

Optimizing Spectral Fresnel Reflectance for Displays

Jonathan Brian Metzgar
University of Alaska Fairbanks
College of Engineering and Mines
PO Box 755960
Fairbanks, Alaska 99775 USA
jmetzgar@acm.org

Sudhanshu Kumar Semwal
University of Colorado at Colorado Springs
Department of Computer Science
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80918 USA
ssemwal@uccs.edu



ABSTRACT

Approximate equations for rendering Fresnel reflectance abound in computer graphics. We take a fresh approach and consider not only the approximation but the display device as well. The sRGB color standard is finally giving way to wide color spaces such as Adobe RGB and DCI P3 which display more color. We present a preprocessing method to use measured spectral index of refraction data and the color space specification to synthesize an RGB complex index of refraction. Metals, in particular, generally require a spectral renderer, but we created a way to sample the complex index of refraction and absorptive index that acts like the color filter built into the display. Our novel contribution uses a normal distribution centered around the ideal display red, green, and blue wavelengths derived from the CIE xy coordinates and respective white point to window sample the complex index of refraction. We created a WebGL experimental platform that uses the Schlick inspired Lazanyi and Szirmay-Kalos (LSK) multispectral Fresnel approximation coupled with modern physically based BRDFs to simulate the appearance of metal. Our application can compare five different Fresnel implementations coupled with physically based Blinn-Phong and GGX microfacet models. We demonstrate that with reasonable filter widths, we eliminate the need for a spectral renderer for real-time rendering. Additionally, we utilize publicly available measurement data to simulate a variety of metals ranging from silver, gold, and copper to silicon, lead, and zinc.

Keywords

Fresnel reflectance modeling, rasterization, physically based rendering, complex index of refraction, absorptive index

1 INTRODUCTION

Fresnel reflectance is a critical component in many microfacet based bidirectional distribution reflectance functions (BRDF). It provides the effect that viewing angles close to the normal provide the highest translucency into a refractive surface or at grazing angles, the highest reflectance of the environment. For conductors

or metals, it provides their natural color and reflective properties. However, the equations are computationally expensive, so a variety of approximations exist for high-performance applications like real-time rendering.

The Fresnel equation is a function using the index of refraction, itself dependent on wavelength. For many materials, the index of refraction is consistent for the visible electromagnetic spectrum. For some metals such as gold and copper, there is significant absorption in the green, blue, and violet bands. Merely using a constant factor for the entire visible spectrum will introduce errors. However, using separate refractive indices for red, green, and blue may introduce color errors when not targeted for the color space of the intended display device. The state-of-the-art is to use spectral rendering,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

impractical for many real-time applications, to accurately reproduce these types of materials.

Because spectral rendering is difficult for real-time applications, we propose synthesizing the complex index of refraction using a chosen color space. Our novel contribution is a preprocessing step that uses window sampling using a Gaussian filter of the refractive index data for the red, green, and blue (RGB) wavelengths. We use the CIE Spectral Locus (or “horseshoe”) and a line-distance function to determine the proper RGB wavelengths. Finally, we implemented an HTML5 application using TypeScript and WebGL to test a variety of exact and approximate Fresnel equations and enable direct comparison for the accuracy of each method. Our results show that using our simple method enables developers to create real-time renderings of metal without needing a spectral renderer.

2 PREVIOUS WORK

Several background ideas must be considered for the realistic rendering of metals and other wavelength dependent materials. There is the Fresnel equation and several approximations which we review in this section. There is also the reflection model or microfacet BRDFs which we cover separately in the next section. Color theory and perception are essential because our method relies on using several color space specifications for determining the appropriate RGB wavelengths. We introduce mathematical equations like the Gaussian filter and relative distance function when they are needed in this paper.

Schlick’s approximation [Sch94],

$$F(\theta_d) = F_0 + (1 - F_0)(1 - \cos^5 \theta_d),$$

or,

$$F(\eta_1, \eta_2, \theta_d) = \frac{(1 - \eta_2)^2 + 4 \cos^5 \theta_d \eta_2}{(1 + \eta_2)^2}$$

where F_0 is the base reflectance at incidence 0° and commonly used in rendering algorithms because it has low computational cost. Recently, this work was extended to support metals [LS05] by including the absorptive index κ ,

$$F(\eta_1, \eta_2, \theta_d) = \frac{(1 - \eta_2)^2 + 4 \cos^5 \theta_d \eta_2 + \kappa_2^2}{(1 + \eta_2)^2 + \kappa_2^2}.$$

It is also common to precalculate F_0 for red, green, and blue in Schlick’s approximation [AMHH08] using the full equation and eliminates the reddish hue that was identified by Lazányi and Szirmay-Kalos [LS05]. For brevity, we refer to their method as the LSK method.

A common approximation used by Glassner [Gla95] and others [PH10] for the Fresnel equations is

$$\rho_{\parallel}^2 = \frac{(\eta_2^2 + \kappa_2^2) \cos^2 \theta_d - 2\eta_2 \cos \theta_d + 1}{(\eta_2^2 + \kappa_2^2) \cos^2 \theta_d + 2\eta_2 \cos \theta_d + 1}, \quad (1)$$

$$\rho_{\perp}^2 = \frac{(\eta_2^2 + \kappa_2^2) - 2\eta_2 \cos \theta_d + \cos^2 \theta_d}{(\eta_2^2 + \kappa_2^2) + 2\eta_2 \cos \theta_d + \cos^2 \theta_d}, \quad (2)$$

$$F = \frac{\rho_{\parallel}^2 + \rho_{\perp}^2}{2}. \quad (3)$$

The full form of the Fresnel equations for an absorbing medium [Mod13, p.51-53] can be written as

$$\rho_{\parallel} = \frac{(p - \sin \theta_1 \tan \theta_1)^2 + q^2}{(p + \sin \theta_1 \tan \theta_1)^2 + q^2} \rho_{\perp},$$

$$\rho_{\perp} = \frac{(\cos \theta_1 - p)^2 + q^2}{(\cos \theta_1 + p)^2 + q^2}$$

where

$$p^2 = \frac{1}{2} \sqrt{(\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2 \theta_1)^2 + 4\eta_2^2 \kappa_2^2} + \frac{1}{2} (\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2 \theta_1),$$

$$q^2 = \frac{1}{2} \sqrt{(\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2 \theta_1)^2 + 4\eta_2^2 \kappa_2^2} - \frac{1}{2} (\eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2 \theta_1).$$

If we let

$$a = \eta_2^2 - \kappa_2^2 - \eta_1^2 \sin^2 \theta_1,$$

then we can shorten the definition for p^2 and q^2 to

$$p^2 = \frac{1}{2} \left[\sqrt{a^2 + 4\eta_2^2 \kappa_2^2} + a \right],$$

$$q^2 = \frac{1}{2} \left[\sqrt{a^2 + 4\eta_2^2 \kappa_2^2} - a \right].$$

Lastly, the angle of refraction is given by the *Generalized Snell’s Law* equation

$$p \tan \theta_2 = \eta_1 \sin \theta_1.$$

Color space theory [JG78, MG80] provides insight into the perception of color and how displays are designed to display color. For example HP and Microsoft introduced the sRGB profile [IEC99] which gives guidance on how to design a display device such that a certain range of colors can be accurately produced. Although sRGB achieved widespread use, it has a limited range of color. Wider color spaces such as Adobe RGB [Ado98] and DCI-P3 [SMP11] are popular amongst artists and for modern consumer devices.

These color spaces use CIE 1931 xy coordinates to define the domain of colors (shown in Figure 1). The white point determines the wavelength of light for each of the three primaries. The wavelengths of light are

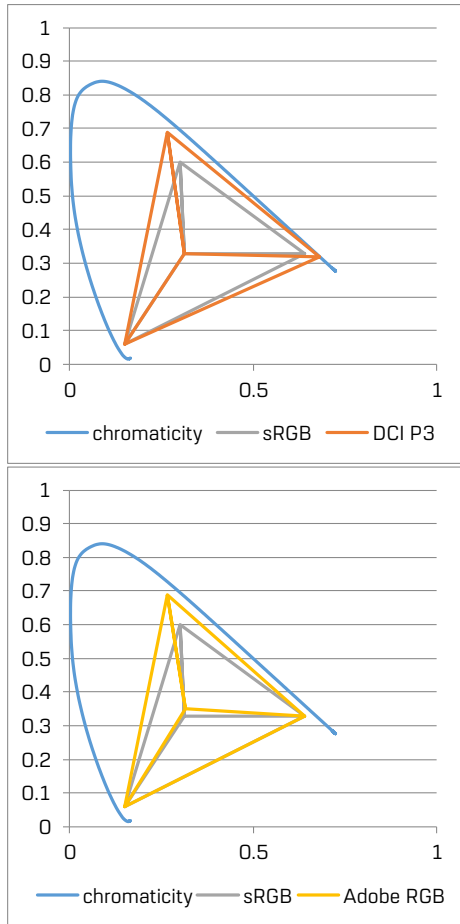


Figure 1: A comparison of sRGB (gray), DCI-P3 (orange), and Adobe RGB (yellow) color spaces.

dependent on the white point and the red, green, and blue coordinates. The line intersecting the spectral locus (the outer curved line representing wavelength), the chromaticity coordinate xyY , and the white point determines the wavelength because we are moving from the white point towards monochromaticity. This is called the *dominant wavelength* or *equi-hue line*. Later, we show that we can apply a relative distance function to find the wavelength if we only have the raw CIE horse shoe data and color space chromaticity coordinates. Figure 2 compares the sRGB and DCI P3 color spaces and the sRGB and Adobe RGB color spaces.

3 APPLYING FRESNEL REFLECTION

The rendering equation [Kaj86] by Kajiya,

$$L_o(\mathbf{x} \rightarrow \omega_o) = L_e(\omega_o \rightarrow \mathbf{x}) + \int_{\Omega} f_r(\omega_i, \omega_o) L_i(\omega_i \rightarrow \mathbf{x}) \langle \omega_i, \omega_o \rangle d\omega_i,$$

says that the light approaching the viewer from point \mathbf{x} is proportional to all the light incident from the positive hemisphere. In other words, every incoming direction at \mathbf{x} is a potential source of light. In turn, the amount of reflected light in the direction ω_o is deter-

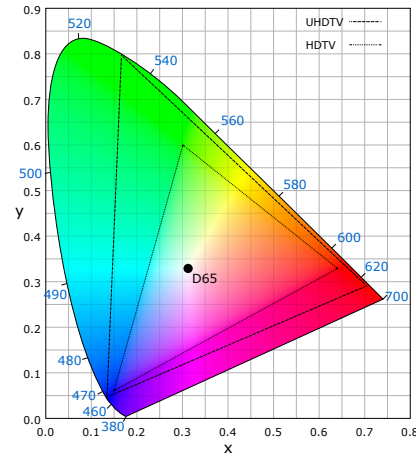


Figure 2: The CIE 1931 xy color space horse shoe [Com16] comparing the Rec 2020 and Rec 709 standards for UHDTV and HDTV.

mined by the bidirectional reflectance distribution function (BRDF) [Nic65]

$$f_r(\omega_i, \omega_o) = \frac{dL_o(\omega_o)}{dE_i(\omega_i)} = \frac{dL_o(\omega_o)}{L_i(\omega_i) \cos \theta_i d\omega_i}.$$

For glossy surfaces using microfacet models, the widely used Cook-Torrance model [CT81] is

$$f_r(\omega_i, \omega_o) = \frac{D(\omega_h) F(\theta_d) G_2(\omega_i, \omega_o)}{4 \cos \theta_i \cos \theta_o},$$

where $D(\omega_h)$ is the microfacet distribution, $F(\theta_d)$ is the Fresnel factor, and $G_2(\omega_i, \omega_o)$ is the masking-shadowing function. The five critical vectors necessary to evaluate this function is the incoming vector ω_i , outgoing vector ω_o , geometric normal ω_g , half angle vector $\omega_h = \omega_i + \omega_o$, and difference angle vector $\omega_d = \omega_i - \omega_o$. We use the *difference angle* $\theta_d = \arccos \omega_i \cdot \omega_h$ is used for Fresnel. To be physically based, it must obey the laws of conservation of energy, Helmholtz reciprocity, and positivity. A thorough analysis of the BRDF and masking-shadowing function is shown by Heitz [Hei14]. Diffuse reflection models, such as the Disney BRDF, also incorporate a Fresnel term [BS12]. This is especially helpful for rendering soft materials like felt or velvet.

We used both the Blinn-Phong and generalized GGX microfacet distributions for testing purposes. The roughness of the surface is given by α . The normalized Blinn-Phong microfacet model is

$$D(\omega_g, \omega_h) = \frac{1}{\pi \alpha^2} (\omega_g \cdot \omega_h)^{\frac{2}{\alpha^2 + \epsilon}} \alpha^{2 + \epsilon} \epsilon^{-2 + \epsilon}$$

where ϵ is a small offset to eliminate division by zero when the roughness is zero. The GGX microfacet distribution [BS12, WMLT07] is

$$D_{GTR}(\omega_g, \omega_h) = \frac{1}{\pi} \left(\frac{1}{(\alpha^2 - 1)(\omega_g \cdot \omega_h)^2 + 1} \right)^\gamma$$



Figure 3: Monochromatic choice of Fresnel sampling can result in deviations. Compare blue wavelengths of 445nm (left) and 500nm (right)



Figure 4: A comparison of gold when rendered with sRGB (left) and Adobe RGB (right) showing that color space is important for rendering.

where γ is an adjustable parameter that closely matches the Beckmann distribution at 10 and the Trowbridge-Reitz at 2.

The masking-shadowing functions use the G_2 version (see Heitz [Hei14] for more information) which take into account both incoming and outgoing directions. The Smith function

$$G_2(\omega_i, \omega_o, \omega_g) = \frac{1}{1 + \Lambda(\omega_i) + \Lambda(\omega_o)}$$

and the GGX Smith function is

$$\Lambda(\omega) = \frac{-1 + \sqrt{1 + \frac{(\omega_g \cdot \omega)^2}{\alpha^2(1 - (\omega_g \cdot \omega)^2)}}}{2}$$

The Blinn-Phong masking-shadowing function [Bli77] is

$$G_2(\omega_i, \omega_o, \omega_g) = \min \left\{ 1, \frac{(\omega_g \cdot \omega_h)(\omega_g \cdot \omega_o)}{\omega_o \cdot \omega_h}, \frac{(\omega_g \cdot \omega_h)(\omega_g \cdot \omega_i)}{\omega_o \cdot \omega_h} \right\}$$

4 THE APPEARANCE OF METAL

The appearance of metal is directly related to the effect of the Fresnel equation. If we have a spectral measurement of the index of refraction η and its absorptive index κ , we can reconstruct its color. So the problem we wish to solve is to choose which wavelength of light to use for display. It might seem implausible, at least initially, to synthesize the index of refraction, but that is exactly what we must do. And therefore we must understand how the display works.

Any modern LCD display is a composition of white light, liquid crystals, and color filters. The liquid crystals control how much light gets to the color filters. We understand that the backlight is not perfectly pure, but it must be good enough for accurate reproduction of

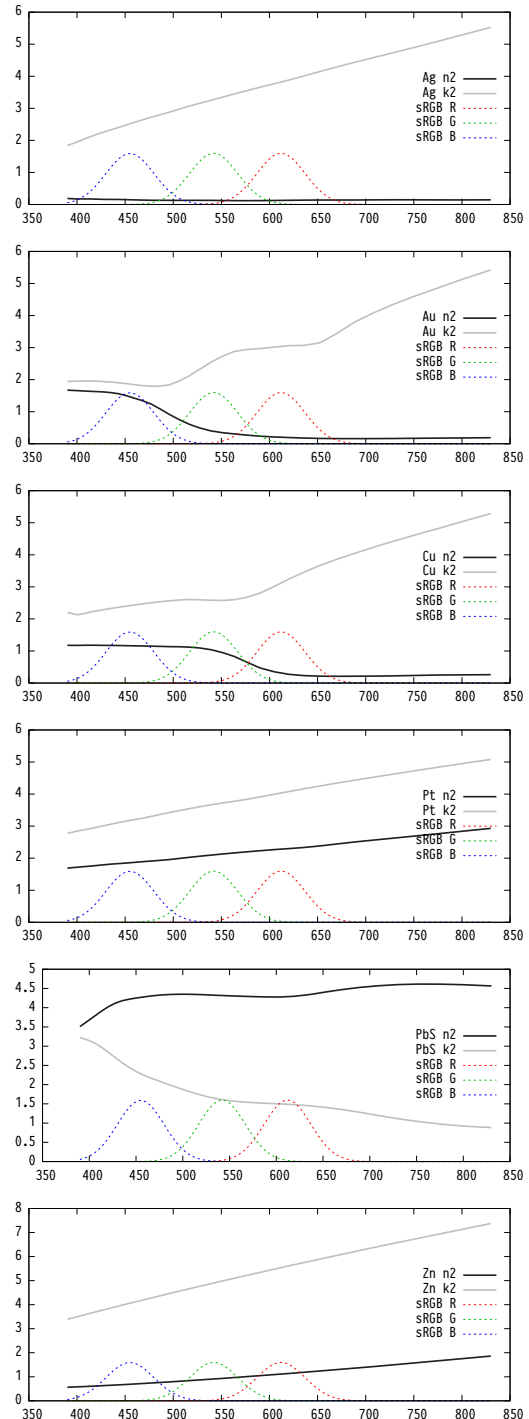


Figure 5: Comparison of index of refraction for materials Ag, Au, Cu, Pt, PbS, and Zn. Dark grey is used for η and light gray for κ . Wavelengths go from 390nm to 830nm.



Figure 6: Comparison of Fresnel using irradiance only. Clockwise from top left: Silver, Gold, Platinum, Zinc, Lead, and Copper.



Figure 7: Comparison of Fresnel with reflections added. Clockwise from top left: Silver, Gold, Platinum, Zinc, Lead, and Copper.

color. The backlight and filters may affect the quality of color reproduction and many professionals choose calibrated monitors for this reason. Calibrated monitors usually have a lookup table so they can target several color spaces. Each designer of any display will choose a light source, liquid crystal display, and color filters depending on factors such as intended audience and target price. Our insight is that the color filters act like Gaussian filters for a given wavelength, rather than just band pass a single wavelength.

So the quality of the display is a function of how good the white light is and the effectiveness of the color

filters. The best display systems are calibrated and measured against standards of gamma correction, and matched against the color space specification such as sRGB, DCI P3, and Adobe RGB. Figure 4 shows that each color space needs to be handled specially if there is a significant change in white point or red, green, and blue xy coordinates.

Let us consider the effect of the color filters on the index of refraction. Consider that instead of the white light and the LCD that we only have a source of light, the metal, and the color filter. The color filter will attenuate the reflection of light from the metal according to its filter shape. A red filter may be a band-pass or low pass filter, the green filter a band-pass filter, and the blue filter either a band-pass or high-pass filter. Let us consider only band-pass filtering as our eyes effectively constrain the lower and upper wavelengths of visible light.

It would be very easy to ignore the color filter design of the display and choose wavelengths based on a set of coordinates. For example, the CIE RGB space defines red as 700nm, green as 546.1nm, and blue as 435.8nm. But sRGB is 612nm, 542nm, and 455nm and Adobe RGB is 612nm, 527nm, and 455nm.

The BRDF in the rendering equation relates how much light is reflected for each of the red, green, and blue wavelengths. The color filter will have the effect of increasingly attenuating how much reflected light is seen as λ moves away from the peak reflection wavelength λ_{filter} . Spectral rendering, which considers the entire visible spectrum, would be the natural choice in ensuring an accurate rendering. To produce a spectral renderer, numerous wavelengths of light are used to evaluate the rendering equation. (As a nod to Newton, we would not be rendering with RGB, we would be rendering with ROYGBIV.) Path tracing is the obvious approach for spectral rendering because it is easy to incorporate into the design. Then once all the samples have been created for the pixel, it is sampled to construct an RGB value. A Gaussian filter is a natural choice to sample the spectral values. More advanced techniques could even pair a tristimulus model to take into account human color perception.

Let us say that λ_c represents the color we want to sample, where c is red, green, or blue. A spectral evaluation for the Fresnel equations for a given wavelength λ_c using the normal Gaussian filter [Bul79] is

$$F_{\lambda_c}(\theta) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\lambda, \theta) \exp\left[-\frac{(\lambda-\lambda_c)^2}{2\sigma^2}\right] d\lambda$$

where σ is the standard deviation of the Gaussian filter. Our filter width is approximately 6σ which is 99.7% of all the area under the Gaussian curve. According to probability theory, 68.27% of the area is between $\pm\sigma$, 95.45% is between $\pm 2\sigma$, and 99.73% is between $\pm 3\sigma$.

If we set σ to be small, only wavelengths around the center wavelength have significant contribution to the final result. This can be seen in Figure 5 where σ is set to 25nm.

We now have two paths we can approach. We can pick three wavelengths and evaluate the Fresnel equations or we can use all wavelengths and spectrally evaluate the Fresnel equations. We propose that instead of evaluating the Fresnel equation at all wavelengths or only choosing three wavelength values, that we instead sample the index of refraction around the wavelengths of the intended display color specification.

The reason for proposing this is twofold. The first reason is that spectral rendering is impractical for many realtime applications, such as mobile devices. And second, for some materials, such as gold or copper, the choice of wavelength for red, green, and blue causes significant changes in appearance. Figure 3 shows that changing blue causes the reflection of the sky to change color.

Before we get into the implementation details, let us quickly describe how our method would be applied. First, you would get access to wavelength dependent index of refraction measurements. Second you would re-sample the data so that it is in even steps (we used steps of $\frac{1}{10}nm$). Third, you would determine the wavelengths of red, green, and blue that are used for the display by using the color specification. Fourth, you would use a Gaussian distribution to sample the index of refraction. Lastly, you could calculate F_0 and use an approximation, like specular color or Schlick's approximation, or you could use a higher precision formula (as we mentioned in section 2). And now let us look at our implementation.

5 IMPLEMENTATION

We implemented a TypeScript and WebGL based renderer in a HTML5 single page application¹ to use all the rendering techniques except for the spectral renderer. We can index an environment map to test both reflections and irradiance. As with most realtime rendering techniques, only a subset of light paths are considered. There is no performance penalty in using our method since it is a preprocessing step for the material. The performance is directly related to the Fresnel equation or approximation chosen.

We use three light paths. First we use the Sun as the primary light source using astronomical calculations to determine a realistic position in the sky. We also calculate the position for the Moon (a simple model without perturbations) for night lighting conditions. This

¹ We extended our HTML5 web application to generate Gnuplot data and charts that can be copy-pasted to create the figures in this paper.

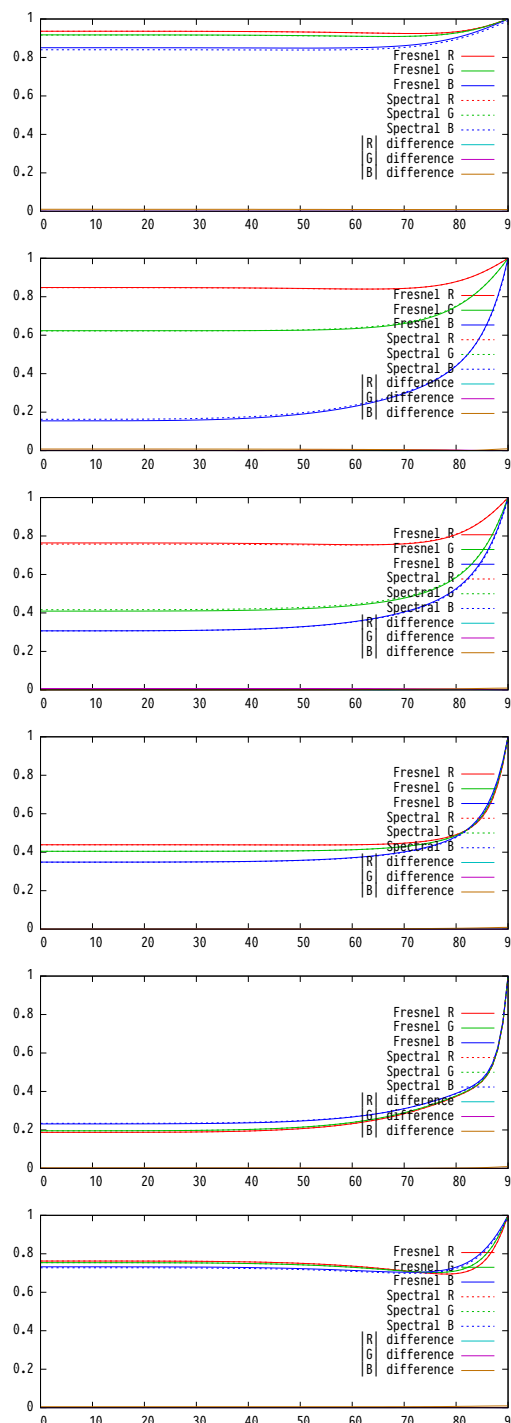


Figure 8: Comparison of relative error of our approximation to the spectral Fresnel functions for materials Ag, Au, Cu, Pt, PbS, and Zn. Wavelengths go from 390nm to 830nm.

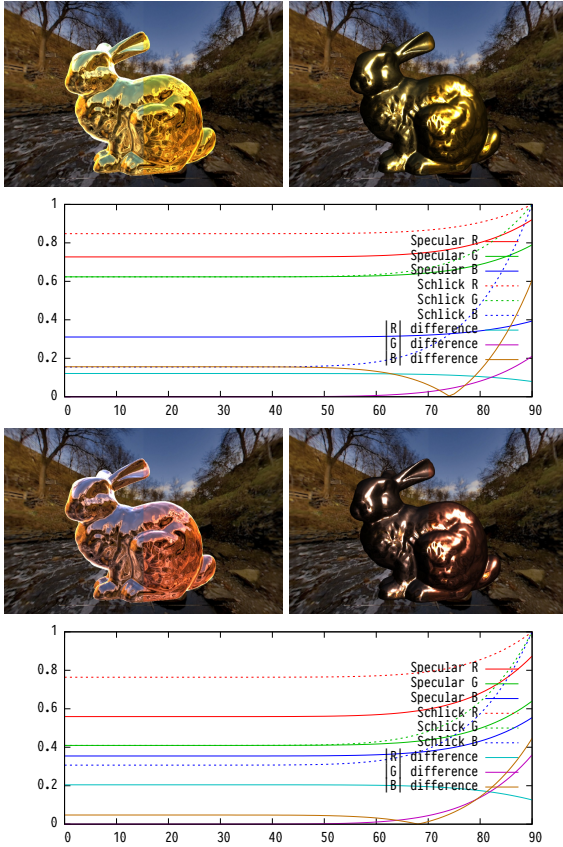


Figure 9: A comparison of using a Fresnel derived specular color k_s (image left halves) and single channel Schlick's approximation (right image halves). The error is shown in the graphs. Gold is top and Copper is bottom with burnished (left) and dull (right) finishes.

also allows for us to animate the motion of the sun to observe the change in reflection. Secondly, we use an environment map to simulate reflection and irradiance. We calculate the reflection vector using the view vector and the normal. We calculate irradiance by using the geometric normal ω_g to index the environment map. Figures 6 and 7 show dull (irradiance) and burnished (reflections) imagery using the sRGB color space for a variety of metals. Charts of their index of refraction are shown in Figure 5.

There are publicly available spectral measurements for a variety of materials [fil18]. Our test application has measurements for many common metals including Silver (Ag), Aluminum (Al), Gold (Au), Chromium (cr), Copper (Cu), Nickel (Ni), Lead Sulfide (PbS), Platinum (Pt), Silicon (Si), Titanium (Ti), Tungsten (W), Zinc (Zn), and Schott SF6HT (glass). We picked these because they have complete spectral profiles.

We allow the user to select from a number of color specifications including sRGB, DCI P3, and Adobe RGB. There is a monochromatic option where the user can set the wavelength individually for red (650 to 740nm),

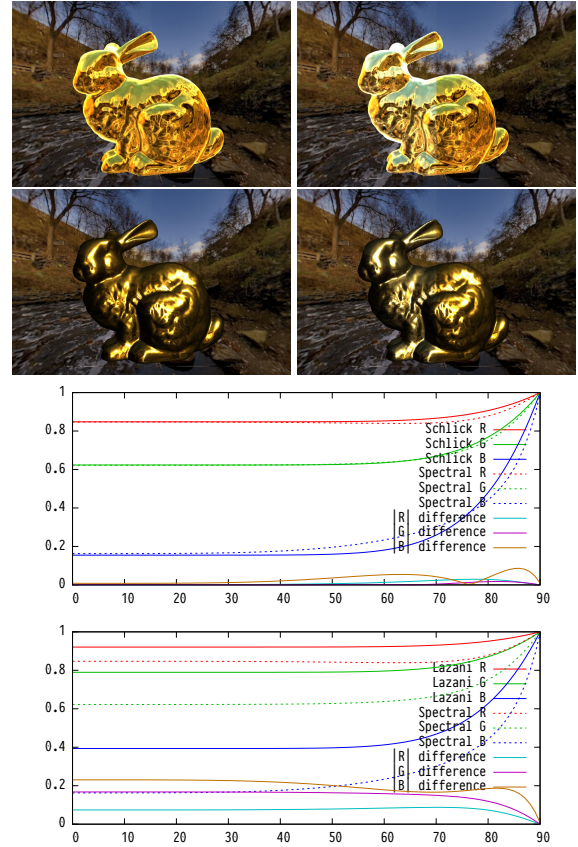


Figure 10: A comparison of Schlick's (left) and LSK method (right) for Gold. LSK requires error compensation, and Schlick's method using precalculated F_0 is very close to reference (right halves of images).

green (520 to 565nm), and blue (435 to 500nm) with a custom width for the Gaussian σ standard deviation. At 0, we use a Dirac delta function to only use the specified wavelength, otherwise we use the normal distribution.

Now we use the color space profile to determine the red, green, and blue wavelengths. We can identify which wavelength of light to use by tracing a line from the white point coordinates w_{xy} through the chromaticity coordinates c_{xy} , where c is red, green, or blue, and find the intersection on the CIE horse shoe which is mapped to wavelength. We can use the relative distance formula

$$d = \frac{y(\mathbf{v}_{2,y} - \mathbf{v}_{1,y}) - x(\mathbf{v}_{2,x} - \mathbf{v}_{1,x}) + \mathbf{v}_{2,x}\mathbf{v}_{1,y} - \mathbf{v}_{2,y}\mathbf{v}_{1,x}}{\|\mathbf{v}_2 - \mathbf{v}_1\|}$$

to calculate the distance to each of the CIE horse shoe points until we minimize d . This method generalizes finding the wavelength for a given white point and chromaticity coordinate. We believe this is a novel contribution of our method.

Next we prepare to resample the wavelengths. We create an array with indexes 3900 to 8300 to correlate to wavelengths 390nm to 830nm, giving 4,401 data points. The data structure contains the wavelength, index of

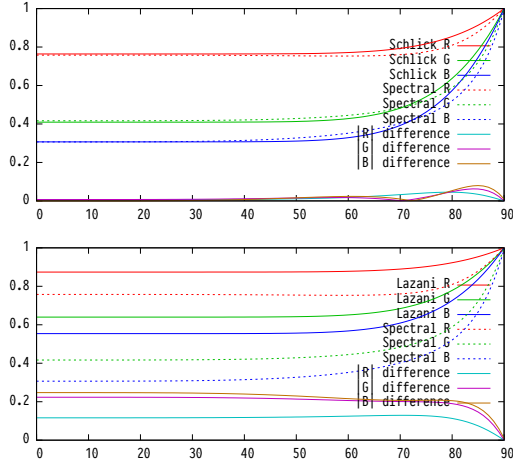


Figure 11: A comparison of Schlick's (left) and LSK method (right) for Copper. LSK requires error compensation, and Schlick's method using precalculated F_0 is very close to reference (right halves of images).

refraction, absorptive index, and RGB weight for the normal distribution function. Our purpose for using 10 samples per nanometer of wavelength is to avoid the situation where we use too few samples statistically. For example, If we use a standard deviation of 12nm, that's only 24 samples—so we increase it by a factor of 10 to avoid that. But first we need to preprocess the input index of refraction data.

We use linear interpolation to resample the original measurements so that they are specified for each tenth of a nanometer in an array. We have not analyzed whether using nearest, linear, or higher forms of sampling would make a huge difference. We think linear is fine unless you need continuous derivatives, which we do not. Then we create a normal distribution calculation for each wavelength data point. If $G(\lambda_i, \sigma^2)$ is the normal distribution function, we then resample the index of refraction

$$\eta' = \frac{1}{10} \sum_{i=3900}^{8300} \eta_i G(\lambda_i, \sigma^2)$$

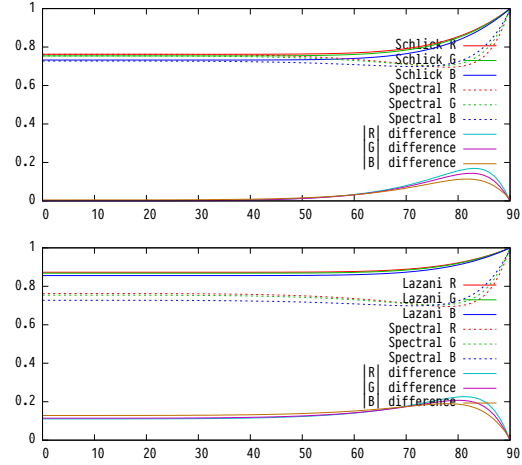


Figure 12: A comparison of Schlick's (left) and LSK method (right) for Zinc. LSK requires error compensation, and Schlick's method using precalculated F_0 is very close to reference (right halves of images).

and absorptive index

$$\kappa' = \frac{1}{10} \sum_{i=3900}^{8300} \kappa_i G(\lambda_i, \sigma^2)$$

for red, green, and blue. We divide by 10 in order to account for using 10 samples per each nm of wavelength. Finally, we use these resampled values for inputs in the Fresnel equation in the shader.

6 RESULTS

If the standard deviation for red, green, and blue is kept reasonable ($\sim 25\text{nm}$), then the error using our method compared with the spectral rendering is acceptable. Figure 8 and Figure 13 shows three separate error lines for red, green, and blue. The error line is the absolute error at each angle for the Fresnel function. Our results show that the index of refraction can be re-synthesized with negligible visible error.

Our method also supports comparing the different approximations to the spectral version. We noticed in Lazányi and Szirmay-Kalos (LSK) [LS05] that they used a multispectral approximation, noting that

Schlick's approximation yielded red tinted results. We understand that this is caused by ignoring the absorptive index which often is significantly greater than the index of refraction. The common way of using Schlick's approximation is to use an accurate Fresnel calculation at 0° , and we show that this is better than the LSK model.

We also note that a common way to approximate gold (i.e. cheating, hacking, or sometimes gross approximations) is to use the specular color k_s . So we present a similar approach and set the specular color $k_s = (F_{0,r}, F_{0,g}, F_{0,b})$ to the base reflection color and then use their approximation using either the average η and κ or even just the green index since our eyes are most sensitive to green. We can use a Dirac-Delta function to eliminate the integral if we are interested in only a single angle (e.g. θ_d) and so the radiance is

$$L_o = k_s L_i(\omega_g \cdot \omega_i) \frac{F(\theta_d) D(\omega_h) G(\omega_i, \omega_o)}{4 \cos \theta_i \cos \theta_o}$$

and

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

where

$$F_0 = \frac{F_r(0^\circ) + F_g(0^\circ) + F_b(0^\circ)}{3}$$

or

$$F_0 = F_g(0^\circ).$$

We show in Figure 9 the copper material using the full Fresnel equation and the specular color approximation. This gets rid of the tinting that Schlick's approximation inherently had (if you mistakenly ignored the complex index of refraction), but at the expense of either underestimating or overestimating reflection at grazing angles. It is not very good and we would not recommend using anything less than the LSK spectral approximation. Perhaps our specular color version may be useful for slow GPUs where only a single channel of Fresnel can be computed.

Schlick's approximation, the approximate, and the exact Fresnel equations look almost identical in practice. LSK is sometimes brighter or dimmer and requires an error compensation factor of

$$-ax(1-x)^\alpha$$

but α was not fully explained in their paper. The idea is to compensate for materials like Zinc or Aluminum which have a dip in their reflectance at near glancing angles. We show the difference between using Schlick's method with F_0 values calculated for red, green, and blue separately and the LSK method in Figures 10, 11, and 12 for gold, copper, and zinc. Their respective comparison with a spectral Fresnel is shown in the graphs, while we use the full Fresnel equation in the images.

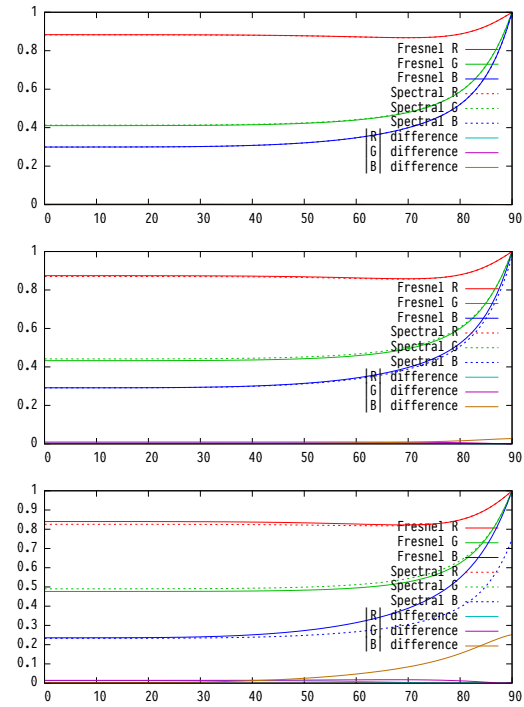


Figure 13: A comparison of setting the filter width to 12nm, 25nm, and 50nm. The error lines show increasing error as filter width increases. The material here is copper.

We find that our method breaks down when setting the filter too wide. Figure 13 shows how setting the filter wide results in greater angle towards incidence. The reason for this is that we are calculating too much blending between red, green, and blue values. If we were to continue widening the filter, we eventually will get a monochromatic reflection value.

7 CONCLUSION AND FUTURE WORK

We have shown that the display is an integral part of getting the Fresnel equations to work correctly for multispectral materials such as metals. We have demonstrated that the complex index of refraction can be sampled so that a spectral rendering is not required for realtime applications. We introduced a novel method of choosing appropriate wavelengths based on the chromaticity specification. We have also created a web based graphics experiment that uses a variety of different materials.

Our results do not have any impact on realtime frame rate performance, but they do enhance the quality of the image because the complex index of refraction is appropriately calculated for RGB color displays. We introduced a specular color method which could be suitable for applications with low CPU/GPU speeds. Finally, we surveyed the available Fresnel equations and

approximations and their suitability in terms of error to a spectral Fresnel equation implementation.

In the future, we hope to continue our work with multispectral materials. This includes work on dispersion and thin films. We also hope to experiment with anisotropic microfacet distribution models. We also plan to extend our sun and moon illumination model to accurately simulate astronomical phenomena to create accurate skylit illumination for both day and night to create realistic illumination estimates for future experiments (the current state of the art is daylight only).

8 REFERENCES

- [Ado98] The adobe rgb (1998) color image encoding. Technical report, Adobe Systems Incorporated, 1998.
- [AMHH08] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-Time Rendering, Third Edition*. CRC Press, 2008.
- [Bli77] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.
- [BS12] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pages 1–7, 2012.
- [Bul79] M.G. Bulmer. *Principles of Statistics*. Dover Books on Mathematics Series. Dover Publications, 1979.
- [Com16] Wikimedia Commons. File:ciexy1931 rec 2020 and rec 709.svg — wikimedia commons, the free media repository, 2016. [Online; accessed 10-March-2018].
- [CT81] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *SIGGRAPH Comput. Graph.*, 15(3):307–316, August 1981.
- [fil18] Refractive index database. Technical report, Filmetrics, Inc., 3 2018. [Online; accessed 23-Jan-2018].
- [Gla95] Andrew S Glassner. *Principles of Digital Image Synthesis*. Elsevier, 1995.
- [Hei14] Eric Heitz. Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques (JCGT)*, 3(2):48–107, June 2014.
- [IEC99] IEC. Multimedia systems and equipment - colour measurement and management - part 2-1: Colour management - default rgb colour space - srgb. Technical report, International Electrotechnical Commission, 1999.
- [JG78] George H. Joblove and Donald Greenberg. Color spaces for computer graphics. *SIGGRAPH Comput. Graph.*, 12(3):20–25, August 1978.
- [Kaj86] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pages 143–150, New York, NY, USA, 1986. ACM.
- [LS05] István Lazányi and László Szirmay-Kalos. Fresnel term approximations for metals. In *The 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2005, WSCG 2005, University of West Bohemia, Campus Bory, Plzen-Bory, Czech Republic, January 31 - February 4, 2005*, pages 77–80, 2005.
- [MG80] Gary W. Meyer and Donald P. Greenberg. Perceptual color spaces for computer graphics. *SIGGRAPH Comput. Graph.*, 14(3):254–261, July 1980.
- [Mod13] Michael F. Modest. *Radiative Heat Transfer*. Academic Press, 3 edition, 2 2013.
- [Nic65] Fred E Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7):767–775, 1965.
- [PH10] Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.
- [Sch94] Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994.
- [SMP11] Rp 431-2:2011 - smpte recommended practice - d-cinema quality - reference projector and environment. *SMPTE RP 431-2:2011*, pages 1–14, April 2011.
- [WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 195–206, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.