{-t}\,dt$. Note that the α_{sgd} and γ_{sgd} parameters are called α and p, respectively, in the original paper. Like the ABC NDF, SGD is isotropic and has two parameters, and was primarily designed to fit measured BRDFs such as the ones in the Matusik database [56]. However, the BRDF it is used in has some unusual qualities—there are separate parameters for the red, green and blue channels, and the "Fresnel" function is a generalized curve that often behaves quite differently from actual Fresnel [4]. For these reasons, it can be difficult to directly compare SGD with other NDFs, such as ABC. For example, we can see in Figure 38 that the SGD NDF quickly goes to 0 even for moderately smooth surfaces and cannot replicate the "long tails" that characterize other NDFs such as ABC; this may be compensated for by some of the unique properties of the BRDF in which it is used. In Figure 38 we also see that unlike ABC, the SGD NDF can model somewhat "super-rough" surfaces, though (as noted earlier) it's unclear how useful this feature is (also, the combination of parameters that produces this behavior is not found in the material fitting performed by the authors [4]). In any case, p22$[x]$ is extremely complex (few shading languages include $\Gamma()$ as a built-in function) but since it is one-dimensional, it is amenable to approximation by a simpler function or a look-up table.

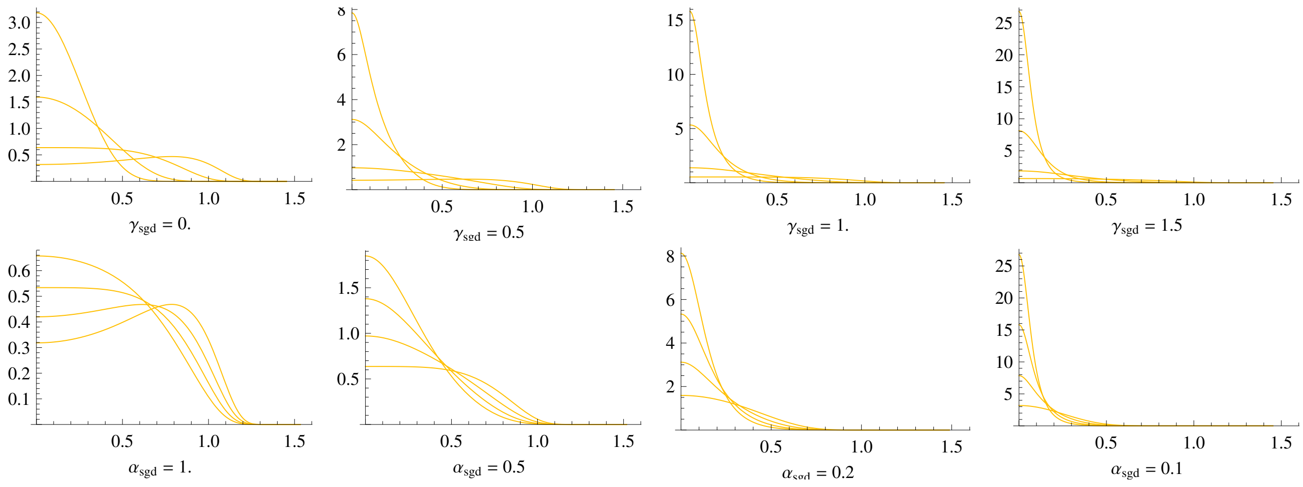

Figure 38: The normalized SGD NDF with various parameter values. Each plot on the top row varies the α_{sgd} (α in the paper) parameter (from 1 to 0.1) while keeping the γ_{sgd} (p in the paper) value constant. Each plot on the bottom row varies the γ_{sgd} (p) parameter (from 0.0 to 1.5) while keeping the α_{sgd} (α) value constant. The range of parameter values used in this plot cover the rough and moderately smooth materials in the Matusik dataset, as fitted by Bagher et al. [4].

The last NDF discussed in this section (and the most recently developed) is a generalized form of the Trowbridge-Reitz NDF proposed by Burley [10]. This proposal was inspired by the fact that the Berry distribution [6], one of the other distributions discussed in the original paper [71], has a very similar form to the Trowbridge-Reitz distribution but with an exponent of 1 instead of 2—resulting in an even longer tail. This led Burley to propose the Generalized-Trowbridge-Reitz (GTR) distribution, with a variable exponent. The unnormalized distribution function has the following form:

$$D_{\text{ugtr}}(\mathbf{m}) = \frac{1}{\left((\mathbf{n}\cdot\mathbf{m})^2\left(\alpha_{\text{gtr}}^2 - 1\right) + 1\right)^{\gamma_{\text{gtr}}}} \tag{25}$$

with two parameters, α_{gtr} (corresponding to the α_{tr} parameter in the original Trowbridge-Reitz distribution) and γ_{gtr}, which when equal to 2 makes the GTR distribution identical to TR, and with other values produces differently shaped distributions. Burley also gave a normalization factor, which in the general case is:

$$k_{\text{gtr}}(\alpha_{\text{gtr}}, \gamma_{\text{gtr}}) = \frac{(\gamma_{\text{gtr}} - 1)\left(\alpha_{\text{gtr}}^2 - 1\right)}{\pi\left(1 - \left(\alpha_{\text{gtr}}^2\right)^{(1-\gamma_{\text{gtr}})}\right)} \tag{26}$$

This function has singularities at $\gamma_{\text{gtr}} = 1$ and $\alpha_{\text{gtr}} = 1$. However (as with ABC) specific normalization factors can be computed for these values. Again, these are the same as the limit values of the general function in the neighborhood of these values, indicating that the singularities are removable:

$$k_{\text{gtr}}(\alpha_{\text{gtr}}, 1.0) = \frac{\left(\alpha_{\text{gtr}}^2 - 1\right)}{\pi \ln\left(\alpha_{\text{gtr}}^2\right)} \tag{27}$$

$$k_{\text{gtr}}(1.0, \gamma_{\text{gtr}}) = \frac{1}{\pi} \tag{28}$$

The normalized BRDF is equal to $k_{\text{gtr}} D_{\text{ugtr}}$. The expression for k_{gtr} is a bit complex (especially when the singularity cases are taken into account), but when plotted it yields smooth curves, making it likely (as for previously discussed NDFs) that a simpler function could be fitted to it.

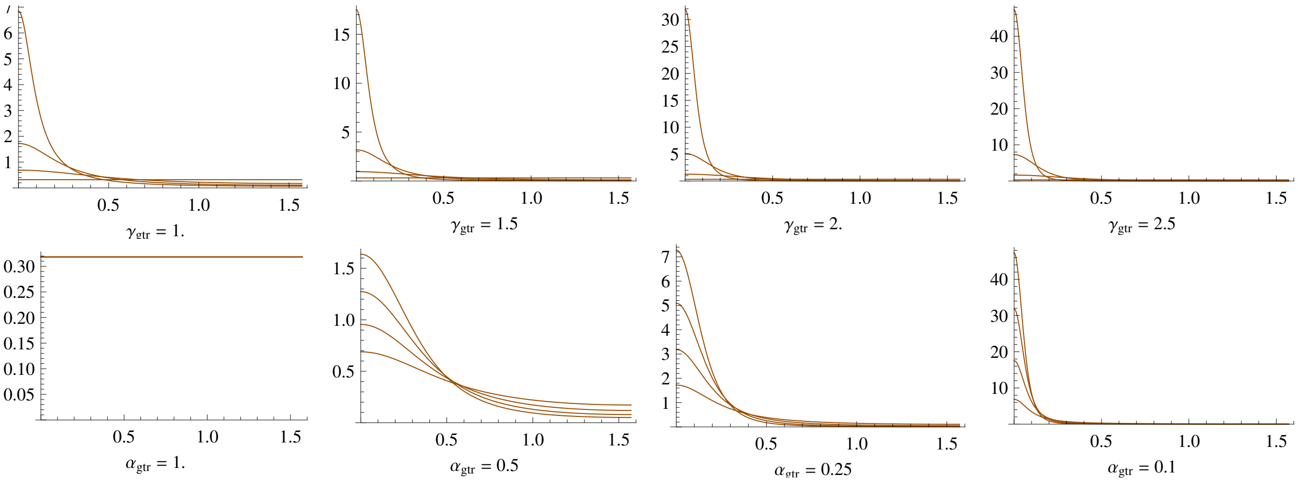

Figure 39: Normalized GTR NDF with various parameter values. Each plot on the top row varies the α_{gtr} parameter (with values of 1, 0.5, 0.25, and 0.1) while keeping the γ_{gtr} value constant. Each plot on the bottom row varies the γ_{gtr} parameter (with values of 1, 1.5, 2, and 2.5) while keeping the α_{gtr} value constant.

Figure 39 shows the GTR NDF with various values for both parameters. GTR can express a wide variety of NDF behaviors. GTR can of course trivially match Trowbridge-Reitz by setting $\gamma_{\text{gtr}} = 2$, and some experimentation shows that by increasing the value of the γ_{gtr} parameter, the GTR distribution can mimic Phong as well (Figure 40). GTR can also match the "spikier" versions of the ABC curve with $\gamma_{\text{abc}} = 0.5$ (seen previously in Figure 37), as Figure 41 demonstrates.

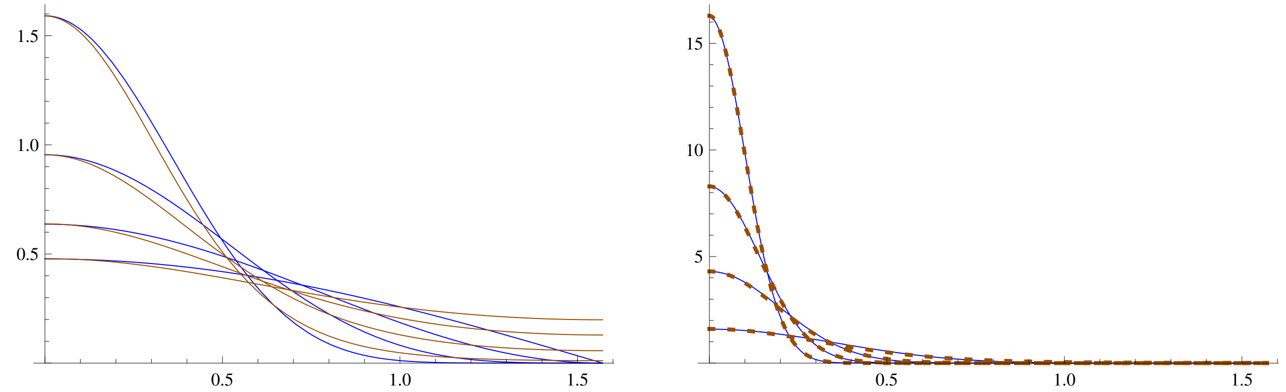

Figure 40: Like ABC, GTR appears to asymptotically approach Phong as the value of γ_{gtr} goes to infinity. Both plots show the GTR distribution with $\gamma_{\text{gtr}} = 10000$ (brown, solid on the left and dotted on the right) compared to the Phong distribution (blue). On the left, we see that the distributions diverge slightly for moderately rough surfaces (values of α_{p} between 1 and 8). On the right, we see that they match almost exactly for moderately smooth surfaces (values of α_{p} between 8 and 100—although smoother surfaces are not shown, the match holds for them as well). In both cases, values of α_{gtr} were manually adjusted to match the peaks of the corresponding Phong curves.

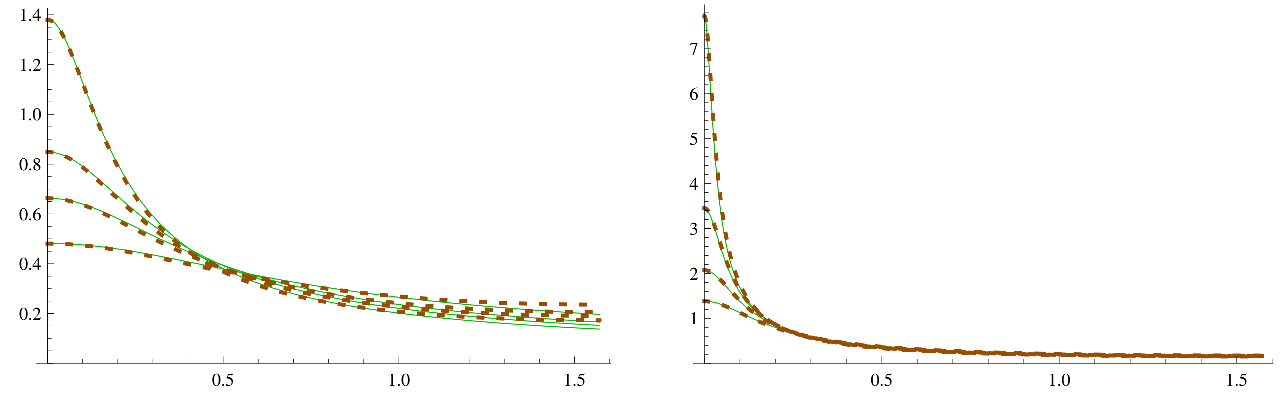

Figure 41: By setting the value of the γ_{gtr} parameter to 0.54, GTR (dotted brown) can achieve a fairly close match to the "spiky" curves of the ABC distribution with $\gamma_{\text{abc}} = 0.5$ (green). On the left, we see a reasonably good match for relatively rough surfaces (values of α_{abc} ranging between 5 and 100), and on the right, an almost exact match for smoother materials (α_{abc} between 100 and 4000). In both cases, values of α_{gtr} were manually adjusted to match the peaks of the corresponding ABC curves.

After listing all these NDFs, the question is which one to recommend. Often spatial variation of specularity is more important than the exact highlight shape at any given point. In these cases, it would likely be simplest to use the Phong NDF, which is computationally simple and reasonably expressive. If more realistic highlight shapes are desired, then Trowbridge-Reitz can be a good fit; its "sombrero" shape is likely to be a better match for real-world materials than the Gaussian Phong lobe. Trowbridge-Reitz has the further advantage of having been successfully used in film and game production [10, 38, 54, 59]. If a more expressive NDF is desired, then out of the two-parameter isotropic NDFs discussed here (ABC, SGD, and GTR) I would recommend using the GTR distribution. It is somewhat simpler than the other two, and has also been used in film production [10]. However, spatial variation of such NDFs can be difficult for artists to manage. One approach that could help is to reduce the dimensionality of the parameter space by defining a one-dimensional line through (α_{gtr}, γ_{gtr}) space—either based on artistic taste or material measurements—and exposing a parameterization of that line to artists. Another approach would be to only expose one of the parameters (probably α_{gtr}) for spatial variation via texture painting and only allow the other (most likely γ_{gtr}) to be set as a per-material constant. In this case a re-parameterization may be in order, to create a cleaner separation between "smoothness" and "shape". One possible approach could be to use statistical measures of probability distributions such as variance and kurtosis (which are likely to somewhat correspond visually to "smoothness" and "spikiness").

Choosing a Geometry Function

Many published microfacet BRDFs replace both the numerator term $G(\mathbf{l}, \mathbf{v}, \mathbf{h})$ and the denominator term $(\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})$ with a single subexpression, so it is useful to have a name for this; since the denominator can be thought of as a "foreshortening factor" and both parts are related to visibility, I like to call it the "visibility term". Some BRDFs (often those used in game or film production) have no visibility term at all; this is equivalent to setting the visibility term to 1, which implicitly defines the following geometry function:

$$G_{\text{implicit}}(\mathbf{l}, \mathbf{v}, \mathbf{m}) = (\mathbf{n} \cdot \mathbf{l_c})(\mathbf{n} \cdot \mathbf{v}) \tag{29}$$

This is actually a plausible geometry function for a heightfield microsurface (which is what the Blinn-Phong normal distribution function corresponds to, since it is zero for all back-facing microgeometry surface regions). $G_{\text{implicit}}()$ is equal to 1 when $\mathbf{l} = \mathbf{n}$ and $\mathbf{v} = \mathbf{n}$, which is correct for a heightfield (no microgeometry surface regions are occluded from the direction of the macrosurface normal). It goes to 0 for either glancing view angles or glancing light angles, which again is correct (the probability of a surface point being occluded by other surface regions increases with viewing angle, going to 100% in the limit). Given that this geometry function actually costs, in effect, less than zero shader cycles to compute (it cancels out the foreshortening factor so we don't need to divide by it), it has very good "bang per buck".

When comparing $G_{\text{implicit}}()$ to geometry functions from the graphics literature, we find that it goes to 0 too quickly—it is too dark at moderately glancing angles. In other words, adding an explicit geometry function will have the result of brightening the specular term (which may seem counter-intuitive, until we recall that we are also introducing the foreshortening factor at the same time). This implicit function is not affected by surface roughness, which is implausible—we would expect a rough surface to have higher shadowing and masking probabilities than a smooth one.

One of the earliest geometry functions in the graphics literature is referred to as "Cook-Torrance", and appeared in the well-known paper by those two authors [15, 16]:

$$G_{\text{ct}}(\mathbf{l}, \mathbf{v}, \mathbf{h}) = \min\left(1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{v})}{(\mathbf{v} \cdot \mathbf{h})}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{l})}{(\mathbf{v} \cdot \mathbf{h})}\right) \tag{30}$$

However, it first appeared several years earlier in a paper by Blinn [7], as a reformulation of a geometry function introduced earlier still by Torrance and Sparrow [70][19]. The Cook-Torrance geometry function has been used a lot over the years (especially in film), but it has some problems: it's based on an unrealistic microgeometry model (an isotropic surface composed of infinitely long grooves) and it's also unaffected by roughness.

[19] Thus it would be more accurate to refer to it as "the Blinn-Torrance-Sparrow geometry function" but the "Cook-Torrance" usage is too heavily established to be changed at this point.

In addition to its other issues, Cook-Torrance is a little on the expensive side for games. However, Kelemen et al. [41] proposed a very cheap and effective approximation for it:

$$\frac{G_{\text{ct}}(\mathbf{l},\mathbf{v},\mathbf{h})}{(\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})} \approx \frac{1}{(\mathbf{l}\cdot\mathbf{h})^2} \tag{31}$$

This is almost as cheap as the implicit geometry function; it approximates the Cook-Torrance geometry function as well as the division by the foreshortening factor and only requires division by the square of a dot product that needs to be computed anyway for the Fresnel term.

When considering accuracy over ultimate performance, I personally recommend the Smith family of geometry functions [66]. These are widely considered to be more accurate than the Cook-Torrance function, and take account of the roughness and shape of the normal distribution. The original Smith function was designed for the Beckmann NDF, but Brown [9] and later Bourlier et al. [8] generalized the Smith function into a method for computing a geometry function to match any NDF. Walter et al. [73] summarize these results and also give efficient approximations to the corresponding Smith functions for the Beckmann and Trowbridge-Reitz (GGX) NDFs[20] and Bagher et al. [4] give an approximation to the Smith function corresponding to their proposed SGD NDF.

The fact that the "Smith geometry function" is actually a family of functions is the motivation behind my earlier statement that the choice of geometry function is "somewhat" independent of the NDF. While choosing the *family* of Smith functions (as opposed to e.g. the Cook-Torrance geometry function) is a decision independent of the NDF used, once this choice is made you should use the correct Smith function for the NDF.

The Smith family of functions has been used to good effect in film production [10, 54], though the author of the former article recommends adjusting the roughness value used. The published approximations to the various Smith functions are still significantly more expensive than the Kelemen function, though it is likely that cheaper approximations could be found for games, in much the same way that the Kelemen function successfully approximates the (much more complex) Cook-Torrance geometry function[21].

Further Reading

Chapter 7 of the 3rd edition of "Real-Time Rendering" [58] provides a broad overview of physically based shading models, going into somewhat more depth than these course notes. For even greater depth, consider reading Glassner's *Principles of Digital Image Synthesis* [25, 26], or *Digital Modeling of Material Appearance* [21] by Dorsey, Rushmeier, and Sillion. Note that these books do not include the research results of the past few years.

Dutré's free online *Global Illumination Compendium* [23] is a useful reference for BRDFs, radiometric math, and much besides.

Finally, the other talks in this course (as well as its 2010 and 2012 predecessors) contain a lot of useful information on production use of physically based shading models.

Acknowledgments

Several thanks are in order. First, I would like to thank Stephen Hill for his considerable assistance in improving these course notes and the accompanying slides. Second, I thank Brent Burley, Paul

[20] Note that the Schlick approximation to the original Smith geometry function (which has been recommended in several places—sadly including my own book) is technically incorrect for use in microfacet BRDFs, since it approximates the wrong version of the function. However, it has been shown [38] that with some parameter remapping it can be used as an effective approximation of the correct function.

[21] A similar approximation for the Schlick-Smith geometry function is described in another talk in this course [50].

Edelstein, Yoshiharu Gotanda, Christophe Hery, Sébastien Lagarde, Dimitar Lazarov, Cedric Perthuis, and Brian Smits for many inspirational and thought-provoking conversations about physically based shading models. Finally, I thank A K Peters for permission to use images from the book *Real-Time Rendering, 3rd edition.*

Bibliography

[1] Ashikhmin, Michael, Simon Premože, and Peter Shirley, "A Microfacet-Based BRDF Generator," *Computer Graphics (SIGGRAPH 2000 Proceedings)*, pp. 67–74, July 2000. `http://www.cs.utah.edu/~michael/brdfs/`

[2] Ashikhmin, Michael, and Peter Shirley, "An Anisotropic Phong Light Reflection Model," Technical Report UUCS-00-014, Computer Science Department, University of Utah, June 2000. `http://www.cs.utah.edu/research/techreports/`

[3] Ashikhmin, Michael, Simon Premože, and Peter Shirley, "An Anisotropic Phong BRDF Model," *journal of graphics tools*, vol. 5, no. 2, pp. 25–32, 2000. `http://www.cs.utah.edu/~michael/brdfs/`

[4] Bagher, Mahdi M., Cyril Soler, Nicolas Holzschuch, "Accurate Fitting of Measured Reflectances using a Shifted Gamma Micro-facet Distribution," *Eurographics Symposium on Rendering (2012)*, 1509–1518, June 2012. `http://hal.inria.fr/hal-00702304/en`

[5] Barzel, Ronen, "Lighting Controls for Computer Cinematography" *journal of graphics tools*, vol. 2, no. 1, pp. 1–20, 1997.

[6] Berry, E. M., "Diffuse Reflection of Light from a Matte Surface," *Journal of the Optical Society of America*, vol. 7, no. 8, pp. 627, August 1923.

[7] Blinn, James F., "Models of Light Reflection for Computer Synthesized Pictures," *ACM Computer Graphics (SIGGRAPH '77 Proceedings)*, pp. 192–198, July 1977. `http://research.microsoft.com/apps/pubs/default.aspx?id=73852`

[8] Bourlier, Christophe, Gérard Berginc, and Joseph Saillard, "One- and Two-Dimensional Shadowing Functions for any Height and Slope Stationary Uncorrelated Surface in the Monostatic and Bistatic Configurations," *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 3, pp. 312–323, March 2002.

[9] Brown, Gary S., "Shadowing by Non-Gaussian Random Surfaces," *IEEE Transactions on Antennas and Propagation*, vol. 28, no. 6, pp. 788–790, November 1980.

[10] Burley, Brent, "Physically-Based Shading at Disney," part of "Practical Physically-Based Shading in Film and Game Production," *SIGGRAPH 2012 Course Notes*. `http://blog.selfshadow.com/publications/s2012-shading-course/`

[11] Chen, Hao, "Lighting and Material of Halo 3," *Game Developers Conference*, March 2008. `http://halo.bungie.net/inside/publications.aspx`

[12] Church, E.L., P.Z. Takacs, and T. A. Leonard, "The Prediction of BRDFs from Surface Profile Measurements," *Proceedings of SPIE*, vol. 1165., pp. 136–150, 1989.

[13] Colbert, Mark, and Jaroslav Krivanek, "GPU-based Importance Sampling," in Hubert Nguyen, ed., *GPU Gems 3*, Addison-Wesley, pp. 459–479, 2007. `http://http.developer.nvidia.com/GPUGems3/gpugems3_ch20.html`

[14] Colbert, Mark, Simon Premože, and Guillaume François, "Importance Sampling for Production Rendering," *SIGGRAPH 2010 Course Notes*. `http://sites.google.com/site/isrendering/`

[15] Cook, Robert L., and Kenneth E. Torrance, "A Reflectance Model for Computer Graphics," *Computer Graphics (SIGGRAPH '81 Proceedings)*, pp. 307–316, July 1981.

[16] Cook, Robert L., and Kenneth E. Torrance, "A Reflectance Model for Computer Graphics," *ACM Transactions on Graphics*, vol. 1, no. 1, pp. 7–24, January 1982. `http://graphics.pixar.com/library/ReflectanceModel/`

[17] Crassin, Cyril, Fabrice Neyret, Miguel Sainz, Simon Green, and Elmar Eisemann, "Interactive Indirect Illumination Using Voxel Cone Tracing," *Computer Graphics Forum* (Proceedings of Pacific Graphics 2011), vol. 30, no. 7, 2011. `http://maverick.inria.fr/Publications/2011/CNSGE11b/`

[18] Cornell University Program of Computer Graphics Reflectance Data. `http://www.graphics.cornell.edu/online/measurements/reflectance/index.html`

[19] Columbia-Utrecht Reflectance and Texture (CUReT) database. `http://www.cs.columbia.edu/CAVE/software/curet/`

[20] Dana, Kristin J, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink, "Reflectance and Texture of Real World Surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, January 1999. `http://www1.cs.columbia.edu/CAVE/publications/`

[21] Dorsey, Julie, Holly Rushmeier, and François Sillion, *Digital Modeling of Material Appearance*, Morgan Kaufmann, 2007.

[22] Drobot, Michal, "Lighting of Killzone: Shadow Fall," *Digital Dragons European Games Festival*, April 2013. `http://www.guerrilla-games.com/presentations/`

[23] Dutré, Philip, *Global Illumination Compendium*, 1999. `http://www.graphics.cornell.edu/~phil/GI`

[24] Dutré, Philip, Kavita Bala, and Philippe Bekaert, *Advanced Global Illumination*, second edition, A K Peters Ltd., 2006.

[25] Glassner, Andrew S., *Principles of Digital Image Synthesis*, vol. 1, Morgan Kaufmann, 1995.

[26] Glassner, Andrew S., *Principles of Digital Image Synthesis*, vol. 2, Morgan Kaufmann, 1995.

[27] Gotanda, Yoshiharu, "Practical Implementation of Physically-Based Shading Models at tri-Ace," part of "Physically Based Shading Models in Film and Game Production," *SIGGRAPH 2010 Course Notes*. `http://renderwonk.com/publications/s2010-shading-course/`

[28] Gotanda, Yoshiharu, "Real-time Physically Based Rendering—Implementation," *Computer Entertainment Developers Conference*, September 2011 `http://research.tri-ace.com/`

[29] Green, Paul, Jan Kautz, and Frédo Durand, "Efficient Reflectance and Visibility Approximations for Environment Map Rendering," *Computer Graphics Forum*, vol. 26, no. 3, pp. 495–502, 2007. `http://people.csail.mit.edu/green/`

[30] Gritz, Larry, and Eugene d'Eon, "The Importance of Being Linear," in Hubert Nguyen, ed., *GPU Gems 3*, Addison-Wesley, pp. 529–542, 2007. `http://http.developer.nvidia.com/GPUGems3/gpugems3_ch24.html`

[31] Hachisuka, Toshiya, Wojciech Jarosz , Guillaume Bouchard, Per H. Christensen, Jeppe Revall Frisvad, Wenzel Jakob, Henrik Wann Jensen, Michael Kaschalk, Matthias Zwicker, Andrew Selle, and Ben Spencer, "State of the Art in Photon Density Estimation," *SIGGRAPH 2012 Course Notes*. `http://users-cs.au.dk/toshiya/starpm2012/`

[32] Hapke, Bruce, "A Theoretical Photometric Function for the Lunar Surface," *J. Geophysical Research*, vol. 68, no. 15, August, 1963.

[33] He, Robert L., Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg, "A Comprehensive Physical Model for Light Reflection," *Computer Graphics (SIGGRAPH '91 Proceedings)*, pp. 175–186, July 1991.

[34] Hoffman, Naty, "Adventures with Gamma-Correct Rendering," Renderwonk Blog, August 3, 2007. `http://renderwonk.com/blog/index.php/archive/adventures-with-gamma-correct-rendering/`

[35] Hoffman, Naty, "Crafting Physically Motivated Shading Models for Game Development," part of "Physically Based Shading Models in Film and Game Production," *SIGGRAPH 2010 Course Notes*. `http://renderwonk.com/publications/s2010-shading-course/`

[36] Isidoro, John R., and Jason L. Mitchell, "Angular Extent Filtering with Edge Fixup for Seamless Cubemap Filtering," *SIGGRAPH 2005 Sketches.* http://developer.amd.com/wordpress/media/2012/10/Isidoro-CubeMapFiltering-Sketch-SIG05.pdf

[37] Kajiya, James T., "The Rendering Equation," *Computer Graphics (SIGGRAPH '86 Proceedings)*, pp. 143–150, August 1986. http://www.cs.brown.edu/courses/cs224/papers/kajiya.pdf

[38] Karis, Brian, "Real Shading in Unreal Engine 4," part of "Physically Based Shading in Theory and Practice," *SIGGRAPH 2013 Course Notes.*

[39] Kautz, Jan, and Michael D. McCool, "Approximation of Glossy Reflection with Prefiltered Environment Maps," *Graphics Interface (2000)*, 119–126, May 2000. http://www0.cs.ucl.ac.uk/staff/j.kautz/publications/

[40] Kautz, Jan, Pere-Pau Vázquez, Wolfgang Heidrich, and Hans-Peter Seidel, "A Unified Approach to Prefiltered Environment Maps," *Eurographics Workshop on Rendering (2000)*, 185–196, June 2000. http://www0.cs.ucl.ac.uk/staff/j.kautz/publications/

[41] Kelemen, Csaba, and Lázló Szirmay-Kalos, "A Microfacet Based Coupled Specular-Matte BRDF Model with Importance Sampling," *Eurographics 2001*, short presentation, pp. 25–34, September 2001. http://www.fsz.bme.hu/~szirmay/scook_link.htm

[42] Keller, Alexander, Simon Premože, Matthias Raab, and Leonhard Gruenschloss, "Advanced (Quasi-) Monte Carlo Methods for Image Synthesis," *SIGGRAPH 2012 Course Notes.* https://sites.google.com/site/qmcrendering/

[43] Křivánek, Jaroslav, Marcos Fajardo, Per H. Christensen, Eric Tabellion, Michael Bunnell, David Larsson, and Anton Kaplanyan, "Global Illumination Across Industries," *SIGGRAPH 2010 Course Notes.* http://www.graphics.cornell.edu/~jaroslav/gicourse2010/

[44] Kurt, Murat, László Szirmay-Kalos, and Jaroslav Křivánek, "An Anisotropic BRDF Model for Fitting and Monte Carlo Rendering," *Computer Graphics*, vol. 44, no. 1, pp. 1–15, 2010. http://www.graphics.cornell.edu/~jaroslav/

[45] Lagarde, Sébastien, "Spherical Gaussian approximation for Blinn-Phong, Phong and Fresnel," *Random Thoughts about Graphics in Games* blog, June 3, 2012. http://seblagarde.wordpress.com/2012/06/03/spherical-gaussien-approximation-for-blinn-phong-phong-and-fresnel/

[46] Lagarde, Sébastien, "AMD Cubemapgen for physically based rendering," *Random Thoughts about Graphics in Games* blog, June 10, 2012. http://seblagarde.wordpress.com/2012/06/10/amd-cubemapgen-for-physically-based-rendering/

[47] Lagarde, Sébastien, and Antoine Zanuttini, "Local Image-Based Lighting With Parallax-Corrected Cubemaps," *SIGGRAPH 2012 Talk.* http://seblagarde.wordpress.com/2012/08/11/siggraph-2012-talk/

[48] Lagarde, Sébastien, and Antoine Zanuttini, "Practical Planar Reflections Using Cubemaps and Image Proxies," in Wolfgang Engel, ed., *GPU Pro 4*, A K Peters, pp. 51–68, 2013.

[49] Lambert, Johann H., "Photometria Sive de Mensure de Gratibus Luminis, Colorum Umbrae," Eberhard Klett, 1760.

[50] Lazarov, Dimitar, "Getting More Physical in *Call of Duty: Black Ops II*," part of "Physically Based Shading in Theory and Practice," *SIGGRAPH 2013 Course Notes.*

[51] Löw, Joakim, Joel Kronander, Anders Ynnerman, and Jonas Unger, "BRDF Models for Accurate and Efficient Rendering of Glossy Surfaces," *ACM Transactions on Graphics*, vol. 31, no. 1, pp 9:1–9:14, January 2012 http://vcl.itn.liu.se/publications/2012/LKYU12/

[52] Ngan, Addy, Frédo Durand, and Wojciech Matusik, "Experimental Analysis of BRDF Models," *Eurographics Symposium on Rendering (2005)*, 117–226, June 2005. http://people.csail.mit.edu/addy/research/brdf/

[53] Marschner, Stephen R., Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg, "Image-Based BRDF Measurement Including Human Skin," *Eurographics Workshop on Rendering (1999)*, 139–152, June 1999. `http://www.cs.cornell.edu/~srm/publications/egrw99-brdf-abstract.html`

[54] Martinez, Adam, "Faster Photorealism in Wonderland: Physically-Based Shading and Lighting at Sony Pictures Imageworks," part of "Physically Based Shading Models in Film and Game Production," *SIGGRAPH 2010 Course Notes.* `http://renderwonk.com/publications/s2010-shading-course/`

[55] Matusik, Wojciech, Hanspeter Pfister, Matt Brand and Leonard McMillan, "A Data-Driven Reflectance Model," *ACM Transactions on Graphics*, vol. 22, no. 3, pp 759–769, July 2003 `http://people.csail.mit.edu/wojciech/DDRM/index.html`

[56] MERL BRDF database. `http://www.merl.com/brdf/`

[57] Mittring, Martin, "The Technology Behind the *Unreal Engine 4 Elemental Demo*," part of "Advances in Real-Time Rendering in 3D Graphics and Games," *SIGGRAPH 2012 Course Notes.*

[58] Akenine-Möller, Tomas, Eric Haines, and Naty Hoffman, *Real-Time Rendering*, third edition, A K Peters Ltd., 2008. `http://realtimerendering.com/`

[59] Neubelt, David, and Matt Pettineo, "Crafting a Next-Gen Material Pipeline for The Order: 1886," part of "Physically Based Shading in Theory and Practice," *SIGGRAPH 2013 Course Notes.*

[60] Oren, Michael, and Shree K. Nayar, "Generalization of Lambert's Reflectance Model," *Computer Graphics (SIGGRAPH 94 Proceedings)*, pp. 239–246, July 1994. `http://www.cs.columbia.edu/CAVE/projects/oren/`

[61] Phong, Bui Tuong, "Illumination for Computer Generated Pictures," *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, June 1975. `http://jesper.kalliope.org/blog/library/p311-phong.pdf`

[62] Reed, Nathan, "How Is the NDF Really Defined?," Nathan Reed's Coding Blog, July 31, 2013. `http://www.reedbeta.com/blog/2013/07/31/hows-the-ndf-really-defined/`

[63] Schlick, Christophe, "An Inexpensive BRDF Model for Physically based Rendering," *Computer Graphics Forum*, vol. 13, no. 3, Sept. 1994, pp. 149–162. `http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/eur2.html`

[64] Schüler, Christian, "An Efficient and Physically Plausible Real Time Shading Model," in Wolfgang Engel, ed., *ShaderX*^{7}, Charles River Media, pp. 175–187, 2009.

[65] Shirley, Peter, Helen Hu, Brian Smits, Eric Lafortune, "A Practitioners' Assessment of Light Reflection Models," *Pacific Graphics '97*, pp. 40–49, October 1997. `http://www.graphics.cornell.edu/pubs/1997/SHSL97.html`

[66] Smith, Bruce G., "Geometrical Shadowing of a Random Rough Surface," *IEEE Transactions on Antennas and Propagation*, vol. 15, no. 5, pp. 668–671, September 1967.

[67] Snow, Ben, "Terminators and Iron Men: Image-Based Lighting and Physical Shading at ILM," part of "Physically Based Shading Models in Film and Game Production," *SIGGRAPH 2010 Course Notes.* `http://renderwonk.com/publications/s2010-shading-course/`

[68] Stam, Jos, "Diffraction Shaders," *Computer Graphics (SIGGRAPH 99 Proceedings)*, pp. 101–110, August 1999. `http://www.dgp.toronto.edu/~stam/reality/Research/Diffraction/index.html`

[69] Tchou, Chris, "HDR The Bungie Way," *Gamefest*, August 2006. `http://microsoftgamefest.com/2006.htm`

[70] Torrance, Kenneth E., and Ephraim M. Sparrow, "Theory for Off-Specular Reflection From Roughened Surfaces," *Journal of the Optical Society of America*, vol. 57, no. 9, pp. 1105–1114, September 1967. `http://www.graphics.cornell.edu/~westin/index.html#educational`

[71] Trowbridge, T. S., and K. P. Reitz, "Average Irregularity Representation of a Roughened Surface for Ray Reflection," *Journal of the Optical Society of America*, vol. 65, no. 5, pp. 531–536, May 1975.

[72] Veach, Eric, “Robust Monte Carlo Methods for Light Transport Simulation,” Ph.D. thesis, Stanford University (1997). `http://graphics.stanford.edu/papers/veach_thesis/`

[73] Walter, Bruce, Stephen R. Marschner, Hongsong Li, Kenneth E. Torrance, “Microfacet Models for Refraction through Rough Surfaces,” *Eurographics Symposium on Rendering (2007)*, 195–206, June 2007. `http://www.cs.cornell.edu/~srm/publications/EGSR07-btdf.html`

[74] Ward, Gregory, “Measuring and Modeling Anisotropic Reflection,” *Computer Graphics (SIGGRAPH ’92 Proceedings)*, pp. 265–272, July 1992. `http://radsite.lbl.gov/radiance/papers/sg92/paper.html`

[75] Westin, Stephen H., James R. Arvo, and Kenneth E. Torrance, “Predicting Reflectance Functions from Complex Surfaces,” *Computer Graphics (SIGGRAPH ’92 Proceedings)*, pp. 255–264, July 1992. `http://www.graphics.cornell.edu/pubs/1992/WAT92.html`

[76] Westin, Stephen H., Hongsong Li, and Kenneth E. Torrance, “A Comparison of Four BRDF Models,” Technical Report PCG-04-2, Program of Computer Graphics, Cornell University, April 2004. `http://www.graphics.cornell.edu/pubs/2004/WLT04a.html`