School of Computing
FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

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 (known as a *scene* in rendering). 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

[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].

empirical model: it is based on human observations of how light interacts with materials, rather 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 Background Research

The research begins with a 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.

1.2.1 Blinn-Phong Shading

Phong Model

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*. Each one describes the contribution of a different lighting component. Splitting shading into the evaluation of a diffuse and specular term is common practice, and the physical basis for doing this is explained in section 1.2.2.

An ideal diffuse surface is one that has a Lambertian response to incident light, where the light is diffused in all directions equally [13]. Therefore, the determining factor in the appearance of such surfaces is the intensity of the incident light, which is a function of the direction of incident light and the orientation of the shaded surface. This is called *Lambert's Cosine Law* [13]. 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:

$$\mathbf{c}_{shaded_{diff}} = \mathbf{c}_{surface_{diff}} \mathbf{c}_{light_{diff}} (\mathbf{n} \cdot \mathbf{l})^{+} \tag{1.1}$$

$\mathbf{c}_{shaded_{diff}}$, $\mathbf{c}_{surface_{diff}}$, 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 [14]. 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 1.1 for an illustration of the principle vectors used in the Phong shading model (and indeed, by most shading models). 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, so the 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 a region of increased illumination, a *specular highlight*. The formula for the specular term is:

$$\mathbf{c}_{shaded_{spec}} = \mathbf{c}_{surface_{spec}} \mathbf{c}_{light_{spec}} ((\mathbf{r} \cdot \mathbf{v})^{+})^{surface_{shininess}} \tag{1.2}$$

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

$$\mathbf{r} = 2(\mathbf{n} \cdot \mathbf{l})\mathbf{n} - \mathbf{l} \tag{1.3}$$

The RGB triplets are similar to those in the diffuse equation, except these are specific to the specular response. Phong's original model had the colour of the specular highlight be the same as the overall colour of the light (not split into separate diffuse and specular components). This gave all materials an overly plastic appearance. Introducing the $\mathbf{c}_{surface_{spec}}$ and $\mathbf{c}_{light_{spec}}$ variables allows for the colour of the specular highlight to be fully configurable, mitigating this issue [14]. The *view vector*, $\mathbf{v}$, is the unit vector pointing in the direction of the viewer. The dot product measures the alignment between the view direction, and the direction of the reflected incident light. The $surface_{shininess}$ parameter determines the concentration of the reflected light rays. The higher the value, the more focused the reflected rays are, the smaller the specular highlight becomes, and the shinier the object appears. Typical values range from 1 to 100.

Finally, we have the ambient term. In the model developed so far, if a shaded point is not directly visible from a light source, then it will be black. In reality, such points are never completely unilluminated - rays from light sources will bounce around the environment, eventually lighting these obscured areas. So far we have only considered *direct illumination*; the ambient term is used to crudely approximate the lighting that is a consequence of this *indirect illumination*:

$$\mathbf{c}_{shaded_{ambi}} = \mathbf{c}_{surface_{ambi}} x \tag{1.4}$$

$\mathbf{c}_{surface_{ambi}}$ controls the colour of the ambient shading. Typically, it is just set equal to $\mathbf{c}_{surface_{diff}}$. x is a constant value defined for the whole scene, rather than per light, and controls the amount of indirect illumination that occurs. A value of $x = 0.2$ is commonly used, with anything greater often giving unrealistic results. Lighting via indirect illumination is usually quite subtle as light loses energy every time it reflects of a surface.

These three terms are summed together to give the overall Phong shading model:

$$\mathbf{c}_{shaded} = \mathbf{c}_{shaded_{ambi}} + \mathbf{c}_{shaded_{diff}} + \mathbf{c}_{shaded_{spec}} \tag{1.5}$$

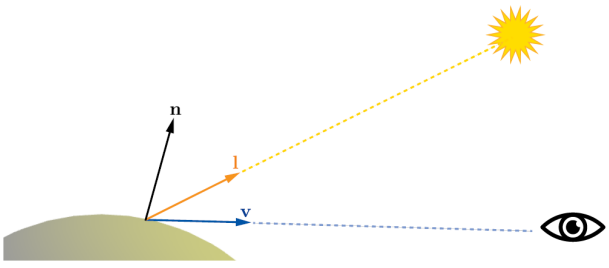


Figure 1.1: The surface normal $\mathbf{n}$, light direction $\mathbf{l}$, and view vector $\mathbf{v}$ [15]

Blinn-Phong Model

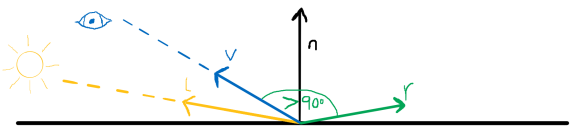
One of the issues with the Phong shading model is made apparent when viewing a rough (low *shininess* value) surface from a direction close to the incident light. The corresponding reflected light vector makes an angle with the view direction that is greater than 90°. In this instance,

the dot product $\mathbf{r} \cdot \mathbf{v}$ evaluates to a negative value, and is thus clamped to 0, leading to no specular contribution. However, for very rough surfaces, the specular highlight is so wide that even at these greater angles, there should still be a specular contribution. See Figure 1.2a for an illustration of the problem.

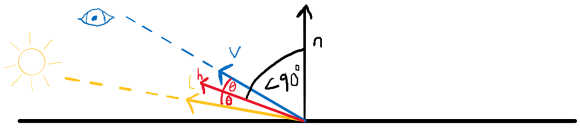
In 1977, Blinn remedied this issue by modifying the Phong shading model with a more accurate specular term [16]. He utilised the half vector, $\mathbf{h}$, dispensing with the reflected light vector, $\mathbf{r}$, and replaced the existing specular dot product with $\mathbf{h} \cdot \mathbf{n}$. $\mathbf{h}$ is a unit vector pointing in the direction that is halfway between the $\mathbf{l}$ and $\mathbf{v}$ vectors. It is calculated as:

$$\mathbf{h} = \frac{\mathbf{l} + \mathbf{v}}{\|\mathbf{l} + \mathbf{v}\|} \tag{1.6}$$

Blinn's specular term emulates the overall behaviour of the original Phong term. As the (now conceptual) reflection vector $\mathbf{r}$ aligns with $\mathbf{v}$, so does the half vector $\mathbf{h}$ align with the surface normal $\mathbf{n}$. Crucially though, $\mathbf{h} \cdot \mathbf{n}$ will only evaluate to a negative value, and subsequently be clamped to 0, if $\mathbf{l}$ or $\mathbf{v}$ is beneath the surface. Therefore, the scenario in which rough surfaces were being shaded incorrectly with no specular contribution, is resolved. See 1.2b for an illustration.



(a) The angle between $\mathbf{r}$ and $\mathbf{v}$ can exceed 90° whilst $\mathbf{l}$ and $\mathbf{v}$ are still above the surface



(b) The angle between $\mathbf{h}$ and $\mathbf{n}$ can't exceed 90° without $\mathbf{l}$ or $\mathbf{v}$ being below the surface

Figure 1.2

Blinn's modification to the Phong model is known as the Blinn-Phong shading model, and it produces more realistic images than the original. The model is formulated as:

$$\mathbf{c}_{shaded} = p_{ambi} + p_{diff}(\mathbf{n} \cdot \mathbf{l})^{+} + p_{spec}((\mathbf{h} \cdot \mathbf{n})^{+})^{surface_{shininess}} \tag{1.7}$$

where

$$\begin{aligned} p_{ambi} &= \mathbf{c}_{surface_{ambi}} x \\ p_{diff} &= \mathbf{c}_{surface_{diff}} \mathbf{c}_{light_{diff}} \\ p_{spec} &= \mathbf{c}_{surface_{spec}} \mathbf{c}_{light_{spec}} \end{aligned}$$

Expressing the model in this way draws attention to the relative proportions of each component that contributes to the overall shading. These proportions, p, are modulated via lighting parameters: *light* variables and x; and material parameters: $surface$ variables.

In scenes with multiple lights, we perform the equation 1.7 for each one and then sum up the intermediate values to get the overall colour of the fragment.

1.2.2 Physics of Light-Matter Interaction

Visible Light

The Electromagnetic spectrum is a distribution of wavelengths, of which a small section is spanned by visible light. More specifically, visible light are those electromagnetic (EM) waves with a wavelength between 400nm, violet, and 700nm, red. Typically, visible light will contain a combination of wavelengths within this range. We can express light waves as a relation between each wavelength and its associated energy, in what are known as Spectral Power Distributions (SPD). Although renderers exist that store light as SPDs, the negative performance implications that accompany this approach mean that in practice, an abstraction is often used [17]. This abstraction exploits the relative imprecision of the human visual perception system. Using a linear combination of three wavelengths, R, G and B, we can accurately represent any visible light wave and colour [18]. Renderers store this combination as an RGB triplet, with each value giving the weight of its associated wavelength.

The area of study concerned with quantifying EM waves is called *Radiometry* [19]. Several *radiometric* measurements exist, with the most commonly used one in rendering being *radiance*, L. Radiance measures the power of a single EM ray. It's defined as energy over time (power), with respect to area and direction: W/m^2sr. Power is expressed in Watts, area in metres squared, and direction in *Steradians*. Steradians are to solid angles, what radians are to normal angles. A solid angle is the 3D version of a normal 2D angle. Evaluating a shading model is equivalent to computing the radiance of a single light ray that goes from the shaded fragment to the viewer. See section 1.2.3 for more details.

Light Interaction with Matter

The way in which light interacts with matter is characterised by the matter's *index of refraction* (IOR). The IOR defines the speed that light travels through the matter, given in relative terms to the speed of light in a vacuum [20]. The IOR can also be extended to describe the absorption qualities of the matter, in what is known as the *complex index of refraction*. Some matter absorbs light waves by converting part of their energy to heat, causing an attenuation of the lights amplitude and intensity. Both forms of refractive index can vary by wavelength.

Light interacts with matter in three ways [21]. When light is travelling through a volume of matter with a uniform IOR, a *homogenous medium*, it will not deviate in its direction. However, the absorption value of the IOR may cause the light to undergo changes in intensity, and if this varies by wavelength, changes in colour also. Water is an example of a homogenous medium that absorbs light, but mostly around the red wavelengths - this gives its intrinsic blue and green colour [20].

Light behaves differently in a *heterogeneous medium*, where the IOR is not uniform. If light encounters a change in IOR that takes place over a distance smaller that a light wavelength, it will *scatter* into multiple directions. The amount of scattering varies by wavelength. The frequency of scattered light will be the same as the original light. The only exception to this is when fluorescence or phosphorescence phenomena is present, but since this is very rare in the real world, we ignore such scenarios when rendering. In the likely case that the original light is comprised of a distribution of different wavelengths and frequencies, each one will interact

independently. If interacting with a single isolated particle, the scattered light will move in all directions, but at different intensities [15].

Finally, light can interact with matter in a manner that is opposite to that of absorption. *Emission* can take place within matter, where other forms of energy are converted to light energy, and light is emitted. Light sources work in this way.

In general, the appearance of most media is as a consequence of both scattering and absorption.

Light Incident to Surfaces

When shading, we are interested in simulating what happens when light is incident upon the surface of an object. When this interaction occurs, two properties effect the outcome: the geometry of the surface, and the nature of the substances either side of the surface [15] [22].

We first consider the substance factor and assume that the surface's geometry is a perfectly flat plane. An object's surface can then be described as a flat two-dimensional boundary separating two volumes with differing IORs. In rendering it is typical that the outer volume is comprised of air, which has an IOR of 1 (or to be exact, 1.0003). The inner volume has an IOR defined by the material of the object. Any light wave that impinges on the boundary will encounter an abrupt change in the IOR, which - as explained above - will result in scattering. Because the boundary is a flat plane, the nature of this scattering is well defined: some of the scattered waves will continue moving into the surface, the *transmitted wave*, and the others will be reflected at the surface and move away, the *reflected wave*. The reflected wave propagates in a direction that is the reflection of the incident wave about the surface normal. The transmitted wave will undergo *refraction* and move at an angle of θ_t, which is defined by the relative IOR of the two volumes and the angle of the incident light, θ_i [23]. See Figure 1.3 for an illustration of this scattering behaviour. The proportion of reflected versus transmitted light follows the Fresnel equations, which are explained in section 1.2.4 [24]. The transmitted wave will interact with the interior of the object in the same way as light interacts with any medium - it will experience some degree of scattering and absorption.

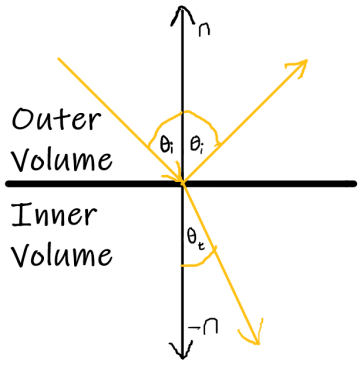


Figure 1.3: The scattering that occurs when light is incident upon a perfectly flat boundary

Now we focus on the effect that geometry plays when light is incident on a surface. The case we have just seen of a perfectly flat surface is of course not possible; all surfaces have some variation in their geometry. Any irregularities in the geometry that are smaller than a single light wavelength do not have any impact. Irregularities larger than 100 wavelengths constitute their own local

plane of flatness, with their own orientation. Irregularities that have a size within this range cause the surface-light interaction to behave differently to that described above, with phenomena such as *diffraction* also being present. However, when shading we typically stay within the realm of *geometrical optics*, which means we ignore the effects of these irregularities[2] [22]. Geometrical optics strictly deals with light as rays, not waves, and these rays always intersect locally flat planes with behaviour described above and illustrated in Figure 1.3. When shading, a single pixel or fragment will span much more than 100 wavelengths, so we need to account for the local planes of flatness that exist in this region. Irregularities at this scale are referred to as *microgeometry*. As a consequence of microgeometry, each point on the surface will reflect light in one direction, and refract it in another. Therefore, when determining the effect of microgeometry over a whole pixel, we can consider the light to be reflected and refracted in many directions. These directions are defined by the individual orientations of the points, and quantified by regarding them as a distribution of normals. The tighter the distribution, the tighter the spread of reflected and refracted directions. The roughness of the material directly controls the variance of this distribution. Section 1.2.4 discusses this in detail.

Transmitted Light Interaction

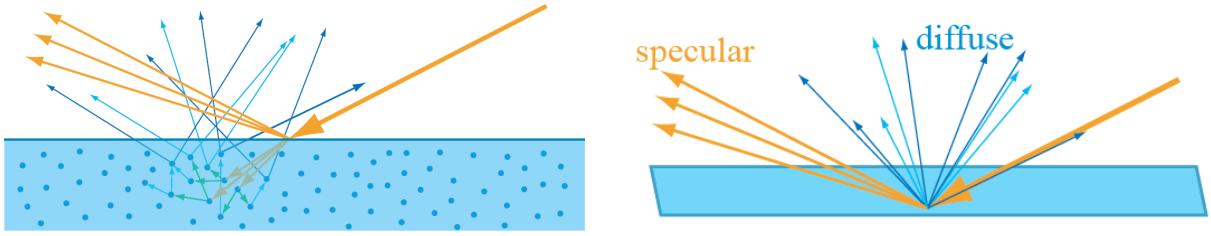
As explained previously, the light that is transmitted into the interior of an object will experience a mixture of scattering and absorption. In some materials, the light will be scattered enough that it is re-emitted at the surface of the object in a process called *subsurface scattering* [26]. See Figure 1.4a for an illustration. The distance between the subsurface scattered light and the original incident light is determined by the nature of the material, and is very important when shading. If the subsurface scattering distances are smaller than the span of one pixel, as is common for most materials, we call it *local subsurface scattering* [24]. In such cases, the subsurface scattered light is combined with the light reflected from the object surface to form a local shading model, where the outgoing light at one point is wholly dependent on the incoming light at that same point. Subsurface scattered light will encounter protracted interactions of absorption and scattering in the interior of the object, before finally being re-emitted. Therefore, it will have a distinctly different colour to the surface reflected light [27]. For this reason, shading is split into two different terms: the *specular term* captures the light reflected at the object surface, whilst the *diffuse term* models the local subsurface scattering. See Figure 1.4b for an illustration of these two shading components.

Shading is more complex when the subsurface scattering distances are larger than a single pixel. *Global subsurface scattering* is often encountered when rendering skin or wax, and although special models have been developed to shade these materials, it is beyond the scope of this report [28].

Metals and Dielectrics

The proportion of scattering and absorption that transmitted light induces varies by material. There are two main categories of materials that we encounter day-to-day: metals and *dielectrics*. Metals immediately absorb any transmitted light, meaning their appearance is solely provided by

[2] Although not widely used in real-time rendering, some shading models do exist for simulating the phenomena that occur when light interacts with geometrical irregularities of this size [25]



(a) Light is re-emitted at the surface of an object by means of subsurface scattering

(b) When shading a point we consider the specular and diffuse terms

Figure 1.4: Taken from [15]

the specular term. Most other materials are non-conductive, called dielectrics. These materials do allow for subsurface scattering, so both the specular and diffuse terms characterise their appearance.

1.2.3 Principles of Shading

The Reflectance Equation

When rendering, we model the viewer as a camera placed at point $\mathbf{c}$. The camera projects a 2D matrix of photosensitive sensors that map to pixels on the display. By combining outputs of all the sensors, we get the final rendered image. For each of the sensors, we have a ray that originates from $\mathbf{c}$, intersects the sensor, and then continues into the scene; this ray propagates in the opposite direction to the view vector, $-\mathbf{v}$. With PBS, we obtain the colour of a sensor by calculating the incoming radiance to $\mathbf{c}$ in the direction $-\mathbf{v}$. Thus, the final rendered image is given by calculating incoming radiance to $\mathbf{c}$ over the set of all $-\mathbf{v}$ vectors.

A scene is comprised of a number of objects separated by media. Typically when rendering, we just consider all media to be air. Air is a medium that exhibits very little scattering nor absorption, so its effect on radiance is minimal and can be ignored. Therefore, incoming radiance to $\mathbf{c}$ in the direction $-\mathbf{v}$, is equivalent to outgoing radiance from point $\mathbf{p}$ along $\mathbf{v}$, where $\mathbf{p}$ is the intersection point that the closest object will make with a ray that travels along $-\mathbf{v}$. This quantity is defined as $L_o(\mathbf{p}, \mathbf{v})$ and is calculated by means of the *reflectance equation*:

$$L_o(\mathbf{p}, \mathbf{v}) = \int_{\mathbf{l} \in \Omega} f(\mathbf{l}, \mathbf{v}) L_i(\mathbf{p}, \mathbf{l})(\mathbf{n} \cdot \mathbf{l})\, d\mathbf{l} \tag{1.8}$$

The $\mathbf{l} \in \Omega$ in the integral subscript says that the integral should be performed over all directions $\mathbf{l}$ that exist in the unit hemisphere Ω, which is centred on $\mathbf{p}$ and orientated along the surface normal $\mathbf{n}$. In this way, the incoming radiance from all possible light sources is considered. The product within the integral gives the outgoing radiance from $\mathbf{p}$ along $\mathbf{v}$ for a singular incident light direction $\mathbf{l}$. The effect of the integration is to sum over all the individual components of outgoing radiance, to obtain the total value. This formulation is consistent with the shading scenario that was introduced in the "Transmitted Light Interaction" section of 1.2.2, except it is expressed from a slightly different perspective: instead of keeping the incoming light direction constant and considering many outgoing light directions, we are now keeping the outgoing light direction constant, and considering all possible incoming light directions. Note that this is indeed a local shading model, with all outgoing light at $\mathbf{p}$ being wholly dependent on incoming light at

p. A physically based shading model computes the reflectance equation using physically based implementations of f and L_i. We now discuss each of the factors of the integral's product in turn.

The BRDF

The *bidirectional reflectance distribution function* (BRDF), $f(\mathbf{l}, \mathbf{v})$, was first introduced by Nicodemus et al in 1977 [29]. For a given $\mathbf{l}$ and $\mathbf{v}$, the BRDF gives the ratio of incident light in direction $\mathbf{l}$, which after striking the surface, is reflected in direction $\mathbf{v}$. The function is a distribution over all possible values of $\mathbf{l}$ and $\mathbf{v}$ that lie in the unit hemisphere introduced above, and thereby it completely describes a surface's local reflectance response to incident light. Local reflectance encompasses both surface reflection and local subsurface scattering. Variations of the BRDF exist which seek to capture other light-matter interactions, such as the BSSRDF which accounts for the influence of global subsurface scattering [30].

As explained previously, scattering and absorption - the two phenomena that underpin local reflectance - vary by wavelength. Therefore, the BRDF also needs to vary by wavelength, so the ratios it returns are given as RGB triplets.

For a BRDF to be considered physically plausible, two constraints must hold. The first is called the Helmholtz reciprocity and states that $f(\mathbf{l}, \mathbf{v}) = f(\mathbf{v}, \mathbf{l})$ [31]. BRDFs used in rendering don't often comply with this equality, but still look physically correct [15]. The second constraint is imposed by conservation of energy: the outgoing light energy cannot exceed the incoming light energy. In real-time rendering, exact adherence to this principle is not required, but respecting it in the approximate sense is very important - otherwise, objects will be rendered as overly bright and unrealistic [15][10].

Constructing a BRDF can be accomplished in two ways: via optical measurements of real materials, or by leveraging mathematical formulas. Ward discusses the use of a *Gonioreflectometer* to measure BRDFs of real world materials [32]. This process is time consuming and thus it's use in rendering is impractical. Instead, we construct BRDFs using parametrised mathematical formulas. The parameters involved are properties of the object's material. If the material properties are specified over every point on the object - which is commonly practiced in graphics by utilising textures - then this is equivalent to defining a BRDF for every point on the object's surface. As mentioned, the local reflectance response of a surface is split into specular and diffuse terms when shading. Naturally, this extends to the BRDF itself, and it is expressed as the sum of a specular BRDF and a diffuse BRDF.

These parametrised mathematical formulas can be categorised as either empirical or physically based. Although not represented explicitly, the Blinn-Phong shading model detailed in section 1.2.1 defines within it an empirical BRDF. Physically based BRDFs lie at the very heart of PBS, and sections 1.2.4, 1.2.5 and 1.2.6 are concerned with investigating them.

Figure 1.5 gives an example of a BRDF. Visualising a BRDF is difficult as it is a function of four parameters (two angles per vector, one for elevation and another for rotation), so the Figure adopts the common approach of keeping the incident light direction constant.

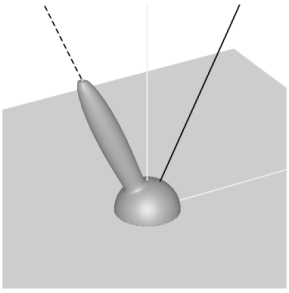


Figure 1.5: The Cook-Torrance-Sparrow BRDF [33]

Incoming Radiance

The incoming radiance term, $L_i(\mathbf{p}, \mathbf{l})$, represents the light that originates in direction $\mathbf{l}$ and strikes the surface at point $\mathbf{p}$. Illumination and light sources are discussed in depth in section 1.2.7.

Dot Product Term

The dot product term, $(\mathbf{n} \cdot \mathbf{l})$, is the same as that discussed in section 1.2.1. It fulfils the same purpose, going from 0 to 1 as the light direction approaches the surface normal. When multiplied by the incoming radiance term, it appropriately scales the amount of illumination. It's also common for this dot product to be clamped to 0, so that contributions from light sources below the surface are ignored.

1.2.4 BRDF Building Blocks

Most physically based BRDFs are built upon the same set of standard mathematical functions and theory. The individual BRDFs then differ from one another in how these functions are implemented and compiled together. We first detail the standard mathematical building blocks, and then specific specular and diffuse BRDFs are discussed in sections 1.2.5 and 1.2.6 respectively.

Fresnel Reflectance

In section 1.2.2, the behaviour of light that impinges upon a surface was said to be dependent on two factors: the substances either side of the surface, and the geometry of the surface. As alluded to, the Fresnel equations describe how light behaves due to the substances. We again assume a perfectly flat boundary separating an outer and inner substance, which have IORs of n_1 and n_2 respectively. The Fresnel reflectance, F, gives the proportion of incoming light that is reflected at the boundary. Due to the conservation of energy, the amount of transmitted light can then also be easily obtained. F varies by wavelength so is expressed as an RGB triplet. Given values of n_1 and n_2, F is then a function of incident light angle, $F(\theta_i)$. As mentioned, in rendering it's typical for the outer substance to be air with an IOR of 1, and thus we will be in a scenario of *external reflection*, where $n_1 < n_2$.

In the case of external reflectance, $F(\theta_i)$ always follows the same pattern [15]. When $\theta_i = 0°$, F will be equal to the intrinsic specular colour of the inner substance, F_0. Between 0° and 90°, F will then increase non-linearly towards white; slowly for most of the interval and then rising rapidly when close to 90°. The tendency for surfaces to exhibit increased reflection at glancing

angles is called the *Fresnel effect*.

Using the Fresnel equations themselves is not possible in real-time rendering because n_1 and n_2 must be known for all wavelengths of visible light, and this data is not available [34]. Instead, approximations that rely on the pattern outlined above are utilised. These are functions that depend in part on F_0, which is known for many materials.

F_0 is defined by the object's material parameters. Dielectrics all have very low F_0 values, grouping around the $(0.04, 0.04, 0.04)$ mark - a fact that is exploited by many physically based shading models. This means dielectrics only reflect a noticeable amount of light at very glancing angles. Metals on the other hand have much higher F_0 values, with each RGB channel typically exceeding 0.5. Akenine-Möller et al give a comprehensive overview of F_0 values for different materials [15].

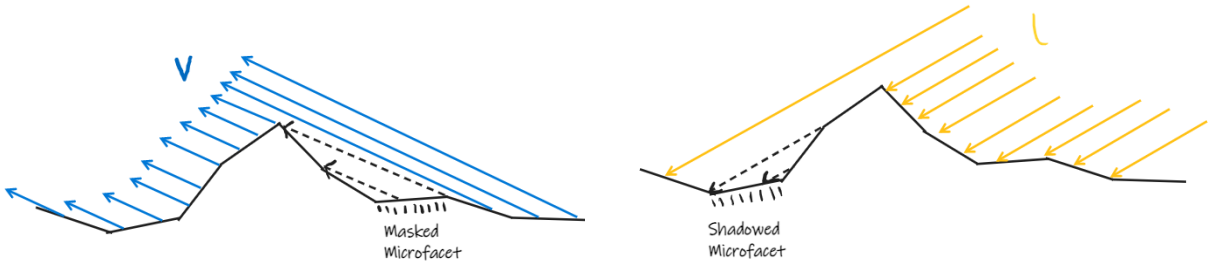
Microfacet Theory

We now revisit the role that the surface geometry plays in local reflectance. Recall that when rendering, a single pixel will lie over many pieces of microgeometry. The reflectance at the shading point is then determined by the aggregation of the individual interactions that light will have with each piece of microgeometry. We reason about these interactions by utilising *microfacet theory*. The theory states that microgeometry be modelled as a collection of perfectly flat planes called *microfacets* [16]. Such a model is effective at representing many real world materials [35]. The microfacets' impact on surface reflectance is described by a *Normal Distribution Function* (NDF), a *masking-shadowing function*, and a *micro-BRDF*.

The NDF, $D(\mathbf{m})$, statistically describes the orientations of the microfacets over the macrosurface (the shaded point when viewed at the pixel scale) [36]. The more microfacets that have a normal of $\mathbf{m}$, the higher the value of $D(\mathbf{m})$. Typically, surfaces posses a distribution of microfacet normals that is concentrated around the normal of the macrosurface $\mathbf{n}$.

Whilst the NDF describes the orientations of the microfacets, it doesn't describe their arrangement, which also plays a significant role in how light reflects off the macrosurface. Some microfacets will not be visible from the view vector $\mathbf{v}$ because they are occluded by other microfacets. This is known as *masking*. Only those microfacets that are not masked by others will be visible to the viewer and thus contribute to the shading. Furthermore, some microfacets will be occluded so they aren't visible to the incoming light direction $\mathbf{l}$. This is known as *shadowing*. Microfacet theory only models the first interaction between light and microfacet, thus, only those microfacets that are not shadowed by others are assumed to contribute to the local reflectance [36]. In reality, incident light will bounce multiple times within a surfaces microgeometry, meaning even microsurfaces that aren't visible directly from the light will exhibit some reflectance. This is known as *interreflection*. Surfaces can look overly dark and conservation of energy will be violated when this phenomena is not modelled [37]. Figure 1.6 illustrates masking and shadowing. The appropriately named *masking-shadowing function*, $G(\mathbf{l}, \mathbf{v}, \mathbf{m})$, accounts for the effects of masking and shadowing. It provides the fraction of microfacets with normal $\mathbf{m}$ that are visible from directions $\mathbf{v}$ and $\mathbf{l}$ [16].

The micro-BRDF, $f_\mu(\mathbf{l}, \mathbf{v}, \mathbf{m})$, describes how light is reflected of an individual microfacet. There are two common choices for micro-BRDF. It's typical in a specular (macro) BRDF, that all the microfacets are modelled as perfect mirrors [38]. This means that the micro-BRDF will



(a) Some microfacets are masked by others (b) Some microfacets are shadowed by others

Figure 1.6

reflect an incident light ray $\mathbf{l}$ in only one direction: $\mathbf{l}$ reflected about the microfacet normal $\mathbf{m}$. Conversely, some diffuse (macro) BRDFs model all the microfacets as perfect diffuse surfaces [39]. As explained in section 1.2.1, in compliance with Lambertian reflection, these surfaces diffuse incoming light equally in all directions and have no specular component; this translates to a micro-BRDF of constant value.

The microfacet theory developed in the above paragraphs can be used to create an expression for the overall (macro) BRDF [15]:

$$f(\mathbf{l},\mathbf{v}) = \int_{\mathbf{m}\in\Omega} f_\mu(\mathbf{l},\mathbf{v},\mathbf{m})G(\mathbf{l},\mathbf{v},\mathbf{m})D(\mathbf{m})\frac{(\mathbf{m}\cdot\mathbf{l})^+}{|\mathbf{n}\cdot\mathbf{l}|}\frac{(\mathbf{m}\cdot\mathbf{v})^+}{|\mathbf{n}\cdot\mathbf{v}|}\,d\mathbf{m} \tag{1.9}$$

Again, Ω is the unit hemisphere centred on $\mathbf{p}$ and orientated along the surface normal $\mathbf{n}$. Heitz gives a thorough explanation as to how this equation is derived [36]. Although not used directly in rendering, given a specific choice of micro-BDRF, equation 1.9 can be simplified to obtain a BRDF suitable for use in a shading model [15]. An example of such a simplification can be seen in section 1.2.5.

1.2.5 Specular BRDFs

Physically based specular BRDFs are built upon microfacet theory, and therefore utilise equation 1.9. This equation integrates over all possible microfacet normals $\mathbf{m}$ that point above the macrosurface. However, because each microfacet is modelled as a perfect mirror, $f_\mu(\mathbf{l},\mathbf{v},\mathbf{m})$ will only be non-zero when $\mathbf{m}$ is of a value such that $\mathbf{l}$ is reflected exactly in direction $\mathbf{v}$. The only instance of $\mathbf{m}$ that satisfies this is when $\mathbf{m} = \mathbf{h}$, the half vector introduced in section 1.2.1. Therefore, we can remove the integration in equation 1.9, simplifying it to the case when $\mathbf{m} = \mathbf{h}$. After some further derivation we arrive at the following equation for specular BRDFs [36]:

$$f_{spec}(\mathbf{l},\mathbf{v}) = \frac{F(\mathbf{h},\mathbf{l})G(\mathbf{l},\mathbf{v},\mathbf{h})D(\mathbf{h})}{4|\mathbf{n}\cdot\mathbf{v}||\mathbf{n}\cdot\mathbf{l}|} \tag{1.10}$$

All the terms in equation 1.10 are familiar, although the Fresnel reflectance, $F(\mathbf{h},\mathbf{l})$, is parametrised slightly different to the one introduced in section 1.2.4 - this is explained in the section below. Given this equation, the construction of a specific specular BRDF then boils down to choosing how these functions are implemented.

Fresnel Reflectance Implementations

As explained in section 1.2.4, we don't make use of the Fresnel equations directly, but rather utilise approximations. An early approximation was given by Cook and Torrance [38]. Schlick then improved upon this with his own function that has since seen widespread use [34]:

$$F(\mathbf{n}, \mathbf{l}) = F_0 + (1 - F_0)(1 - (\mathbf{n} \cdot \mathbf{l})^+)^5 \tag{1.11}$$

This function approximates the Fresnel reflection with less than a 1% error [34]. Figure 1.7 shows the difference between the actual Fresnel reflectance and this approximation for several materials. The equation interpolates between F_0 and white in a non-linear manner that emulates the description given in section 1.2.4. When the light is directly incident to the surface, $\mathbf{n} \cdot \mathbf{l} = 0$ so the Fresnel reflectance will be F_0. As the angle between the microfacets' normal and the incident light increases, $\mathbf{n} \cdot \mathbf{l}$ decreases and so the overall value of F tends towards white.

Lagarde created a more optimised version of Schlicks approximation by using *Spherical Gaussians* [40]:

$$F(\mathbf{n}, \mathbf{l}) = F_0 + (1 - F_0)2^{(-5.55473(\mathbf{n}\cdot\mathbf{l}) - 6.98316)(\mathbf{n}\cdot\mathbf{l})} \tag{1.12}$$

Unreal engine makes use of this formulation for their Fresnel term [9]. Frostbite on the other hand, have adapted the Schlick approximation into a more flexible form [10]:

$$F(\mathbf{n}, \mathbf{l}) = F_0 + (F_{90} - F_0)(1 - (\mathbf{n} \cdot \mathbf{l})^+)^{\frac{1}{p}} \tag{1.13}$$

F_{90} can be used to define the colour that the reflectance tends towards at glancing angles - it doesn't have to be white. Although this is physically incorrect, it does give artists more control. The p variable is used to modify the steepness of the transition to F_{90}.

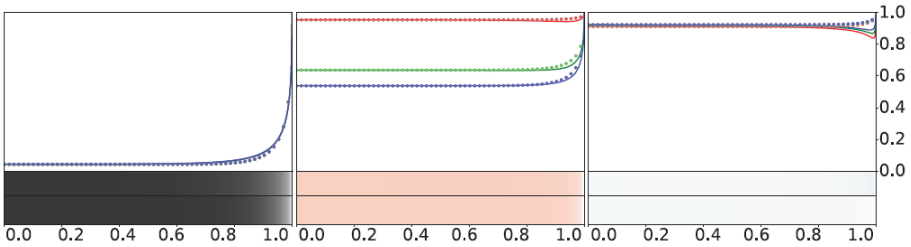


Figure 1.7: From left to right the materials are, glass, copper and aluminium. Along the x-axis is plotted $\sin(\theta_i)$, and along the y is the intensities of each RGB channel. The solid lines are the actual Fresnel reflectance, and the dotted lines are the values given by the Schlick approximation. Similarly, the upper colour strip is the actual Fresnel reflectance, and the lower is that yielded from the Schlick approximation. Taken from [15].

NDF Implementations

In this section we will only concern ourselves with *isotropic* NDFs, which are those that are invariant when rotated about n. *Anisotropic* NDFs do exist and are crucial when rendering surfaces like brushed metal. A prominent isotropic NDF is the Beckmann distribution [41]. It is widely used in the field of optics and is the NDF of choice in the Cook-Torrance BRDF [42] [38]. Walter et al give an equation for the Beckmann distribution [42]. After using some identities

to manipulate it, the trigonometric functions can be replaced with dot products to yield the following:

$$D(\mathbf{m}) = \frac{\mathcal{X}^+(\mathbf{m}\cdot\mathbf{n})}{\pi\alpha_b^2(\mathbf{m}\cdot\mathbf{n})^4}\exp\left(\frac{(\mathbf{m}\cdot\mathbf{n})^2-1}{\alpha_b^2(\mathbf{m}\cdot\mathbf{n})^2}\right) \tag{1.14}$$

α_b controls the roughness of the surface and thus the spread of the specular response [38].

The other NDF that has seen extensive use in the graphics community is the GGX distribution. Originally formulated by Trowbridge and Reitz in 1975, the GGX distribution only started to see use after it was reformulated by Walter et al in 2007 [43][42]. It has the following form:

$$D(\mathbf{m}) = \frac{\alpha_g^2\mathcal{X}^+(\mathbf{m}\cdot\mathbf{n})}{\pi(1+(\mathbf{m}\cdot\mathbf{n})^2(\alpha_g^2-1))^2} \tag{1.15}$$

α_g is similarly used to control the roughness of the surface, although it scales in a different manner. Burley proposes mapping $\alpha_g = r^2$ where r provides a perceptually linear change in roughness as it moves between 0 and 1 [12]. Exposing parameters to artists in this linear manner is a common trope in graphics. Karis outlines it as one of the key goals they pursued when migrating the Unreal Engine over to a PBR pipeline [9].

Masking-shadowing Function Implementations

Implementations of the masking-shadowing function depend on the NDF used. There are two main options to choose from. The first is a function created by Torrance and Sparrow, which considers adjacent microfacets to form a "V-groove cavity" [44]. The second option is the Smith masking-shadowing function [45]. Heitz thoroughly analyses the two models and concludes that whilst they are both physically based, the Smith function is more accurate to real materials, with the Torrence-Sparrow function producing too small a specular component for rough materials viewed at glancing angles [36]. The most accurate version of the Smith function is the *Smith height-correlated masking-shadowing function.* This takes into account the height of surface points relative to the rest of the surface. This is significant because points that are lower in the surface are more likely to experience masking and shadowing [15].

The Smith height-correlated masking-shadowing function that is intended for use with the GGX NDF is [15]:

$$G(\mathbf{l},\mathbf{v},\mathbf{m}) = \frac{\mathcal{X}^+(\mathbf{m}\cdot\mathbf{v})\mathcal{X}^+(\mathbf{m}\cdot\mathbf{l})}{1+\Lambda(a_\mathbf{v})+\Lambda(a_\mathbf{l})} \tag{1.16}$$

where

$$\Lambda(a) = \frac{-1+\sqrt{1+\frac{1}{a^2}}}{2} \tag{1.17}$$

and

$$a_\mathbf{v} = \frac{\mathbf{n}\cdot\mathbf{v}}{\alpha_g\sqrt{1-(\mathbf{n}\cdot\mathbf{v})^2}} \quad a_\mathbf{l} = \frac{\mathbf{n}\cdot\mathbf{l}}{\alpha_g\sqrt{1-(\mathbf{n}\cdot\mathbf{l})^2}}$$

1.2.6 Diffuse BRDFs

We now focus on diffuse BRDFs, which model the local subsurface scattering component of local reflectance. Similar to the intrinsic specular colour of a material, F_0, we also define an intrinsic diffuse colour called the *subsurface albedo*. Denoted by ρ_{ss}, this is the proportion of transmitted light that is re-emitted at the surface via subsurface scattering. Since this amount is a distribution of wavelengths, ρ_{ss} is represented as an RGB triplet. The more ρ_{ss} tends towards $(0, 0, 0)$, the more absorption takes place within the interior of the associated material.

Diffuse BRDFs can be placed into two categories: those that are built upon microfacet theory and account for surface roughness; and those that don't utilise microfacet theory and just assume the surface is flat. As explained by Akenine-Möller et al, the type of model to use is determined by the differences between the size of the microgeometry, and the subsurface scattering distances [15]. If the microgeometry is larger than the scattering distances, then a rough-surface diffuse BRDF should be used. If the microgeometry is smaller than the scattering distances, then a flat-surface diffuse BRDF can be used.

Flat-surface Diffuse BRDFs

The most common diffuse BRDF is based on Lambertian reflection and has the following formulation:

$$f_{diff}(\mathbf{l}, \mathbf{v}) = (1 - F(\mathbf{h}, \mathbf{l}))\frac{\rho_{ss}}{\pi} \tag{1.18}$$

The π is obtained by setting the *directional-hemispherical reflectance function* to a constant value and integrating it over all values of $\mathbf{v}$ [15]. The inclusion of the Fresnel reflectance ensures that only light that was transmitted into the interior of the object is available to the diffuse term [46]. This helps to comply with the law of conservation of energy.

Equation 1.18 has a constant value for all view directions. The physical basis for such a formulation is rooted in the observation that, prior to re-emission, subsurface scattered light will undergo many scattering events in the interior of the object (as explained in section 1.2.2). This has the effect of randomising the directions of outgoing light, which can be intuitively modelled as a constant value over all $\mathbf{v}$ [47]. However, real materials will show some directional bias, whether that's due to refraction or the need to respect Helmholtz reciprocity (which equation 1.18 violates) [15]. Shirley et al present a more accurate diffuse BRDF that is partially coupled with the specular component [24].

Rough-surface Diffuse BRDFs

The most notable microfacet-based, rough-surface diffuse BRDF is that created by Oren and Nayar [39]. It is comprised of: a gaussian NDF; a Torrence-Sparrow masking-shadowing function; and a constant micro-BRDF that models each microfacet as a perfectly diffuse surface. The Oren-Nayar BRDF increases in intensity as $\mathbf{v}$ approaches $\mathbf{l}$, a property that is coherent with the reflectance characteristics of rough surfaces like clay [39]. As a result, their BRDF produces much more realistic results for these rough surfaces when compared to the Lambertian BRDF. Alternative diffuse BRDFs that are based on microfacet theory are given by Gotanda, and another by Hammon [48][47]. These both make use of the more modern GGX distribution and Smith

height-correlated masking-shadowing function.

Burley presents a BRDF that accounts for roughness, but does not use microfacet theory [12]. Instead, it is empirically constructed by observing how light responds to different materials in the *MERL* database. The Frostbite Engine utilises a slightly modified version of Burley's BRDF, where it has been tweaked to respect energy conservation [10].

1.2.7 Illumination

Having thoroughly examined the BRDF, we now study the other function in the reflectance equation, the incoming radiance, $L_i(\mathbf{p}, \mathbf{l})$.

In section 1.2.1, a distinction was made between direct and indirect illumination. Direct illumination is light that arrives directly from a light source. Indirect illumination is light that only strikes point $\mathbf{p}$ after first undergoing collisions with other objects and media within the scene. When considering the unit hemisphere of incoming light, direct illumination contributes high levels of radiance, over small solid angles. Indirect illumination then spans the rest of the hemisphere, with small to moderate levels of radiance. This difference means that when integrating the reflectance equation over all possible incident light directions, it is common to deal with direct and indirect light sources separately.

Indirect Illumination

Indirect illumination impinging upon $\mathbf{p}$ can be computed using *Global Illumination* (GI) or *Local Illumination* techniques. GI algorithms determine $L_i(\mathbf{p}, \mathbf{l})$ by backtracking, and explicitly simulating the previous light-matter interactions that have led to that illumination. Such techniques are dependent on having a global view of the objects in the scene at each shading point. The ray tracing algorithm briefly discussed in the introduction is a GI technique. The tremendous amount of realism and detail that GI techniques yield, unfortunately demand an equally tremendous amount of compute time. The frame times given for ray tracing was indicative of this. As a result, GI algorithms cannot be computed per frame in real-time rendering. However, for static environments, GI can be used to precompute some aspects of lighting, which are then retrieved at run time. This process is known as *baking* [49].

Local Illumination techniques do not require a global view of scene objects; they only need to be aware of the current object that is being shaded. We have already examined a crude Local Illumination technique for modelling indirect illumination: the ambient term presented in section 1.2.1. More advanced techniques are Image Based Lighting and Irradiance Mapping [21][50].

Direct illumination

Computing the direct illumination incident to $\mathbf{p}$ involves integrating the reflectance equation over specific, known values of $\mathbf{l}$. There are three types of light sources: area, punctual and directional. Area lights, as the name suggests, have a size, as well as a location. When projected onto the unit hemisphere, this size translates to a solid angle. Therefore, to work out the contribution of incoming radiance from that area light, an integration is done over values of $\mathbf{l}$ that lie within its projected solid angle.

Punctual lights are similar to area lights, except they have an infinitesimally small size. This

simplifies the evaluation of the reflectance equation because only one direction $\mathbf{l}$ needs to be integrated for each punctual light source. However, punctual lights are an abstraction of the real world - all lights occupy some area - and therefore the simplified computation they provide comes at a cost. Point lights will produce very small specular highlights on shiny materials, which looks unrealistic. To combat this, artists will often increase the roughness of the material to spread the highlight out, but in doing so they couple the specific lighting environment and material properties together [10] [9]. In the introduction we discussed why such a practise can be problematic.

Directional lights are then a further simplification, where they are assumed to be positioned so far away, that their direction is constant over all objects in the scene. Thus, they are not associated with a particular location.

Point Lights

The most common form of punctual light is the point light. A point light emits constant radiance over all directions. The colour of the point light is denoted as $\mathbf{c}_{PL}$, which is an unbounded RGB triplet [21]. $\mathbf{c}_{PL}$ is then defined as the reflected radiance from a perfectly diffuse white surface that directly faces the light. Given this definition, Hoffman derives a simplified version of the reflectance equation for a single point light [21]. Extending this for multiple point lights, the reflectance equation becomes:

$$L_o(\mathbf{p}, \mathbf{v}) = \pi \sum_{i=0}^{N-1} f(\mathbf{l}_{PL_i}, \mathbf{v}) \mathbf{c}_{PL_i} (\mathbf{n} \cdot \mathbf{l}_{PL_i})^+ \tag{1.19}$$

where N is the number of point lights.

Light intensity is attenuated as the distance between the light source and shaded point $\mathbf{p}$ increases. For point lights, this attenuation is equal to the inverse of the squared distance. Therefore, $\mathbf{c}_{PL_i}$ will be proportional to $1/(\mathbf{p} - \mathbf{p}_{PL_i})^2$, where $\mathbf{p}_{PL_i}$ is the position of point light i.

The dot product term has been clamped to 0, so that the contribution of point lights that are located below the surface is discarded.

1.2.8 HDR

- 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 What physical phenomena am I going to use for comparison?

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.

This is the key point: in the real world the relative proportions of matte and specular appearance change with viewing angle. - Practitioners guide to reflectance models

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] Johann Heinrich Lambert and David L DiLaura. *Photometry or On The Measure and Gradations of light, Color, and Shade*. A translation of Lambert's Photometria, which was originally published in 1760. Illuminating Engineering Society of North America, 2001. ISBN: 978-0-87995-179-5.

[14] 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.

[15] Tomas Akenine-Möller et al. *Real-Time Rendering 4th Edition*. Boca Raton, FL, USA: A K Peters/CRC Press, 2018, p. 1200. ISBN: 978-1-13862-700-0.

[16] James F. Blinn. "Models of Light Reflection for Computer Synthesized Pictures". In: *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '77. San Jose, California: Association for Computing Machinery, 1977, 192–198. ISBN: 9781450373555. DOI: 10.1145/563858.563893. URL: https://doi.org/10.1145/563858.563893.

[17] David Murray, Alban Fichet, and Romain Pacanowski. "Efficient Spectral Rendering on the GPU for Predictive Rendering". In: *Ray Tracing Gems II*. Springer, 2021, pp. 673 –698. DOI: 10.1007/978-1-4842-7185-8_42. URL: https://hal.inria.fr/hal-03331619.

[18] M.C. Stone. "Representing colors as three numbers [color graphics]". In: *IEEE Computer Graphics and Applications* 25.4 (2005), pp. 78–85. DOI: 10.1109/MCG.2005.84.

[19] Raju Datla and Albert Parr. "1. Introduction to Optical Radiometry". In: *Experimental Methods in the Physical Sciences* 41 (Dec. 2005), pp. 1–34. DOI: 10.1016/S1079-4042(05)41001-2.

[20] Victor frederick Weisskopf. "How Light Interacts with Matter". In: *Scientific American* 219 (1968), pp. 60–71.

[21] Naty Hoffman. "Background: physics and math of shading". In: *Physically Based Shading in Theory and Practice* 24.3 (2013), pp. 211–223.

[22] Bram van Ginneken, Marigo Stavridi, and Jan J. Koenderink. "Diffuse and Specular Reflectance from Rough Surfaces". In: *Appl. Opt.* 37.1 (1998), pp. 130–139. DOI: 10.1364/AO.37.000130. URL: http://opg.optica.org/ao/abstract.cfm?URI=ao-37-1-130.

[23] Wikipedia contributors. *Snell's law — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-March-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Snell%27s_law&oldid=1074492276.

[24] Peter Shirley et al. "A Practitioners' Assessment of Light Reflection Models". In: Sept. 1997. ISBN: 0-8186-8028-8. DOI: 10.1109/PCCGA.1997.626170.

[25] Nicolas Holzschuch and Romain Pacanowski. "A Two-Scale Microfacet Reflectance Model Combining Reflection and Diffraction". In: *ACM Transactions on Graphics* 36.4 (July 2017). Article 66, p. 12. DOI: 10.1145/3072959.3073621. URL: https://hal.inria.fr/hal-01515948.

[26] Gustav Kortüm. *Reflectance spectroscopy: principles, methods, applications*. Springer Science & Business Media, 2012, p. 3.

[27] Pat Hanrahan and Wolfgang Krueger. "Reflection from Layered Surfaces Due to Subsurface Scattering". In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '93. Anaheim, CA: Association for Computing Machinery, 1993, 165–174. ISBN: 0897916018. DOI: 10.1145/166117.166139. URL: https://doi.org/10.1145/166117.166139.

[28] Jorge Jimenez et al. "Separable Subsurface Scattering". In: *Computer Graphics Forum* (2015).

[29] Fred E Nicodemus et al. "Geometrical considerations and nomenclature for reflectance". In: *National Bureau of Standards (US) monograph* 160 (1977).

[30] Henrik Wann Jensen et al. "A Practical Model for Subsurface Light Transport". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, 511–518. ISBN: 158113374X. DOI: 10.1145/383259.383319. URL: https://doi.org/10.1145/383259.383319.

[31] Bruce Hapke. *Theory of Reflectance and Emittance Spectroscopy*. 2nd ed. Cambridge University Press, 2012, pp. 264–265. DOI: 10.1017/CBO9781139025683.

[32] Gregory J. Ward. "Measuring and Modeling Anisotropic Reflection". In: *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '92. New York, NY, USA: Association for Computing Machinery, 1992, 265–272. ISBN: 0897914791. DOI: 10.1145/133994.134078. URL: https://doi.org/10.1145/133994.134078.

[33] Szymon M. Rusinkiewicz. "A New Change of Variables for Efficient BRDF Representation". In: *Rendering Techniques*. 1998.

[34] Christophe Schlick. "An Inexpensive BRDF Model for Physically-based Rendering". In: *Computer Graphics Forum* 13.3 (1994), pp. 233–246. DOI: https://doi.org/10.1111/1467-8659.1330233. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.1330233. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.1330233.

[35] Addy Ngan, Frédo Durand, and Wojciech Matusik. "Experimental Analysis of BRDF Models". In: *Eurographics Symposium on Rendering (2005)*. Ed. by Kavita Bala and Philip Dutre. The Eurographics Association, 2005. ISBN: 3-905673-23-1. DOI: 10.2312/EGWR/EGSR05/117-126.

[36] Eric Heitz. "Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs". In: *Journal of Computer Graphics Techniques (JCGT)* 3.2 (2014), pp. 48–107. ISSN: 2331-7418. URL: http://jcgt.org/published/0003/02/03/.

[37] Eric Heitz et al. "Multiple-Scattering Microfacet BSDFs with the Smith Model". In: *ACM Trans. Graph.* 35.4 (2016). ISSN: 0730-0301. DOI: 10.1145/2897824.2925943. URL: https://doi.org/10.1145/2897824.2925943.

[38] R. L. Cook and K. E. Torrance. "A Reflectance Model for Computer Graphics". In: *ACM Trans. Graph.* 1.1 (1982), 7–24. ISSN: 0730-0301. DOI: 10.1145/357290.357293. URL: https://doi.org/10.1145/357290.357293.

[39] Michael Oren and Shree K. Nayar. "Generalization of Lambert's Reflectance Model". In: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '94. New York, NY, USA: Association for Computing Machinery, 1994, 239–246. ISBN: 0897916670. DOI: 10.1145/192161.192213. URL: https://doi.org/10.1145/192161.192213.

[40] S Lagarde. *Spherical Gaussian approximation for Blinn-Phong, Phong and Fresnel*. June 3, 2012. URL: https://seblagarde.wordpress.com/2012/06/03/spherical-gaussien-approximation-for-blinn-phong-phong-and-fresnel (visited on 04/06/2022).

[41] P. Beckmann, J.M. Coulson, and A. Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. A Pergamon Press book. Pergamon Press; [distributed in the Western Hemisphere by Macmillan, New York], 1963. ISBN: 9780080100074. URL: https://books.google.co.uk/books?id=QBEIAQAAIAAJ.

[42] Bruce Walter et al. "Microfacet Models for Refraction through Rough Surfaces". In: *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. EGSR'07. Grenoble, France: Eurographics Association, 2007, 195–206. ISBN: 9783905673524.

[43] T. S. Trowbridge and K. P. Reitz. "Average irregularity representation of a rough surface for ray reflection". In: *J. Opt. Soc. Am.* 65.5 (1975), pp. 531–536. DOI: 10.1364/JOSA.65.000531. URL: http://opg.optica.org/abstract.cfm?URI=josa-65-5-531.

[44] K. E. Torrance and E. M. Sparrow. "Theory for Off-Specular Reflection From Roughened Surfaces*". In: *J. Opt. Soc. Am.* 57.9 (1967), pp. 1105–1114. DOI: 10.1364/JOSA.57.001105. URL: http://opg.optica.org/abstract.cfm?URI=josa-57-9-1105.

[45] B. Smith. "Geometrical shadowing of a random rough surface". In: *IEEE Transactions on Antennas and Propagation* 15.5 (1967), pp. 668–671. DOI: 10.1109/TAP.1967.1138991.

[46] Peter S. Shirley. "Physically Based Lighting Calculations for Computer Graphics". UMI Order NO. GAX91-24487. PhD thesis. USA, 1991, pp. 47–48.

[47] Earl Hammon Jr. "PBR Diffuse Lighting for GGX+Smith Microsurfaces". Game Developers Conference (GDC). 2017. URL: https://www.gdcvault.com/play/1024478/PBR-Diffuse-Lighting-for-GGX (visited on 04/08/2022).

[48] Yoshiharu Gotanda. "Designing Reflectance Models for New Consoles". In: 2014.

[49] Dario Seyb et al. "The Design and Evolution of the UberBake Light Baking System". In: *ACM Trans. Graph.* 39.4 (2020). ISSN: 0730-0301. DOI: 10.1145/3386569.3392394. URL: https://doi.org/10.1145/3386569.3392394.

[50] Ravi Ramamoorthi and Pat Hanrahan. "An Efficient Representation for Irradiance Environment Maps". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: Association for Computing Machinery, 2001, 497–500. ISBN: 158113374X. DOI: 10.1145/383259.383317. URL: https://doi.org/10.1145/383259.383317.

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.>

Appendix C

Mathematical Notation

$\mathbf{n}$	Normal vector
$\mathbf{l}$	Light direction
$\mathbf{v}$	View vector
$\mathbf{h}$	Half vector
$\mathbf{a} \cdot \mathbf{b}$	The dot product of vectors $\mathbf{a}$ and $\mathbf{b}$
$\|\mathbf{a}\|$	The norm of vector $\mathbf{a}$
$\|x\|$	The absolute value of x
x^+	Clamp x to 0 if $x < 0$
$\mathcal{X}^+(x)$	Returns 1 if $x > 0$, else returns 0