Practical Physically Based Shading in Film and Game Production:

Beyond a Simple Physically Based Blinn-Phong Model in Real-Time

Yoshiharu Gotanda
tri-Ace, Inc.
http://research.tri-ace.com/

1. Introduction

We presented our implementation for a physically based Blinn-Phong model [Blinn 1977] at SIGGRAPH 2010, which was:

$$f_r(\mathbf{x},\omega',\omega) = \frac{R_d}{\pi}(1-F_0) + (0.0397436 shininess + 0.0856832)\frac{F_{spec}(F_0)(\mathbf{n}\cdot\frac{\omega'+\omega}{|\omega'+\omega|})^{shininess}}{\max(\mathbf{n}\cdot\omega',\mathbf{n}\cdot\omega)}, \quad (1)$$

where R_d is the diffuse albedo and $F_{spec}(f_0)$ is the Fresnel function with Schlick's approximation using f_0 as the specular reflectance for the normal direction. $\mathbf{n}$ is the normal vector, ω is the outgoing direction and ω' is the incident direction. Since then, physically based shading models have been rapidly adopted in the game industry. There are and will continue to be many titles that use energy-conserving and/or normalized shading models that include physically based parameters. Our physically based Blinn-Phong has similarities to the Cook-Torrance model in [Cook and Torrance 1982]. It is not enough, however, to reproduce the complex appearances of materials in the real world; for example, translucent, rough, layered, and retro-reflective materials. In this course, we would like to introduce our newer models that represent more complex appearances, which could not only be useful for next-generation consoles but also current-generation consoles.

2. Layered Blinn-Phong Model

The first model we implemented is a layered shading model. Layered materials are very important to reproduce realistic results because, in the real world, there are lots of materials that have combinations of different properties consisting of different BRDFs. If artists try to reproduce these materials with a single physically based shading model, there are cases in which it would be more difficult than using an ad-hoc model. This is because in an ad-hoc model the artists are able to adjust the parameters for the desired layered appearance with non-layered shading models without physical limitations. A simple solution for this issue is to implement a layered shading model. Typically, off-line renderers support very flexible layered shading models that can be combinations of any number of layers and BRDF models. However, due to performance issues, for real-time rendering on not-so-powerful GPUs, we chose only dual-layered and limited shading models. Even under these restrictions, a layered model can reproduce much more realistic results than any other single-layered shading model.

We referred to [Weidlich and Wilkie 2009] and [Weidlich and Wilkie 2011] for our layered model. Their implementation was very simple, even for real-time rendering, and we were able to approximate some elements to meet the performance requirements for current-generation consoles. In the original implementation,

they used the Bouguer-Lambert-Beer law to represent one light-absorption phenomenon by the top layer. However, it uses an exponential function and refracted distances, which are computationally expensive to evaluate in the shader. Therefore, we firstly approximated them with a linear function and non-refracted distance and used our own physically based Blinn-Phong instead of Cook-Torrance. According to the original implementation, the light absorption can be computed as:

$$I = I_0 e^{-\alpha l} \tag{2}$$

$$l = d \cdot \left(\frac{1}{\cos\theta_{i'}} + \frac{1}{\cos\theta_{r'}}\right),$$

where I is the intensity, I_0 is the incident intensity, α is the absorption coefficient, d is the thickness of the layer, $\theta_{i'}$ is the angle of refracted incident light, and $\theta_{r'}$ is the angle of outgoing refracted light. Our implementation approximates these equations with the following equation:

$$I = I_0 \cdot \max\left(0, 1 - \alpha'\left(\frac{1}{N \cdot L} + \frac{1}{N \cdot E}\right)\right), \tag{3}$$

where N is the normal vector, α' is the modified absorption coefficient representing a combination and adjustment of the original coefficient and thickness, L is the light vector, and E is the eye vector.

Secondly, we approximated the Fresnel term computation in the bottom layer. Since the bottom-layer BRDF is evaluated with the light passing through the top layer, the Fresnel term in the bottom layer is computed with the following equation:

$$F_{bottom}(n_1, n_2) = \left(1 - F(n_1)\right)\left(F\left(\frac{n_2}{n_1}\right)\right)\left(1 - F\left(\frac{1}{n_1}\right)\right), \tag{4}$$

where $F(n)$ is the Fresnel equation, n_1 is the refractive index of the top layer and n_2 is the refractive index of the bottom layer. Figure 1 shows an example graph of the Fresnel component in the top and bottom layers. The red line represents the component in the bottom layer. Notice that it is almost a flat line and can therefore be approximated with a constant such as:

$$F_{transmittance}(n_1) = \left(1 - F(n_1)\right) \tag{5}$$

$$F_{bottom}(n_1, n_2) = \left(F_0\left(\frac{n_2}{n_1}\right)\right)\left(1 - F_0\left(\frac{1}{n_1}\right)\right),$$

where $F_0(n)$ is the Fresnel equation for the normal direction, $F_{transmittance}(n)$ is the transmittance from the top layer by the Fresnel term, and $F_{bottom}(n_1, n_2)$ is the reflectance of the bottom layer and the Fresnel term.

Figure 2 shows a comparison between the original absorption implementation and our approximated implementations.

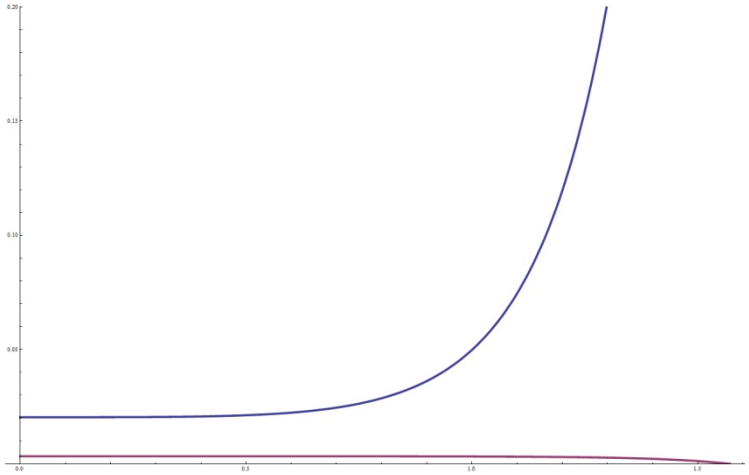


Figure 1: *A graph showing the Fresnel components. The blue line is the component of the top layer with a refractive index of 1.33333. The red line is the bottom layer with 1.5.*

Figure 2: *A comparison with different shaders. The piece on the left is rendered with the original implementation taking 0.53ms on a PlayStation 3 at 1280x720 with a single directional light. The piece in the middle is rendered with our approximated implementation, taking 0.37ms, and the piece on the right is rendered with our approximated implementation and no absorption, taking 0.34ms. A single-layered Blinn-Phong (standard physically based shader) takes 0.33ms. These numbers demonstrate that our approximation provides good results for both rendering and performance.*

Supporting image-based lighting for layered materials is very important. In typical cases, image-based lighting with layered materials shows more impressive results than a punctual light with layered materials. The absorption component must be integrated with a BRDF according to the rendering equation. However, because it is also computationally expensive, it should be approximated. We used the same approach as AmbientBRDF as described in [Gotanda 2010].

For the specular component of the bottom layer, although the integral is dependent on the shininess value in our Blinn-Phong model, we decided to coarsely approximate the integral with the following equation, which

regards the specular component as a perfect mirror reflection. In this case, $N \cdot L$ is equivalent to $N \cdot E$. Therefore, the reciprocal term in our approximated absorption equation of Equation 3 becomes:

$$(\frac{1}{N \cdot L} + \frac{1}{N \cdot E}) = (\frac{1}{N \cdot E} + \frac{1}{N \cdot E}), \tag{6}$$

Thus, the specular intensity from the bottom layer by image-based lighting can be computed with:

$$I_s = IBL_{specular} \cdot F_{bottom}(n_1, n_2) \max\left(0,1 - \alpha'(\frac{2}{N \cdot E})\right). \tag{7}$$

To compute the diffuse component, we analytically integrated the reciprocal term in Equation 3 over the hemisphere with Lambert as follows:

$$\int_\Omega \left(\frac{1}{\cos\theta_r} + \frac{1}{\cos\theta_i}\right)\frac{1}{\pi}\cos\theta_i d\omega = \frac{1}{\pi}\int_{-\pi}^{\pi}\int_0^{\frac{\pi}{2}}\left(\frac{1}{\cos\theta_r} + \frac{1}{\cos\theta_i}\right)\cos\theta_i \sin d\theta_i d\theta_i d\varphi_i \tag{8}$$

$$= \frac{1}{\pi}\int_{-\pi}^{\pi}\frac{1}{2}(2 + \frac{1}{\cos\theta_r})d\varphi_i$$

$$= 2 + \frac{1}{\cos\theta_r}.$$

As a result, the diffuse intensity from the bottom layer by image-based lighting can be computed with:

$$I_d = IBL_{diffuse} \cdot \left(1 - F_{bottom}(n_1, n_2)\right)\max\left(0,1 - \alpha'(2 + \frac{1}{N \cdot E})\right). \tag{9}$$

Figure 3 shows the results and performance of our layered material implementation for image-based lighting.

Figure 3(a): *Images with a material that represents lacquer and coating. Our layered-material system represents this kind of a complex appearance very well, even with image-based lighting.*

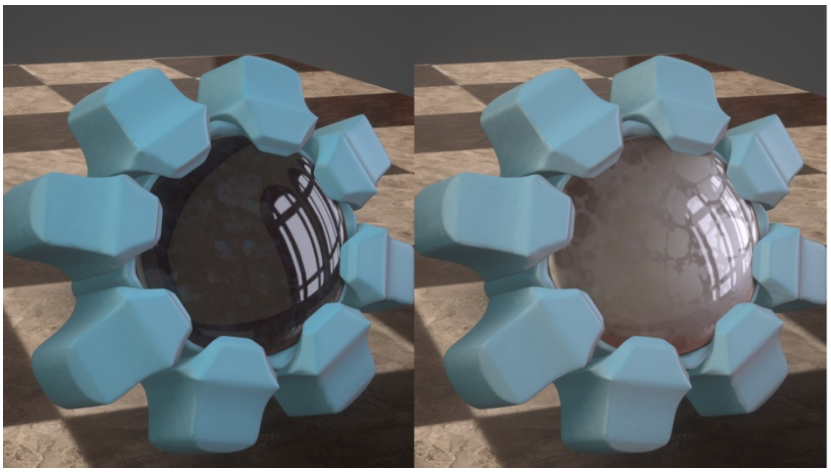


Figure 3(b): *A performance comparison. The image on the left is rendered with the single-layered Blinn-Phong model and image-based lighting, taking 3.00ms on a PlayStation 3. The image on the right is rendered with the dual-layered Blinn-Phong model with the same lighting, taking 4.28ms.*

3. Oren-Nayar Implementation

The second model we implemented is an Oren-Nayar model, because pure Lambertian diffuse surfaces infrequently exist in the real world. Especially for rough materials (e.g. when shininess is less than 30), a matte diffuse representation is more important than the specular component. Oren-Nayar was introduced in [Oren and Nayar 1994] and was designed as a diffuse reflectance model with a Lambertian surface and Torrance-Sparrow V-cavity model. Cook-Torrance is a specular model using the Torrance-Sparrow V-cavity model and it only takes into account reflections from the microfacets towards the viewer. As a result, Cook-Torrance is a relatively simpler model than Oren-Nayar, because Oren-Nayar has to take into account Lambertian diffuse reflection (hemispherical reflections) from all microfacets. Since the integral becomes much more complicated, the Oren-Nayar model was numerically fitted to the integral in the paper. The Oren-Nayar model (Equation 27 in the original paper) is too complicated to evaluate in real-time, so we used the "Qualitative Model" offered in Equation 30 in the paper as follows:

$$L_r(\theta_r,\theta_i,\phi_r-\phi_i;\sigma)=\frac{\rho}{\pi}E_0\cos\theta_i\left(A+B\,\mathrm{Max}\left(0,\cos(\phi_r-\phi_i)\right)\sin\alpha\tan\beta\right) \tag{10}$$

$$A=1-0.5\frac{\sigma^2}{\sigma^2+0.33},\quad B=0.45\frac{\sigma^2}{\sigma^2+0.09},\quad \alpha=\mathrm{Max}(\theta_r,\theta_i),\quad \beta=\mathrm{Min}(\theta_r,\theta_i),$$

where ϕ_r and ϕ_i are projected angles onto the tangent plane, ρ is the diffuse albedo, E_0 is the incoming irradiance, and σ is roughness. In order to make this equation shader-friendly, the equation can be rewritten with familiar expressions and dot products as follows:

$$L_r=\frac{\rho}{\pi}E_0(N\cdot L)\left((1.0-0.5\frac{\sigma^2}{\sigma^2+0.33})+(0.45\frac{\sigma^2}{\sigma^2+0.09})\right.$$
$$\left.\mathrm{Max}(0,\frac{E-N(N\cdot E)}{\|E-N(N\cdot E)\|}\cdot\frac{L-N(N\cdot L)}{\|L-N(N\cdot L)\|})\frac{\sqrt{(1-\mathrm{Max}(N\cdot L,N\cdot E)^2)(1-\mathrm{Min}(N\cdot L,N\cdot E)^2)}}{\mathrm{Max}(N\cdot L,N\cdot E)}\right), \tag{11}$$

As this equation still seems too expensive to evaluate in a shader at run-time, typically it is approximated or some terms are baked into textures. However, before falling back on these methods, we try to simplify the equation instead.

In Equation 11, there are max and min functions in the square root in the numerator of the last term. When $N \cdot L$ is greater than $N \cdot E$, the max chooses $N \cdot L$ and the min chooses $N \cdot E$. In the other case, the max chooses $N \cdot E$ and the min chooses $N \cdot L$. Since these functions always choose $N \cdot L$ and $N \cdot E$, the equation can be simplified as follows:

$$L_r = \frac{\rho}{\pi} E_0 (N \cdot L)((1.0 - 0.5\frac{\sigma^2}{\sigma^2 + 0.33}) + (0.45\frac{\sigma^2}{\sigma^2 + 0.09}) \\ \text{Max}(0, \frac{E - N(N \cdot E)}{\left\|E - N(N \cdot E)\right\|} \cdot \frac{L - N(N \cdot L)}{\left\|L - N(N \cdot L)\right\|}) \frac{\sqrt{(1 - (N \cdot L)^2)(1 - (N \cdot E)^2)}}{\text{Max}(N \cdot L, N \cdot E)}) \quad . \tag{12}$$

For the later operation, the first $N \cdot L$ is expanded as follows:

$$L_r = \frac{\rho}{\pi} E_0 ((N \cdot L)(1.0 - 0.5\frac{\sigma^2}{\sigma^2 + 0.33}) + (0.45\frac{\sigma^2}{\sigma^2 + 0.09}) \\ \text{Max}(0, \frac{E - N(N \cdot E)}{\left\|E - N(N \cdot E)\right\|} \cdot \frac{L - N(N \cdot L)}{\left\|L - N(N \cdot L)\right\|}) \frac{(N \cdot L)\sqrt{(1 - (N \cdot L)^2)(1 - (N \cdot E)^2)}}{\text{Max}(N \cdot L, N \cdot E)})) \quad . \tag{13}$$

The fractional term excluding the square root term can be simplified as follows:

$$\frac{N \cdot L}{\text{Max}(N \cdot L, N \cdot E)} = \text{Min}(\frac{N \cdot L}{N \cdot L}, \frac{N \cdot L}{N \cdot E}) \tag{14}$$

$$= \text{Min}(1, \frac{N \cdot L}{N \cdot E}) .$$

Substituting this formula into Equation 13, the equation then becomes:

$$L_r = \frac{\rho}{\pi} E_0 ((N \cdot L)(1.0 - 0.5\frac{\sigma^2}{\sigma^2 + 0.33}) + (0.45\frac{\sigma^2}{\sigma^2 + 0.09}) \\ \text{Max}(0, \frac{E - N(N \cdot E)}{\left\|E - N(N \cdot E)\right\|} \cdot \frac{L - N(N \cdot L)}{\left\|L - N(N \cdot L)\right\|}) \sqrt{(1 - (N \cdot L)^2)(1 - (N \cdot E)^2)}\ \text{Min}(1, \frac{N \cdot L}{N \cdot E})) \quad . \tag{15}$$

Recalling the relation of $\cos(\phi_r - \phi_i)$ in Equation 10 and 11, the relational expression becomes:

$$\frac{(E - N(N \cdot E)) \cdot (L - N(N \cdot L))}{\left\|E - N(N \cdot E)\right\| \cdot \left\|L - N(N \cdot L)\right\|} = \cos(\phi_r - \phi_i) . \tag{16}$$

In order to simplify the left side of the equation, we derive another relational expression from Equation 37 and 38 in the original paper as follows:

$$\cos\theta_r \cos\theta_i + \sin\theta_r \sin\theta_i \cos(\phi_r - \phi_i) = \cos(\theta_r - \theta_i) . \tag{17}$$

Rewriting this equation using vector expressions, the equation becomes:

$$(N \cdot E)(N \cdot L) + \sqrt{1 - (N \cdot E)^2}\sqrt{1 - (N \cdot L)^2}\cos(\phi_r - \phi_i) = E \cdot L \tag{18}$$

$$\cos(\phi_r - \phi_i) = \frac{E \cdot L + (N \cdot E)(N \cdot L)}{\sqrt{(1-(N \cdot E)^2)(1-(N \cdot L)^2)}}.$$

This relational expression can be substituted into Equation 15:

$$L_r = \frac{\rho}{\pi} E_0 ((N \cdot L)(1.0 - 0.5\frac{\sigma^2}{\sigma^2 + 0.33}) + (0.45\frac{\sigma^2}{\sigma^2 + 0.09}) \\ \text{Max}(0, \frac{E \cdot L - (N \cdot E)(N \cdot L)}{\sqrt{(1-(N \cdot L)^2)(1-(N \cdot E)^2)}}) \sqrt{(1-(N \cdot L)^2)(1-(N \cdot E)^2)}\ \text{Min}(1, \frac{N \cdot L}{N \cdot E})) \quad (19)$$

And then, since $\sqrt{(1-(N \cdot L)^2)(1-(N \cdot E)^2)} \geq 0$, Equation 19 can be simplified as follows:

$$L_r = \frac{\rho}{\pi} E_0 \left[(N \cdot L)(1.0 - 0.5\frac{\sigma^2}{\sigma^2 + 0.33}) + \left((0.45\frac{\sigma^2}{\sigma^2 + 0.09}) \text{Max}(0, E \cdot L - (N \cdot E)(N \cdot L))\ \text{Min}(1, \frac{N \cdot L}{N \cdot E}) \right) \right]. \quad (20)$$

Finally, this equation has been dramatically simplified from Equation 11 and it seems simple enough to evaluate in real-time. The first term containing $N \cdot L$ is just a weighted Lambert and the next term is the key component of Oren-Nayar. The last min function is the same term as in our Blinn-Phong model, which appears in the Neumann-Neumann geometry attenuation factor [Neumann and Neumann 1999]. Therefore, many terms in Equation 20 are reused from other parts of the shader. As a result, this model is computationally inexpensive enough for real-time.

Both the specular and diffuse components are based on the same microfacet model as the "Torrance-Sparrow" V-cavity model. So, if the same shininess (roughness) value can be used for both components, both the data size and computational time are reduced at run-time. When the sizes of microfacets are close to the wavelengths of the visible light, the behaviors of the specular and diffuse components are different. However, for other cases, they behave the same way, so the shininess (specular power) values can be converted to roughness (σ) values in Equation 20 using following equation:

$$\sigma = \sqrt{\frac{2}{shininess}}. \quad (21)$$

This equation can be substituted into Equation 20 to simplify it further:

$$L_r = \frac{\rho}{\pi} E_0 \left[(N \cdot L)(1 - 0.5\frac{\frac{2}{shi}}{\frac{2}{shi} + 0.33}) + \left((0.45\frac{\frac{2}{shi}}{\frac{2}{shi} + 0.09}) \text{Max}(0, E \cdot L - (N \cdot E)(N \cdot L))\ \text{Min}(1, \frac{N \cdot L}{N \cdot E}) \right) \right]. \quad (22)$$

Then, this equation is again simplified to make it GPU-shader friendly as follows:

$$L_r = \frac{\rho}{\pi} E_0 \left[(N \cdot L)(1 - \frac{1}{2 + 0.33shi}) + \left(\frac{1}{2.22222 + 0.1shi} \text{Max}(0, E \cdot L - (N \cdot E)(N \cdot L))\ \text{Min}(1, \frac{N \cdot L}{N \cdot E}) \right) \right]. \quad (23)$$

As a result, all squares and two multiplications can be removed from Equation 20.

Equation 23 has been simplified enough to use in real-time. However, it is based on the qualitative model of Oren-Nayar. Figure 4 shows the difference between the qualitative and original models. This graph reveals problems in the qualitative model. The first problem is that the qualitative model behaves the same as the Lambert model causing a flat appearance in the case of backward lighting. The second problem is that the

model doesn't contain an inter-reflection component, which causes a slightly darker appearance.

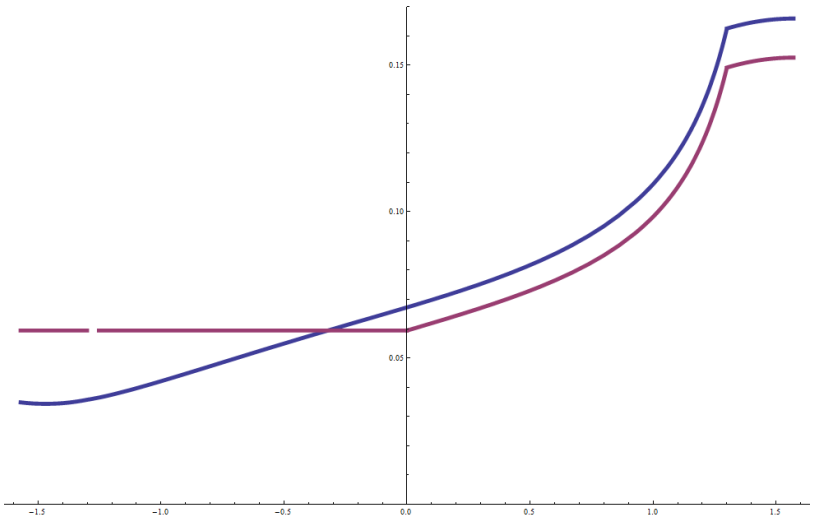


Figure 4: *The graph of the difference between the qualitative (red) and the original models (blue) of the Oren-Nayar model. The horizontal axis is (E·N), which means that the qualitative model behaves the same as the Lambert model in the case of backward lighting.*

To solve this problem, we adjust Equation 23 as follows:

$$L_r = \frac{\rho}{\pi} E_0 \left[(N \cdot L)(1 - \frac{1}{2 + 0.65 shi}) + \left(\frac{1}{2.22222 + 0.1 shi} (E \cdot L - (N \cdot E)(N \cdot L)) \operatorname{Min}(1, \frac{N \cdot L}{N \cdot E}) \right) \right]. \quad (24)$$

We adjusted the coefficient in the first fraction and removed the first max function in Equation 23. Figure 5 shows the difference between Equation 24 and the original Oren-Nayar model.

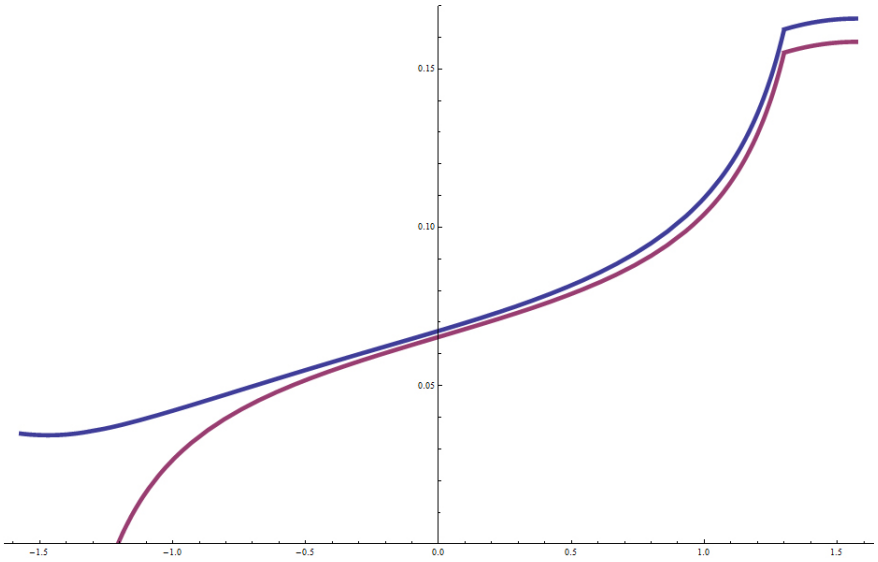


Figure 5: *The graph of the difference between our slightly improved (red) and the original (blue) Oren-Nayar models. Our slightly improved model still has a problem on the left side of the graph.*

When carefully looking at Equation 27 in the original paper, the coefficient C_2 is changed with respect to the

sign of cos(ϕ_r-ϕ_i). Following suit, Equation 24 can be modified to:

$$L_r = \begin{cases} \text{if } (E\cdot L-(N\cdot E)(N\cdot L))\geq 0 \\ \dfrac{\rho}{\pi}E_0\left[(N\cdot L)(1-\dfrac{1}{2+0.65shi})+\left(\dfrac{1}{2.22222+0.1shi}(E\cdot L-(N\cdot E)(N\cdot L))\,\mathrm{Min}(1,\dfrac{N\cdot L}{N\cdot E})\right)\right] \\ \text{otherwise} \\ \dfrac{\rho}{\pi}E_0\left[(N\cdot L)(1-\dfrac{1}{2+0.65shi})+\left(\dfrac{1}{2.22222+0.1shi}(E\cdot L-(N\cdot E)(N\cdot L))(N\cdot L)\right)\right] \end{cases} \quad (25)$$

Figure 6 shows the result of Equation 25. Compared to the flat look within the qualitative model, our improved model can represent smoother appearances closer to those of the original Oren-Nayar model.

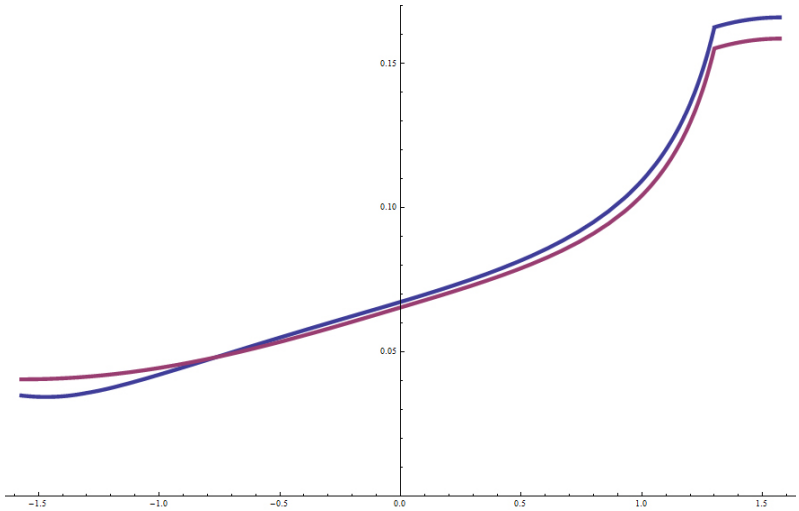


Figure 6(a): *The graph of the difference between our improved (red) and the original (blue) Oren-Nayar models. Our model seems to be able to be further improved by adjusting the coefficient in the first fraction in order to get closer to the original model. However, this graph shows the value of functions in the case where ρ is 0.9. The amount of the inter-reflection component changes with respect to the albedo, so this is plausible in practice.*

Figure 6(b): *A comparison between the qualitative (left) and our improved (right) models. It may be difficult to distinguish the difference at a glance. Compared to the qualitative model on the left, the bottom side of the road with our improved model looks darker, as shown in the graph in the Figure 4(a). Not only is there a difference in brightness, but our improved model also makes the shading result richer than the qualitative model.*

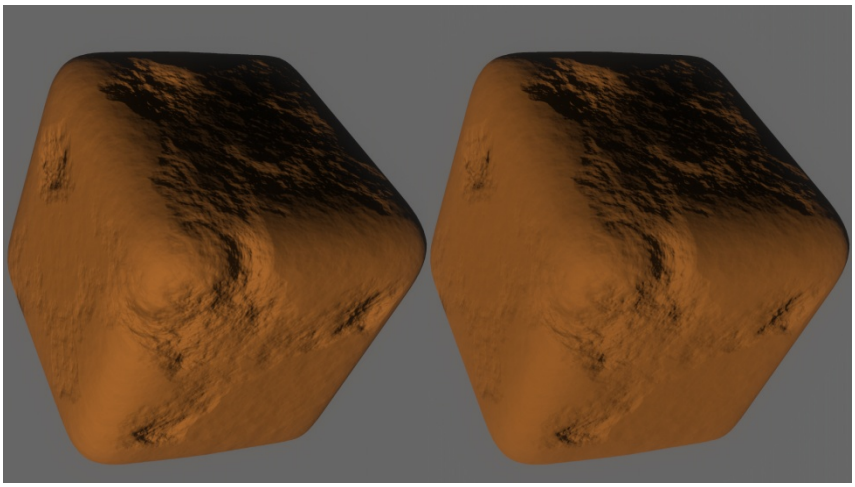


Figure 6(c): *A performance comparison between our physically based Blinn-Phong-Lambert model (left) and our Blinn-Phong-Oren-Nayar (right). Scenes are rendered with one directional light on a PlayStation 3 at 1280x720. The left box is rendered in 0.97ms and the right is rendered in 1.25ms.*

As mentioned in the layered Blinn-Phong section, image-based lighting for Oren-Nayar is also important because it is used within our rendering pipeline to achieve area lighting. However, it is difficult to achieve image-based lighting for Oren-Nayar using environment mapping, because Equation 25 is computed with respect to $N \cdot L$, $N \cdot E$ and $E \cdot L$. If it is implemented using environment mapping within the shader, a multi-dimensional cube map is required. In our case, since diffuse shading is approximated with spherical harmonic lighting, we can use spherical harmonic coefficients to reproduce the behavior of the Oren-Nayar model.

The Oren-Nayar model has some characteristics that Lambert doesn't, such as a more "matte" appearance, view-dependency, a shadowing/masking factor, and an off-peak diffuse (retro-reflective) component. We chose a few notable behaviors that can be reproduced with spherical harmonic lighting. A view-dependent component is very important for representing certain distinctive characteristics compared to Lambert, because it gives the feel of a very rough specular component, despite being only diffuse. The shadow and masking factors change the brightness of the shading result. A retro-reflective component is difficult to represent with low-order spherical harmonic coefficients. If there are high-order spherical harmonic lights, bending the normal vector used for fetching the environmental map would work to reproduce the retro-reflective characteristic.

Firstly, to reproduce the shadowing and masking factors, a scale factor for the DC component in the spherical harmonics is computed with a later equation. To that end, integrating the shading model to analyze the behavior of shadowing and masking factors is as follows:

$$f(\theta_r) = \int_{\Omega} L_r(\theta_i, \theta_r) d\omega_i \tag{26}$$

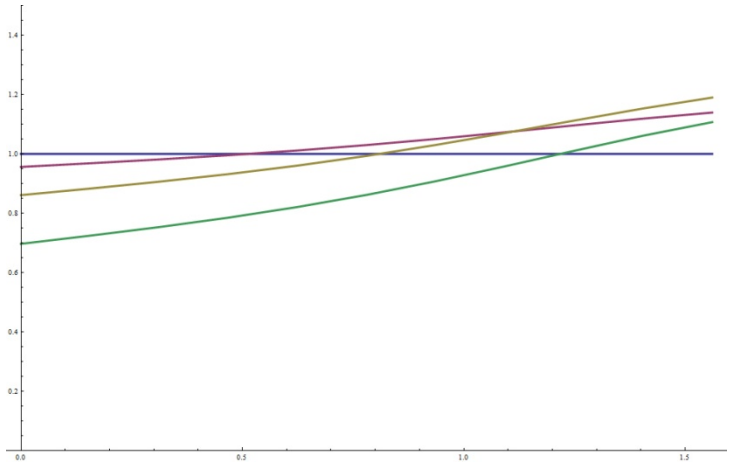


Figure 7: *Results of the integral in Equation 26 with different σ (0.0:blue, 0.25:red, 0.5:yellow, 1.0:green).*

Figure 7 shows the result of this integral[1]. We looked for an equation to fit this integral and the result is as follows:

$$SH'_{DC} = \left(f(shi) + \frac{1}{2} g(shi) \left(1 + \frac{2\arccos(N \cdot E)}{\pi} - N \cdot E \right) \right) SH_{DC}, \tag{27}$$

where $f(shi) = (1 - \frac{1}{2 + 0.65shi})$ and $g(shi) = (\frac{1}{2.22222 + 0.1shi})$.

Figure 8 shows the result of this equation. Though the result of our fitted model is good enough, it is computationally expensive because of the arccos in the equation. The following equation can be used as a compromise for a real-time implementation:

$$SH'_{DC} = \left(f(shi) + g(shi)(1 - N \cdot E) \right) SH_{DC}. \tag{28}$$

Figure 9 shows the result of this equation. The fit is not as good compared to that of Equation 27, but it is faster to evaluate and close enough to reproduce the characteristics of Oren-Nayar.

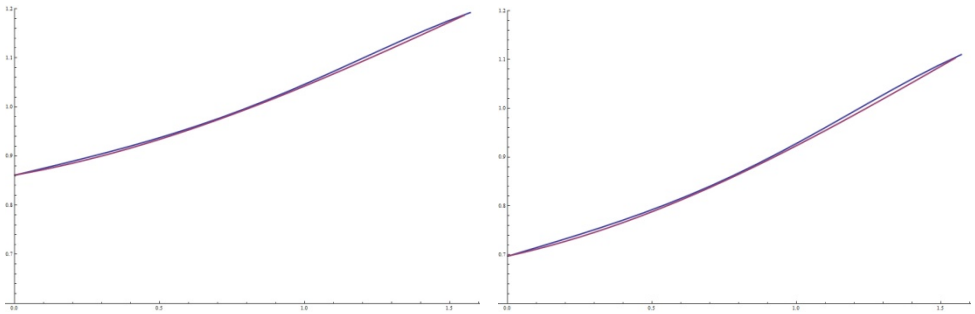


Figure 8: *The graph of the integral result with σ=0.5 (left) and σ=1.0(right). The blue line shows the integral result and the red line shows the fitted model.*

---

[1] When looking at the figure, you can recognize that this shading model violates the law of energy conservation. Although you can adjust the coefficients in the shading model to follow it, as a physically based shading model, it may break the relationship between real roughness values and the shading result of the Oren-Nayar model. This graph shows the integral results with ρ = 1.0. This model violates the law of energy conservation with high ρ values and a narrow range of values for σ. If you use physically reasonable values (especially for ρ) this model doesn't violate the law of energy conservation.

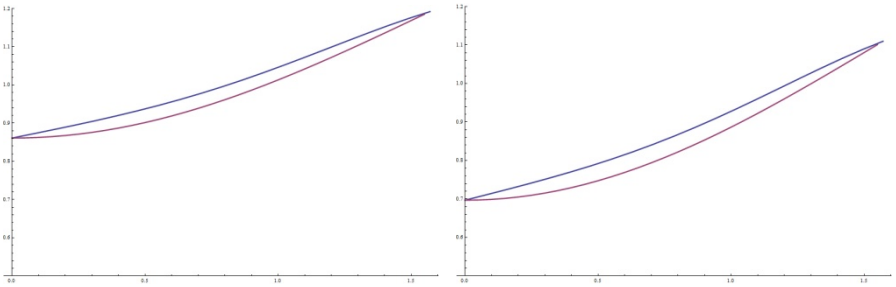


Figure 9: *The graph of the compromised equation (red line) with σ=0.5 (left) and σ=1.0(right) compared to the integral result (blue line).*

Secondly, to reproduce the characteristic of the matte appearance of Oren-Nayar, we analyze the behavior (shape) of Oren-Nayar curves with different parameters. With the analysis in Figure 10, the variance of Oren-Nayar changes from Lambert with respect to $\sigma = 0$ to a flat curve with respect to $N \cdot E = 1$ and $\sigma = 1$. In order to reproduce this characteristic, we derive the following equation for the linear components in spherical harmonics:

$$SH'_{linear} = S(shi, N \cdot E) SH_{linear} \tag{29}$$

$$S(s,x) = f(s) + \big(f(s) - 1\big)\big(1 - x\big).$$

This function S has following characteristic; it becomes 1.0 in the case where $\sigma = 0$ with $N \cdot E = 0$ or $N \cdot E = 1$. It becomes 0.7 in the case where $\sigma = 1$ with $N \cdot E = 0$. It becomes 0.4 in the case where $\sigma = 1$ with $N \cdot E = 1$. The value is linearly interpolated between parameters. If the linear coefficients in spherical harmonics become zero, spherical harmonic lighting becomes a constant, which means that the shading becomes flat and matte. This equation controls how matte the result of spherical harmonic lighting is with respect to σ and $N \cdot E$. Therefore, this equation can represent the view-dependent matte characteristic of Oren-Nayar. Figure 11 shows the results of our implementation of Oren-Nayar with image-based lighting.

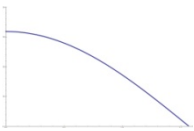
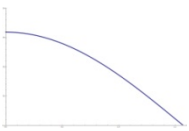
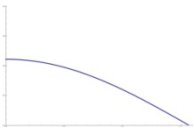
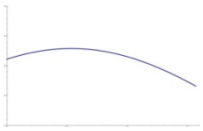
	N·E = 1	***N·E = 0***
σ = 0	*Equivalent to Lambert* 	*Equivalent to Lambert* 
σ = 1	*0.7 * Lambert* 	*Comparatively Flat* 

Figure 10: *Analysis of Oren-Nayar with different parameters. This table represents the relationship between different $N \cdot E$ and σ values in the case where $\phi_r - \phi_i$ is small enough (inside the plane of incidence). This table demonstrates this characteristic in extreme cases. If the average case is required, you can modify Equation 29 to halve its effect.*

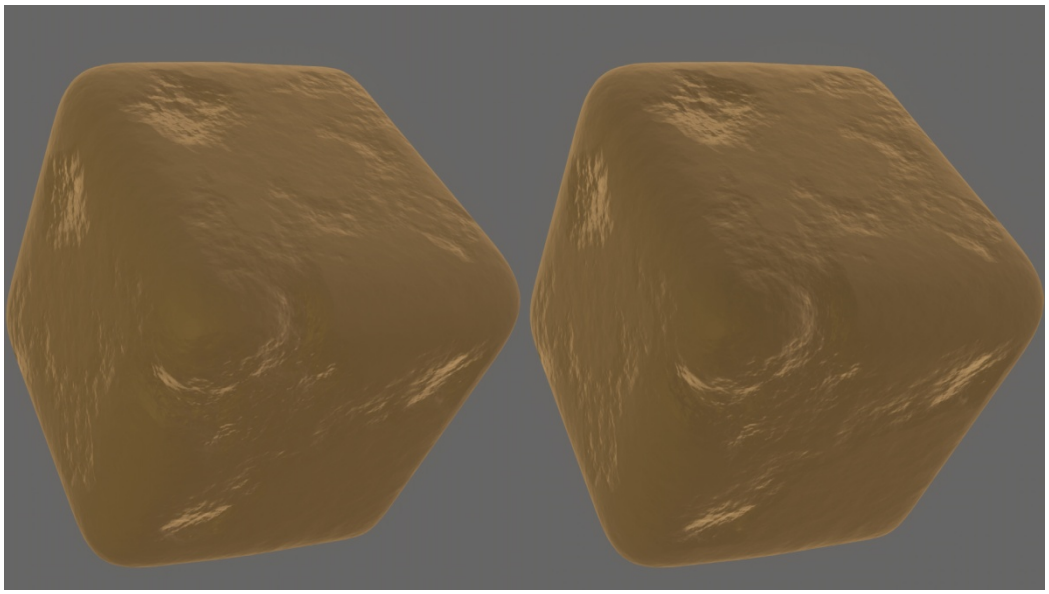


Figure 11: *A comparison of image-based lighting with our physically based Blinn-Phong-Lambert (left) and Blinn-Phong-Oren-Nayar (right). It may be difficult to distinguish the difference with this screenshot. The result varies with respect to the view angle, so you would be able to see the difference as the view angle changes. The performance is 1.35ms (left) and 1.62ms (right) on a PlayStation 3 at 1280x720.*

4. Applications

A layered shading model, Oren-Nayar, and the combination of these models is both very useful and practical to represent the realistic appearance of real-world materials. For example, though our physically based Blinn-Phong has similar characteristics to those of Cook-Torrance and is enough to represent human skin, some subsurface scattering methods such as texture-space or screen-space diffusion are still useful to represent human skin shading. These methods have become very popular for video games recently. However, in the real world, human skin has a much rougher look than Lambertian surfaces. If artists try to represent this matte appearance just with Lambert and subsurface scattering, they may apply too much translucency, causing a blurry appearance. The result looks like a wax figure. Figure 12 shows an extreme example of this case.

Our layered shading model and the Oren-Nayar model both easily solve this problem. As a simple method to simulate human skin, the top layer is used to represent sweat or oil, which can produce a shiny appearance, and the bottom layer is used to represent the matte appearance of human skin itself, with a matte specular component and matte view-dependent diffuse components due to the rough surface. Even though this model can achieve more natural results than with only our physically based Blinn-Phong-Lambert or with heavy subsurface scattering, adding a small amount (almost invisible) of subsurface scattering helps add more realism. Figure 13 shows these results.

Finally, Figure 14 shows a performance comparison with different configurations of shaders among our proposed shading models.

Figure 12: *A comparison of our physically based Blinn-Phong model (left) and screen-space subsurface scattering (right). Too much translucency seems too blurry and much like a wax figure. (This is an extreme example, exaggerated for ease in order to distinguish the problem).*

Figure 13: *A comparison of our layered model with our modified Oren-Nayar model (left), adding a small amount of subsurface scattering (right). Both have more natural results than the results in Figure 12, and if saving computational time is necessary, no subsurface scattering (left) is more acceptable than too much subsurface scattering (the image on the right in Figure 12).*

Blinn-Phong-Lambert	*Blinn-Phong-Oren-Nayar*	*Layered Blinn-Phong-Lambert*	*Layered Blinn-Phong-Oren-Nayar*
6.87ms	7.17ms	7.55ms	7.82ms

Figure 14: *A comparison of our layered model and our modified Oren-Nayar model with different configurations. The scene is rendered with one directional light and one image-based light on a PlayStation 3 at 1280x720.*

5. Conclusion

Our proposed approximated layer model and modified Oren-Nayar model are both very practical even for current-generation consoles, which implies that these models could be used as default models for next-generation consoles. The combination of these models with many light methods and image-based lighting can drastically improve the realism of rendering results with an inexpensive rendering cost. Moreover, our methods are very easy to implement.

Acknowledgements

The author would like to thank Tatsuya Shoji for helping with the research and implementation of these physically based shading models, and Elliott Davis, Shawn Wilcoxen and Stephen Hill for reviewing the paper and slides.

References

[Blinn 1977] James F. Blinn. Models of Light Reflection for Computer Synthesized Pictures. *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, 1977.

[Cook and Torrance 1982] Robert L. Cook and Kenneth E. Torrance. A reflectance Model for Compute Graphics. *SIGGRAPH,* 1982.

[Oren and Nayar 1994] Michael Oren and Shree K. Nayar. Generalization of Lambert's Reflectance Model. *SIGGRAPH,* 1994.

[Neumann and Neumann 1999] László Neumann, Attila Neumann, László Szirmay-Kalos. Compact Metallic Reflectance Models. *Computer Graphics Forum*, 1999

[Weidlich and Wilkie 2009] Andrea Weidlich, Alexander Wilkie. Exploring the Potential of Layered BRDF Models. *SIGGRAPH Asia,* 2009.

[Gotanda 2010] Yoshiharu Gotanda. Practical Implementation of physically based shading models at tri-Ace. *SIGGRAPH,* 2010.

[Weidlich and Wilkie 2011] Andrea Weidlich, Alexander Wilkie. Thinking in layers: modeling with layered materials*, SIGGRAPH Asia,* 2011.

Yasuhiro Fujii. A tiny improvement of Oren-Nayar reflectance model. *http://mimosa-pudica.net/improved-oren-nayar.html*, 2012.