

Designing Reflectance Models for New Consoles

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

1 Introduction

Physically based shading models are now standard for real-time rendering and the video game industry now has the promise of using powerful GPUs thanks to the new generation consoles, the PlayStation 4 and Xbox One. Compared to previous generation consoles, the GPUs now available can use far more ALU instructions¹, since they have, in relative terms, many more instruction units than texture units. However, if a rendering pipeline is designed based on the old architecture, it can't utilize the maximum performance of the GPU. Therefore, higher resolutions, faster framerates and higher quality post-processing can be candidates to utilize the GPU's capability. On the other hand, in typical shading (e.g. both forward and deferred) algorithms, including physically based shading, there are a lot of texture samples, fetching information such as albedo, reflectance, normal, height, roughness and ambient occlusion maps. Additionally, they require more texture samples from the G-Buffer, image-based lighting, screen-space lighting and more global illumination information. Therefore, considering the ALU instruction and memory bandwidth ratio, using a more complex shading model is a good candidate to use this extra power. Specifically, the old normalized Blinn-Phong model only uses 20 to 30 instructions, though new generation consoles are capable of using 100 or more instructions depending on shading complexity.

2 Specular Model

Nowadays, there is a popular physically based model which uses the microfacet model with the GGX (Trowbridge-Reitz) distribution for D instead of Blinn-Phong, the Schlick approximation for F and Smith shadowing and masking function for G instead of Torrance-Sparrow V-cavity model in Equation 1.

$$f_r(\mathbf{l}, \mathbf{e}) = \frac{D(\mathbf{l}, \mathbf{e})F(\mathbf{l}, \mathbf{e})G(\mathbf{l}, \mathbf{e})}{4(\mathbf{n} \cdot \mathbf{e})(\mathbf{n} \cdot \mathbf{l})}. \quad (1)$$

Although the GGX and Smith functions have been frequently referenced and details are easy to find, for the Fresnel term F , the Schlick approximation is still widely used because it is a computationally inexpensive function and easy to implement. However, do we still have to approximate the function with a powerful GPU? Figure 1 shows the difference between the original Fresnel equation and the Schlick approximation with different refractive indices.

Equation 2 is a Fresnel equation using the reflectance instead of a refractive index.

$$F(\theta) = 0.5 \left\{ \left(\frac{(\sqrt{f_0} + 1) \cos \theta + (\sqrt{f_0} - 1)\alpha}{(\sqrt{f_0} + 1) \cos \theta - (\sqrt{f_0} - 1)\alpha} \right)^2 + \left(\frac{(\sqrt{f_0} - 1) \cos \theta + (\sqrt{f_0} + 1)\alpha}{(\sqrt{f_0} - 1) \cos \theta - (\sqrt{f_0} + 1)\alpha} \right)^2 \right\}, \quad (2)$$

$$\text{where } \alpha = \sqrt{1 - \frac{(\sqrt{f_0} - 1)^2(1 - \cos^2 \theta)}{(\sqrt{f_0} + 1)^2}}.$$

¹Trying to reduce the number of GPRs is still very important for performance because of multi-threading in the GPU. More complex shaders can increase the number of GPRs.

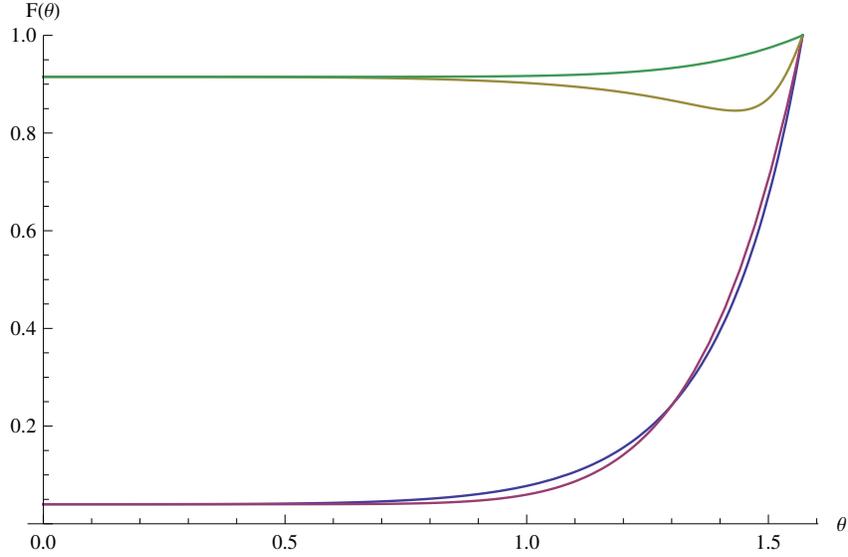


Figure 1: The yellow curve is calculated with the Fresnel equation for aluminum at 610nm. The green curve is calculated with the Schlick approximation for aluminium at 610nm. Blue (Fresnel) and Purple (Schlick) are curves for a refractive index of 1.5.

This equation is more useful than the original Fresnel equation, because a reflectance (map) is more intuitive than a refractive index and easier for artists to handle. However, this equation doesn't work correctly for a large f_0 . For real world materials, a large f_0 isn't just a large real number, but a complex number with a large imaginary part, as it is calculated from a complex refractive index. In this case, the Fresnel equation behaves differently, compared to using a real reflectance value, but the Fresnel equation with a complex refractive index is much more complicated than Equation 2. Therefore, it is difficult to directly use the Fresnel equation in a shader, even with new generation consoles. Figure 2 shows this problem.

Since the behaviour of the Fresnel equation is complicated, we need to implement the Fresnel equation with a complex refractive index or a complicated approximation to reproduce the behavior. Therefore, we tried to improve the accuracy with the most frequently used refractive index of 1.5, using the spherical Gaussian approximation shown in Equation 3,

$$F(\theta) = f_0 + (1 - f_0) \cdot 2^{-9.60232 \cos^8 \theta - 8.58092 \cos \theta}. \quad (3)$$

As shown in Figure 3, the Schlick approximation is a good enough approximation, although there is a slight difference to the size of the specular highlight. However, the Schlick approximation still has a problem with complex refractive indices with a large imaginary part. If you need to improve quality in these cases, you can use [9] with a complex refractive index as a parameter. In these course notes, the Schlick approximation is used for the F term, though you can choose better approximations including the improved spherical Gaussian.

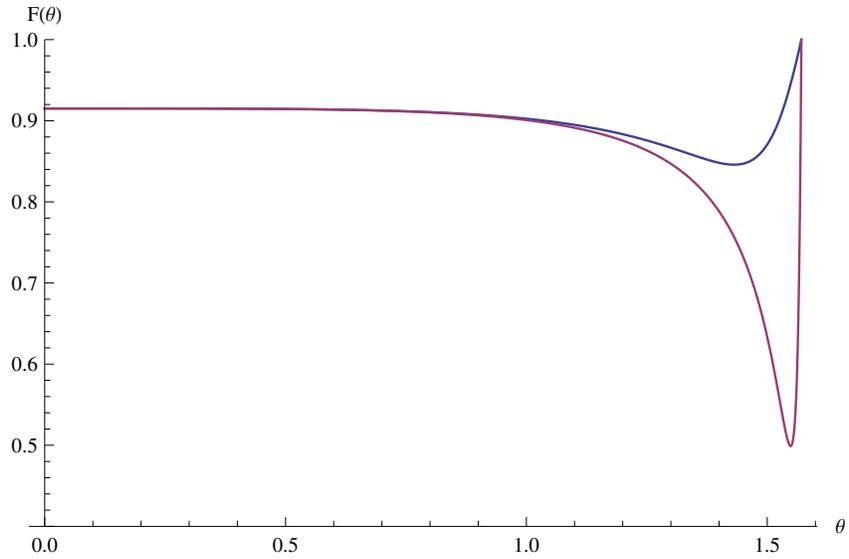


Figure 2: *The blue curve is the Fresnel equation with a refractive index of $1.3 + 7.48i$. The purple curve is Equation 2 with the real reflectance from the same refractive index.*

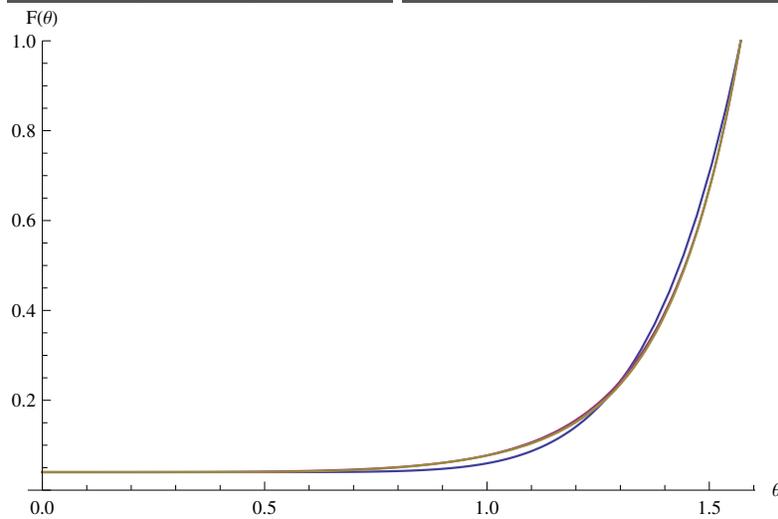
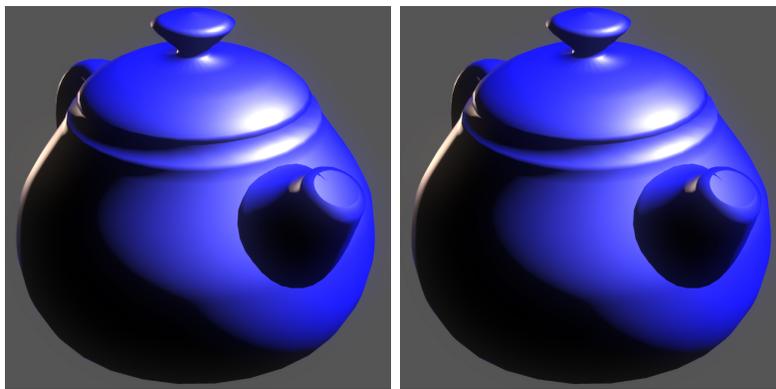


Figure 3: *All curves are calculated with a refractive index of 1.5. The blue curve is the Schlick approximation, purple is the Fresnel equation, yellow is the improved spherical Gaussian. The teapot on the left is rendered with the improved spherical Gaussian and the right one is rendered with the Schlick approximation. The sizes of specular highlight are slightly different.*



Figure 4: Comparison between different diffuse models. The teapot on the left is rendered with the Lambert model. The middle one is with the improved Oren-Nayar model. The right one is with the improved Oren-Nayar model with Fresnel (Equation 7). All teapots only have the diffuse component rendered.

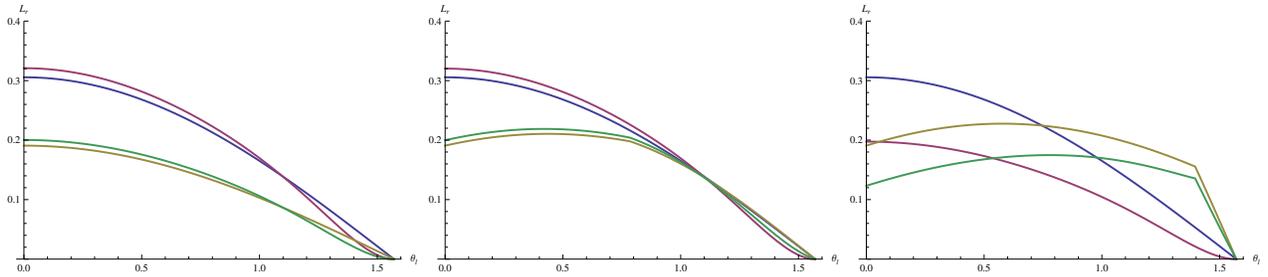


Figure 5: Oren-Nayar and Oren-Nayar with Fresnel(Equation 7) model analysis. Blue ($\sigma = 0$) and yellow ($\sigma = 1$) curves are Oren-Nayar. Purple($\sigma = 0$) and green($\sigma = 1$) curves are Oren-Nayar with Fresnel. From the left, $\cos \theta_e$ is 0, 45 and 80 degrees respectively. $\cos \phi$ is 0 degrees and f_0 is 0.04.

$$L_r = \frac{\rho}{\pi} E_0 (1 - f_0) \left[F_{\text{diff}}(\mathbf{l}, \mathbf{e})(\mathbf{n} \cdot \mathbf{l}) \left(1.0 - 0.5 \frac{\sigma^2}{\sigma^2 + 0.65} \right) + \left(0.45 \frac{\sigma^2}{\sigma^2 + 0.09} \right) (\mathbf{e} \cdot \mathbf{l} - (\mathbf{n} \cdot \mathbf{e})(\mathbf{n} \cdot \mathbf{l})) \min\left(1, \frac{\mathbf{n} \cdot \mathbf{l}}{\mathbf{n} \cdot \mathbf{e}}\right) \right]. \quad (7)$$

Figure 4 shows the rendering results with Equation 7 and the graphs in Figure 5 show analysis of different models.

The problem is that this new model is still a compromise. The Fresnel component is added to the geometry surface instead of the microfacet. Ideally, it should be added to each microfacet. Additionally, the D and G terms are different from the terms used in the specular model. As mentioned earlier, they should be the same terms. To solve this problem, we have to derive a new model from scratch instead of using existing models. To that end, we analyzed the diffuse microfacet model in Equation 4, and simply integrated the model numerically. Figures 6 to 10 analyze the new model and compare it to the original Oren-Nayar and Oren-Nayar with Fresnel.

This model doesn't have a closed form for the given D, F and G terms³. So we need to bake this model to textures or fit it to an equation to implement this model in a shader. However, if this model

³This time, the model uses GGX for the D Term, the Schlick approximation for the F term and Smith Height-Correlated Masking and Shadowing Function introduced in [6] for the G term.

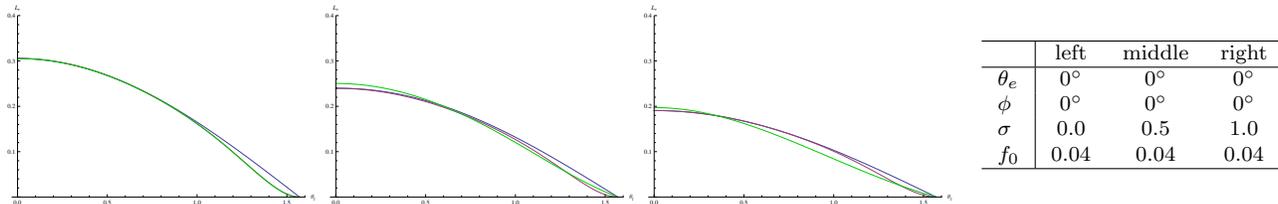


Figure 6: Comparison with different roughness values. The blue curve is Oren-Nayar, the purple is Oren-Nayar with Fresnel and the green is the new model. As roughness increases, the shading result gets more flat. In addition, even with roughness at 1.0, Oren-Nayar with Fresnel (purple) has a sharp fall-off at grazing angles, caused by Fresnel. However, the new model (green) doesn't exhibit such an effect.

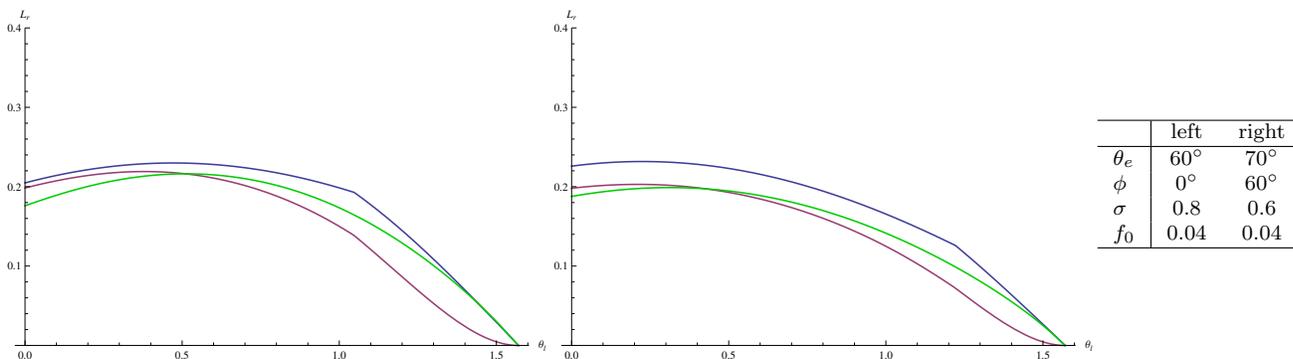


Figure 7: Comparison with different settings. With these settings, three models show completely different results. Discontinuities in Oren-Nayar based models (blue and red) are caused by the Torrance-Sparrow V-cavity model, but the new model doesn't have such a discontinuity as it uses the Smith shadowing and masking function.

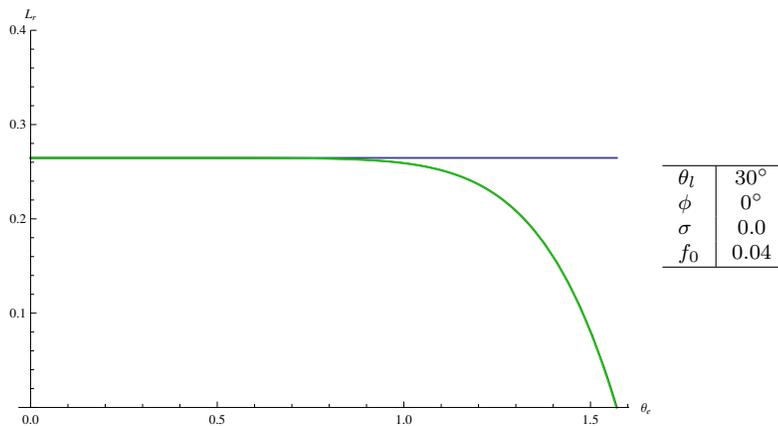


Figure 8: This graph is sliced by θ_e instead of θ_l . The purple (Oren-Nayar with Fresnel) and green (new model) curves completely overlap. This graph shows simple results which are that the original Oren-Nayar has no Fresnel effect at all, and the others have the Fresnel effect.

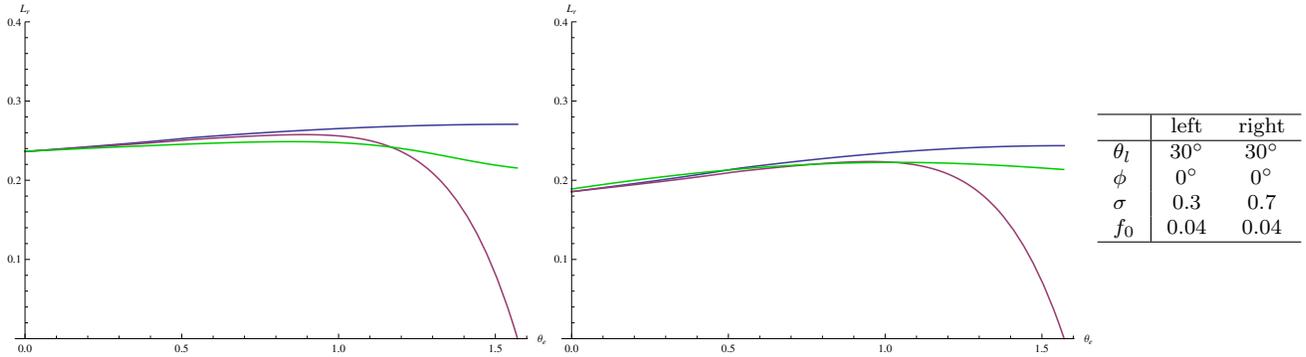


Figure 9: These graphs are also sliced by θ_e with different roughness settings. The three models show completely different results. The original Oren-Nayar (blue) doesn't have the Fresnel effect at all. Oren-Nayar with Fresnel (red) has a strong Fresnel effect regardless of roughness. The new model (green) has an appropriate Fresnel effect depending on roughness.

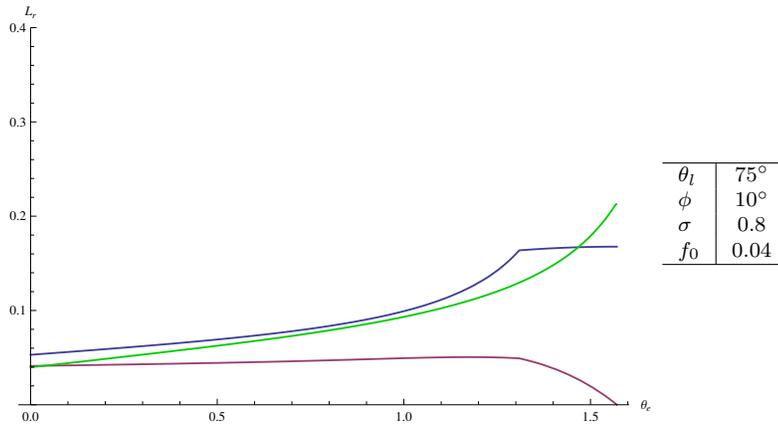


Figure 10: This graph has an out-of-plane setting and has θ_l at close to grazing angles. Again, the discontinuities in the original Oren-Nayar (blue) and Oren-Nayar with Fresnel (red) disappear in the new model (green). For Oren-Nayar with Fresnel, the Fresnel effect is too strong and causes strongly decreasing shading results at grazing angles for θ_e despite a very rough material ($\sigma = 0.8$). The other models don't have such results.

is directly baked to into textures, they need three vectors and two scalars as parameters⁴, even though f_0 can be factored out. Therefore, this model would be difficult to implement as a texture lookup. Moreover, as mentioned earlier, recent shaders may be texture-bound, thus we should be able to afford to implement this model as a fitted model, even if the fitted model is complicated. As a result, we tried to find a model and found the following approximation:

$$L_r(\mathbf{e}, \mathbf{l}, \sigma, f_0) = \frac{21}{20\pi}(1 - f_0)(F_r(\mathbf{e}, \mathbf{l}, \sigma)L_m(\mathbf{l}, \sigma) + V_d(\mathbf{e}, \mathbf{l}, \sigma)B_p(\mathbf{e}, \mathbf{l})), \quad (8)$$

$$F_r(\mathbf{e}, \mathbf{l}, \sigma) = \left(1 - \frac{0.542026\sigma^2 + 0.303573\sigma}{\sigma^2 + 1.36053}\right) \left(1 - \frac{(1 - \mathbf{n} \cdot \mathbf{e})^{5-4\sigma^2}}{\sigma^2 + 1.36053}\right) \\ \left((-0.733996\sigma^3 + 1.50912\sigma^2 - 1.16402\sigma)(1 - \mathbf{n} \cdot \mathbf{e})^{(1 + \frac{1}{39\sigma^4 + 1})} + 1\right),$$

$$L_m(\mathbf{l}, \sigma) = (\max(1 - 2\sigma, 0)(1 - (1 - \mathbf{n} \cdot \mathbf{l})^5) + \min(2\sigma, 1))((1 - 0.5\sigma)\mathbf{n} \cdot \mathbf{l} + 0.5\sigma(\mathbf{n} \cdot \mathbf{l})^2),$$

$$V_d(\mathbf{e}, \mathbf{l}, \sigma) = \left(\frac{\sigma^2}{(\sigma^2 + 0.09)(1.31072 + 0.995584(\mathbf{n} \cdot \mathbf{e}))}\right) \left(1 - (1 - \mathbf{n} \cdot \mathbf{l})^{\left(\frac{1 - 0.3726732(\mathbf{n} \cdot \mathbf{e}^2)}{0.188566 + 0.388410\mathbf{n} \cdot \mathbf{e}}\right)}\right),$$

$$B_p(\mathbf{e}, \mathbf{l}) = \begin{cases} 1.4(\mathbf{n} \cdot \mathbf{e})(\mathbf{n} \cdot \mathbf{l})(\mathbf{e} \cdot \mathbf{l} - (\mathbf{n} \cdot \mathbf{e})(\mathbf{n} \cdot \mathbf{l})), & \text{if } (\mathbf{e} \cdot \mathbf{l} - (\mathbf{n} \cdot \mathbf{e})(\mathbf{n} \cdot \mathbf{l})) < 0 \\ \mathbf{e} \cdot \mathbf{l} - (\mathbf{n} \cdot \mathbf{e})(\mathbf{n} \cdot \mathbf{l}), & \text{otherwise.} \end{cases}$$

If you think this model is too complicated to implement into your rendering engine, you can simplify some terms in Equation 8. For example, this model uses rational interpolations and a cubic function and these functions can be simplified to linear functions or quadratic functions at the sacrifice of accuracy.

4 Limitations and results of the new diffuse model

There are limitations in this fitted model. The first limitation is that the accuracy of the fitted model at grazing angles for both view and light vectors is not enough. This may cause some visual differences on the edges. The second limitation is that roughness values are limited from 0 to 1. The last one may not be a limitation, though this model shouldn't be used to compute an AmbientBRDF table which was introduced in [3]. In that course, the diffuse component in the AmbientBRDF was not stored in the AmbientBRDF texture for performance reasons. Then, [4] described how to compute the diffuse component for image based lighting with Oren-Nayar without the diffuse AmbientBRDF texture. However, new generation consoles have enough computational capability to store integrated diffuse components into the AmbientBRDF texture. Therefore in [5], the new diffuse model was introduced for image based lighting with an AmbientBRDF. That model can be also used for the new model shown in Equation 8 with the AmbientBRDF table which is computed using Equation 4. However, as mentioned, the fitted model shouldn't be used for this purpose, numerical integration should be used to compute the AmbientBRDF for better accuracy.

⁴Three vectors in the model can be reduced to three scalars by projecting the model onto θ_l , θ_e and ϕ . Even with this operation, we still need four scalars including roughness, and it is difficult to bake the model into one volume texture. However, using BTF-like techniques, the model can be baked to multiple textures, if necessary.

Figure 11 to Figure 15 describe observations of the fitted model. Figure 16 and 17 show visual comparisons between different diffuse models.

In Figure 17, you will notice that the results of the new diffuse model are darker, especially with rougher settings, than the other models. This is caused by the diffuse model taking Fresnel on each facet into account. All facets have a specular component, which doesn't contribute to the model and in fact causes a decrease in the diffuse component to conserve energy. As a result, the new diffuse model gets darker than models which don't take Fresnel on each facet into account. This is another effect of Fresnel for diffuse, not only on edges (at grazing angles).

5 Conclusion

From a physically based perspective, the same microfacet model should obviously be used for diffuse and specular. New consoles have enough computational power to implement such complex models. These course notes describe how to design a model with given functions. If you use some different models to derive a new reflectance model with the microfacet model, you should care about not only a specular model, but also a diffuse model.

Acknowledgements

The author would like to thank Tatsuya Shoji for helping with the research and implementation of these physically based shading models, and Stephen Hill and Stephen McAuley for reviewing the course notes and slides.

References

- [1] BLINN, J. F. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.* 11, 2 (July 1977), 192–198.
- [2] BURLEY, B. Physically-Based Shading at Disney. In *ACM SIGGRAPH 2012 Courses* (New York, NY, USA, 2012), SIGGRAPH '12, ACM, pp. 10:1–7. <http://selfshadow.com/publications/s2012-shading-course/>.
- [3] GOTANDA, Y. Practical Implementation of Physically-Based Shading Models at tri-Ace. In *ACM SIGGRAPH 2010 Courses* (New York, NY, USA, 2010), SIGGRAPH '10, ACM. <http://renderwonk.com/publications/s2010-shading-course/>.
- [4] GOTANDA, Y. Beyond a Simple Physically Based Blinn-Phong Model in Real-Time. In *ACM SIGGRAPH 2012 Courses* (New York, NY, USA, 2012), SIGGRAPH '12, ACM, pp. 10:1–7. <http://selfshadow.com/publications/s2012-shading-course/>.
- [5] GOTANDA, Y. Real-time Physically Based Rendering. CEDEC 2013. <http://research.tri-ace.com>.
- [6] HEITZ, E. Understanding the masking-shadowing function in microfacet-based BRDFs. *Journal of Computer Graphics Techniques (JCGT)* 3, 2 (June 2014), 32–91.
- [7] HOFFMAN, N. Background: Physics and Math of Shading. In *ACM SIGGRAPH 2012 Courses* (New York, NY, USA, 2012), SIGGRAPH '12, ACM, pp. 10:1–7. <http://selfshadow.com/publications/s2012-shading-course/>.
- [8] LAGARDE, S. Spherical Gaussian approximation for Blinn-Phong, Phong and Fresnel. <http://seblagarde.wordpress.com/2012/06/03/spherical-gaussian-approximation-for-blinn-phong-phong-and-fresnel/>.
- [9] LAZÁNYI, I., AND SZIRMAY-KALOS, L. Fresnel term approximations for metals. In *WSCG (Short Papers)* (2005), pp. 77–80.
- [10] OREN, M., AND NAYAR, S. K. Generalization of lambert’s reflectance model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 239–246.
- [11] SHIRLEY, P., SMITS, B., HU, H., AND LAFORTUNE, E. A practitioners’ assessment of light reflection models. In *Proceedings of Pacific Graphics 97* (1997), IEEE Computer Society, p. 40.
- [12] SMITH, B. Geometrical shadowing of a random rough surface. *IEEE Trans. on Antennas and Propagation* 15 (1967), 668–671.
- [13] TORRANCE, K., AND SPARROW, E. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America* 57, 9 (1967), 1105–1114.
- [14] WALTER, B., MARSCHNER, S. R., LI, H., AND TORRANCE, K. E. Microfacet models for refraction through rough surfaces. pp. 195–206.
- [15] WOLFF, L. B., NAYAR, S. K., AND OREN, M. Improved diffuse reflection models for computer vision. *Int. J. Comput. Vision* 30, 1 (Oct. 1998), 55–71.

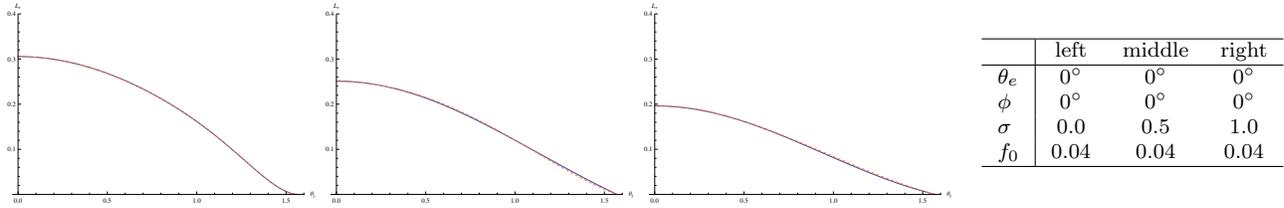


Figure 11: *Evaluations of the fitted model (blue) compared to the ground truth (orange dash) with different parameters. The blue curve is the fitted model and the orange dash is numerically integrated. As shown in these graphs, the model fits well for these settings.*

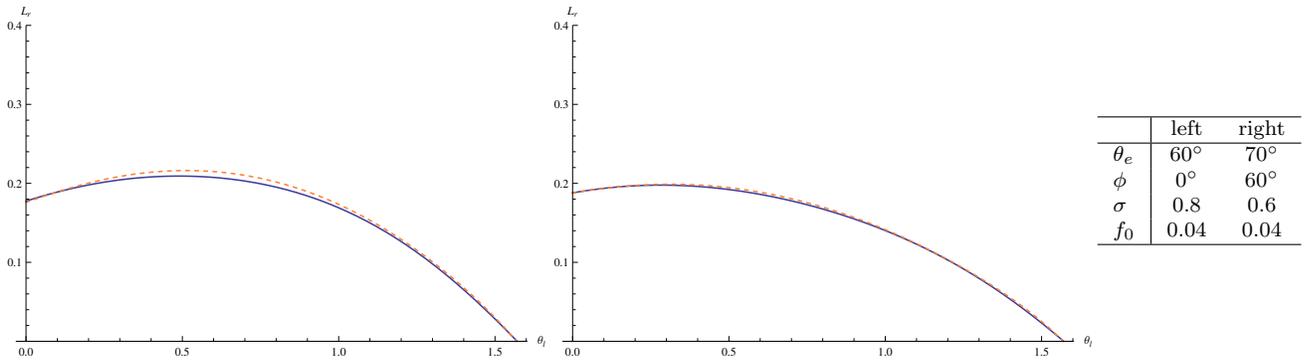


Figure 12: *Other observations with different parameters. There are slight differences with these settings. However the fitted model still represents characteristics of the new model well.*

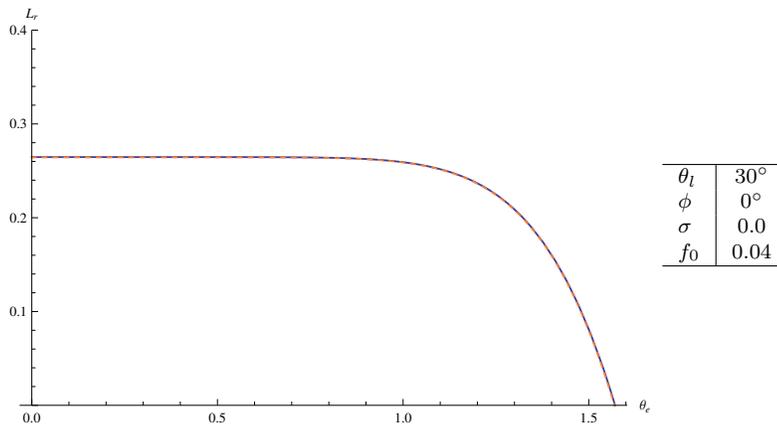


Figure 13: *This graph is sliced by θ_e instead of θ_l . The fitted model accurately traces the new model even with a different parametrization.*

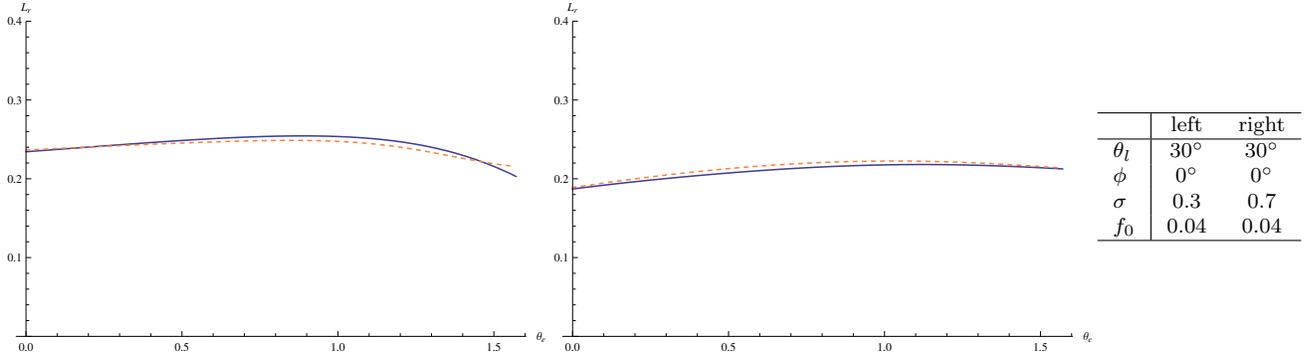


Figure 14: *These graphs are also sliced by θ_e with different roughness settings. As shown in the left graph, there are small errors at grazing angles in some cases.*

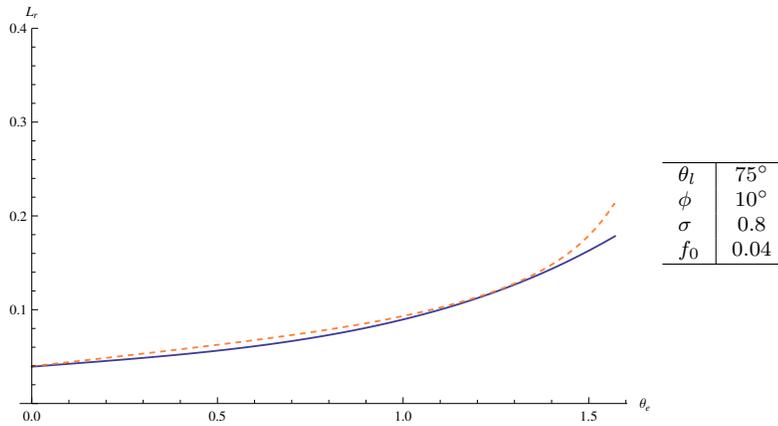


Figure 15: *This graph has an out-of-plane setting and θ_l at close to grazing angles. As shown in this graph, when both θ_l and θ_e are close to grazing angles, the error increases. This is the one of the limitations of this model.*



Figure 16: *Model comparisons with a character and multiple lights. From the left, the diffuse shading models are Lambert, Oren-Nayar with Fresnel and the (fitted) new model respectively. The specular shading model is same for all images. Differences between the three models are easily distinguishable. With the Lambert model, shading gradually varies. However, with other two models, shading is flatter and suddenly drops at grazing angles. This effect is more strongly apparent for the new model.*

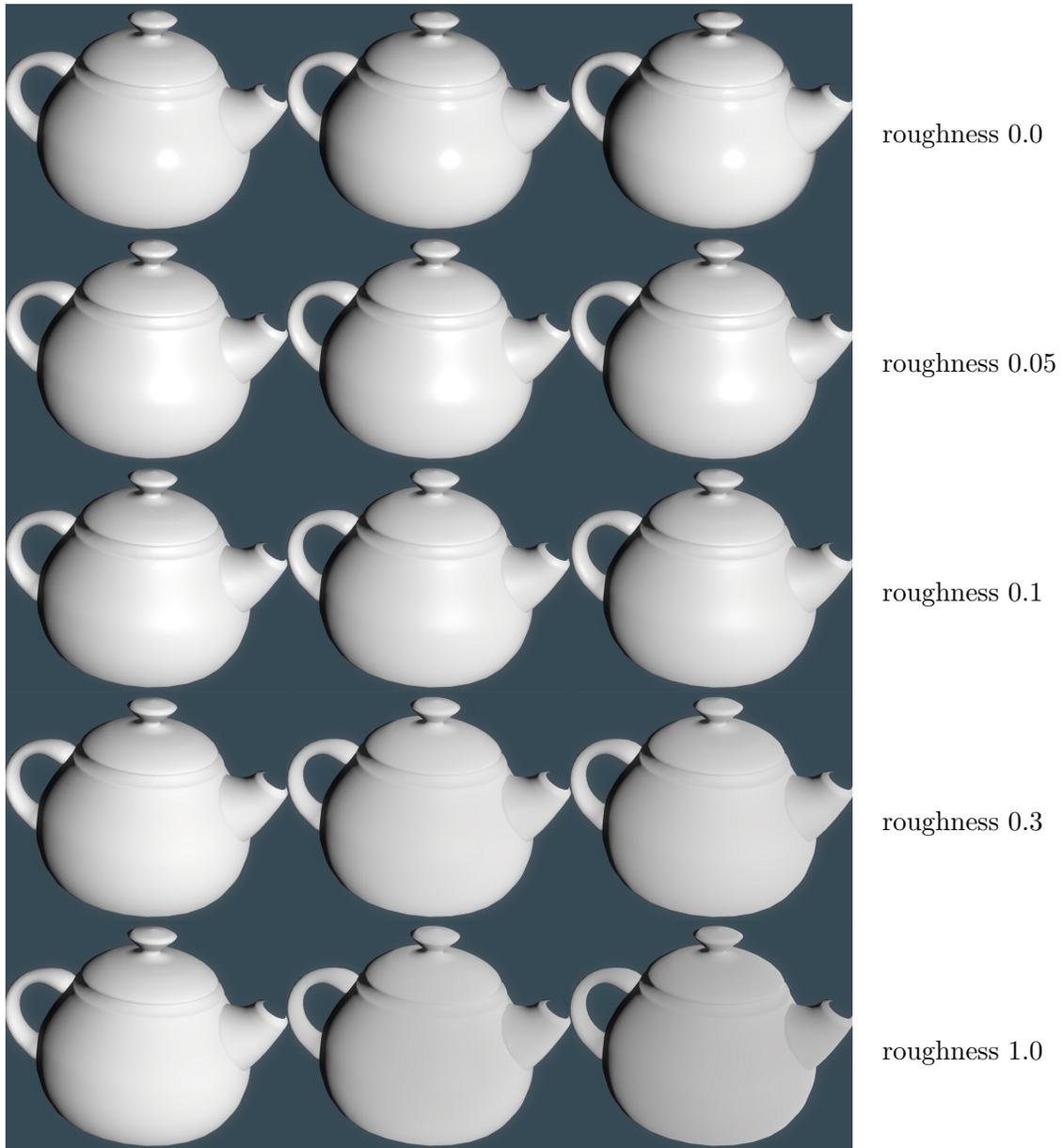


Figure 17: *Model comparisons with simple teapots with different roughness values. From the left, the diffuse shading models are Lambert, Oren-Nayar with Fresnel and the (fitted) new model respectively. The specular shading model is same for all teapots. With the Lambert model, only the specular highlight varies according to roughness. The new model has a relatively more matte (flat) look than Oren-Nayar with the same roughness value. Moreover, the Fresnel effect on the edge rapidly disappears even with a roughness of 0.05 for the new model, though it remains for Oren-Nayar with Fresnel with the same roughness.*