*ICIC Express Letters*
*Part B: Applications*
Volume **7**, Number **2**, February **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 351–356

# INTERACTIVE REFLECTION SIMULATION VIA PHYSICAL SHADING MODEL AND FAST ENVIRONMENT MAPPING

Jie Xu[1] and Jinhua Fu[2,3,*]

[1]School of Software
[2]School of Computer and Communication Engineering
Zhengzhou University of Light Industry
No. 5, Dongfeng Road, Zhengzhou 450002, P. R. China
*Corresponding author: fujinhuazz@qq.com

[3]State Key Laboratory of Mathematical Engineering and Advanced Computing
Zhengzhou Information Science and Technology Institute
Zhengzhou 450000, P. R. China

Received August 2015; accepted November 2015

Abstract. *In order to enhance realistic reflection effects on some objects and realize faster rendering to meet interactive requirements, we present a novel method to produce interactive reflection simulation using physical shading model and fast environment mapping. The physical shading model could efficiently express ambient lighting, specular lighting and diffuse lighting. Also the model is based on the microfacet distribution function, which can describe physics-based characteristics. Fast environment mapping is realized by Spherical Harmonics (SH) transformation, and the rendering equation can be rewritten as a simple dot product of SH coefficients, which fits for GPU (Graphics Processing Unit) computation. The results show that we can realize faster rendering and acquire realistic reflection effects such as glossy and mapping scene on the surface.*
**Keywords:** Reflection simulation, Shading model, Fast environment mapping, Spherical harmonics

1. **Introduction.** Environment lighting plays a crucial role in the photo-realistic rendering of virtual scenes and can greatly enhance the realistic effects. With the rapid development of graphics hardware and human vision needs, simulations of shading model and environment lighting have become increasingly attractive for real-time applications nowadays.

Many methods for realistic simulations via different shading models and environment lighting have been presented in the past. IBR (Image Based Rendering) [1] is a first visualization technique that utilizes captured HDR (High Dynamic Range) images as light sources in the rendering process, and makes objects appear as if they are actually in a real-world scene. When integrating IBR into the rendering system, the employment of a good sampling strategy for illumination is an important issue to speed up rendering. Hence, Kollig and Keller [2] propose an algorithm for determining quadrature rules and compute the direct illumination of predominantly diffuse objects by HDR images. On the other hand, Lawrence et al. [3] seek to reduce the diffuse illumination from HDR environments into a set of directional light sources using the BRDF (Bidirectional Reflectance Distribution Function) importance sampling. Recently, with the increasing requirements of modern 3D technology, the main challenge of rendering is not only to approximate the complex effect of environment lighting to give realistic looking appearance, but also develop sufficiently fast method to allow for real-time rendering based on GPU rendering pipelines [4]. Based on GPU platform, Wang et al. [5] adopt complex environment lighting to illuminate the synthetic objects naturally in the dynamic scenes and the results show their algorithm renders the synthetic objects at rates of over 30 FPS (Frames

Per Second). Similarly, using CUDA (Compute Unified Device Architecture) ray tracing is implemented by Shih et al. [6] and they focus on the performance in various optimization, but the sense of reality is not very high. Also, Yang et al. [4] consider the practical application and the lighting in real scene, and exploit environment lighting for illumination to make the human face more realistic via GPU rendering. More recently, Kronander et al. [7] present a GPU-based rendering system based on the NVIDIA OptiX framework, and enable real-time ray tracing of scenes illuminated with environment maps. They give a GPU implementation for multiple importance sampling and provide better results. Iwasaki et al. [8] propose an interactive rendering method of cloth fabrics under environment lighting. Their GPU implementation enables interactive rendering of static cloth fabrics with dynamic viewpoints and lighting. In order to acquire a lower computation complexity and faster rendering, HDR images are usually transformed and projected onto a group basis functions such as wavelet [9] and spherical harmonics [10], and then simulated by some transformation coefficients. However, here we choose the SH to transform HDR image based on its significant advantages: the rendering equation using SH can be rewritten as a simple dot-product which allows more easily fast evaluation and combination with practical application (e.g., 3D game development). This is the main reason why SH is so attractive.

In this paper, our focus differs in that we present a physical shading model which can efficiently express ambient lighting, specular lighting and diffuse lighting. Also, we realize fast environment mapping by implementing SH transformation on GPU, and acquire a much more fast rendering. The result shows that not only could we acquire realistic reflection simulations, but also the rendering speed is faster than the classic rendering method.

The rest of this paper is organized as follows. The physical shading model is established in Section 2. Fast environment mapping via SH transformation is given in Section 3. The results are discussed in Section 4 and conclusions are drawn in Section 5.

2. **Physical Shading Model.** Classic lighting model such as Blinn-Phong model can give nice results, which is fast and easy to be implemented, but it is far from the reality of light reflection. For more realistic appearance, we should use more sophisticated models and design the physical shading model.

The shading equation in this paper is the following in case of one light source.

$$I_{finalColor} = I_{ambient} + I_{diffuse} + I_{specular} \tag{1}$$

where $I_{ambient}$, $I_{diffuse}$ and $I_{specular}$ are ambient, diffuse and specular component of intensity of light reflecting from the object surface respectively.

The ambient lighting refers to the natural light in a scene and provides an area with overall illumination, so it can be evaluated as

$$I_{ambient} = k_d \times I_a \tag{2}$$

where $k_d$ is the diffuse reflectance of the object surface and $I_a$ is the ambient light intensity in a certain scene.

The specular lighting is the bright, shiny spots that appear on smoother surfaces. Usually they are the color of the light, but occasionally will take on a component of the material they are reflecting off. We consider surface's microstructure to give a more physically accurate approximation for the specular term as

$$I_{specular} = k_d \times \max(N \cdot L, 0) \times I/r^2 \tag{3}$$

where $N$ is the normal vector of the object surface. $L$ is the light direction. $N \cdot L$ expresses the surface's microstructure. $I$ is the light intensity. $r$ is the shaded point's distance from the light source.

To compute diffuse lighting efficiently, our shading model starts with the same Lambertian diffuse shading as Blinn-Phong model, but it models the surface's microstructure. The microfacet model used in the physical shading model assumes that the surface consists of randomly aligned small smooth planar facets. The diffuse term also takes the Fresnel refraction-reflection term and the shadowing effect of the microfacets into account as follows.

$$I_{diffuse} = \left(k_s \frac{F(\beta)}{\pi} \frac{D(\alpha_h)G}{(N \cdot V)(N \cdot L)}\right) \times \max(N \cdot L, 0) \times I/r^2 \tag{4}$$

where $F$ is the Fresnel term. $D$ is the microfacet distribution. $G$ is the geometric attenuation. $k_s$ is specular reflectance of the surface. $V$ is the viewing direction. $\beta$ is the angle between the viewing direction $N$ and the half vector $H$, which can be evaluated as $H = \frac{L+V}{\|L+V\|}$. $\alpha_h$ is the angle between the normal $V$ and the half vector $H$. The relationship of these directions and angles are clearly shown in Figure 1.

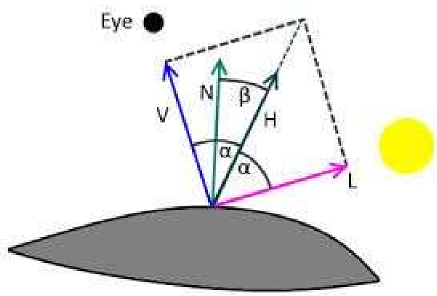


FIGURE 1. Relationship of reflectance directions and angles

The Fresnel term $F$ describes the reflective or refractive behavior of the light when it reaches a smooth surface. In this paper we use an approximation of the reflectance part here and ignore refraction as

$$F(\beta) = F_0 + (1 - F_0)(1 - \cos\beta)^5 = F_0 + (1 - F_0)(1 - (V \cdot H))^5 \tag{5}$$

where $F_0$ is the specular reflectance when light arrives perpendicularly to the surface.

To evaluate microfacet term $D$, we use a physically based model of microfacet distribution called Beckmann distribution function which can be defined as

$$D(\alpha_h) = \frac{e^{\lambda}}{\pi m^2 \cos^4 \alpha_h} \tag{6}$$

where $m$ is the RMS (Root Mean Square) slope of the surface microfacets (the roughness of the material). $\lambda$ is the wavelength of incoming light.

The term of geometric attenuation $G$ captures the self-shadow of the microfacets which can be written as

$$G = \min\left(1, \frac{2(N \cdot H)(N \cdot V)}{V \cdot H}, \frac{2(N \cdot H)(N \cdot L)}{V \cdot H}\right) \tag{7}$$

Hence, the Fresnel term $F$, the microfacet distribution $D$ and the geometric attenuation $G$ can be respectively evaluated according to Equation (5), Equation (6) and Equation (7), and further the diffuse lighting $I_{diffuse}$ can be obtained by Equation (4). Finally, the physical shading equation Equation (1) can be computed by Equation (2), Equation (3) and Equation (4).

3. **Fast Environment Mapping via SH Transformation.** When using environment lighting, the image looks more realistic if you can see the environment being reflected in the background of the scene. That is that environment lighting can greatly enhance the sense of reality. We use the environment map represented by HDR images to add the background, which is shown in Figure 2 [11].

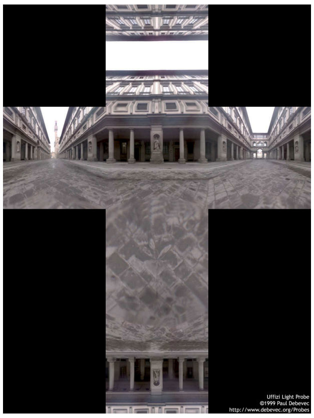


FIGURE 2. One HDR image used in this paper [11]

We consider that an environment map has $k$ texels and each texel can be thought of being a single light source. Therefore, for each surface point of the object, the irradiance $E$ can be computed as follows.

$$E = surfaceAlbedo \times \sum_{i=1,2,\ldots,k} lightColor_i \cdot \max(0, L_i \cdot N) \tag{8}$$

where $L_i$ is the light direction of texel $i$ and $N$ is the surface normal. It is obvious that if the number of texels $k$ is large, computation of this sum for each point of the object surface is really expensive.

More generally, this is the same as computing the irradiance $E$ with the integral over an upper hemisphere $\Omega(N)$ of light directions $\omega$ as follows:

$$E(N) = \int_{\Omega(N)} L(\omega)V(\omega)(N \cdot \omega)d\omega \tag{9}$$

where $L(\omega)$ is the amount of light coming from the direction $\omega$. $V(\omega)$ is the visibility which can be evaluated by ray tracking.

We define the soft shadow function as $G(\omega) = V(\omega)(N \cdot \omega)$, so Equation (9) can be further written as

$$E(N) = \int_{\Omega(N)} L(\omega)G(\omega)d\omega \tag{10}$$

Now, we can see that $L$ is a function that depends only on the environment map and that $G$ is a function that depends only on the geometry of the 3D object mesh. Thus, at each point of the our object surface, we need to compute this integral of the product of the functions $L$ and $G$. However, we have seen that integrating the product of two SH functions is equivalent to evaluating the dot product of their spherical harmonics coefficients. Therefore, we can precompute (by projection) the SH coefficients $L_i$ and $G_i$, and then at rendering time, for each point of the surface of the object, we only have to compute

$$E(N) = \int_{\Omega(N)} L(\omega)G(\omega)d\omega = \sum_{i=0}^{n} L_i G_i \tag{11}$$

The coefficients $L_i$ and $G_i$ come from the projection of the environment lighting function and shadow function onto SH basis functions. They are the same 9 coefficients that can be precomputed given an environment map.

4. **Rendering Results.** We have implemented the fast environment lighting to give interactive reflection simulation. Our final rendering clearly shows the effective implementation of the above mentioned technique. With AMD Athlon II X4 Four Cores and NVIDIA GeForce GT430, and $2048 \times 2048$ resolution of environment maps from Paul Debevec's HDR Shop [11], we have achieved frame rates of approximately 106 FPS which is a much more fast speed and can meet requirements of interactive rendering.

Polished metal, glass and more others have a good ability to reflect environment light. It means that besides their own color they also project surrounding light on their surface. If you do not have an HDR environment (including sky, light sources, etc. with realistic brightness levels), rendering result will lead to unnaturally dark, washed-out-looking reflections. However, in this paper with an HDR environment, the reflection looks natural and realistic.

Firstly we give realistic reflection simulation of glass balls under environment lighting, and the full glossy effects and mapping scene on glass balls can be observed in Figure 3 and Figure 4.

FIGURE 3. Reflection simulation of glass balls in one view

FIGURE 4. Reflection simulation of glass balls in another view

From Figure 3 and Figure 4, we can see that color of glass balls is formed from color of object's material and color of the environment reflected on the ball surface. Also, we can observe that some scenes in this environment such as mountain, cloud and sky could be seen on the glass balls, which accords with physical phenomena.

Further, in order to prove the effectiveness of our method, we render a teapot model (polished metal material) which has 42739 vertices, and give a rendering comparison between ray tracing method [6] and our method as shown in Figure 5 and Figure 6.

From Figure 5 and Figure 6, we can produce more accurate reflection simulations of lighting in some effects such as glossy and mapping scene on the teapot surface. So reflection effects obtained using our technique are more plausible.

In the performance of implementation, we also provide a cheap and easy way to render compared to ray tracing method [6]. We use the HDR image which is discretely sampled as $2048 \times 2048$ pixels, so the number of operations for nine lighting coefficients using Equation (11) is $2048 \times 2048 \times 9 = 36M$ ($M = 1024^2$) and the time complexity is $O(NP)$ ($P = 9$ is number of lighting coefficients). However, the number of operations using classic method such as ray tracing method [6] is $2048 \times 2048 \times 42739/4 = 42739M$ (4 cores on our platform and $M = 1024^2$) and its time complexity is $O(NH)$ ($H$ is pixel numbers of

FIGURE 5. Reflection simulation of teapot via ray tracing method

FIGURE 6. Reflection simulation of teapot via our method

3D model). Hence, it is clear $36M << 42739M$, and the rendering speed of our method is increased by $H/P \approx 1187$ times which has a practical value for interactive rendering such as 3D game.

5. **Conclusions.** We have implemented interactive reflection simulation via physical shading model and fast environment lighting. The physical shading model is deduced and established, which can efficiently express different lighting and describe the physical characteristics via microfacet distribution function. Environment lighting is transformed by SH and fast rendering is acquired by dot production of SH coefficients on GPU. The rendering results show that we produce more accurate reflection simulation effects such as glossy and mapping scene on the object surface. Also lower computation complexity and faster rendering are achieved and analyzed finally.

Although we have achieved a step in interactive reflection simulation, how the more realistic details can be approximated is still a challenge issue which we are currently working on.

## REFERENCES


[1] P. Debevec, Image-based lighting, *IEEE Computer Graphics & Applications*, vol.22, no.2, pp.26-34, 2002.

[2] T. Kollig and A. Keller, Efficient illumination by high dynamic range images, *Eurographics Symposium on Rendering*, pp.45-50, 2003.

[3] J. Lawrence, S. Rusinkiewicz and R. Ramamoorthi, Efficient BRDF importance sampling using a factored representation, *ACM Trans. Graphics*, vol.23, no.3, pp.496-505, 2003.

[4] M. Z. Yang, K. Q. Wang and W. M. Zuo, Realistic real-time rendering of human face with environment lighting, *IEEE International Conference on Systems and Informatics*, pp.2130-2134, 2012.

[5] Z. D. Wang, X. H. Ma and Z. G. Pan, Real-time illumination computing algorithm mixing of virtual and actual reality under complex environment lighting, *Journal of Image and Graphics*, vol.14, no.10, pp.2118-2121, 2009.

[6] M. Shih, Y. Chiu, Y. Chen and C. Chang, Real-time ray tracing with CUDA, *Lecture Notes in Computer Science*, pp.327-337, 2009.

[7] J. Kronander, J. Dahlin and D. Jonsson, Real-time video based lighting using GPU ray tracing, *Proc. of the 22nd European Signal Processing Conference*, pp.1627-1631, 2014.

[8] K. Iwasaki, K. Mizutani, Y. Dobasbi and T. Nisbita, Interactive cloth rendering of micro-cylinder appearance model under environment lighting, *Computer Graphics Forum*, vol.33, no.2, pp.333-340, 2014.

[9] A. M. Heidrich, Efficient wavelet rotation for environment map rendering, *Eurographics Symposium on Rendering Eurographics Association*, pp.173-182, 2006.

[10] S. Ling and X. T. Jiang, Environment mapping technology based on spherical harmonics function, *Information and Electronic Engineering*, vol.8, no.4, pp.420-424, 2010.

[11] *HDR Shop*, http://www.pauldebevec.com/Probes/.