School of Computing
FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

# Final Report

## Physically Based Shading Models in Real-time Rendering

**Evan James**

**Submitted in accordance with the requirements for the degree of**
**Computer Science (Digital & Technology Solutions) BSc**

2021/22

COMP3932 Synoptic Project

The candidate confirms that the following have been submitted.

| Items | Format | Recipient(s) and Date |
|---|---|---|
| Final Report | PDF file | Uploaded to Minerva (DD/MM/YY) |
| <Example> Scanned participant consent forms | PDF file / file archive | Uploaded to Minerva (DD/MM/YY) |
| <Example> Link to online code repository | URL | Sent to supervisor and assessor (DD/MM/YY) |
| <Example> User manuals | PDF file | Sent to client and supervisor (DD/MM/YY) |

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of Student) ____________________

## Summary

<Concise statement of the problem you intended to solve and main achievements (no more than one A4 page)>

**Acknowledgements**

<The page should contain any acknowledgements to those who have assisted with your work. Where you have worked as part of a team, you should, where appropriate, reference to any contribution made by other to the project.>

Note that it is not acceptable to solicit assistance on 'proof reading' which is defined as the "the systematic checking and identification of errors in spelling, punctuation, grammar and sentence construction, formatting and layout in the test"; see
`https://www.leeds.ac.uk/secretariat/documents/proof_reading_policy.pdf`

# Contents

# Chapter 1

# Introduction and Background Research

## 1.1 Introduction

Rendering is the process of generating images, or *frames*, of a virtual world. Real-time rendering requires that the generation of these frames is done at a fast enough rate so that the viewer feels they are taking part in an immersive, dynamic experience. Typically, this rate needs to be at least 30 FPS (Frames Per Second), with 60 FPS and beyond being desirable [1]. This imposes a maximum time budget of 33 to 16 milliseconds in which each frame must be generated, the *frame time*. Real-time rendering presents a compelling problem: how can the visual fidelity of a rendered scene be maximised, whilst adhering to this strict computational budget.

Rendering can be performed using one of two techniques, ray tracing or rasterization. Ray tracing is based on a model that is analogous to how humans perceive light and colour in the real-world. In the real-world, rays of light are produced from many sources, bounce from one object to the next, and eventually reach the viewers eyes. Ray tracing models this same process, but in reverse, with the rays originating from the views eyes, and being traced back to their sources. Provided enough rays are sampled, this approach produces very realistic images. Although ray tracing is the standard in the realm of movie production, its expensive computational requirements lead to frame times in the region of minutes instead of milliseconds [2]. Aside from so notable exceptions[1], this prohibits its use in real-time applications. As a result, real-time rendering employs another technique, rasterization.

With rasterization, each object in the world is composed of an arrangement of primitive shapes, most commonly, triangles, and their material is described through a number of parameters. When rendering, the world is transformed and projected onto a 2D plane. Within this plane, a fixed region maps to the space of the output image; all triangles that lie outside this region are clipped. The remaining triangles are then split into granular pieces, called *fragments*. A colour is calculated for each fragment by evaluating the amount of light that shines on that fragment in the world, and then how that light interacts with the material of the object that fragment belongs to. Performing this calculation is called *shading*, and how it is done is defined by a *shading model*. After resolving which fragments lie on top of which others, the final image is presented to the user. This whole rasterization process is referred to as the graphics rendering pipeline, and dedicated hardware has been developed to carry it out, the *Graphics Processing Unit* (GPU).

The appearance of the final rendered frames is largely determined by the shading model, and therefore the choice of such a model is crucial. For a long time, the standard shading model used for photo-realistic real-time rendering was Blinn-Phong; it was utilised in popular game engines, and was the default model used in OpenGL's fixed function pipeline [5] [6] [7]. Blinn-Phong is an empirical model: it is based on human observations of how light interacts with materials, rather

[1] With the introduction of hardware accelerated ray tracing on consumer GPUs [3], the use of ray tracing to render specific visual phenomena, such as reflections, has seen use in some modern games [4].

than the underlying real-world physical rules that govern those interactions [8]. Blinn-Phong can produce reasonably realistic images, and is computationally inexpensive - a very desirable trait for real-time rendering. However, due to its non-physically based nature, Blinn-Phong has many issues. Paramount amongst which is its inability to render certain physical phenomena, which limits the realism of rendered frames. Furthermore, the parameters of Blinn-Phong that are used to specify material properties, bear little relation to the characteristics of physical materials. This problem manifests itself in a tight coupling between material parameters and lighting conditions. In order to accurately depict the same physical material under different lighting conditions, it may be necessary to specify differing values for these parameters. This reduces the reusability of assets, making artist workflow more difficult.

In an effort to alleviate these issues, the replacement of Blinn-Phong in favour of physically based shading models has seen widespread adoption. Such models work by evaluating equations that simulate the real world physical interaction of light and objects. Using these models for shading is known as *Physically Based Shading* (PBS), and their use in the wider rendering pipeline is called *Physically Based Rendering* (PBR). PBS represented a seismic shift in the real-time rendering industry, with major game engines migrating to a PBR pipeline [9] [10].

The aim of this project is to investigate the use of physically based shading models in real time rendering. Specifically, I will seek to highlight the benefits of PBS when compared to the technology is superseded, Blinn-Phong shading.

The advantages of using PBS over Blinn-Phong shading can be broadly categorised into two groups: the improvements to artist workflow; and the improved photorealism. As mentioned previously, because of how materials are defined in Blinn-Phong shading they are often not portable between different lighting environments. In contrast, the parameters that determine materials in PBS are based on physical properties. This permits the reuse of materials and assets over different lighting configurations [10] [11]. Burley outlines how this reduction in the need for "material 're-do's" yields an extremely significant improvement to artist workflow [12]. Although these benefits are an important motivating factor for using PBS, the practical issues that arise from trying to investigate and quantify them (I don't have access to a team of artists) mean that this report will focus solely on exploring those advantages in the latter category – how does PBS render frames that are more photorealistic than Blinn-Phong?

Answering this question by simply commenting on the general perceived realism of a frame when compared to another, is a largely subjective exercise. Instead, in a concerted effort to be as objective as possible, I will examine the benefits of PBS by identifying physical phenomena that it models in its rendered frames, but that are absent when using Blinn-Phong shading. To this end, I will be developing a piece of software that can render scenes using both Blinn-Phong shading, and PBS.

## 1.2 Mathematical Notation

When reasoning about different mathematical expressions and models, it's of vital importance that a standard, coherent notation is used throughout.

| | |
|---|---|
| $\mathbf{n}$ | Normal vector |
| $\mathbf{l}$ | Light direction |
| $\mathbf{v}$ | View vector |
| $\mathbf{c}_x$ | The RGB triplet vector representing the colour of $x$ |
| $\mathbf{a} \cdot \mathbf{b}$ | The dot product of vectors $\mathbf{a}$ and $\mathbf{b}$ |
| $x^+$ | Clamp $x$ to 0 if $x < 0$ |

## 1.3 Literature review

The review begins with a brief discussion of the Blinn-Phong shading model; this provides the necessary background knowledge for the later comparison with physically based shading models to be performed. We then delve into the physics of how light interacts with matter, and how this pertains to shading. An exploration of the theory underpinning physically based shading models follows, and then we present several such models. After, we consider how lights can be represented in a physical manner. Finally, we finish by focusing on the wider PBR aspect with a discussion on how we store pixel colours, and the transformations that need to be applied before passing those colours to the display.

### 1.3.1 Blinn-Phong Shading

In 1975, Phong introduced a simple shading model for rendering realistic images [8]. The original model is parametrised as the sum of two terms, *diffuse* and *specular*, but in practice it is commonly supplemented with a third term, *ambient* [citation]. Each one describes the contribution of a different lighting component.

The diffuse term encodes the lighting effects of parity between a primitive's surface orientation and the direction of the light, with surfaces facing the light being illuminated more intensely than those facing away from it. This behaviour is formulated as:

$$\mathbf{c}_{shaded_{diff}} = \mathbf{c}_{surface_{diff}} \mathbf{c}_{light_{diff}} (\mathbf{n} \cdot \mathbf{l})^+$$

$\mathbf{c}_{shaded_{diff}}$, $\mathbf{c}_{surface_{diff}}$, and $\mathbf{c}_{light_{diff}}$ are the RGB triplets that represent the diffuse colour of the shaded fragment, the diffuse colour of the surface, and the diffuse colour of the incident light respectively. This separation of the light and surface colours into separate components was not present in Phong's original model. However, many implementations have increased the flexibility of the model by exposing these additional parameters [13]. The *normal*, $\mathbf{n}$, is the unit vector pointing away from the surface at the shaded point, giving the orientation of the surface. The *light direction*, $\mathbf{v}$, is the unit vector pointing in the direction of the incident light. See [Figure] for an illustration. The dot product of two unit vectors is equivalent to taking the cosine of the angle between them. Therefore, $(\mathbf{n} \cdot \mathbf{l})^+$ will increase from 0 to 1 as the angle between the incident light direction and the surface normal decreases. Thus, the more aligned the surface orientation and light direction are, the greater the intensity of the diffuse term. Negative values of the dot product indicate that the light direction is underneath the surface. In these cases the light is not incident upon the surface at all, dot product is clamped to 0.

The specular term of the model captures the ability for surfaces to exhibit highlights due to surface reflections. When light is incident upon a surface, it will experience some reflection, and when the reflected light is aligned with the direction of the viewer, this is perceived as small region of high intensity colour, a *specular highlight*. The formula for the specular term is:

$$\mathbf{c}_{shaded_{spec}} = \mathbf{c}_{surface_{spec}} \mathbf{c}_{light_{spec}} ((\mathbf{r} \cdot \mathbf{v})^{+})^{shininess}$$

Where $\mathbf{r}$ is the reflection of the incident light about the surface normal, and is defined as:

$$\mathbf{r} = 2(\mathbf{n} \cdot \mathbf{l})\mathbf{n} - \mathbf{l}$$

[Explain how the formula works]

[the shininess variable controls the concentration/focus of the reflection]

[Ambient term - crude simulation of GI; can get much more complex]

[Blinn improved with the half vector]

The area of this highlighted region is variable, and depends on the material of the surface.

An illustration of this specular behavior can be seen in [Figure].

[Variables would be defined by the material of the surface.]

Splitting shading into the evaluation of a diffuse and specular term is common practice, and the physical basis for such a practice is explained in section 1.3.2.

~~seminal paper in which he proposed a simple shading model. He also introduced the idea of computing this model per pixel rather than per vertex. As explained before, an object in the scene is made up of a collection of primitive shapes, this is called the mesh. Before we just calculated lighting per vertex of those primitives, in what is known as goroud shading, and inteprolated those colours across the mesh. Phong proposed that we interpolate the necessary data over the vertices but compute the colour for each pixel.~~

### 1.3.2 Physics of Light-Matter Interaction

- Blinn-Phong Shading
- Physics of light?
- Reflectance equation
- BRDFs
  - What makes a good BRDF (energy conservation and the reihtsz... reciprocity)
  - Fresnel Reflectance
  - Microfacet theory
  - Surface reflection
  - Subsurface reflection
- Lights
- HDR, tonemapping and gamma correction (display encoding)

# Chapter 2

# Methods

<Everything that comes under the ‘Methods’ criterion in the mark scheme should be described in one, or possibly more than one, chapter(s).>

## 2.1 A section

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

### 2.1.1 A sub-section

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## 2.2 Another section

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

# Chapter 3

# Results

<Results, evaluation (including user evaluation) *etc.* should be described in one or more chapters. See the 'Results and Discussion' criterion in the mark scheme for the sorts of material that may be included here.>

Fresnel effect shown; energy conservation shown; more artist options shown; show the specular lobe is more accurate using PBR approaches (page 338 of the real-time rendering book)? Take pictures from papers - reconstruct scene and show how PBR is close to the paper image and Blinn-Phong isn't.

## 3.1 A section

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

### 3.1.1 A sub-section

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

## 3.2 Another section

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

# Chapter 4

# Discussion

<Everything that comes under the 'Results and Discussion' criterion in the mark scheme that has not been addressed in an earlier chapter should be included in this final chapter. The following section headings are suggestions only.>

## 4.1 Conclusions

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

## 4.2 Ideas for future work

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

# Bibliography


[1] Mark Claypool, Kajal Claypool, and Feissal Damaa. “The effects of frame rate and resolution on users playing first person shooter games”. In: *Multimedia Computing and Networking 2006*. Ed. by Surendar Chandra and Carsten Griwodz. Vol. 6071. International Society for Optics and Photonics. SPIE, 2006, p. 8. DOI: 10.1117/12.648609. URL: https://doi.org/10.1117/12.648609.

[2] Per H. Christensen et al. “Ray Tracing for the Movie ‘Cars’”. In: *2006 IEEE Symposium on Interactive Ray Tracing*. 2006, p. 5. DOI: 10.1109/RT.2006.280208.

[3] NVIDIA. *NVIDIA Turing GPU Architecture*. Tech. rep. 2018, p. 31.

[4] Johannes Deligiannis and Jan Schmid. *It Just Works: Ray-Traced Reflections in 'Battlefield V'*. EA DICE and GDC. Mar. 18, 2018. URL: https://www.gdcvault.com/play/1026282/It-Just-Works-Ray-Traced (visited on 03/13/2022).

[5] Unity Technologies. *Specular*. URL: https://docs.unity3d.com/560/Documentation/Manual/shader-NormalSpecular.html (visited on 03/15/2022).

[6] Unreal Engine. *Materials Overview*. URL: https://docs.unrealengine.com/udk/Three/MaterialsOverview.html (visited on 03/15/2022).

[7] Khronos. *Fixed Function Pipeline*. URL: https://www.khronos.org/opengl/wiki/Fixed_Function_Pipeline (visited on 03/15/2022).

[8] Bui Tuong Phong. “Illumination for Computer Generated Pictures”. In: *Commun. ACM* 18.6 (1975), 311–317. ISSN: 0001-0782. DOI: 10.1145/360825.360839. URL: https://doi.org/10.1145/360825.360839.

[9] B Karis. “Real Shading in Unreal Engine 4”. In: *SIGGRAPH 2013 Course: Physically Based Shading in Theory and Practice*. Anaheim, California, 2013, p. 2.

[10] S Lagarde and C Rousiers. “Moving Frostbite to Physically Based Rendering 3.0”. In: *SIGGRAPH 2014 Course: Physically Based Shading in Theory and Practice*. Vancouver, 2014.

[11] Stephen Hill et al. “Physically based shading in theory and practice”. In: *ACM SIGGRAPH 2020 Courses*. 2020, p. 2.

[12] Brent Burley and Walt Disney Animation Studios. “Physically-based shading at disney”. In: *ACM SIGGRAPH*. Vol. 2012. vol. 2012. 2012, p. 18.

[13] P.S. Strauss. “A realistic lighting model for computer animators”. In: *IEEE Computer Graphics and Applications* 10.6 (1990), pp. 56–64. DOI: 10.1109/38.62696.

# Appendix A

# Self-appraisal

<This appendix should contain everything covered by the 'self-appraisal' criterion in the mark scheme. Although there is no length limit for this section, 2—4 pages will normally be sufficient. The format of this section is not prescribed, but you may like to organise your discussion into the following sections and subsections.>

## A.1 Critical self-evaluation

## A.2 Personal reflection and lessons learned

## A.3 Legal, social, ethical and professional issues

<Refer to each of these issues in turn. If one or more is not relevant to your project, you should still explain *why* you think it was not relevant.>

### A.3.1 Legal issues

### A.3.2 Social issues

### A.3.3 Ethical issues

### A.3.4 Professional issues

# Appendix B

## External Material

<This appendix should provide a brief record of materials used in the solution that are not the student's own work. Such materials might be pieces of codes made available from a research group/company or from the internet, datasets prepared by external users or any preliminary materials/drafts/notes provided by a supervisor. It should be clear what was used as ready-made components and what was developed as part of the project. This appendix should be included even if no external materials were used, in which case a statement to that effect is all that is required.>