Journal of Korea Game Society 2021 Aug; 21(4): 109-116
http://dx.doi.org/10.7583/JKGS.2021.21.4.109

Surface Model and Scattering Analysis for Realistic Game Character

Seongdong Kim*, Myounjae Lee**
Kaywon University of Art Design, Department of Game Media,
Euiwhang Kyunggi, Korea*
Baekseok University, Division of Computer Engineering,
Cheonan ChungNam, Korea**
sdkim@kaywon.ac.kr, davidlee@bu.ac.kr

ABSTRACT

In this paper, we considered that recently 3D game characters have been almost alike realistic expression because of a great mathematical computation and efficient techniques on GPU hardware. We presented the rendering technique and analysis for 3D game characters to simulate and render mathematical approach model from recent researches to perform the game engine for the surface reflection of lighting model.

We compare our approach with the existing variant rendering techniques here using Open GL shader language on game engine. The experimental result will be provided the view-dependent visual appearance of variant and effective modeling characters for realistic expression using existing methods on the GPU for effective simulations and rendering process. Since there are many operations that are used redundantly while performing mathematical operations, the necessary functions and requirements have been to compute in advance.

Keywords : Lambertian model, Oren-Nayar model, reflection model, BRDF, BSSRDF

Received: Jun. 22. 2021　　Revised: Aug. 13. 2021
Accepted: Aug. 19. 2021
Corresponding Author: Seongdong Kim(Kaywon University of Art & Design)
E-mail: sdkim@kaywon.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

1. Introduction

Mobile devices are rapidly developing, starting with smartphones. There is also a significant increase in control applications. The main factor in determining the quality of a computer graphic data is the light calculation for determining the color of an object surface (Illumination Method)[1,2]. That is, it is an optical phenomenon simulation method. In the field of computer graphics, rendering technique refers to the process or technique of creating a picture or image using a computer based on a two-dimensional or three-dimensional scene. Usually, to improve 3D rendering, it is related to 3D character appearances and technological developments. In fact, it is used for building design, games, and animation. Strictly, the 3D shape production process is divided into modeling and rendering processes[11]. While modeling is the process of creating a basic skeleton, rendering is the process of computing the skeleton surface. Of course, there are cases where the modeling process is finished without going through the rendering process, but in general video and game fields, the rendering process is also performed. Rendering includes Projection, Clipping, Hidden Surface, Shading, and Mapping[3].

Artists and researchers realize that the diffuse BRDF (Bidirectional Reflectance Distribution Function) and is something odd. Physically based shading means estimating what kind of material behaviors works, rather than intuitively estimating what the light's reflection should be. Physically based rendering (PBR) refers to the concept of using realistic shading / lighting models along with measured surface values to accurately represent real-world materials[6]. PBR is more of a concept than a strict set of rules, and such as the exact implementations of PBR systems tend to vary. However, as every PBR system is based on the same principal idea, many concepts will transfer easily from project to project or engine to engine. The result is a more accurate and natural look. Physical-based materials work equally well in lighting environments. In addition, material values become less complex and interdependent, making the interface more intuitive. These advantages also apply to non-photorealistic renderings, as can be seen clearly in Pixar's or Disney's recent work.

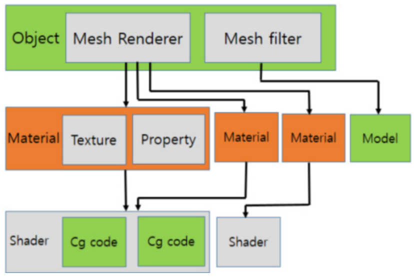
Probably the most assumption made about diffuse reflection is that its reflected radiance distribution is described by the Lambert model [1], whether the surface is rough or smooth [2,3]. To render accurate images reliably and easily, the reflectance of surfaces must be rendered accurately. The most direct way to ensure correct simulation is to use great reflectance measurements. Such measurements can guide the choice of parameters for existing reflectance models, and if they are sufficiently complete, they can be used as input for renderers or provide the basis for entirely new models. If you look at [Fig. 2-1] (a) (b), you can see that the light goes out in all directions. Light going in the other direction cannot reach the observer. Subsurface Scattering is a rendering technique for realistic simulation of translucent materials based on BSSRDF (Bidirectional Subsurface Reflectance Distribution Function)[3]. This method can be applied to organic materials such as skin, bones and plant materials, and to materials such as ice and marble that are scattered by light. The BSSRDF can describe light transport between any two rays that hit a surface, whereas the BRDF assumes that light entering material leaves the material at the same position[3].

In this paper, the shader language that can be used in the latest graphics libraries such as OpenGL can be applied by mixing several rendering techniques. Based on this, this study proposes a mixed rendering technique based on a mathematical model that may be utilized all the advantages of various techniques.

2. Rendering structure theory

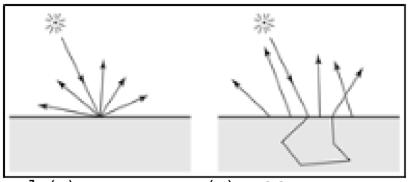
It is important for game Engine developers to be familiar with the concept of game components. All objects that make up the game contain components that affect their appearance or behavior. If it is the script that determines the behavior of the game object, the renderer determines how the object will appear on the screen. There are several renderers in game engine, but the renderers are slightly different depending on the type of object you want to visualize. And for 3D models, it will be usually used a mesh renderer. An object uses only one renderer, but a renderer can visualize multiple materials on screen. If it might be compared 3D graphics to a food chain, the single shader that each material encloses will be the ultimate predator. The structure correlation between these components is presented in Figure[1-1]. By understanding the differences between these components, you can understand exactly how the material shaders work.

As this rendering technology is the most noticeable technology in computer and video games, it is a function that almost all game engines provide. Although the current game engine began in the era of 3D and first-person shooter games, a few 2D game engines appeared due to the demand for game developers and 2D graphics[2]. Since the appearance of games expressing polygonal objects in the mid- to late 1990s, rendering engines have progressed a lot [1,2].



[Fig. 1-1] System structure of Renderer

In 2007, game engines tend to support technologies such as high dynamic range rendering, pixel shaders, vertex shaders, normal mapping, and depth of field, level of detail, light scattering simulation, and soft shadow. The other paper explains a semi-Lagrangian approximation scheme for the Oren - Nayar and the Phong model[9]. In addition, today's game engines provide a scene graph that depicts an object-oriented 3D game world. The scene graphic simplifies the design of the game and makes it possible to render a huge virtual world more efficiently[3]. Some engines offer only real-time 3D graphics rendering, leaving out all other necessary features. These engines require game developers to write the rest of the functions themselves or combine them with other middleware components. Engines of this type are commonly referred to as graphics engines and rendering engines. Graphics engines include Reel Forge, Auger, and Crystal Space, Genesis 3D, and Unity.

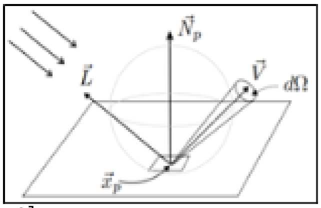


[Fig. 2-1] (a) BRDF and (b) BSSRDF radiances of light [3,6]

The rendering technology is a process of converting a prepared model into a picture and obtaining a photorealistic image by calculating the flow of light or obtaining the results through non-photorealistic rendering (NPR) using a specific technique. The basic computational processes that implement photorealistic rendering technology include transmission of light (how much light is transmitted from one site to another) and scattering (how the surface reacts to light transmission). This process is usually done using 3D computer graphics software or 3D graphics APIs. In addition, transforming scene into an appropriately shaped result in rendering includes a 3D projection that displays a 3D image on a 2D plane. Subsurface Scattering is a phenomenon that occurs mainly on a surface where light penetrates the inside of the surface and then bounces out of the surface. Skin, marble, milk, candles, etc. are the substances to be understood. The BRDF that we mainly use is made under the assumption that the incident point and the reflecting point of light are the same as shown in [Fig. 2-1](a). The problem of BRDF for a rough surface is much simplified when the various global geometrical and radio metrical effects are confined to finite regions[10]. Several experiments on synthetic and real images with different reflectance properties have been carried out to assess the effectiveness of SFS approach[11].

2.1 Lambertian surface model

This diffuse reflection model was proposed by Johann Heinrich Lambert in 1760 and is probably the most widely used reflection model in computer vision and graphics. To model light reflections in computer graphics, BRDF is mainly used [5]. It is a function representing the relationship between light reflected along the outgoing direction and light coming in from the incoming direction. A perfect diffuse surface has a BRDF that has the same value for all incidence and egress directions. This significantly reduces the computation at the processing stage mathematically, so it is commonly used to model physically plausible diffuse surfaces even if there is no pure diffuse material in the real world.



[Fig. 2-2] Reflectance model of Lambertian[4]

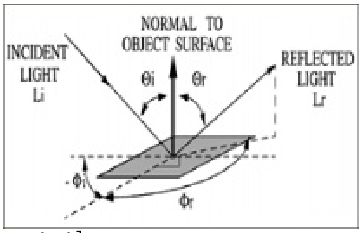
Most reflectance models can be broadly classified into two categories: diffuse and reflective. In computer vision and computer graphics, the diffuse component is often considered of Lambert model. A surface that follows Lambert's Law looks equally bright in all fields of view. However, for many real surfaces such as concrete, plaster, sand, etc., the Lambertian model is an inadequate approximation of the dispersive component. This is mainly idea because the Lambertian model does not consider the roughness of the surface. This BRDF is called to Lambertian reflection because it follows Lambert's law of cosine as [Fig. 2-2]. Diffusely reflecting objects appear similarly from all viewing direction. The radiance finally is not dependent on the viewing direction, $\overrightarrow{V}$. This technique causes all closed polygons (such as a triangle within a 3D mesh) to reflect light equally in all directions when rendered. In effect, a small flat region rotated around its normal vector will not change the way it reflects the light[3]. However, the region will change the way it reflects light if it is tilted away from its initial normal vector because the area is illuminated by a smaller fraction of the incident radiation. The reflection is calculated by taking the dot product of the surface's normal vector, $\overrightarrow{N_p}$ and a normalized light-direction vector, $\overrightarrow{L}$ pointing from the surface to the light source[4].

2.2 Oren-Nayar surface model

This is a lighting model developed in 1994 in collaboration with Michael Oren, a computer science professor named Shree K. Nayar in India. It is a lighting model for reflectance designed to express detailed graphics more realistically in a state where the basics of computer graphics have already been made. Their ultimate goal was to create a diffuse reflection model as close to the actual diffuse reflection as possible. The Lambert method was very good at handling diffuse reflections properly in the low performance of computers in the past. However, as computer structure performance increased, higher graphics performance began to be demanded from developers.

Reflectance is a physical property of a material that describes how it reflects incident light. The shape of various materials is largely determined by their reflectivity properties. A rough surface can be modeled as a series of faces with different slopes, each face being a small flat patch.

Both the camera's retina and the pixel's photoreceptor are finite-domain detectors, so the actual macroscopic (much larger than the wavelength of the incident light) surface roughness is often projected onto a single sensing element, which in many respects produces aggregated brightness values. Lambertian expressed very well when observing a single planar face, but a collection of faces with different directions is guaranteed to violate Lambert's law. The main reason for that is that the reduced face area changes according to the different viewing directions, so the surface shape varies with the view.



[Fig. 2–3] The surface standard model of Oren–Nayar[1]

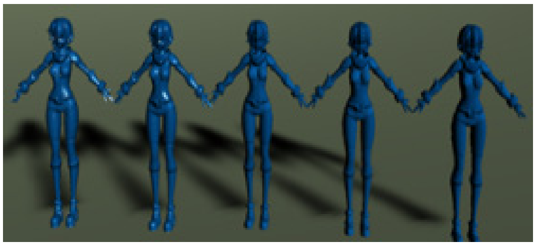
[Fig. 2–4] A face rendered using the BRDF model (left) and the BSSRDF model (right)[3]

3. Proposed reflective idea

Oren and Nayar indicated that Lambert's model for diffuse reflection is extensively used in computation vision. It is used explicitly by methods such as shape from shading and photometric stereo, and implicitly by methods such as binocular stereo and motion detection. For several real-world objects, the Lambert model can prove to be a very inaccurate approximation to the diffuse component. While the brightness of a Lambert surface is independent of viewing direction, the brightness of a rough diffuse surface increases as the viewer approaches the source direction. The simulations showed that coefficient makes a relatively small contribution to the total radiance as the model equation[8]. 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[7].

Assuming the facets to be Gaussian distributed and considering an additional light attenuation that models the inverse square law known from light transport, the final brightness equation for all facets of the Oren-Nayar surface reflectance model is proposed as [7,8], and ρ is the albedo value of the surface, and σ is the roughness of the surface. If the σ = 0 is in the same plane, the following equation (1) will be A=1 and B=0, and thus the Oren-Nayar model simplifies to the Lambertian model.

$$L_r = \frac{\rho}{\pi} E_0((N \bullet L)(1.0 - 0.5\frac{\sigma^2}{\sigma^2 + 0.33})$$
$$+ (0.45\frac{\sigma^2}{\sigma^2 + 0.09})$$
$$MAX(0, \frac{E \bullet L - (N \bullet E)(N \bullet L)}{\sqrt{(1-(N \bullet L)^2 + (1-(N \bullet E)^2}}) X$$
$$\sqrt{(1-(N \bullet L)^2 + (1-(N \bullet E)^2)} \quad MAX(1, \frac{N \bullet L}{N \bullet E}) \tag{1}$$



[Fig. 3–1] The surface rendering reflections (Oren–Nayar) of transformation with roughness (σ = 0, 0.2, and 0.33) of last 3 characters.

4. Conclusion

We have implemented the surface rendering analysis for useful characters including Oren-Nayar mathematical diffuse, and hybrid models using Unity engine on GPU. Henrick rendered a practical model of face for subsurface light simulation as shown in [Fig. 2-4][3]. We simulated a modeling character behavior and visualized the rendering reflection on surface with the proposed mathematical method as shown in [Fig. 3-1]. The techniques have been used for surface rendering with roughness values, σ = 0, 0.2, and 0.3 of last 3 characters with main 9.7ms, render thread 1.7ms, 103.4 FPS and NVIDIA Quadro 600 through Intel Xeon 3.2GHZ 4GB. The first 2 characters were simulated by standard specular and Lambert model. It is true that research and development of rendering techniques are in progress with high computation. The rendering process is almost impossible to improve the performance manually, so the rendering quality is influenced by the renderer function used by previous model or the development of rendering program. Currently, commercial renderers that are widely used continue to strive to achieve both real-time and high-quality purposes at once, and with the development of hardware performance, the rendering speed is rapidly increasing. However, a considerable amount of rendering time is still required to create a realistic image that is indistinguishable from the real one.

For fast rendering and realistic process for 3D game characters, it was utilized by the shader codes, supporting both declarative programming of the fixed-function pipeline and shader codes written in CG. Diffuse reflection scatters light in center, assume equally all direction called the improved Oren-Nayar surface model. When a light hits a rough surface of the game character, it rebounds in a direction not much related to its incoming direction. When the new direction is statistically independent of the old, the new direction has a very specific distribution with some roughness. The Oren-Nayar model applied, center to right, is shown slightly darker than Lambertian model. Thus, we believe that our work can be considered a nice example for the successful analysis and rendering model.

In fact, it is not a lighting model that is often used in game graphics. In most cases, it can be replaced by the Lambert model, because it has an excessive amount of computation compared to other lighting models. Since there are many operations that are used redundantly while performing other operations anyway, you can use the ones that have been calculated in advance, but performing other operations is complicated for just one diffuse. So, in general, this lighting model can be used in games that require extreme photorealism to achieve a very good effect.

REFERENCES

[1] Wolff, L.B., Nayar, S.K. & Oren, M. "Improved Diffuse Reflection Models for Computer Vision", International Journal of Computer Vision 30. 1998; 55 - 71.

[2] https://graphics.stanford.edu/papers/bssrdf/bssrdf.pdf.

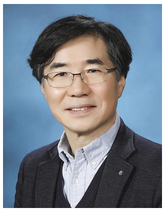
[3] Henrik Wan Jensen, Stephen R. Marcher, Marc Levey and Pat Hanrahan. A Practical Model for Subsurface Light Transport. Proceedings of SIGGRAPH. 2001 August; 511-18.

[4] http://www.cs.toronto.edu/~fleet/courses/2503/fall11/Handouts/light.pdf

[5] Seongyeon Moon at al. "Cinematic Wound Synthesis Optimized for Real-time Gameplay" International J of Software Engineering and its Applications. 2013; 7(1): 221 - 28.

[6] Olga Souring at al. Subconscious Learning via Games and Social Media. Singapore: Springer ;2015.p.1 -10.

[7] https://en.wikipedia.org/wiki/Oren%E2%80%93Nayar_reflectance_model (accessed 2020).
[8] Yong Chul Ju, Michael Breuss. Andres Bruhn and Silvano Galliani. "Shape from Shading for Rough Surfaces. Analysis of the Oren-Nayar Model" In Proceedings British Machine Vision Conference, 2012 January; 104.1-104.11. http://dx.doi.org/10.5244/C.26.104.
[9] Tozza, Silvia & Falcone, Maurizio. "Analysis and Approximation of Some Shape-from-Shading Models for Non-Lambertian Surfaces" Journal of Mathematical Imaging and Vision. 2016 Feburay;153-78.
[10] Jan J. Koenderink and Andrea J Van Doorn. "Bidirectional Reflection Distribution Function of Thoroughly Pitted Surface", International Journal of Computer Vision.1999. 31(2/3): 129 - 44.
[11] Guohui Wang, Xuan Zhang, Jin Cheng. "A Unified Shape-From-Shading Approach for 3D Surface Reconstruction Using Fast Eikonal Solvers" International Journal of Optics. 2020 May.
[12] Leiming Li, Wenyao Zhu, Hongwei Hu, "Multivisual Animation Character 3D Model Design Method based on VR Technology", Complexity, Vol.2021, Article ID 9988803, 12 pages



김 성 동 (Seongdong Kim)

약 력 : 2021. 8 Professor, Kaywon Univ. of Art & Design, Dept of Game Media
2001. 6 Ph.D. Kwangwoon University
2002. 8 Research Scientist University of Notre Dame
2011. 3 Visiting Research Scholar of Purdue University

관심분야 : 3D Graphic Programming, PBR, XR

이 면 재 (MyounJae Lee)

약 력 : 2006. 2 Ph.D. Hongik University
2021. Current: Professor, Baekseok University, Division of Computer Engineering

관심분야: Computer Graphics, Game Engine