Real-Time Physically Based Clay Rendering using a multi-layered BRDF

Kieran Bowsher
19036291

University of the West of England

March 18, 2022

The task was to create a shader inside the Unity or Unreal engine that could re-create the appearance of clay. Two approaches were used to re-create the appearance of clay. These were: Multi-layered BRDF and Disney's Clear-coat gloss. The Multi-layered BRDF was chosen since it allows for simulating multi-layers material like clay as it's made up of layers of water and clay. Finally, the shader was used to re-create reference images found online and comparison were made to summarize how well it did.

1 Introduction

Physically based rendering (PBR) was introduced into computer graphics to allow surfaces to react to lighting more realistically. While this method does an excellent job at replicating dialectic and conductive materials, it struggle when replicating surfaces that have multiple layers. An example of this could be wet concrete.

2 Background and Research

2.1 Previous work

There are already examples online of 3D models and animations done using clay materials. Many 3D artist on sites like Artstation and Sketchfab have showcased their rendered work. However, nearly all of their artworks use renderers like Blender or VRay which uses more advance version of Physically based rendering with Ray-tracing/Path-Tracing lighting.

2.2 Clay types

As explained by (Marie, 2021) and (ikitmovi, 2018), there are different types of clay. For pottery, ceramic clay like Earthenware, Ball, Stoneware and Porcelain are used. From **Figure 1**, each ceramic clay has a sheen/glossy look. This glossy effect is seen in the raw clay balls but is even more present in the final baked pottery.
For 'Claymation', softer more malleable clay like oil-based, water-based and polymer clay are used for there ease of use and lack of drying. Similar to ceramic clay, these also have a sheen/glossy look as well.

Figure 1: *Left to right. Earthenware, Ball, Stoneware, Porcelain clay*

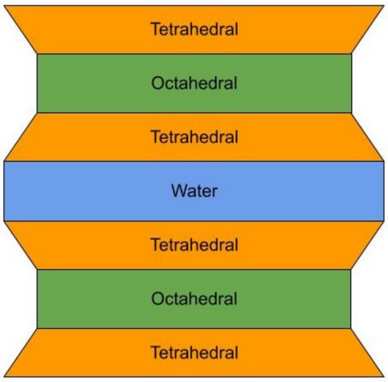

Figure 2: *Diagram showing the sheet structure of clay*

2.3 Molecular structure of Clay

(Pūtaiao, 2010) and (Neeraj et al., 2021) explain that clay is made up of sheets that have a Tetrahedral/Octahedral/Tetrahedral structure. These sheets are separated by molecules like water or oil as shown in **Figure 2**. Clay families like smectite and kaelintie have different sheet structures which define their surface appearance. Since clay has water that surrounds the Tetrahedral/Octahedral/Tetrahedral sheets, this allows clay to have a rough appearance but with a glossy highlight.

2.4 Light behaviour on clay surfaces

Since the microscopic surface of clay is very rough, light bounces at different angles when hitting the surface. This is also known as diffuse reflectance allowing the clay to have a rough appearance. However, since water and oils are used for clay, the appearance of the Clay's surfaces can change. As shown in **Figure 3**, when a thin layer of water is applied to the surface of a clay ball, it surface gets darker. This is caused by a phenomenon called Total Internal reflection where the layer of water allows more light to be absorbed into the clay's surface making it appear darker.

2.5 Physically Based Rendering

Physically Based Rendering (PBR) has been used for real-time graphics since the early 2010s. It approximates light behaviour on different surfaces like dialectic and conductive materials. It uses a simplified version of the rendering equation also called the reflectance equation.

$$L_o(p, \omega_o) = \int_{\Omega} f_r(p, \omega_i, \omega_o) L_i(p, \omega_i) n \cdot \omega_i d_{\omega i} \tag{1}$$

For PBR, most implimentation like (Karis, 2013) and (Open-Gl, n.d.) use the Cook-Torrance BRDF with the reflectance equation as shown in equation 2

Figure 3: *Left: Dry ball of clay. Right: Wet ball of clay*

$$L_o(p, \omega_o) = \int_{\Omega} (k_d \frac{c}{\pi} + \frac{DFG}{4(\omega_o \cdot n)(\omega_i \cdot n)}) L_i(p, \omega_i) n \cdot \omega_i d_{\omega i} \tag{2}$$

The Cook-Torrance Di-Directional Reflectant Distribution Function (BRDF) is made up of three functions that control the micro-facets of a surface.

2.5.1 Micro-facet Model

Figure 4: *Diagram showing the distribution of micro-facets to describe rough or smooth surfaces. (Open-Gl, n.d.)*

Figure 5: *Diagram showing light bouncing off different micro-facet surfaces. (Open-Gl, n.d.)*

The Micro-facet Model is a theory that describes the microscopic surface of an object as small perfectly reflective mirrors as demonstrated in **Figure 4**. The angle and alignment of these micro-facets are controlled by the roughness of the surface. This causes the incoming light to scatter more at different angles resulting in a more widespread specular reflection as shown in **Figure 5**.

Micro-facets can be calculated using a halfway vector which is a vector between the light and view vector.

This can be described as

$$h = \frac{l + v}{||l + v||} \quad (3)$$

Essentially, the more micro-facets that are aligned with the halfway vector, the sharper and strong the specular reflection is.

2.5.2 BRDF

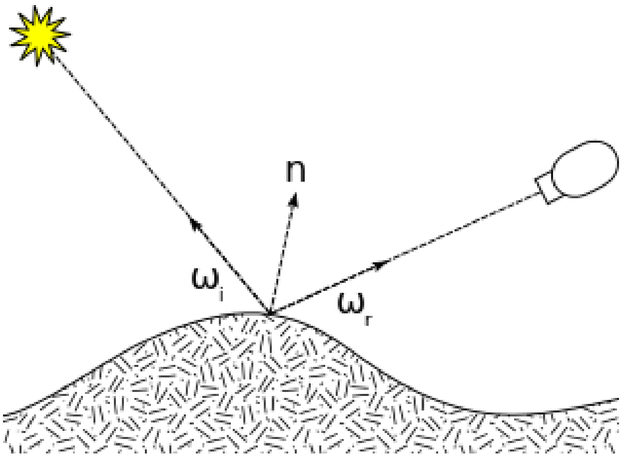

Figure 6: *Diagram showing the vectors that are used for BRDF. (Wikipedia, 2021)*

The BRDF is a function that approximates how each light ray contributes to the reflected light on a surface based on the surface properties. The function uses information like:

- The incoming light direction w_i.
- The outgoing view direction w_o.
- The surface normal n.
- Surface parameter α which represents the roughness of a surface.
- The halfway vector between w_i and w_o

Currently, most real-time usage of PBR like (Karis, 2013) uses the Cook-Torrance BRDF (Robert, 1982). This is mainly due to that model having better performance than other models.

The Cook-Torrance BRDF model uses three functions. These functions are:

- F is the Fresnel function that describes the ratio of surface reflection at different angles.
- D is the normal distribution function that approximates the surface's micro-facets that are aligned to the halfway vector. These micro-facets are controlled by the roughness of the surface.
- G is the geometry function that influences self-shadowing on micro-facets based on the roughness of the surface.

The Cook-Torrance BRDF is calculated as

$$f_r = \frac{FDG}{4 \cdot (N \cdot L)(N \cdot V)} \quad (4)$$

Where FDG are the functions of the BRDF and $(N \cdot L)$ is the dot product of the surface normal and light direction with $(N \cdot V)$ being the dot product of the surface normal and view direction. This function produces the specular component of the surface.

2.5.3 Multi-layered BRDF

While physically based rendering is used heavily throughout games, physically based rendering with multi-layered BRDF are not so widely used. There has been work done by (Belcour and Barla, 2017) and (Belcour, 2018) which explores using multi-layered BRDF for various effects like iridescent in the Unity engine.

As well, multi-layered BRDF have been used for rendering surfaces in Call Of Duty Advance Warfare (Michal, 2017) by using surface, micro and macro properties to define a material surface. All the citations in this section base their works and findings from (Andrea et al., 2007) which has become the basis for most multi-layered BRDF implementation.

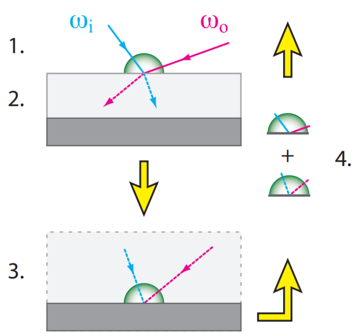

Figure 7: *Diagram from (Andrea et al., 2007) showing how two BRDF are evaluated*

(Andrea et al., 2007) explains how two BRDF layers can be used together to create multi-layered materials. The process works by evaluating the top most layer BRDF with the incoming light direction w_i and the outgoing view direction w_o and producing two refracted direction of $w_{i'}$ and $w_{o'}$. The two refracted directions are then evaluated in the next layer BRDF as shown in **Figure 7**. Finally, these are combined together using transmission, absorption and internal reflectance.

As mentioned in section 2.3, clay is made up of layers of clay and water. This multi-layered BRDF would work for simulating this type of surface.

3 Method

This section will explain the methods used to produce the clay shader. There were two clay shader made using Disney's clear-coat gloss and a multi-layered brdf. While the multi-layered brdf shader is more accurate, this section will still explain the implementation for Disney's clear-coat gloss. Comparison between the two are made in section 4.
The clay shader was developed using Unity 2021.1.21f1 and the project uses the Standard Rendering Pipeline (SRP). The features that the multi-layered brdf clay shader has include:

- Blended normal maps
- Control of two BRDF with exposed parameters like specular intensity and roughness intensity
- Image Based Lighting (IBL) for the top layer
- Roughness textures for extra artistic control

The clay shader only has two layers which use the Cook-Torrance BRDF similar to the papers referenced in this section with (Karis, 2013) and (Open-Gl, n.d.).

3.0.1 Cook-Torrance BRDF

The Cook-Torrance BRDF is describes in equation (4). For each function, a model is used.

3.0.1.1 Frensel Function

For the Fresnel function F, Schlick Fresnel approximation is used and this is described as:

$$F(\theta) = F_0 + (1 - F_0)(1 - N \cdot V)^5 \tag{5}$$

where F_0 is the specular reflectance at normal incidence and its value is depended on the index of refraction. By default, F_0 is usually valued as 0.04.

3.0.1.2 Normal Distribution Function

For the normal distribution function D, Disney's GGX/Trowbridge-Reitz is used which is calculated as:

$$D(h) = \frac{\alpha^2}{\pi((n \cdot h)^2(\alpha^2 - 1) + 1)^2} \tag{6}$$

Here α is the value of the surface roughness.

3.0.1.3 Geometry Function

For the geometry function G, a combination of GGX and Schlick-Beckmann approximation also known as Schlick-GGX is used and is described as:

$$G(n, v, \alpha) = \frac{n \cdot v}{(n \cdot v)(1 - k) + k} \tag{7}$$

where k is the remapped value of α.

$$k = \frac{(\alpha + 1)^2}{8} \tag{8}$$

Note that $(\alpha + 1)$ is a modification done by (Burely, 2012) to reduce roughness "hotness". Finally, Smith's method is used to combine two Schlick-GGX function to create the geometry function.

$$G_{Smith}(n, v, l, a) = G(n, v, a)G(n, l, a) \tag{9}$$

3.0.2 Disney's clear-coat Gloss

As mentioned in section 2.2, clay appearance tend to have a rough but glossy look. Originally, during research, it was decided to use Disney's clear-coat gloss since that could produce the glossy effect on-top of the rough surface.
To calculate Disney's clear-coat gloss, the three functions as explained in section 3.0.1 are used. However, the distribution function is replaced by a General Trowbridge-Reitz (GTR) function. As explained by (Burely, 2012) this function is summarized as:

$$D_{GTR}(\theta_h) = \frac{(\alpha^2 - 1)}{\pi \log \alpha^2} \frac{1}{(1 + (\alpha^2 - 1)\cos^2 \theta h)} \tag{10}$$

where α is the roughness of the surface and $\cos^2 \theta h$ is the dot product of the surface normal and half vector $N \cdot H^2$.

For the fresnel function, Schlick Fresnel is used with a fixed F_0 of 0.04. For the geometry function, G_{Smith} is used with a fixed α value of 0.25.

Combining these function together can be described as:

$$ClearCoatGloss = 1 * F * D_{GTR} * G \tag{11}$$

3.0.3 Layering multiple BRDF

The implementation of the multi-layered BRDF is based off (Andrea et al., 2007), (Andrea et al., 2009) and (Elek, 2010). Calculating each layer BRDF is similar to any PBR shader that uses Cook-Torrance BRDF as shown in equation 4. However, the view direction and light direction will need to be refracted for the second layer BRDF as explained by (Andrea et al., 2007).

HLSL already has a function for this called $refract(i, n, \eta)$. Using this function and inverting it will produce the refracted view and light direction and applying them to the next BRDF looks like this:

$$f_{r2}(L\prime, V\prime) = \frac{FDG}{4 \cdot (N \cdot L\prime)(N \cdot V\prime)} \tag{12}$$

Its the same as the first layer with the only difference is using $L\prime$ and $V\prime$ which are the refracted light and view direction. To combine both layers, a recurrent equation from (Andrea et al., 2007) is used as shown in equation 13

$$f_r = f_{r1}(L, V) + T_{12} \cdot f_{r2}(L\prime, V\prime) \cdot a \cdot t \tag{13}$$

where:

- $f_{r1}(L, V)$ is Layer 1 BRDF
- $f_{r2}(L\prime, V\prime)$ is Layer 2 BRDF
- $T12$ is the transmission from layer 1 to layer 2.
- a is the amount of light absorbed between layer 1 and layer 2
- t is the internal reflectance of light between layer 1 and layer 2

$T12$ can be summarized as $T12 = 1 - f_{r1}F$ with $f_{r1}F$ being layer 1's Fresnel function.

For absorption, the top layer will allow the bottom layer to absorb more light using the constant wavelength-dependent absorption coefficient α which defines how much light the material absorbs. For this shader, α is set as 0.8. Absorption a is calculated as

$$a = e^{-\alpha l} \tag{14}$$

where l is the length at which the light travels through the layer. l is determined by the density d of the top layer. This can be summarized as

$$l = d(\frac{1}{N \cdot L\prime} + \frac{1}{N \cdot V\prime}) \tag{15}$$

with d as the density of the top layer and $L\prime$ and $V\prime$ being the refracted light and view directions.

t which is the internal reflectance of the light passing through layer 1 is calculated as $t = (1 - G) + T_{21} \cdot G$ with G being layer 1's Geometry function. For simplicity, T_{21} is summarized as $T_{21} = T_{12}$

3.1 Diffuse

For the diffuse lighting, Lambert's Cosine Law (Lambert, 1760) $\cos\theta = N \cdot L$ is mostly used in any implementation of a surface shader. Most of the time the implementation look like equation 16.

$$DiffuseSurfaceColour = \frac{pd}{\pi} * L_i * \cos\theta \tag{16}$$

where $\frac{pd}{\pi}$ is the surface colour/albedo texture divided by π for normalizing and L_i is light intensity multiplied by light colour.

The reason Lambert's Cosine Law is mainly used is due to its simplicity and high performance. However, it makes the presumption that all surfaces are equally rough and for that reason equal diffuse.

As explained by (Elek, 2010), the Oren–Nayar reflectance model (Oren and Nayar, 1994) is used for surfaces with high diffuse back-scattering like clay or the moon's surface and for that reason, Oren-Nayar is used in the shader.

Oren-Nayar reflectance mode can be describe as:

$$L_r = \frac{p}{\pi} \cdot \cos\theta_i \cdot (A + (B \cdot \max[0, \cos(\phi_i - \phi_r)] \cdot \sin\alpha \cdot \tan\beta)) \cdot E_0 \tag{17}$$

where

- $\alpha = \max(\theta_i, \theta_r)$
- $\beta = \min(\theta_i, \theta_r)$
- $A = 1 - 0.5\frac{\alpha^2}{\alpha^2 + 0.33}$
- $B = 0.45\frac{\alpha^2}{\alpha^2 + 0.09}$

Combining both the specular and diffuse components together is summarized mathematically as

$$finalColor = Diffuse \times a \times f_r \tag{18}$$

with

$$Diffuse = (1 - f_{r2}F) \times OrenNayar(p, l, n) \tag{19}$$

4 Evaluation

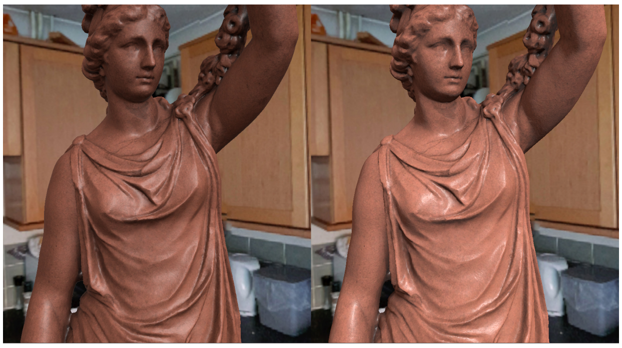

Figure 8: *Left: Mutli-layerd brdf. Right: Disney's clear coat gloss*

Figure 8 shows the differences between a shader using Disney's clear coat gloss and a multi-layered brdf shader. A big difference between the two shaders is the colour of the surface. Since Disney's clear coat gloss only adds a secondary specular lope, it doesn't account for the light getting trapped by the top layer and being absorbed into the clay surface. The shader can be edited to include absorption but this won't be a accurate as the multi-layered brdf.
Another difference is the specularity between both shaders. Disney's clear coat gloss produces a brighter and much sharper specular highlight whereas the multi-layered brdf produces a flatter and more spread out specular highlight. This is true for **Figure 9** as the earthenware clay ball has a much flatter and spread out specular highlight.

Figure 9 is another comparison between the multi-layered brdf and a ball of earthenware clay. Both pictures show a similar glossy effect and the mutli-layered brdf produces similar results to the earthenware clay.

Figure 9: *Comparison between multi-layered brdf and a ball of earthenware clay*

Figure 10: *Comparison between a clay model (left) and the clay shader (right)*

Both **Figure 10** and **Figure 11** showcase the results from the multi-layered clay shader with an image that it was referenced from. Results from **Figure 10** do show that the shader can produce similar features like the diffuse lighting and specular highlights similar to the reference image. While the lighting in both pictures is different, the general surface appearance does seem to be similar.

However, an attempt was made to re-create a ball of clay from **Figure 11** using Photogrammetry and capturing the light of the scene into a cube map. While the clay ball does look similar in appearance, the shader results seem to have a high amount of diffuse lighting unlike the reference image where the diffuse lighting is flat.
Overall, the shader does re-create the appearance of clay well but improvement into better lighting in scene could help create a more convincing effect.

5 Conclusion

Using the multi-layered BRDF for simulating surfaces that have multiple layers like clay seems to work well. While other methods like Disney's clear coat gloss can somewhat reproduce similar effects. These effects are not physically accurate. While the appearance is not

Figure 11: *Another comparison between clay model (left) and clay shader (right)*

completely accurate, what the shader produces does look convincing.

Bibliography

Andrea, W. et al. (2007). *Arbitrarily Layered Micro-Facet Surfaces*. URL: `https://www.cg.tuwien.ac.at/research/publications/2007/weidlich_2007_almfs/weidlich_2007_almfs-paper.pdf`. (accessed: 29.12.2021).

– (2009). *Exploring the Potential of Layered BRDF Models*. URL: `https://publik.tuwien.ac.at/files/PubDat_182027.pdf`. (accessed: 29.12.2021).

Belcour, Laurent (2018). "Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators". In: *ACM Transactions on Graphics* 37.4, p. 1. DOI: `10.1145/3197517.3201289`. URL: `https://hal.archives-ouvertes.fr/hal-01785457`.

Belcour, Laurent and Pascal Barla (July 2017). "A Practical Extension to Microfacet Theory for the Modeling of Varying Iridescence". In: *ACM Transactions on Graphics* 36.4, p. 65. DOI: `10.1145/3072959.3073620`. URL: `https://hal.archives-ouvertes.fr/hal-01518344`.

Burely, Brent (2012). *Physically Based Shading at Disney*. URL: `https://media.disneyanimation.com/uploads/production/publication_asset/48/asset/s2012_pbs_disney_brdf_notes_v3.pdf`. (accessed: 29.12.2021).

Elek, Oskar (2010). *Layered Materials in Real-Time Rendering*. URL: `https://old.cescg.org/CESCG-2010/papers/ElekOskar.pdf`. (accessed: 24.01.2022).

ikitmovi (2018). *Best Clay To Use For Claymation Animation*. URL: `https://www.ikitmovie.com/best-clay-to-use-for-claymation-animation/`. (accessed: 29.11.2021).

Karis, Brian (2013). *Real Shading in Unreal Engine 4*. URL: https://cdn2.unrealengine.com/Resources/files/2013SiggraphPresentationsNotes-26915738.pdf. (accessed: 29.12.2021).

Lambert, Johann Heinrich (1760). *I. H. Lambert... Photometria sive de mensura et gradibus luminis, colorum et umbrae*. URL: https://www.e-rara.ch/zut/doi/10.3931/e-rara-9488. (accessed: 17.02.2022).

Marie (2021). *What Are the Four Types of Clay?* URL: https://potterycrafters.com/what-are-the-four-types-of-clay/. (accessed: 29.11.2021).

Michal Drobot. Adam, Micciulla (2017). *Practical Multilayered Materials*. URL: https://www.activision.com/cdn/research/s2017_pbs_multilayered_slides_final.pdf. (accessed: 12.01.2022).

Neeraj, K. et al. (2021). *Basics of Clay Minerals and Their Characteristic Properties*. URL: https://www.intechopen.com/online-first/76780. (accessed: 01.12.2021).

Open-Gl (n.d.). *Theory of PBR*. URL: https://learnopengl.com/PBR/Theory. (accessed: 15.12.2021).

Oren, Michael and Shree K. Nayar (1994). *Generalization of Lambert's Reflectance Model*. URL: https://doi.org/10.1145/192161.192213.

Pūtaiao, Science Learning Hub – Pokapū Akoranga (2010). *What is clay?* URL: https://www.sciencelearn.org.nz/resources/1771-what-is-clay. (accessed: 29.11.2021).

Robert Cook. Kenneth, Torrance (1982). *A Reflectance Model for Computer Graphics*. URL: https://graphics.pixar.com/library/ReflectanceModel/paper.pdf. (accessed: 02.01.2022).

Wikipedia (2021). *Bidirectional reflectance distribution function*. URL: https://en.wikipedia.org/wiki/Bidirectional_reflectance_distribution_function. (accessed: 11.02.2022).