<< RETURN

Advanced Real-Time Reflectance

Dan Baker
Microsoft
danbaker@windows.microsoft.com

Naty Hoffman
Naughty Dog, Inc.
naty@naughtydog.com

Peter-Pike Sloan
Microsoft
ppsloan@windows.microsoft.com

Introduction

One of the primary goals in graphics is generating compelling images of real objects. There are three big challenges that must be overcome to achieve compelling images. First, real objects exist in general lighting environments (area lighting). Second, the light from the environment can be shadowed and reflected by the objects in the scene, e.g. global illumination. Finally, the surface material properties in the scene all have to be represented and modeled. This paper strictly deals with the last category – how to represent a broad class of surface materials. Addressing one of these challenges to the exclusion of others will not result in realistic images. For example, extremely realistic reflectance properties on an object lit by a single directional light with no shadows or interreflections will not look realistic. Techniques that address several of these challenges will be briefly mentioned later in this paper.

One of the main mathematical tools for describing surface materials is called the *Bidirectional Reflectance Distribution Function* (BRDF). A BRDF models how a material reflects light, and though it has several simplifying assumptions, it can represent a broad class of materials. For example, wood is a material, representable by a BRDF, in which reflectance is not homogenous since it varies over the surface due to pigment, grain and possibly the amount of varnish. Additionally, varnished wood exhibits strong reflection of the environment at grazing angles and almost none at incident angles, i.e. looking at the surface head on. Some materials like plastic are *isotropic*, meaning that there is no visible grain on the surface. Other materials having a visible grain, such as hair and brushed metal, are termed *anisotropic*.

This paper accompanies the GDC lecture “Advanced Real-Time Reflectance”. It will describe some of the underlying physics and mathematics behind reflection models, detail the limitations of the models and describe several powerful building blocks that can be used to describe a large set of materials. Finally, some implementation considerations will be discussed as well.

Measuring Light

Computer graphics is fundamentally about light and its interaction with objects. Light is electromagnetic radiant energy (we concern ourselves with the visible part of the spectrum - wavelengths from 380 to 780 nanometers). In this paper and lecture, we will stick strictly to *geometric optics*, which treats light as linear rays (as opposed to *physical optics*, which treats light as waves, or *quantum optics*, which treats light as dual waves/particles). This works well for most scenes, in which the scale of objects is much larger than light wavelengths (those interested in the exceptions can look at research [He 1991; Stam 1999] using physical optics to model effects such as pearlescence, diffraction patterns such as on optical discs, etc.). Also, we ignore light polarization (which can be visually important in some scenes, such as the sky reflecting off glass).

The physical measurement and study of light is *radiometry*. The fundamental quantity in radiometry is power (energy over time), which is *radiant flux* (Φ), measured in Watts. For example, the light pouring through a window (in all directions) would be measured as radiant flux. *Radiosity* (B) is the light *exitant* (going out of) a single point (in all directions), and is measured in Watts per square meter. *Radiance* (L) is the light in a single ray, and is measured in Watts per square meter[1] per steradian[2]. *Irradiance* (E) can be seen as the opposite of radiosity – it is the light going into and illuminating a surface point. It includes light hitting that point from all *incident* (incoming) directions. It is also measured in Watts per square meter.

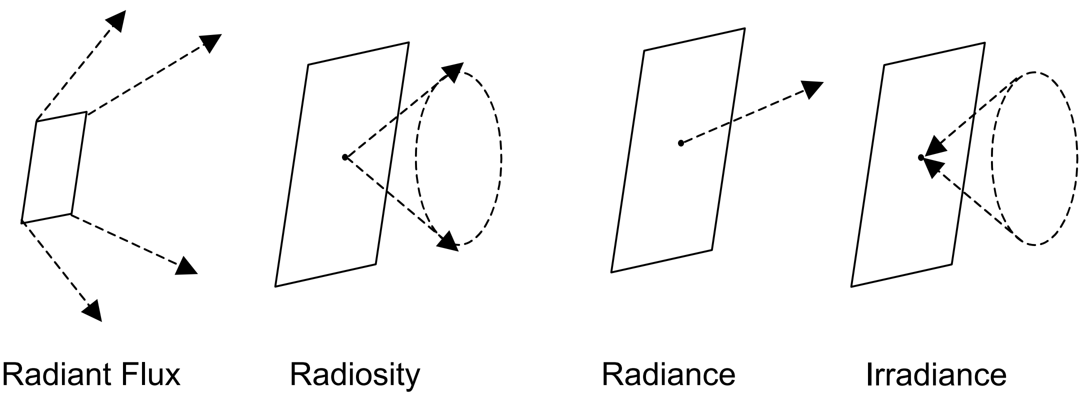

Figure 1. Radiometric quantities

Note that although the radiosity and irradiance diagrams in figure 1 show a cone of directions for clarity, they are measured over a full hemisphere (centered on the surface normal).

[1] Unlike radiosity and irradiance which are defined in terms of area measured relative to a given surface, radiance is defined in terms of *projected area* measured perpendicularly to the ray direction.
[2] The steradian is a unit of solid angle, which is a 3D extension of the 2D concept of an angle. Like an angle, a solid angle is a continuous range of directions. As a 2D angle can be thought of as the length of an arc on a (unit) circle, measured in radians (of which there are 2π in a full circle), equivalently a solid angle can be thought of as the area of a patch on a (unit) sphere, measured in steradians (of which there are 4π in a full sphere).

An important relationship between radiance and irradiance is $E = \int_{\Omega} L_i \cos\theta_i d\omega_i$ or

$dE = L_i \cos\theta_i d\omega_i$ (see figure 2).

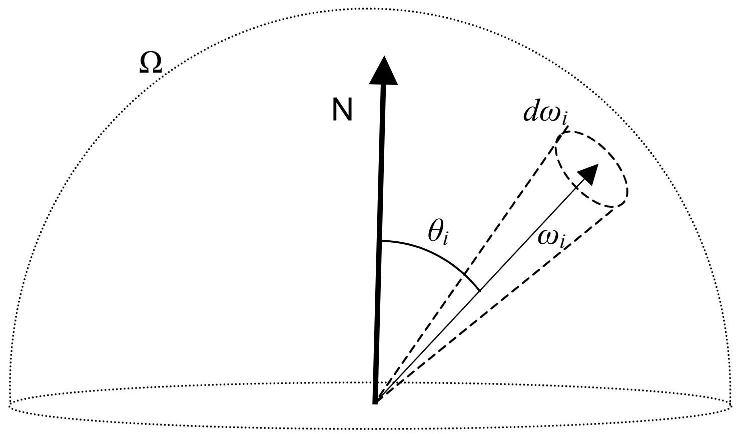

Figure 2. Radiance and irradiance

This means that if we take a tiny patch of incident directions with solid angle $d\omega_i$ (small enough to be represented by a single incident direction ω_i without loss of accuracy), the radiance incident from ω_i, times $d\omega_i$, times the cosine of the angle θ_i between ω_i and the normal N gives us the contribution of that patch to the irradiance. If we integrate this over the hemisphere Ω (centered on the normal), we get the total irradiance. The cosine is there because radiance is defined relative to an area perpendicular to the ray, and irradiance is defined relative to an area parallel to the surface. Another way of looking at it is that the same irradiance, coming in at a more oblique angle, contributes a smaller amount to the irradiance because it is 'spread out' more. Note that although we talk about radiance incident from ω_i, the direction actually points outwards. This can be a little confusing but it is established usage in the computer graphics community – it might be clearer to think of ω_i as the direction in which the point "sees" the incident light.

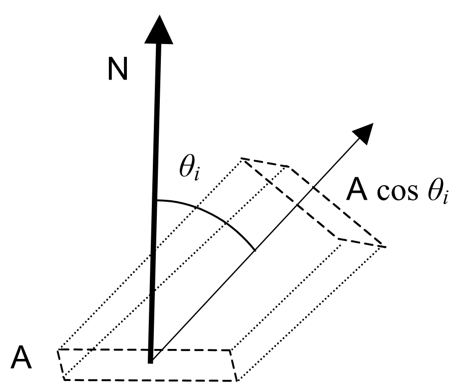

Figure 3. Projected area

Note that to illuminate a surface point, a light source needs to emit a nonzero radiance and it needs to subtend a nonzero solid angle from the illuminated point. For example, the sun emits a radiance of about 20 million W·sr^{-1}·m^{-2}, and subtends (from Earth) a solid angle of about

0.00007 sr (a cone about ½ degree across). Multiplying the two gives us the irradiance contribution of the sun to a surface which is perpendicular to its direction (about 1400 $W{\cdot}m^{-2}$). The point and directional light sources commonly used in real-time computer graphics are non-physical, because they “squeeze” their irradiance contribution into a zero solid angle, which implies an infinite radiance.

Radiometric quantities such as radiant flux, radiance and irradiance are *spectral* quantities. Their exact distribution over the visible spectrum determines the light color. For example, the sun’s irradiance contribution to a perpendicular surface on Earth is 1400 W/m^2 but this tells us nothing about its color. This scalar value also has a *spectral power distribution* (SPD) which looks like this:

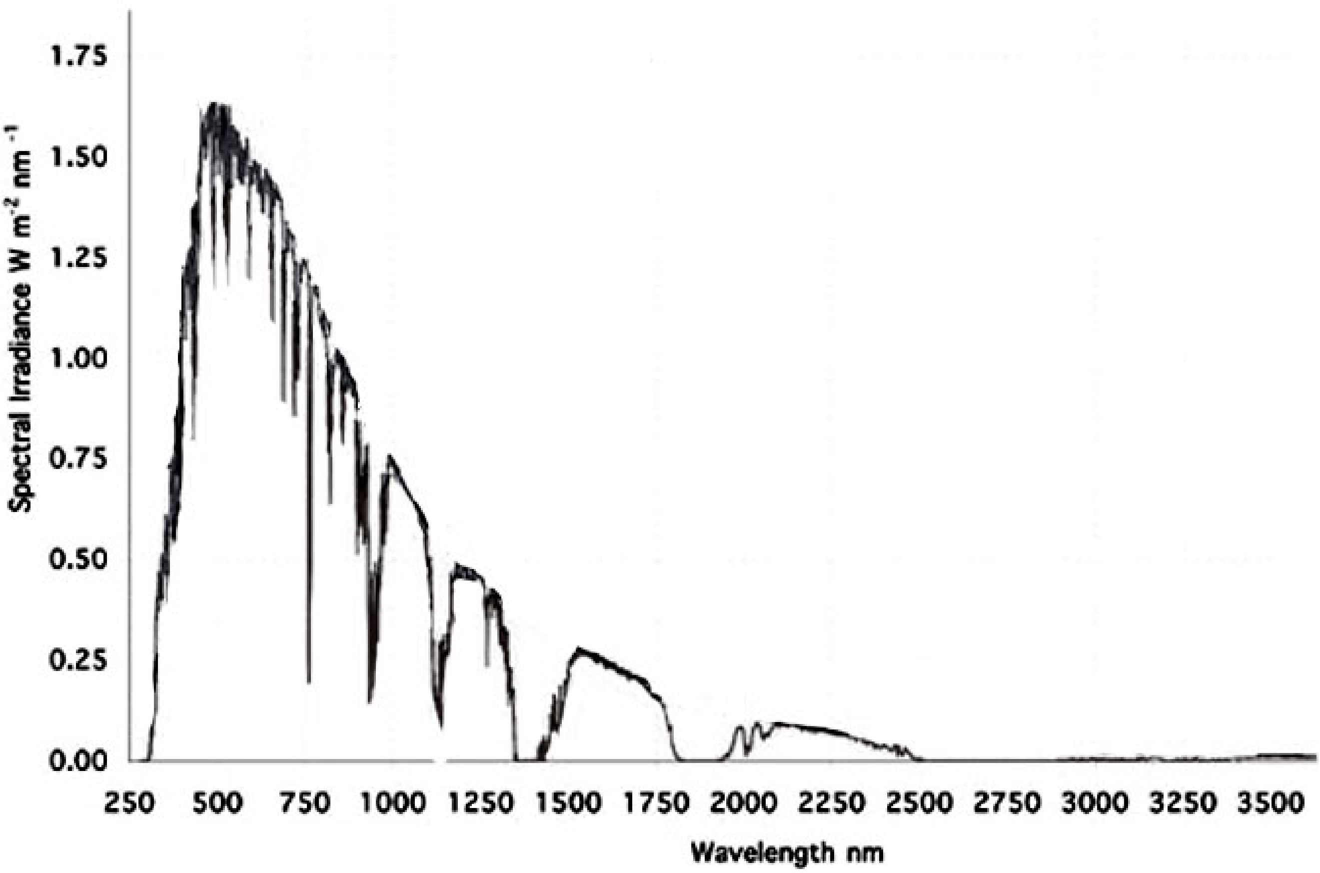

Figure 4. SPD of solar irradiance

For rendering purposes, we only care about the part of the SPD between 380 and 780 nm (note that the sun’s SPD has most of its energy in this region). Note that the spectral irradiance, unlike the scalar irradiance, is measured in $W{\cdot}sr^{-1}{\cdot}m^{-2}{\cdot}nm^{-1}$. The integral of this SPD over all wavelengths will yield the scalar irradiance value.

How do these quantities relate to frame buffer RGB values used to display the final result? The final pixel colors are derived from the radiance values along the rays from the center of projection through each pixels sample points. For display, these need to be converted to RGB values by integrating the radiance SPD multiplied by the SPDs of the RGB primaries. The resulting RGB values need to be converted from unbounded radiance values (which can have a very high dynamic range) to frame buffer values (which are between 0 and 1). This process is known as *tone mapping* and it is outside the scope of this paper.

Modern graphics hardware now enables us to perform our rendering using HDR (high dynamic range) radiometric quantities, performing tone mapping when writing the results out for final display. Physically-based offline rendering packages can also work on spectral quantities represented as large vectors of spectral samples (50 or more), but graphics hardware is not yet fast enough to do this in real-time. Instead RGB triples are typically used for real-time rendering (and most production offline rendering). The errors introduced by this approximation are probably small, especially compared to other common approximations.

The BRDF

A *BRDF* (Bidirectional Reflectance Distribution Function) is a function which relates incident irradiance to exitant reflected radiance. More precisely:

$$f_r(\omega_i, \omega_e) = \frac{dL_e(\omega_e)}{dE(\omega_i)}$$

Where ω_i is the direction to the incident irradiance, and ω_e is the direction to the exitant reflected radiance.

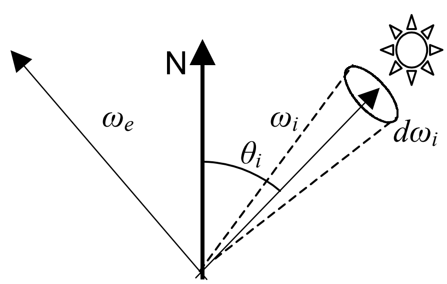

Figure 5. BRDF

Thus for every pair of directions (incident and exitant), the BRDF gives us the ratio between incident irradiance and exitant radiance. Since the incident direction and the excitant direction are both 2D quantities (a common parameterization is to use two angles: elevation θ relative to the surface normal and rotation φ about the normal), the BRDF is a 4D function. For most types of surface materials, the BRDF does not vary if the incident and exitant directions are rotated around the normal (while keeping the relative rotation between them). This reduces the number of variables from 4 to 3. BRDFs with this property are called *isotropic* BRDFs, and those without it are *anisotropic* BRDFs (two examples are brushed metal and hair).

If we take this equation and combine it with the previous relation between radiance and irradiance, we get:

$$f_r(\omega_i, \omega_e) = \frac{dL_e(\omega_e)}{L_i(\omega_i)\cos\theta_i d\omega_i}$$

which gives us:

$$dL_e(\omega_e) = f_r(\omega_i, \omega_e)L_i(\omega_i)\cos\theta_i d\omega_i$$

or:

$$L_e(\omega_e) = \int_\Omega f_r(\omega_i, \omega_e)L_i(\omega_i)\cos\theta_i d\omega_i$$

This means if we want to know the contribution of light from a tiny patch of incident directions (characterized by $d\omega_i$ and ω_i) to the reflected light exitant in a direction ω_e, we take the value of the BRDF at ω_i and ω_e, times the incident radiance from ω_i, times $d\omega_i$, times $\cos\theta_i$. If we integrate this over the hemisphere (centered on the normal), we get the total exitant radiance in direction ω_e. This is the *reflection equation*.

In real-time rendering, a simplified model of incident radiance is used (environment maps allow complex incident radiance, but they are a special case). In this model the incident radiance is equal to an ambient constant L_{iA} from all directions, except for a small number of directions in which we have directional or point lights. Those lights are characterized by their irradiance contribution E (if they were physical lights, this would be equal to their solid angle times their radiance). Since radiance is constant but the solid angle varies as the square of the distance between the light source and the surface point it illuminates, this causes the irradiance contribution to follow an inverse square law. This incident radiance model enables converting the integral into a sum over the light sources, which simplifies the reflection equation:

$$L_e(\omega_e) = \sum_l f_r(\omega_{il}, \omega_e)E_l \cos\theta_i + L_{iA}\int_\Omega f_r(\omega_i, \omega_e)\cos\theta_i d\omega_i$$

This can also be written:

$$L_e(\omega_e) = \sum_l f_r(\omega_{il}, \omega_e)E_l \cos\theta_i + L_{iA}R_A(\omega_e)$$

Usually, for the purpose of simplicity, we ignore the dependency of R_A (the *ambient reflectance*) on ω_r and assume that it is constant. This may introduce significant errors in many cases, but does simplify the equation further:

$$L_e(\omega_e) = \sum_l f_r(\omega_{il}, \omega_e)E_l \cos\theta_i + L_{iA}R_A$$

Not every arbitrary 4D (or 3D for isotropic BRDFs) function can be a BRDF. To be physically plausible, a BRDF has to follow certain rules:

Reciprocity: $f_r(\omega_i, \omega_e) = f_r(\omega_e, \omega_i)$. This means that the surface reflects light the same when the directions are reversed, and is a basic law of physics (*Helmholtz reciprocity*). It is also required for bidirectional ray tracing algorithms to work, since they implicitly depend on this property.

Energy Conservation: $\forall \omega_e : \int_{\Omega} f_r(\omega_i, \omega_e) \cos\theta_i d\omega_i \leq 1$. If the BRDF does not obey this condition, then there will be more reflected light energy than incident light energy which is a violation of conservation of energy. It is also required for many global illumination algorithms to converge.

Note that the BRDF in this form cannot model certain classes of physical phenomena, such as effects relating to light entering the surface at one point and leaving at a different point (subsurface scattering / translucency), light which enters the surface from one side and is emitted out the opposite side (transmittance). Note that the BRDF usually is dependent on wavelength, but it cannot model effects in which the light changes wavelength (fluorescence). Finally, due to energy conservation it cannot model surfaces which emit light on their own (phosphorescence).

Most real objects are not made out of one homogeneous pure material with no scratches or blemishes. Most surfaces have BRDFs which are *shift-variant* (which vary from point to point on the surface). This adds an additional two dimensions to the BRDF which now depends on a surface parameterization as well.

BRDF data can be measured from real surfaces, which is useful in comparing or fitting BRDF models. The device most commonly used for this is a *gonioreflectometer*, which measures the reflectance of a surface under multiple incident and exitant directions. Usually only one point on the surface can be measured, so shift-variant BRDFs cannot be captured with this device. There are large collections of BRDF data measured from various surfaces which are available online.

Lambertian BRDF

The simplest reflectance function is the pure diffuse or *Lambertian* function. This BRDF is simply a constant, which means that incident light from any direction is reflected equally over the entire exitant hemisphere (the well-known cosine factor is part of the reflectance equation and not part of the BRDF).

Note that the Lambertian BRDF does not correspond to any real-world surface; in fact a perfect Lambertian surface is physically impossible (the paint industry spends a lot of effort trying to get as close to this ideal as possible). However, it can be a reasonable approximation for surfaces which are extremely rough or which exhibit a lot of subsurface scattering (although a BRDF cannot model subsurface scattering in the general case, it can model the restricted case where the entry and exit point are the same). It is also often used in BRDFs in addition to other terms.

What values can this constant BRDF have? Let us imagine a perfectly reflective (white) Lambertian surface which reflects all incident light energy without absorbing any. In this case, the irradiance will be reflected out as radiosity and spread out over the π radians of the hemisphere. The exitant radiance in any given direction will be equal to the irradiance divided by π, so the BRDF is equal to a constant $1/\pi$. In general, the Lambertian BRDF is equal to the *bihemispherical reflectance* (the ratio between radiosity and irradiance, which must be between 0 and 1) divided by π. Since BRDFs vary by wavelength, the bihemispherical reflectance is represented as an RGB triple which is usually thought of as the diffuse color of the material.

Note that the light intensity values in most real-time rendering systems are multiplied directly by the diffuse color to get the output color (radiance), without applying a $1/\pi$ factor. This means that the light intensity value used in these systems is actually equal to E/π. Care must be taken when implementing BRDFs for use in these systems, to multiply the BRDF by π before multiplying the light color (most BRDFs have a $1/\pi$ factor somewhere in them which this will cancel out).

Fresnel

Another common reflectance case is that of the perfect mirror, which reflects all incoming radiance into a direction ω_r which is the incident light direction mirrored about the surface normal. Due to the fact that incident light is reflected into only one exitant direction, this reflectance function cannot be expressed in BRDF form without recourse to special functions such as Dirac delta functions. It is commonly used with environment maps (if it were used with directional lights pixels of infinite brightness would result).

Perfectly flat surfaces exhibit mirror-like behavior; however, they do not usually reflect all incident light. The *Fresnel equations* describe the amount of light which is reflected in the reflection direction (the remaining light is refracted into the material where it undergoes subsurface scattering and absorption). The original form of these equations is fairly complex and involves light polarization and complex indices of refraction; fortunately [Schlick1994] has a much simplified version which is a good approximation. This version captures the main feature of the Fresnel effect, which is that perfectly smooth surfaces have a given spectral reflectance for light which is incident parallel to the surface normal, and this reflectance gradually increases for all wavelengths as the angle of incidence increases, becoming 1 at all wavelengths in the limit case where the incident light is parallel to the surface. Qualitatively, this means that reflections off smooth surfaces become brighter and shift towards the light color at increasingly grazing angles. If F_0 is the reflectance at normal incidence, then according to the Schlick approximation the reflectance at other incidences is:

$$F(\omega_e) = F_0 + (1 - F_0)(1 - \cos\theta_i)^5$$

Note that this reflectance is not the same as the BRDF; it is the ratio between incident *radiance* and exitant radiance, rather than between incident *irradiance* and exitant radiance. It is either used as a component of more complex BRDFs, or it can be used directly to modulate environment maps.

F_0 must be between 0 and 1 for all wavelengths, due to energy conservation. It tends to be high for metals, and relatively low for *dielectric* (non-conductive) materials. In metals the light which is not reflected is absorbed, in dielectrics it can be transmitted to the other side (e.g. glass) or it may undergo subsurface scattering and exit close to the original entry point (e.g. plastics). F_0 is usually thought of as the specular color of a material.

Micro-Geometry

Besides its physical properties (such as F_0 or subsurface scattering properties), a material's reflectance is affected by its *micro-geometry*. This is geometry which is too small to see – the color we see is the average color of many microscopic elements. This micro-geometry affects the reflectance in various ways. One is that the microscopic elements may have normals which differ from the macroscopic surface normal. This causes light to be reflected in directions other than the reflection direction ω_r. Another way in which the micro-geometry affects the reflectance is by *shadowing* (some microscopic elements may block other ones from receiving light), and *masking* (some elements may block other ones from being visible). Finally, *interreflection* causes light to bounce between the various elements, finally exiting in some direction other than ω_r.

Micro-geometry can be used to generate a BRDF in two ways. It can be explicitly modeled, and a ray-tracer used as a *virtual gonioreflectometer* to simulate various incident and exitant directions [Cabral1987; Westin1992; Gondek1994]. These results can be used in a table or fit to a function for later use in rendering. Or the micro-geometry can be used to analytically create a BRDF [Blinn1982; Kajiya1985] of which a special case is *micro-facet theory* [Torrance1967; Blinn1977; Cook1981; Oren1994; Ashikhmin2000a].

Micro-facet theory is a method of generating a BRDF from micro-geometry. It models normal variance, shadowing and masking, but not interreflection. The surface is assumed to be composed of micro-facets. Normal variance is modeled by means of a *normal distribution function* (NDF) $p(\omega)$. This has the property that for sufficiently small $d\omega$, $p(\omega)d\omega$ gives the fraction of micro-facets which have normals in the solid angle $d\omega$ around ω. The NDF is formulated with ω in the local frame of the surface.

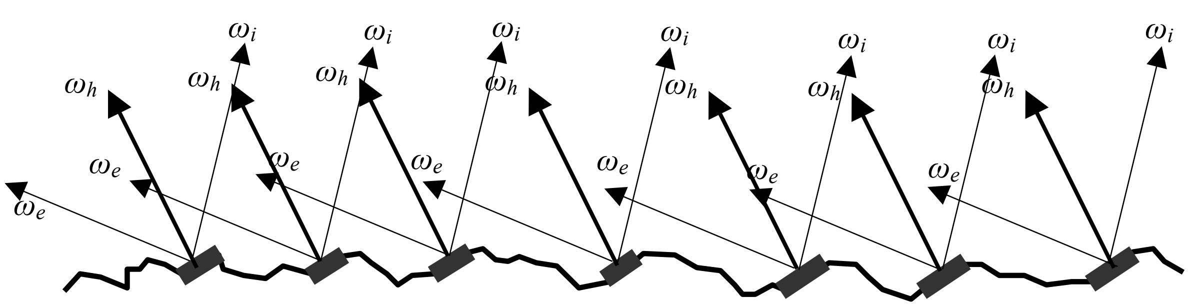

Figure 6. Micro-facets

In figure 6 we can see an example micro-facet structure. Each facet is assumed to be a perfect Fresnel mirror. For a given incident direction ω_i and exitant direction ω_e, only those facets which are facing in the right direction to reflect ω_i into ω_e can contribute to the reflectance. These are the facets which have their normal equal to ω_h, which is the direction which bisects ω_i and ω_e and is known as the *half-angle direction*. Evaluating the NDF for the half-angle direction gives us $p(\omega_h)$, which is used to determine how many microfacets are oriented such that they can contribute to the reflectance. We will use α_h to denote the angle between ω_i and ω_h (which is also the angle between ω_e and ω_h). Then the Fresnel reflectance of each of the contributing microfacets is $F(\alpha_h)$.

A micro-facet BRDF is derived in the following manner. Remember the BRDF definition:

$$f_r(\omega_i,\omega_e)=\frac{dL_e(\omega_e)}{L_i(\omega_i)\cos\theta_i d\omega_i}$$

And from the definition of radiance, we have the following expression for dL_e:

$$dL_e=\frac{d\Phi_e}{dA\cos\theta_e d\omega_e}$$

Where dA is the area of a small (but large enough to contain many micro-facets) surface patch. Note that dA is the nominal macro-scale area of the patch – the area of the patch on the micro-facet scale (the sum of areas of all micro-facets in the patch) is equal to dA/K_p, which is larger. K_p is a constant with a value that depends on the micro-facet structure. If the micro-geometry is a heightfield then it depends only on $p(\omega)$ and is equal to the cosine of the angle between ω and the macro-scale surface normal, averaged over the distribution (more details, as well as the derivation for non-heightfield geometry, are in [Ashikhmin2000a]). Combining the last two equations gives us:

$$f_r(\omega_i,\omega_e)=\frac{d\Phi_e}{dA\, L_i(\omega_i)\cos\theta_i\cos\theta_e d\omega_i d\omega_e}$$

This describes a situation where the surface patch dA is illuminated by incident radiance L_i, coming from a solid angle $d\omega_i$ around the incident direction ω_i. As a result this patch is reflecting radiant flux $d\Phi_e$ in a solid angle $d\omega_e$ around an exitant direction ω_e. There will be a certain number of micro-facets in the patch which are actively participating in reflecting this radiant flux. We will use $dA_{\mu Active}$ to denote the total area of these micro-facets. All these micro-facets have their normals oriented in the direction ω_h. Each of these micro-facets will have Fresnel reflectance $F(\alpha_h)$. Also, the total active projected area in the direction ω_i is $dA_{\mu Active}\cos\alpha_h$. This gives us the following expression for $d\Phi_e$:

$$d\Phi_e=L_i(\omega_i)d\omega_i dA_{\mu Active}\cos\alpha_h F(\alpha_h)$$

Yielding the following for the BRDF:

$$f_r(\omega_i,\omega_e)=\frac{dA_{\mu Active}\cos\alpha_h F(\alpha_h)}{dA\cos\theta_i\cos\theta_e d\omega_e}$$

The total area of all microfacets oriented in the right direction is $dA\ p(\omega_h)d\omega_h/K_p$. Out of these micro-facets, some will be masked or shadowed. We use the *geometry factor* $G(\omega_i,\omega_e,\omega_h)$ (which is always between 0 and 1) to represent the fraction of micro-facets which are not masked and/or shadowed. Various forms for $G(\omega_i,\omega_e,\omega_h)$ can be found in [Torrance1967; Blinn1977; Cook1981; Ashikhmin2000a]. Then:

$$dA_{\mu Active}=\frac{dA\ p(\omega_h)d\omega_h G(\omega_i,\omega_e,\omega_h)}{K_p}$$

From [Torrance1967; Ashikhmin2000a] we learn that:

$$d\omega_h=\frac{d\omega_e}{4\cos\alpha_h}$$

Which combined with the previous two equations gives us the final form of the micro-facet BRDF:

$$f_r(\omega_i,\omega_e)=\frac{p(\omega_h)G(\omega_i,\omega_e,\omega_h)F(\alpha_h)}{4K_p\cos\theta_i\cos\theta_e}$$

The NDF $p(\omega)$ is the most important parameter of the micro-facet model. Various alternative NDFs were proposed in [Torrance1967; Blinn1977; Cook1981; Ashikhmin2000a]. Isotropic NDFs yield isotropic BRDFs; anisotropic NDFs yield anisotropic BRDFs. NDFs can even be generated from hand-painted 2D images [Kautz2002b; Ashikhmin2000a]. These images are 2D tables which are indexed by ω_h projected into the local surface frame. It should be noted that such 2D images are very closely related to the shape of the specular highlight generated by the resulting BRDF, so painting such an image is equivalent to painting the highlight.

Composite Materials

These various building blocks can be combined in various ways to form BRDFs. Micro-facet theory is useful for creating specular terms which model the surface interaction of the light. Light which is not reflected off the surface can undergo subsurface scattering which can be modeled by a diffuse term (Lambertian or other). Metals do not have subsurface scattering but can use diffuse terms to model interreflection. Most materials have similar colors for their various reflectance components, but some composite materials may have components which vary in color. A common example is plastic, which is usually composed of a smooth clear substrate in which colored pigment particles are embedded, leading to a colored diffuse term and a white specular term.

The Blinn-Phong BRDF

This [Phong1975; Blinn1977] is the reflectance function most commonly used in graphics, usually in this form:

$$f(\mathbf{V},\mathbf{L}) = k_d dot(\mathbf{N},\mathbf{L}) + k_s dot(\mathbf{N},\mathbf{H})^n$$

Where $\mathbf{V}$ is the view vector, $\mathbf{L}$ is the light vector, $\mathbf{N}$ is the surface normal and $\mathbf{H}$ is the half-angle vector. The parameters of the model are k_d the diffuse reflectivity (or albedo) of the surface, k_s the specular reflectivity and n, the specular power. This reflectance function is not in the form of a BRDF. We will transform it into one, using our usual notation (remember that the BRDF is always multiplied by $\cos\theta_i$, which is the same as $dot(\mathbf{N},\mathbf{L})$):

$$f_r(\omega_i,\omega_e) = \frac{k_d}{\pi} + \frac{k_s}{\pi\cos\theta_i}(\cos\alpha_h)^n$$

Note that we have also divided by π to take account of the difference between the light intensity values commonly used in game engines and physical irradiance.

While this model violates some of the basic properties of BRDFs that were described earlier, it does in fact have a loose connection with a physically plausible model. It roughly models a dielectric material, composed of a smooth glossy layer over a diffuse substrate. The $(\cos\alpha_h)^n$ term can be seen to represent a microfacet NDF. If k_s is colored this can be used to model metals (if k_d is near zero). If k_s is white and k_d is colored it can model polished surfaces like plastic.

Because of the $\cos\theta_i$ term in the denominator of the specular part the BRDF is not reciprocal. It does not conserve energy and can strictly represent isotropic materials. It also does not handle facet shadowing and masking that are modeled in traditional microfacet models mentioned earlier. It also does not model Fresnel reflectance properly – the diffuse term represents transmitted light scattered in a matte layer from Fresnel's formula and should be dependent on the view/light directions.

Some of these issues can be addressed by multiplying the specular term by $(n+4)/8$ (which would normalize it, ensuring energy conservation) and eliminating the divide by $\cos\theta_i$ (which would ensure reciprocity). However this last change in particular will result in a qualitatively different model and it would make more sense to just use a more realistic reflection model.

Other Reflectance Functions

There are many BRDFs that are commonly used. The most simple anisotropic model [Banks1994] is very simple to evaluate and can be easily implemented using graphics hardware. Another notable anisotropic BRDF is in [Ward1992]. There have been several generalizations of Blinn-Phong to handle anisotropy, one that is simple and can easily model spatially variation even on DX7 level hardware is [Kautz2000b]. A very compelling recent model [Ashikhmin2000b] has all of the desired physical properties of BRDFs, has intuitive parameters and correctly models Fresnel effects. A model that is computationally very light weight, but has less intuitive parameters is [Lafortune1997]. It can handle effects like retro-reflection (light reflecting back in the direction of the light source) which can be difficult in general. There is a rich set of BRDF models to choose from, these are just some of the ones that might be more attractive for game developers. An interesting comparison of several BRDf models is in [Shirley1997].

Implementation

Each point on a surface (e.g. an object in a game) has a BRDF which represents it. When performing direct illumination, we are interested in the light which makes its way to viewer's eye. In other words, the exitant direction will always be the (negative) eye vector. With discrete, directional lights, we evaluate each light source using the incident angle of the light and sum the total energy.

BRDFs are usually implemented via programmable shaders. Because a BRDF is a 4 dimensional function, it is typically implemented through a parameterized model (e.g. the Blinn-Phong BRDF). Since a BRDF exists at an infinitely small point, it is evaluated at a frequency which is high enough to prevent aliasing.

A basic implementation of a BRDF model is the translation of the mathematical formulation into shader code. This creates a lighting model which takes a discrete light as input (multiple lights would be summed). This can be implemented at either a vertex or a pixel level. This Blinn-Phong function, for instance, is a parameterized model which takes *Normal*, *Kd*, *Ks*, *n* as parameters which can vary for each point on a surface. When *Ks* changes at a per pixel level, this is known as gloss mapping, when *Normal* varies at a per pixel level, this is known as bump mapping, etc.

Unfortunately, using discrete lights limits lighting to scenes in which everything is lit only by a handful of bright light sources. This is analogous to lighting a single object in a dark room with a bright flashlight. Though physically correct, lighting objects in this manner produces results that would be unrealistic in real-world settings. In real world scenes much of light comes from indirect illumination. For indoor lighting direct lighting is generally minimized by design (e.g. lamp shades.)

This presents one of the greatest challenges for using BRDFS: accounting for area lighting. Since integrating against the environment is prohibitive, models need to be fitted with some kind of area lighting approximation. The simplest approximation is known as ambient light, and it is usually added directly to the exitant light.

Ambient light is a poor looking approximation, however, and does not produce compelling results. With a simple BRDF, e.g. a constant BRDF (Diffuse) [Ramamoorthi2001], area lighting can be precomputed by preintegrating the environment. As BRDF's become more complex, however, it becomes more difficult to precompute the environment reflections because of the high dimensionality: 3 DOF represent a normal and tangent frame of a surface (2 for an isotropic BRDF), 2 DOF represent the lighting environment and a DOF is needed for each non-linear parameter of the model to handle spatial variation [Kautz2000].

A common approach is to factorize the BRDF into lower-dimensional functions which are stored in lookup tables [Heidrich1999; Kautz1999; McCool2001; Kautz2000b; Steigleder2003]. However, care needs to be taken not to perform the factorization in a way that would “bake” the BRDFs parameters into the tables, thus preventing shift-variance.

Environment Mapping

Environment mapping [Blinn1976] is a technique used for pure reflectance-based BRDFs which gives sharp reflections. With a general BRDF, an environment map represents the environment as an arbitrary dense set of weak lights. For a very restricted set of BRDFs prefiltering can be performed using a single environment map (BRDFs that convolve a spherical function with a radially symmetric kernel, e.g. Phong, but even for those cases it will not clamp values under the hemisphere to zero.) Some recent papers handle a slightly richer set of BRDFs, but are still limited to isotropic ones or purely anisotropic ones [Ramamoorthi2002]. If the lighting is restricted to be low frequency more general BRDFs can be represented [Kautz2002a]. If the parameters of the BRDF vary spatially multiple pre-filtered environments are needed, the simplest case is to just bias the LOD of the environment map to model a blurrier kernel [McAllister2002; Ashikmin2003],

Filtering

When a BRDF is sampled at a pixel level (as it usually is done for games), it is easy to see that as a model becomes smaller the BRDF evaluation will quickly begin to alias. Normal MIP-map filtering techniques filter linearly, or rather under the assumption that the per-pixel evaluation will be a linear one. Unfortunately, for all but the most basic BRDFs, this is not true. To achieve correct filtering, the BRDF must be evaluated at the highest MIP level. This, however, is not normally feasible on most hardware. An alternative technique is for each MIP level to adjust the parameters of the BRDF to approximate the reflectance properties of the higher MIP level’s BRDFs for that sample location [Fournier1992; Olano1997; Kautz2001]. One common technique to avoid aliasing with bump mapping is to shift the normal closer to the actual surface normal for each MIP level so by the lowest MIP level it becomes the surface normal.

Reciprocity and Conservation

Since we are interested in BRDFs primarily for direct lighting, it is not necessary to strictly enforce reciprocity and energy conservation. Indeed, as we begin to factor in global illumination approximations it will be impossible to maintain either constraint. Often, these terms can be compensated via author time adjustment of the BRDF parameters.

Conclusions

To generate images of realistic objects, it is important to understand some of the underlying physics that describe every day phenomena. For game production, it is important to expose the reflectance model parameters so that artists can easily control them and vary them over surfaces. It is also important for the reflectance models to be expressive and computationally efficient enough for real time use. Finally for realistic image generation, general lighting environments and complex transport should be handled as well.

References

[Ashikhmin2000a] ASHIKHMIN, M., PREMOZE, S., AND SHIRLEY, P. A microfacet-based BRDF generator. In Proceedings of SIGGRAPH 2000, pp. 65-74.

[Ashikhmin2000b] ASHIKHMIN, M. AND SHIRLEY, P. An anisotropic Phong BRDF model. In Journal of Graphics Tools, v.5 no. 2 (2000).

[Ashikhmin2003] ASHIKHMIN, M. AND GHOSH, A. Simple blurry reflections with environment maps. In Journal of Graphics Tools, v.7 no. 4 (2003).

[Banks1994] BANKS, D. Illumination in diverse codimensions. In Proceedings of SIGGRAPH 1994, pp. 327-334.

[Blinn1976] BLINN, J. F. AND NEWELL, M. E. Texture and reflection in computer generated images. Communications of the ACM Vol. 19, No. 10 (October 1976), 542-547.

[Blinn1977] BLINN, J. F. Models of light reflection for computer synthesized pictures. In Proceedings of SIGGRAPH 1977, pp. 192-198.

[Blinn1982] BLINN, J. F. Light reflection functions for simulation of clouds and dusty surfaces. In Proceedings of SIGGRAPH 1982, pp. 21-30.

[Cabral1987] CABRAL, B., MAX, N., AND SPRINGMEYER, R. Bidirectional reflectance functions from surface bump maps. In Proceedings of SIGGRAPH 1987, pp. 273-282.

[Cook1981] COOK, R. L., AND TORRANCE, K. E. A reflectance model for computer graphics. In Proceedings of SIGGRAPH 1981, pp. 307–316.

[Fournier1992] FOURNIER, A. Normal distribution functions and multiple surfaces. In Proceedings of graphics Interface Workshop on Local Illumination 1992, pp. 45-52.

[Gondek1994] GONDEK, J.S., MEYER, G.W., AND NEWMAN, J.G. Wavelength dependent reflectance functions. In Proceedings of SIGGRAPH 1994, pp. 213-220.

[He1991] HE, X. D., TORRANCE, K. E., SILLION, F. X., AND GREENBERG, D. P. A comprehensive physical model for light reflection. In Proceedings of SIGGRAPH 1991, pp. 175-186.

[Heidrich1999] Heidrich, W. and Seidel, H.-P. Realistic, hardware-accelerated shading and lighting. In Proceedings of SIGGRAPH 1999, pp. 171-178.

[Kajiya1985] KAJIYA, J. Anisotropic reflectance models. In Proceedings of SIGGRAPH 1985, pp. 15-21.

[Kautz1999] KAUTZ, J. AND MCCOOL, M. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In Proceedings of Eurographics Workshop on Rendering 1999, pp. 255-268.

[Kautz2000] KAUTZ, J. AND MCCOOL, M. Approximation of glossy reflection with prefiltered environment maps. In Proceedings of Graphics Interface 2000, pp. 119-126.

[Kautz2001] KAUTZ, J., HEIDRICH, W., AND SEIDEL, H.-P. Real-time bump map synthesis. In Proceedings of Graphics Hardware 2001, pp. 109-114.

[Kautz2000b] KAUTZ, J. AND SEIDEL, H.-P. Towards interactive bump mapping with anisotropic shift-variant BRDFs. In Proceedings of Graphics Hardware 2000, pp. 51-58.

[Kautz2002a] KAUTZ, J., SLOAN, P.-P., SNYDER, J. Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In Proceedings of Eurographics Rendering Workshop 2002, pp. 301-308.

[Kautz2002b] KAUTZ, J. Rendering with handcrafted shading models. In Game Programming Gems 3, Editors Mark DeLoura and Dante Treglia, Charles River Media, July 2002.

[Lafortune1997] LAFORTUNE, E. P. F., FOO, S.-C., TORRANCE, K. E., AND GREENBERG, D.P. Non-linear approximation of reflectance functions. In Proceedings of SIGGRAPH 1997, pp. 117-126.

[McAllister2002] MCALLISTER, D. K., LASTRA, A., AND HEIDRICH, W. Efficient rendering of spatial bi-directional reflectance distribution functions. In Proceedings of Graphics Hardware 2002, pp. 79-88.

[McCool2001] MCCOOL, M. D., ANG, J., AND AHMAD, A. Homomorphic Factorizations of BRDFs for High-Performance Rendering. In Proceedings of SIGGRAPH 2001, pp. 171-178.

[Olano1997] Olano, M. and North, M. Normal distribution mapping. *UNC Computer Science Technical Report 97-041*, Department of Computer Science, University of North Carolina, Chapel Hill, North Carolina, 1997.

[Oren1994] OREN, M., AND NAYAR, S. K. Generalization of lambert's reflectance model. In Proceedings of SIGGRAPH 1994, pp. 239–246.

[Phong1975] PHONG, B.-T. Illumination for computer generated images. In Communications of the ACM, 18, 6 (June 1975), pp. 311-317.

[Ramamoorthi2001] RAMAMOORTHI, R. AND HANRAHAN, P. An efficient representation for irradiance environment maps. In Proceedings of SIGGRAPH 2001, pp. 497-500.

[Ramamoorthi2002] RAMAMOORTHI, R. AND HANRAHAN, P. Frequency space environment map rendering. In Proceedings of SIGGRAPH 2002, pp. 517-526.

[Schlick1994] SCHLICK, C. An inexpensive BRDF model for physically-based rendering. In Proceedings of Eurographics 1994, pp. 233-246.

[Shirley1997] Shirley, P., Hu, H., Smits, B., and Lafortune, E. A practitioners' assessment of light reflection models. In Proceedings of Pacific Graphics 1997, pp. 40-49.

[Stam1999] STAM, J. Diffraction shaders. In Proceedings of SIGGRAPH 1999, pp. 101-110.

[Steigleder2003] Steigleder, M. and McCool, M. Factorization of the Ashikhmin BRDF for real-time rendering. In In Journal of Graphics Tools, v.7 no. 4 (2003).

[Torrance1967] TORRANCE, K. E., AND SPARROW, E. M. Theory for off-specular reflection from roughened surfaces. Journal of Optical Society of America 57, 9 (1967), pp.1105-1114.

[Ward1992] WARD, G. Measuring and modeling anisotropic reflection. In Proceedings of SIGGRAPH 1992, pp. 265-272.

[Westin1992] WESTIN, S. H., ARVO, J. R., AND TORRANCE, K.E. Predicting reflectance functions from complex surfaces. In Proceedings of SIGGRAPH 1992, pp. 255-264.