Physically Based Shading Models in Film and Game Production:

Practical Implementation at tri-Ace

Yoshiharu Gotanda
tri-Ace, Inc.

1. Introduction

In this paper, we present our practical examples of physically based shading models that we implemented. In the game industry, traditional ad-hoc shading models are mainly used because of performance, though recently, physically based models have been attracting more attention. In our studio, artists had some difficulty setting parameters for physically correct materials using the following ad hoc shading model:

$$\sigma = R_d (N \cdot L) + R_s F(f_0)(N \cdot L)(N \cdot H)^n G , \tag{1}$$

where R_d is the diffuse color, R_s is the specular color, N is the normal vector, L is the light vector, H is the halfway vector, $F(f_0)$ is the Fresnel function, and n is the cosine power that is often called shininess which represents the roughness of a surface. Lastly, G is the geometry attenuation factor. Some of the above parameters are typically stored in textures.

The human eye doesn't perceive light linearly and the brain recognizes the brightness of light in non-linear space (similar to logarithmic space) with a high dynamic range. On the other hand, display devices used for video games typically have an 8-bit color resolution which is insufficient to represent real world, high dynamic range, light intensity. Due to these two reasons, artists can set material parameters incorrectly via their intuition, even though the real world material properties have more dynamic variance. For example, if there is an object that has a 1,000 times stronger specular than another object, an artist may set only a 10 times stronger specular parameter, because the artist felt that was the correct value.

In our case, this problem often happened by the Fresnel parameter being incorrectly set. For our implementation, we used Schlick's approximation[1] as shown in Equation 2. The approximation is faster than the original equation and produces a good shading result.

$$F(f_0) = f_0 + (1 - f_0)(1 - E \cdot H)^5 . \tag{2}$$

f_0 is the normal specular reflectance. This can be computed with:

$$f_0 = \left(\frac{1-n}{1+n} \right)^2 , \tag{3}$$

where n is the complex refractive index[1]. For typical dielectric materials, n is between 1.3 and 1.7. Using the average value of 1.5, f_0 evaluates to 0.04 and Equation 2 evaluates to 0.04 in the normal direction and 1.0 in the direction perpendicular to the normal. As a result, the reflection ratio between the normal and glancing directions becomes 25. Since such a large specular variance is not intuitively acceptable for artists, they incorrectly set a value like 0.3 or 0.5 to f_0. This causes the specular in the normal direction to become too

[1] If n is a complex number, it is important that this equation be calculated using n as a complex number.

strong compared to reality. The artist then senses something isn't right and reduces specular intensity to try and compensate. Consequently, this leads to specular at glancing angles becoming too weak. As a typical example, because the specular at glancing angles is weak, the highlight on the edge of a back lit object cannot be accurately represented. As a solution, we changed parameters in our tool to use either complex refractive indices, n , or prebuilt material templates such that artists do not have the opportunity to provide incorrect inputs for the Fresnel equation.

Compared with these issues, physically based shading models allow us to easily manipulate the parameters. We can reduce the number of parameters or textures and the shader still produces physically correct results.

2. Customized Blinn-Phong model

The following equation is our BRDF model based on the Blinn-Phong model[2]:

$$\rho = \frac{R_d}{\pi}\left((1 - F_{diff}(f_0)\right) + \frac{(n+2)}{4\pi(2 - 2^{-\frac{n}{2}})} \cdot \frac{F_{spec}(f_0)(N \cdot H)^n}{\max(N \cdot L, N \cdot E)} . \tag{4}$$

$F_{diff}(f_0)$ and $F_{spec}(f_0)$ are Fresnel functions. We used Schlick's approximation for them:

$$F_{diff}(f_0) = f_0 + (1 - f_0)(1 - N \cdot L)^5 , \tag{5}$$

$$F_{spec}(f_0) = f_0 + (1 - f_0)(1 - E \cdot H)^5 . \tag{6}$$

The BRDF shown in Equation 4 basically follows the laws of energy conservation. However, the function violates reciprocity in the diffuse term for performance.

The normalization factor $(n+2)/4\pi(2 - 2^{-\frac{n}{2}})$ is calculated with our specular BRDF with Neumann-Neumann BRDF[3] as:

$$f_{r\_spec} = \frac{(H \cdot N)^n}{\max(N \cdot E, N \cdot L)} . \tag{7}$$

In order to acquire the normalization factor, this BRDF with the cosine term is integrated and must satisfy the laws of energy conservation as follows:

$$c \cdot \int_{\Omega} \frac{(H \cdot N)^n (N \cdot L)}{\max(N \cdot E, N \cdot L)} d\omega \le 1 \quad . \tag{8}$$

Choosing c in this inequality to satisfy the energy conservation requirements, the normalization factor can be computed. Since integrating the integral analytically over all light (L) directions is impossible, we assume that the maximum reflected energy occurs when $L = N$. Therefore the inequality becomes:

$$c \cdot \int_{\Omega} \frac{(H \cdot N)^n}{\max(N \cdot E, 1)} d\omega \le 1 . \tag{9}$$

However, $N \cdot E$ is always less than or equal to 1 and the inequality can be simplified to:

$$c \cdot \int_{\Omega} (H \cdot N)^n \, d\omega \leq 1 . \tag{10}$$

Then the reciprocal of c becomes:

$$\begin{aligned} &\int_{\Omega} (H \cdot N)^n \, d\omega \\ &= \int_{-\pi}^{\pi} \int_{0}^{\frac{\pi}{2}} (\cos\frac{\theta}{2})^n \sin\theta d\theta d\varphi \\ &= \frac{4\pi(2 - 2^{-\frac{n}{2}})}{n+2} . \end{aligned} \tag{11}$$

This normalization factor is a little expensive to compute in real-time, therefore we approximate it with a linear function. The coefficients of the linear function are determined using the least square method by fitting them in the range of n = 0 to 1,000:

$$\frac{n+2}{4\pi(2 - 2^{-\frac{n}{2}})} \approx 0.0397436n + 0.0856832 . \tag{12}$$

Figure 1 shows the difference between the original normalization factor and the approximated one.

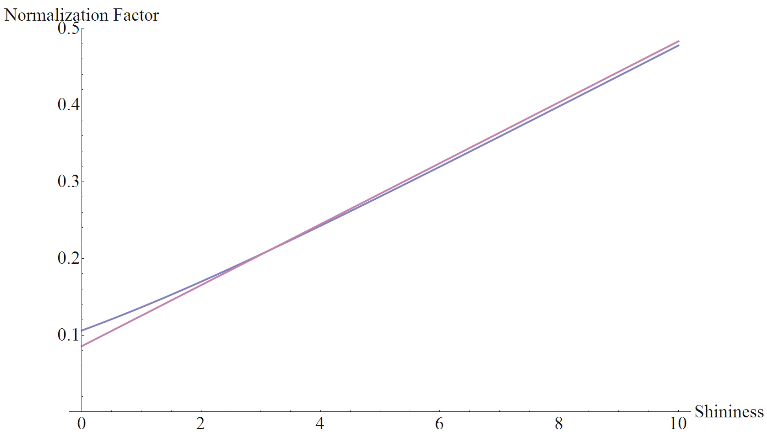


Figure 1: *Normalization Factor Comparison. The blue line is the original factor and the red line is the approximated factor with the linear function. The graph only shows the factor between shininess of 0 and 10. The error between shininess of 0 and 1,000 is almost negligible.*

The diffuse term in our BRDF model is an approximation of a more physically correct model. First we assume that the diffuse component is a perfect lambertian[2]. Then, the incident light reflects as specular and diffuse reflections at the shading point. (In this case, we don't think of other types of reflections.) The amount of specular reflection is decided by the Fresnel equation with respect to the reflection angle. Due to energy conservation, the diffuse component can be computed as the rest of specular reflection. The reflection angle is equivalent to the incident angle, so the Fresnel equation in the diffuse term is calculated with respect to the incident angle.

---

[2] This is obviously a wrong assumption because it violates the reciprocity. It is not an assumption from physics, from simplicity for performance.

However, the diffuse term is a little expensive and approximating it only produces subtle differences. Therefore, the diffuse term is also approximated in our final implementation as:

$$\rho = \frac{R_d}{\pi}(1 - f_0) + (0.0397436n + 0.0856832)\frac{F_{spec}(f_0)(N \cdot H)^n}{\max(N \cdot L, N \cdot E)}. \quad (13)$$

Figure 2 shows rendering results with our customized BRDF.

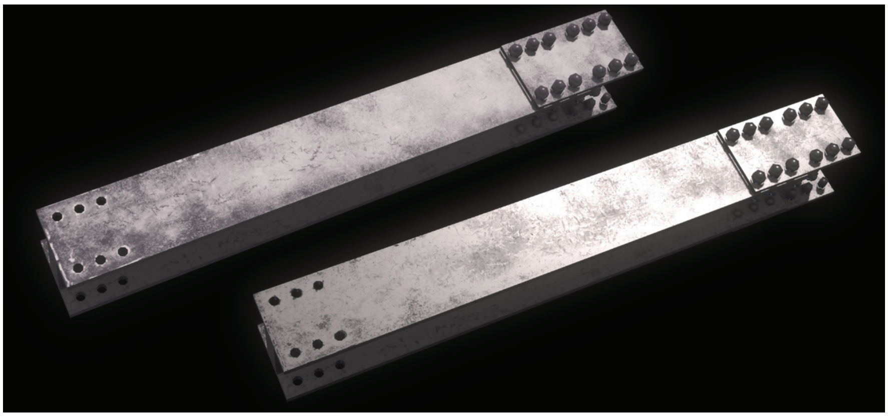


Figure 2 (a): *A comparison only with a shininess map. The left steel frame is rendered with an ad-hoc model and the right one is rendered with our model. The right frame looks more natural compared to the left frame, though it looks a flat material.*

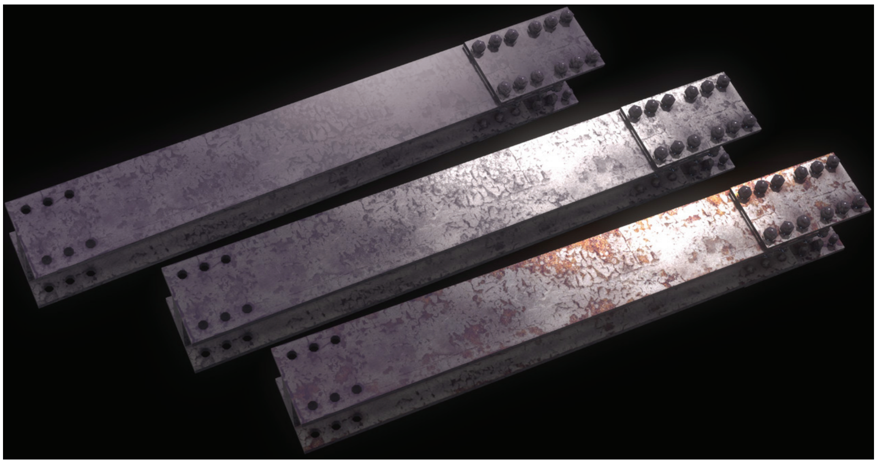


Figure 2 (b): *A comparison only with a reflectance map. The steel frames from the top left are rendered with an ad-hoc model, our model and our spectral model. The texture is used as a specular color map for the ad-hoc model that looks like it is just varying the design of the steel frame. Compared with it, our model looks to have more varied surface materials. Moreover, spectral model represents rust well.*

The BRDF model shown in Equation 13 is isotropic. In order to cover other types of materials, we implemented other variations of models which are anisotropic, spectral and metal versions. Because of the performance reasons, the basic model only covers isotropic, monochrome (non-spectral) and dielectric materials. The anisotropic model supports two different shininess parameters along with the tangent vector and binormal vector. The spectral model supports three different f_0 parameters, especially for materials which have different refractive indices at different wavelengths. The metal model[4] is designed for materials which have a complex refractive index especially with a large imaginary number. The Fresnel function for these kinds of metals has a distinctive curve compared to other materials. Figure 3 shows rendering results of these variation shaders.

In addition to the BRDF model based on the Blinn-Phong model, we also implemented other BRDF models such as Ashikhmin-Shirley[5] model, Marschner[6] model and Kajiya-Kay[7] model. Figure 4 shows rendering results of these shaders and comparisons with our Blinn-Phong based shaders.

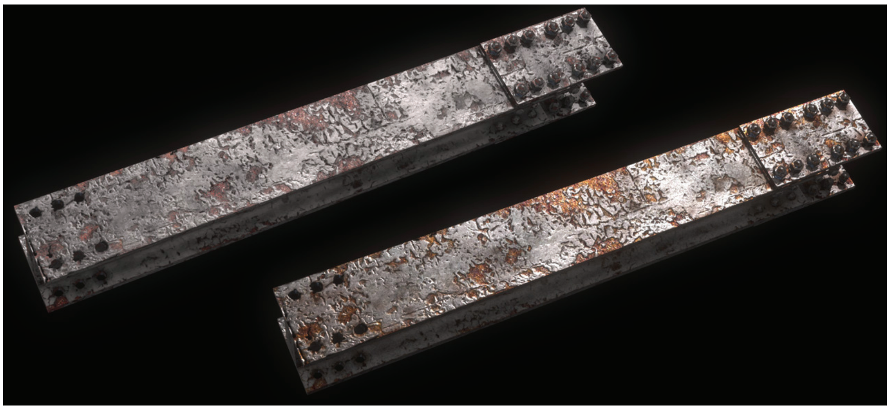


Figure 3(a): *The left steel frame is rendered with an ad-hoc model and the right one is rendered with our spectral model. Both frames have all textures such as reflectance, shininess maps, etc. With respect to the steel part and rusty part of both steel frames, it's easier to notice the difference between the materials on the right one.*

Figure 3 (b): *From the left: (a) ad-hoc model vs. our model, (b) ad-hoc model vs. spectral model.*
The left object uses an ad-hoc model and the right object uses our model in both images. In the image (a), reflectance is changed with respect to the black and yellow pigments in the sign. However, with the ad-hoc model, the black and yellow pigments look like they use the same material. Different materials are rendered well for the drum on the right using a reflectance map.

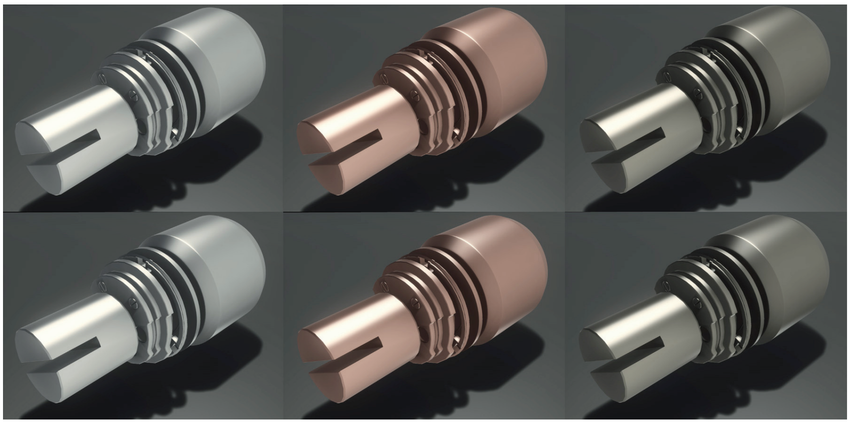


Figure 3 (c): *Comparisons of our metal model and spectral model. The top bolts are rendered with our spectral model and the bottom bolts are rendered with our metal models. Materials from the left are aluminum, copper and titanium. Specular reflection looks slightly different.*

Figure 4 (a): *The hair of the left character is rendered with Marschner model and the hair of the right character is rendered with Kajiya-Kay model.*

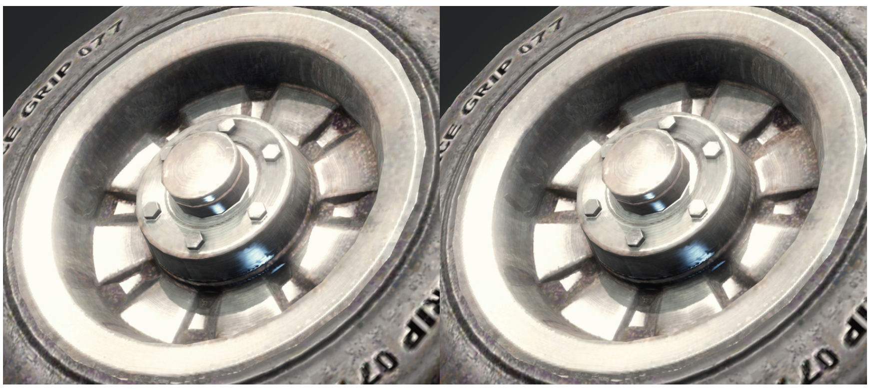


Figure 4 (b): *The wheel on the left is rendered with Ashikhmin-Shirley model, the wheel on the right is rendered with our anisotropic model. Our anisotropic model can achieve the close result with a less computational cost than Ashikhmin-Shirley model.*

3. Ambient Shading Improvement

Implementing the physically-based shading models introduced in the previous section, the quality of typical materials can be drastically and easily improved. However, in typical game engines, these BRDF models are used for direct (punctual) light sources such as directional, point and spot lights. Although deferred based shading or lighting techniques are becoming popular, multiple BRDF models don't get along well with deferred based techniques because different BRDF models increase the number of deferred shading passes or require dynamic branches in the deferred shader. Whether using forward shading or deferred shading, ambient lighting is still important for real-time rendering in games. However, because ambient lighting has been invented for performance, ambient lighting typically has a different pass from direct light shading.

There are two standard implementations for ambient shading. One is only treating only the diffuse term with ambient light. With this simplification, objects that only have a specular component such as metals are rendered completely black in a scene that only uses ambient lighting. The second implementation solves the problem of the first by using one arbitrary constant between diffuse and specular terms. In both implementations, the ambient term is computed as a constant and is view independent; though if it existed in the real world, it should be computed as spherical area lighting with a proper BRDF.

As a result of the ambient shading simplification, rendering quality degrades in scenes mainly lit by ambient light such as shadowed areas, the inside of a house lit only by daylight (no artificial lights), cloudy outside areas, etc. Figure 5 shows a sample scene with this problem.

Figure 5: *A scene with only ambient lighting. Because there are no specular components, material differences are difficult to see.*

On the other hand, ambient lighting has been improved. In the past, ambient light was a constant color vector in the scene and the color vector was same for every pixel and/or vertex. Recently, ambient lighting color has been changed to be dependent on location[3]. Examples would be Hemisphere lighting, Spherical Harmonic (SH) lighting[8][9], and Ambient Occlusion[10]. Hemisphere lighting or SH are one of the most reasonable approximations of spherical area lighting and are widely used for real-time rendering. SH coefficients,

[3] If the calculation is done on the CPU then the object location is used. If done on the GPU then the vertex or pixel location is used.

irradiance[11], or values similar to irradiance are stored in a voxel structure and lighting vectors are interpolated according to shading position. These kinds of techniques are widely used in pseudo real-time global illumination. However, these improvements for ambient lighting only change incoming light according to the shading point. Even if physically correct material parameters are set for every object, they seem to have the same material in a scene dominated by ambient lighting. Figure 6 shows objects with variety of materials using ambient lighting and ambient occlusion.

Figure 6: *Screenshots rendered with only ambient lighting. Compared to other screenshots with direct lighting, it is difficult to distinguish material differences. In the real world, though it is also difficult to distinguish.*

Artists noticed this problem from the beginning. In one project, even though they tried painting ambient occlusion or normal mapping textures carefully, quality only marginally improved. Therefore, they solved the problem by placing secondary lights, like rim lights or fill lights, manually. In another project, an artist built an original shader that improved quality using our highly flexible shader system[12] [13]. However, both solutions were based on artists' intuition and were not physically correct.

For solving the previously mentioned problem, we came up with a novel BRDF model called Ambient BRDF. After implementing this model, ambient shading computes the specular and diffuse components properly and the ambient term is no longer a constant. In other words, ambient lighting is regarded as area lighting in our shader. In practice, first we integrated a BRDF integral as:

$$f(n, f_0, \omega) = \int_{\Omega} \rho(n, f_0, \omega, \omega')(N \cdot L) d\omega' , \tag{14}$$

where ρ is a BRDF model, n is shininess, f_0 is normal specular reflectance, ω is the eye vector and ω' is the light vector. In our case, the specular term in Equation 13 is used for computing the specular component of the function ρ. Since this integral is difficult to integrate analytically, we integrated it numerically offline and stored it in a linear (non-swizzled) volume texture. For creating the texture, we developed an application that was firstly used for the experiment of our ambient BRDF models. After that, we tried both a linear texture and swizzled texture, and the linear texture was faster. With our texture, the U coordinate represents $E \cdot N$, the V coordinate represents shininess, and W coordinate is used for f_0 . Figure 7 shows the volume texture computed by our implementation. The volume texture stores the specular term itself and is directly used as the specular term in a pixel shader. The diffuse term computation is approximated with $R_d \cdot (1 - \text{specular term})$ but ideally, it should be stored in another texture because due to geometry attenuation, (specular term + diffuse term) does not always equal one. If this approximation is used, the reflectance gets too strong at glancing angles; however, we used it for performance.

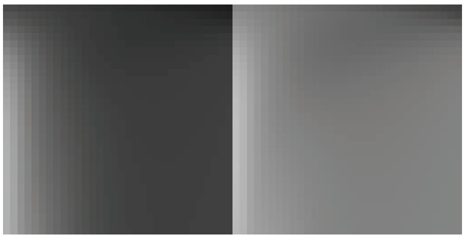


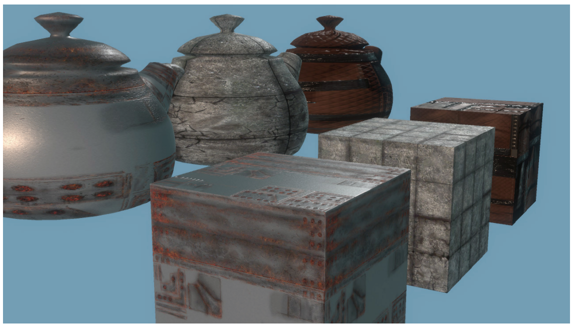
***Figure** 7*: *Part of the volume texture computed by our application.*

With the integral, only specular and diffuse terms can be computed. Therefore, the ambient shading result will vary due to material parameters and viewing angle. For a more realistic result, we computed the color terms of ambient light. When using Spherical Harmonics, the diffuse lighting color vector is evaluated with a normal vector and the specular lighting color vector is evaluated with a reflection vector. For both vectors, the color could be approximated depending on the number of SH coefficients. Since the specular component requires a lot of coefficients and no specular cosine lobe is taken into account, the specular color in ambient shading is coarsely approximated. When using image based lighting (environment map), the diffuse lighting color vector is computed with a diffuse cube map or SH coefficients which are calculated from an environment map. The specular color vector is computed with a pre-filtered mipmapped environment map.

For spectral materials, we fetch the texture three times with different f_0. For anisotropic materials, we fetch the texture with the average of two different shininesses. If we implement anisotropic ambient BRDF with texture, we need 4D texture. Therefore, we decouple the 4D texture to two 3D textures. However, compared to the difference between this experiment and the average version, we thought that it was too costly. As a result, we decided to use the current implementation.

Figure 8 shows results of rendering with Ambient BRDF and Figure 9 shows a performance comparison.

Figure 8: *The image on the left was rendered without Ambient BRDF and the image on the right was rendered with Ambient BRDF. The both images were rendered with only ambient lighting. In the right image, the shading result from the specular component was added on the edges of the tire or wheel. As a result, the right image's material differences are slightly more recognizable than the left image.*



	Ad-hoc model	*Customized model*	*Anisotropic model*	*Spectral model*	*Metal model*	*Ashikhmin*
Ambient BRDF off	3.13	3.71	4.67	4.02	3.48	5.15
Ambient BRDF on	N/A	4.37	5.13	4.61	3.83	5.76

Figure 9: *Performance of different shaders is compared with the scene shown on the image. The numbers indicate rendering times in millisecond on GPU for PlayStation 3. Each object has an albedo map, normal map and combined map (R: Reflectance, G: Shininess). The scene is rendered by 1280x720.*

4. Limitations and future work

Our customized physically-based Blinn-Phong model doesn't handle reciprocity and roughness in the diffuse component. The diffuse roughness component is very important for some materials. Oren-Nayer or other approximation models would be necessary to achieve higher quality results.

Additionally, multi-layered BRDF is not supported. A single layer BRDF is not enough to represent objects with coating, organic materials, and so on. We will need to implement multi-layered BRDF models in real-time.

Our Ambient BRDF model is only an approximation; in the future we can compute more accurate real time representations on more powerful GPUs.

5. Conclusion

With physically based shading models, artists can easily manipulate parameters and textures compared to ad-hoc models. Physically based shading models are fast enough to run in real-time, allow us to render realistic images, and give us true HDR images. The Ambient BRDF model automatically improves image quality in a scene dominated by ambient lighting instead of using ad-hoc methods such as placing secondary lights. As a result secondary lights like rim lights or fill lights can be used for their original purposes. Lastly, Figure 10 shows rendering results.

Acknowledgements

The author would like to thank to Tatsuya Shoji for helping with the research and implementation of physically based shading models, Bart Sekura and Elliott Davis for reviewing the paper and slides, Kenichi Kanekura, Kazuki Shigeta, Kenichi Kaneko and Ryo Mizukami for creating beautiful graphics samples, and my fellow course speakers, especially Nathaniel Hoffman for reviewing.

References

[1] Christophe Schlick. An Inexpensive BRDF Model for Physically-based Rendering. *Computer Graphics Forum,* 13(3):233-246, 1994.

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

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

[4] István Lazányi, László Szirmay-Kalos. Fresnel Term Approximation for Metals. In *Short Paper Proceedings of WSCG*, pp.77-80, 2005.

[5] Michael Ashikhmin and Peter Shirley. An Anisotropic Phong BRDF Model. *UUCS-00-014*, 2000.

[6] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley and Pat Hanrahan. Light Scattering from Human Hair Fibers. *ACM Transactions on Graphics*, 22, 3(July), 281-290, 2003.

[7] James T. Kajiya. Anisotropic Reflection Models. In *Proceedings of SIGGRAPH 85*, ACM Press, 15–21, 1985.

[8] Robin Green. Spherical Harmonic Lighting: The Gritty Details. *http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.html*, 2003.

[9] Peter-Pike Sloan, Jan Kautz, John Snyder. Precomputed Radiance Transfer for Real-Time Rendering in Dynamics, Low-Frequency Lighting Environments. *ACM Transactions on Graphics* 21, 3, 527–536, 2002.

[10] S. Zhukov, A. Iones, G.Kronin. An ambient light illumination model. In *Rendering Techniques '98 (Proceedings of the Eurographics Workshop on Rendering)*, 45–55, 1998.

[11] Gene Greger, Peter Shirley, Philip M. Hubbard, Donald P. Greenberg. The Irradiance Volume. *IEEE Compute Graphics & Applications*, 1998

[12] Yoshiharu Gotanda, Tatsuya Shoji. Shader Kanrijirei – Jiyudoto Hikikaeni – (Postmortem of Shader Management). In *CESA Developers Conference*, 2008.

[13] Yoshiharu Gotanda. STAR OCEAN 4: Flexible Shader Management and Post-Processing. In *Game Developers Conference*, 2009.

Figure 10: *The images on the left are rendered with our physically based shading models and the other images on the right are rendered with ad-hoc models. Due to our artists' good work, the right images still look very good. The left images keep consistent material appearance with all kinds of lighting such as daylight, sunset, or under shadow.*

Appendix

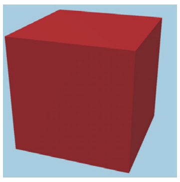
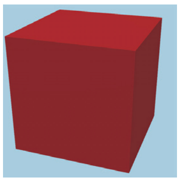
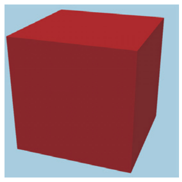
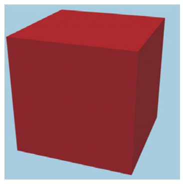
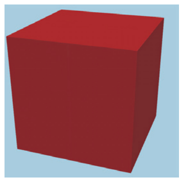
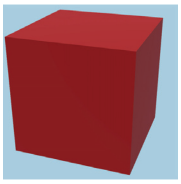
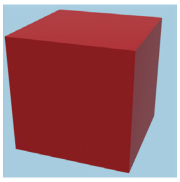
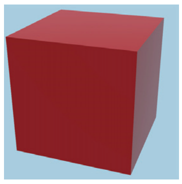
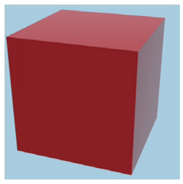
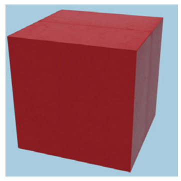
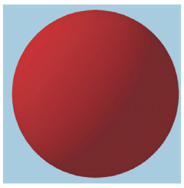
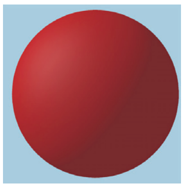
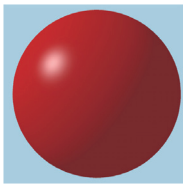
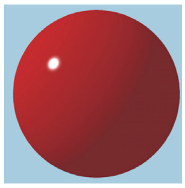
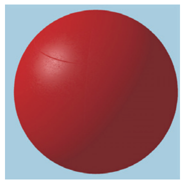
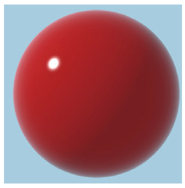
	Shininess 1	Shininess 10	Shininess 100	Shininess 1000	Shininess Map
Box (AB OFF)					
Box (AB ON)					
Sphere (AB OFF)					
Sphere (AB ON)					
Teapot (AB OFF)					
Teapot (AB ON)					

Table 1: *Ambient BRDF comparison with different settings. Each column shows different shininess settings. Each row shows different objects and Ambient BRDF settings (ON or OFF). Images with Ambient BRDF look surrounded by the area light.*