Introduction to Artist Workflow of Physically Based Rendering in Real Time Graphics

Lucas Lundin
2016

Bachelor of Fine Arts
Computer Graphic Arts

Luleå University of Technology
Department of Arts, Communication and Education
Introduction to artist workflow of Physically Based Rendering in real time graphics

*Thesis in computer graphics*

Lucas Lundin

Art, communication and learning (KKL)
Luleå tekniska universitet, Skellefteå, 2015
Thesis work 15 hp
Computer Graphics, Bachelor of Fine Arts, 180 hp
Preface

This thesis contains the thesis that concludes my bachelor’s degree in computer graphics at Luleå Technical University, LTU. The thesis was written during a combination of practical embodiment in the industry and literature research. The project was conducted at a small advertising company in Stockholm called Inrederiet. Special thanks to my fellow students, family and teachers for the help during this project.

Lucas Lundin
Sammanfattning


Abstract

Physically Based Rendering has spread rapidly and been adopted well within the real time graphics industry. During the last years this technique has become increasingly more popular and the development have come far within real time graphics. Several major companies in the industry have changed their applications and workflow to handle the Physically Based Rendering pipeline, such as Epic Games Inc. and Unity Technologies. This could be a new breaking point in the industry, not necessarily in the graphical quality but in the process of creating art content in a new modern way. This thesis will present the basics foundation of Physically Based Rendering for real time graphics, followed by discussions around the workflow, benefits and drawbacks in production for artists.
Table of Contents

1 Introduction .................................................................................................................................... 1
  1.1 Background ..................................................................................................................................... 1
    1.1.1 The evolution and history of Physically Based Rendering ...................................................... 1
  1.2 Questions of issue ............................................................................................................................ 3
  1.3 Purpose ........................................................................................................................................... 3
  1.4 Limitations ...................................................................................................................................... 3

2 Glossary ........................................................................................................................................... 4

3 Theory ............................................................................................................................................. 2
  3.1 Diffuse and specular reflection ......................................................................................................... 2
  3.2 Translucency and transparency ........................................................................................................ 3
  3.3 Energy conservation .......................................................................................................................... 3
  3.4 Metal – Electric conductors and insulators ...................................................................................... 5
  3.5 Fresnel ............................................................................................................................................. 5
  3.6 Microfacet ....................................................................................................................................... 6
  3.7 Scene lighting and tone mapping ..................................................................................................... 7
  3.8 Physically Based Rendering attributes .......................................................................................... 8
    3.8.1 Albedo ........................................................................................................................................ 8
    3.8.2 Specular ..................................................................................................................................... 8
    3.8.3 Metal ......................................................................................................................................... 9
    3.8.4 Roughness ................................................................................................................................. 9
    3.8.5 Ambient Occlusion .................................................................................................................... 9
    3.8.6 Other maps ............................................................................................................................... 9

4 Methodology .................................................................................................................................... 9
  4.1 Industry practice .............................................................................................................................. 9
  4.2 Literature ....................................................................................................................................... 10
  4.3 Production tools comparison .......................................................................................................... 10
  4.4 Empirical Research .......................................................................................................................... 10
    4.4.1 Shading and rendering comparison .......................................................................................... 11
  4.5 Method Critique ............................................................................................................................... 13

5 Empirical Research ......................................................................................................................... 14
  5.1 Shading and rendering comparison ................................................................................................. 14

6 Result ............................................................................................................................................... 20

7 Discussion ....................................................................................................................................... 22
  7.1 Workflow and pipeline ..................................................................................................................... 22
  7.2 Development and utilization of Physically Based Rendering ....................................................... 23

8 Conclusion ....................................................................................................................................... 25

9 References ....................................................................................................................................... 26
  9.1 Literature ....................................................................................................................................... 26
  9.2 Figures ............................................................................................................................................ 29
1 Introduction

This thesis will present an introduction to an artist’s workflow of Physically Based Rendering in real time graphics based on research done on several industry leading companies and personal testing of production tools and principles. The major changes from a standard AD-HOC system to a Physically Based Rendering does not necessarily lie in the graphical quality but in the process of creating art content in a new way, with another workflow and a more optimized pipeline. The artist’s texturing workflow will be covered within Physically Based Rendering with emphasis on the benefits and drawbacks of each change. The thesis is written with Epic Games Inc.’s implementation of Physically Based Rendering within the game engine Unreal Engine as basis, which may differ from other implementations.

1.1 Background

In real time graphics, there have been many significant improvements over the past few years. Creating believable skin with plausible subsurface scattering, advanced particle systems and even anamorphic lens flares has been brought alive to the world of real time graphics. However rendering materials and handling lights properly have still not been contested with great success until now.

1.1.1 The evolution and history of Physically Based Rendering

The basic technology about Physically Based Rendering has been known and been running on offline rendering systems before this suddenly increased use within the real time graphics industry [6]. To be able to make any kind of statement about the future development and career for Physically Based Rendering the first step is to look at the evolution so far.

Mitsubishi Electric Research Laboratories published a study in 2005 [14] which became famous within the field of computer graphics, see Figure 1. The study compared 100 different materials in five popular shading models. All of the models had trouble to represent the same types of materials. This raised questions like: what are the models missing to represent in these materials? This was an early cause to the development of a new rendering system [7], which became the Physically Based Rendering system eventually. Walt Disney Animation Studios developed a Physically Based Rendering system for their hair shading in Tangled (2010) for a more artist friendly production [15]. With the success from Tangled, Walt Disney further developed the system in order to be able to have more materials with the same model that would give the same visual richness and smoother production with improved artistic control. Walt Disney’s new Physically Based Rendering system is showcased in Wreck-It Ralph (2012) and Monsters University (2013) [16]. This was the big break through that was seen as a source of inspiration and proof of the physically based model working. Walt Disney’s work with the physically based shading model became the basis in the field. Around 2012 Epic Games Inc. started their research in the field of Physically Based Rendering with Walt Disney’s shading model as
inspiration, creating a stripped real time variation of Walt Disney’s model. Epic Games took great care in putting up a set of guidelines or principles for the development of their game engine, just like Walt Disney had done. Using Physically Based Renderings full potential that not only lies in visual richness but in several other categories as well. Epic Games developed the game engine with the following keywords in mind: real time performance, reduced complexity, intuitive interface and flexibility, to mention some. Creating not only an efficient system, but a system with a wide range of features encouraging the artists as well. By the launch of Epic Games new game engine Unreal Engine 4, several other major companies in the industry had developed or experimented within the field. A new modern standard of texturing materials emerged with several companies developing new tools. Quixel and Allegorithmic as example, with entire suites of tools for texturing and shading. More game engines are supporting Physically Based Rendering and even more are to come. Unity Technologies is releasing Unity 5 with support for Physically Based Rendering. Another notable fact is the fairly new progress of research within real world light behavior, such as deeper understanding of fresnel and energy conservation.

(Figure 1. A study comparing 100 different materials in five shading models. Sorted left to right by relative errors. This was an early cause to the development of Physically Based Rendering [7].)
1.2 Questions of issue
This thesis will discuss the following questions of issue:

- What does the modern production pipeline look like in the Physically Based Rendering workflow within real time graphics for an artist and how does it work?
- What are the benefits and drawbacks of using Physically Based Rendering in real time graphics in production as an artist, compared to the traditional diffuse and specular shading model?

1.3 Purpose
The purpose of this report is to discuss Physically Based Rendering in real time graphics and evaluate the advantages and disadvantages within the field compared to the traditional diffuse and specular shading model.

- Clarify what Physically Based Rendering means for a graphical artist. Since there is some general misunderstanding regarding Physically Based Rendering and how it works [17].
- Explain how the modern production pipeline looks like in the Physically Based Rendering workflow within real time graphics for a graphical artist.
- Evaluate Physically Based Rendering within real time graphics from an artist point of view.

1.4 Limitations
The Physically Based Rendering model in focus is primarily Epic Games Inc.’s implementation in Unreal Engine 4. This thesis will focus on real time graphics but will use pre-rendered graphics within the field of Physically Based Rendering for comparison and reflection, since Unreal Engine 4 uses a simplified variant of the pre-rendered solution Walt Disney Animation Studios uses and the gap between these two rendering methods is getting smaller [5]. No mathematical equations or algorithms will be discussed or presented since the main target group for this report is artists. Since the readers are assumed to have basic understandings of geometry and computer graphics this thesis will exclude such theory.
2 Glossary

**Physically Based Rendering** – Rendering system which uses real world measured data.

**PBR** – Physically Based Rendering

**PBS** – Physically Based Shading, similar meaning as PBR.

**Normal** – The perpendicular direction from a surface.

**Diffusion** – Light that penetrates and scatters across a surface.

**Reflection** – Light that returns in the opposing angle of the surface normal intact.

**Specular** – Surface reflected light, highlights.

**Albedo** – Base color of a surface without any shadow information or highlights.

**Metal map** – A gray scale bit map which defines if the material should be treated as electrical conducting or not (a metal or non-metal).

**Translucency** – A phenomenon where some light penetrate and passes through a surface intact.

**Transparency** - A phenomenon where entire light images penetrate and pass through a surface intact.

**Energy conservation** – Light is conserved within a surface. No more light will ever leave a surface than the light that originally fell upon the surface.

**Microfacet** – Small irregularities in a surface that makes light scatter in different directions.

**Fresnel** – A phenomenon which makes surfaces observed at grazing angle reflecting more light rather than from a flat on angle.

**Real time graphics** – Graphics which is rendered at real time.

**Pre-rendered graphics** – Graphics which is not rendered at real time (offline).

**BRDF** - Bidirectional Reflectance Distribution Function

**IOR** - Index Of Reflection

**HDR** – High Dynamic Range

**AD-HOC system** – The traditional rendering system for real time graphics which uses diffuse and specular.
3 Theory
In order to analyze, compare, and be able to answer the question of issue a thorough theoretical background is required to understand what the subject is about. This section explains the foundation of Physically Based Rendering with a basic comparison of how light behaves in the real world to be able to later discuss the questions in more detail.

3.1 Diffuse and specular reflection
Diffuse reflection also called diffusion and specular reflection describes the foundations of light interaction with surfaces. There may be a difference between the tales terms that are often used and the physical process that occurs. When light hit a surface it will reflect, portions of the light bounces further in the opposite direction of the surface normal. This phenomenon is comparable to throwing a bouncing rubber ball on the ground (See Figure 2). If the light hits a surface which is very even and smooth the reflection will become stronger and a mirror like effect appears [1]. This is the specular reflection which also is known as the highlight on a surface. Just as in the example with the rubber ball, the specular reflection will reflect in a specific angle making it view-dependent for the camera and the appearance will change when the viewpoint vary [29].

In reality, when light hits a surface not all of the light reflects. Portions of the light are absorbed by the surface and can form heat or scattering internally in the surface. Light can also get up to the surface again and becomes visible for the camera or the eye. This effect is called inter alia subsurface scattering or diffuse reflection, see Figure 2. Diffuse reflections are light that scatter in every different direction, this makes the surface look the same from different viewpoints [29].

Different types of materials and surfaces absorb and scatter diffuse light in different ways internally. As an example: an object absorbing light with all colors except red light, and only scattering out the red light from the surface, would make the object appear red.

To determine the diffuse light color that scatter back, you only need one attribute. Commonly known as Albedo, diffuse color and base color, these different words are all describing the same thing which is the base color of the surface without shadow information.
3.2 Translucency and transparency

In some material the diffusion is more complex and can not only be determined by a single attribute. If a material has a wide scattering distance, light could pass through the object and come out of the backside intact. This effect is called translucency and can be found in material such as skin and wax. The effect is seen clearer if the object is thin since light can pass through easier. This forces the shading system to take both the thickness and shape of objects into account in order to render accurate translucency [1].

With an even wider scattering distance where almost no scattering is found, there is possible for entire images of light to pass through intact. This is called transparency and can be found in water and glass. Transparency is generally controlled by a grayscale opacity map.

3.3 Energy conservation

Diffusion and reflection has an important relationship which is one of the corner stones of physically based rendering. For incoming light to be diffused it must penetrate the surface in order to be absorbed and scatter through the object. The light that does penetrate the surface is not a part of the reflecting light. This is called energy conservation and means that no more light will ever leave the surface then the original amount of light that fell upon the surface [30]. This also means that a highly reflective surface will only show a small amount of diffuse light [1]. This is also true conversely. A surface with high diffusion will be less reflective, see Figure 3. To calculate the diffuse shading you simply subtract the reflecting light and what is left is the absorbed and scattered light within the object’s surface. In many cases a surface with low amount of reflection will appear brighter overall compare with a highly reflective surface, due to the fact that the surface is having the same amount of light now scattered and illuminated more evenly across the surface which illuminates the surface as a whole, so called diffuse reflection or diffusion, mentioned in 3.1.
Increasingly reflective

(Figure 3. Increasingly more reflective from left to right. Demonstrating energy conservation.)
3.4 Metal – Electric conductors and insulators

Materials that are electrical conducting have a distinct different behavior when interacting with light compared to insulating materials (sometimes mentioned as dielectric materials). The most common electrically conducting materials are metals, and therefore generally dividing the materials into either metals or non-metals. Conductors or metals are in general far more reflective than isolators. Metal is usually between 60 to 90 percent reflective, when on the other hand insulators or non-metals don’t exceed far over 20 percent reflectivity [1]. The high amount of reflectivity in the metals only leaves a small portion of light to penetrate the surface, making the material appear shinier. The small amounts of light that do penetrate the surface tend to be mostly absorbed rather than scattered within metal materials. The lack of scattering light, also called diffusion makes metal materials to only show a small amount of diffuse light. In other words, metal only present a small portion of the base color of the material.

Another characteristic behavior of metals is that the specular reflection can alter and sometimes has a tinted color [29]. This effect can be found in metals such as gold and copper.

For game engines and applications to be able to render metal or non-metal materials correct, a gray scale bit map is used to determine what properties different surfaces shall have. This attribute is often called “metal” or “metallic” where 1 represent electrical conducting materials and 0 represents insulating materials. Since most materials are either conductors or insulators the values in the bitmap tends to be either 1 or 0 with few exceptions of float values. If a surface would be determined as metal (metallic: 1) the specular intensity and color is picked from the diffuse color or albedo map, giving the reflection of the surface a tint [10]. A metallic value sometimes changes the fresnel factor to an average appropriate for metals or non-metals in an application or game engine.

3.5 Fresnel

In this thesis fresnel refers to a reflection phenomenon which changes based upon the different angles an object surfaces is observed. Smooth objects are more reflective at the edges where you see the surface at a grazing angle compared with an angle straight on, see Figure 4. A smooth object can also appear brighter at the edges if the reflecting light is scattered. At maximum grazing angle of a smooth object the reflection should be corresponding, reflecting a perfect image, looking like a mirror[2][3]. This is true no matter what the material is, although this is not very clear to observe in reality [4]. This fresnel effect has a curve of the influence based on the view angle. A smooth non-metal surface which is facing the observer straight on reflects about 4% of the light, where at grazing angle the same material would reflect almost 100% of the light [10]. This curve changes slightly between material, with two groups of materials being most divergent, metals and non-metals. The fresnel factor changes so little between different materials within these two groups that some game engines and applications have preset fresnel values triggered by the metal map. The fresnel reflection is added on top of the original reflection during rendering where the conditions for the effect are met.
3.6 Microfacet

Both specular reflection and diffuse reflection takes the objects shape and angles into account when reflecting light on a surface. The reflections will be determined from all shapes and angles of the object, even down to a microscopic level. In computer graphics an objects is many times described by several different attributes. The main features of an object are represented by a 3D mesh made up by polygons, where increased accuracy is necessary, a normal map is commonly used to describe smaller details for the shading system. Although we cannot see it with the naked eye, even more irregularities and small bumps can be found in any surfaces in reality. These small irregularities are called microfacets, commonly mentioned as the microsurface. The shape of the microsurface affects the diffusion and specular reflection in a similar way as the overall shape of an object but on a very small scale, still performing an effect essential for the surface appearance. The amount and scale of microfacets can vary widely between surfaces and materials. A surface with a high amount of irregularities and small bumps in its microsurface will mostly affect the reflection, making the incoming light reflect in unpredictable directions [1], see figure 5. This creates a more dim reflection appearing blurry. Contra a smooth surface with low amount of irregularities in the microsurface, which would have a sharper reflection appearing more intense due to the predictable and concentrated direction of the reflection. Light hitting a more irregular microsurface have a larger chance of bouncing several times on the same surface due to the cracks and peaks in the surface, trapping the light. This can make the surface more likely to absorb the light.

In order to manage to describe the detail in the microsurface of a material you determine average values of the amount of irregularities rather than describing every single bump. This works out very well and is much more efficient. You
control this via a gray scale bit map called roughness map. Where the value 0 represents no roughness (smooth) and the value 1 represents maximum roughness.

![Figure 5. Incoming light reflects in unpredictable directions on a surface with a high amount of irregularities and small bumps in its microsurface.](image)

3.7 Scene lighting and tone mapping
In order to get the most benefits from Physically Based Rendering the system will need several basic features to handle the shading properly. The scene will need to have physically accurate materials properly set up for the specific application determined to use, using the PBR workflow. These materials won’t perform as expected if you don’t follow the basic lighting guideline suggested for setting up Physically Based Rendering scenes. With proper lighting but incorrect materials the final shading will also appear inaccurate. One of the major differences to the older real time lighting workflows is the increased use of Image Based Lighting with environment maps, ambient light and usage of directional light [3]. Another aspect is the lights fall off and intensity values. A light intensity fall off rate that follows the inverse square law provides a more accurate lighting condition where it is bright close to the light source and becomes dim quickly further away from the source [10], see figure 7. Some applications provide light intensity in lumens to easier apply measured light values.
In order to handle environmental IBL and the intense values from the inverse square falloff the system required handling High Dynamic Range (HDR) values which extends the color values displayed by monitors. These HDR values are not only used for lighting and shading, but used to drive other elements such as bloom effects during the post process as an example. The rendering system is thereby requiring tone mapping. Tone mapping is the process in which the HDR values become converted to displayable values for the monitor [3].

3.8 Physically Based Rendering attributes

With the modern Physically Based Rendering production pipeline a new set of attributes are required. There are some notable guidelines that are worth mentioning about the different texture maps, attributes and the way artists are supposed to handle them in the general set of application used today.

3.8.1 Albedo

_Also referred as base color, diffuse color and diffuse._

Albedo is the base color of a surface with no shadow information or highlights. Real world materials can be measured and converted into RGB values and intensity values for the albedo which can be found in PBR charts online. Using a polarizing filter when photographing real world non-metal materials removes specular highlights for textures [13].

3.8.2 Specular

_Also referred as reflection. (0 = no reflection   1 = mirror)_

Specular map controls the amount of reflectivity on non-metal. Specular attribute has no effect when the metallic is set to 1, but is instead driven by the values form the albedo. The specular attribute should in some applications be left unchanged, the default value will be 0.5 which works in most cases. Real world materials can be measured and be converted into intensity values for specular based on IOR measurements, which can found in PBR charts online.
3.8.3 Metal
\(0 = \text{non-metal} \quad 1 = \text{metal}\)
The metal map gives material metallic properties if set to 1 and no metallic properties if set to 0. Very few gray scale values should be found in this texture map since materials is mostly either metal or not. If the metal map has a value of 1, the specular values will be blended with the albedo values, and the albedo maps values will be lowered. Sometimes the fresnel factor will change based on the metal maps value.

3.8.4 Roughness
\(0 = \text{rough/matte} \quad 1 = \text{smooth/shiny}\)
Roughness map controls whether the material is smooth or rough. With a low roughness the material will spread the reflecting light making a subtle and blurry reflection, contra a low value which would give a concentrated and sharp reflection like a mirror. The roughness map should be a gray scale bit map and can include a lot of variation in it.

3.8.5 Ambient Occlusion
Also referred as Cavity. \(0 = \text{max shadow} \quad 1 = \text{no shadow}\)
Ambient occlusion is used to add extra shadow information which the shading system cannot manage to produce itself [11]. Possible shadow information removed from the albedo can be added here to regain detail.

3.8.6 Other maps
Most other maps in the process of creating physical based material are having the original standard procedure, such as normal map, emissive map, opacity map and so on.

4 Methodology
This chapter presented the methods that were used.

To answer the questions at issue, several methods were used to gather information, data and examine theories. The largest source of information and knowledge for this report has come from studying literature and the works of others with in the same field. Empirical research and testing theories and new knowledge in practice and to observe the result have been a part of the method.

4.1 Industry practice
The project was conducted at a small advertise company in Stockholm called Inrederiet. The task I was designated to do was to create a library of 3D model of selected props according to the industry's standard set for Physically Based Rendering in real time running on web browsers. This experience and visit taught the basis of Physically Based Rendering and was the basic foundation for the project.
4.2 Literature
This thesis is having its basis from studying literature and comparing others work within the same field. To mention a few: Crafting Physically Motivated Shading Models for Game Development by Naty Hoffman[3], Theory of Physically-Based Rendering by Jeff Russell[1], Real Shading in Unreal Engine 4 by Brian Karis[5]. The topic in focus has well documented material. However, most of the documented reports are handling a technical aspect such as implementation of Physically Based Rendering and not about usage, utility or future of Physically Based Rendering. Studies within Physically Based Rendering that are not about real time graphics are still relevant for this thesis in many ways since many things apply to both real time and pre-rendered graphics.

4.3 Production tools comparison
Analyzing companies within the field regarding their working methods and production tools is a part of finding out and establishing the modern production pipeline and sort out the benefits and drawbacks of Physically Based Rendering. There are multiple examples of companies in the industry that have tried out or entirely developed a Physically Based Rendering workflow. Epic Games Inc. have their latest game engine (Unreal Engine 4) using Physically Based Rendering which is one of the motives for this thesis and which the analysis is mostly based on. Allegorithmic and Marmoset have recently developed tools for a Physically Based Rendering workflow. Disney is a company that has been working with Physically Based Rendering in pre-rendered graphics and has several publications within the field, which have been useful reading for this thesis.

4.4 Empirical Research
The empirical research consists of a comparison between four commonly used game engines with two using Physically Based Rendering and the other two not. The comparison will focus on the usage and workflow of PBR versus previously standards. The test is primary targeting shading in different lighting conditions

(Figure 8. A portion the models in a scene lit using Image Based Lighting, Physically Based Rendered in real time on a web browser.)
in order to see consistency in the shading model, in other words; how well the reproduced materials performs in different lighting condition.

4.4.1 Shading and rendering comparison
This section describes the method of the shading and rendering comparison made in the four game engines in more detail.

4.4.1.1 Objective
The objective for this comparison is to analyze the behavior of real time graphics rendered with Physically Based Rendering and without. The comparison is also made to look for differences in the workflow and to see benefits and drawbacks in a practical demonstration.

4.4.1.2 Approach
To reach the objective in an equivalent and as accurate way as possible, the comparison is done by comparing the same mesh-model which will be textured and shaded specially for each rendering model. They will be lit with as identical light conditions as possible. The texturing and shading will be examined in several lighting conditions. The comparison is done in four game engines in total, two with Physically Based Rendering models and two without. The two game engines using Physically Based Rendering are Unreal Engine 4 from Epic Games Inc. and Unity 5 from Unity Technologies. The two game engines without Physically Based Rendering are Unreal Development Kit Unity from Epic Games Inc. and Unity 4 from Unity Technologies.

4.4.1.3 Preparation
To get as accurate result as possible in this comparison it is important to have the same conditions for all game engines. This will eliminate the amount of errors and give a more truthful picture. The same mesh model (see Figure 9.) will be used in both of the engines with the same normal map applied (see figure 10).

(Figure 9. Mesh model with normal map applied.)
In order to be able to more easily have the same light condition in the different game engines the choice of lighting set up became the following. Using a High Dynamic Range image to drive a Skybox for reflection and an Image Based Lighting for the ambient light effect. To produce cast shadows a directional light is placed in the scene in the direction of the main light with a tint of the overall HDR image. In Unity the Image Based Lighting was done by Marmosets plugin, called Skyshop. In Unreal Engine I decided to create my own blueprint for the Skybox and Image Based Lighting. Once everything is sorted out the light set up should be easy to handle and to import the lights to the second engine once it is set up in the first. The first HDR image which the texture and shading will be adapted specifically for can be seen in figure 11.
4.5 Method Critique

The empirical research was done on a low number of different game engines and it is desirable to compare more game engines and applications to get a more truthful picture. The main reason why the comparison only was done in four game engines was the time limitation and secondly that some of the major game engines is outside the budget for the report. Smaller game engines may be less expensive or free but would give inaccurate results since they are on a completely different quality level and are not a part of the industry standard which this report focus on. Having more in depth knowledge about the game engines and how the rendering and lighting systems in each of them are working would also have been of great benefit to be able to do a more precise light setup and comparison.
5 Empirical Research

5.1 Shading and rendering comparison
The workflow of handling the textures and shading when working with Physically Based Rendering is quite different from the ordinary non-PBR workflow. With other attributes for your shader you have to think differently when texturing. For the albedo map you have to remove some of the shadow information and highlights in order to get better results. Compared to a diffuse map which many times can benefit from keeping detailed shadow and light information. The albedo map uses fixed values for materials as a reference point, which in general has a brighter intensity than the diffuse maps materials will have, see Figure 12 and 13. The roughness map can sometimes be seen used as an inverted specular map, but this is wrong. A roughness map only contains gray scale values, and the specular tint will be picked from the albedo map if the material is metallic, see Figure 14 and 15. To add in the small detail lost when removing the shadow information from the albedo map, an ambient occlusion map or cavity map can be applied, see Figure 17.

(Figure 12. Left, Albedo map for Unreal Engine.)
(Figure 13. Right, Diffuse map for Unity.)
Handling the light setup between the four different game engines turned out to be harder than expected. The game engines use different scales for controlling the light intensity and cube maps are treated differently and have to be in different formats. Textures and shading got compared in three different lighting conditions, first light setup being the one the textures and shading was specifically adapted for, see figure 18.

The behaviors of the materials and textures in the different lighting conditions can be seen in figure 21 and 22. It is clearly notable that the Physically Based Rendering model performs consistent in different lighting scenarios. The specular on the chrome ring around the turntable is a mentionable example of
the specular reflection not reflecting enough of the surrounding IBL in Unity’s model without Physically Based Rendering. Materials that have high specular values tend to get a peaking saturation in some light conditions making the texturing phase unpredictable for the traditional ad-hoc system, this can be seen in figure 21 and 22 at the center of the shield and in the left barrel.

(Figure 21. Left, Unreal Engine 4 with PBR model. Right, Unreal Development Kit without a PBR model. The top row showing the light condition which the textures and shaders were specifically made for.)
(Figure 22. Left, Unity 5 with PBR model. Right, Unity 4 without a PBR model. The top row showing the light condition which the textures and shaders were specifically made for.)

By looking at how companies in the industry uses their production tools, (Crytek [20], Marmoset [2][19], Epic Games Inc. [12][11][10], Allegorithmic [8][9], Quixel [21]) manages their workflow and by the observations during the empirical research it is shown that the workflow many times can vary. There is no strict standard on how to work with Physically Based Rendering when it comes to texturing and shading in the industry since companies are doing things differently. There is a wide range of software available on the market which provides the industry with different approaches and solutions, and companies often develop their own tools. Some solutions are more automated using presets and existing libraries, while others are more manually operated. Some prefers to paint straight on a 3D model when texturing, while others paint on 2D UV maps. However, you can find some similarities in the workflow. The Physically Based Rendering workflow is looking different in several ways compared to the traditional ad-hoc system, which uses diffuse and specular. First of all, the material model has changed and the attributes must be fed with correct data, see figure 23. With different attributes, new methods are required for the artists to
create these textures. In addition to the new attributes, the general guidelines of thinking regarding the texturing process has changed slightly, see figure 24.

(Figure 23. To the left, attributes of a traditional ad-hoc system using diffuse and specular. To the right, attributes of a Physically Based Rendering system. [2][11][18])

The greatest difference between the traditional system and the Physically Based Rendering system is of course the roughness and metal map. Another big change is trying to reduce specific light and shadow information from the texture maps and instead have an external ambient occlusion map if necessary. It is clearly seen in the industry that the overall workflow of controlling multiple texture maps simultaneously within Physically Based Rendering has increased. This is done in different ways in practice within companies, but is intended toward the same goal, an easier workflow for the artists. Some companies uses different layered methods, while other uses color id maps to distinguish each material and use these as masks. This is sometimes handled automatically by software or by more manual methods.
To be able to deliver correct textures for each attributes you will need to know what game engine or application you ultimately will use, since different implementations of Physically Based Rendering prefers slightly different attributes. Some companies even develop tools to calibrate and review their art content in separate passes for each attribute, Dontnod Entertainment for their title Remember Me as an example [22]. In addition to content in the texture files, optimization is also handled a bit different. In the traditional ad-hoc system you will usually have three textures with three channels in each, ending up with three image files. When it comes to the Physically Based Rendering system you have six textures with a total of ten channels, this makes it possible to stack textures in empty RGBA channels to reduce the amount of image files down to three even though you have six different textures [20].

<table>
<thead>
<tr>
<th>General guidelines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ad-hoc system</strong></td>
</tr>
<tr>
<td>Observe reference footage.</td>
</tr>
<tr>
<td>Trial and error- Experiment to get the right color and intensity values.</td>
</tr>
<tr>
<td>Use specific light and shadow information in the diffuse and specular map to achieve more details and look more plausible.</td>
</tr>
<tr>
<td>Textures might need adjustments if the objects lighting condition would change in order to look plausible.</td>
</tr>
<tr>
<td><strong>Physically Based Rendering</strong></td>
</tr>
<tr>
<td>Observe reference footage.</td>
</tr>
<tr>
<td>Use reference color and intensity values from real world measurements as a starting point.</td>
</tr>
<tr>
<td>Avoid using specific lighting and shadow information in the texture maps.</td>
</tr>
<tr>
<td>Define which surfaces are metal and non-metal.</td>
</tr>
<tr>
<td>Check the shading in different lighting conditions.</td>
</tr>
<tr>
<td>Optimize the use of the RGBA channels</td>
</tr>
</tbody>
</table>

(Figure 24. To the left, guidelines for a traditional ad-hoc system using diffuse and specular. To the right, guidelines for a Physically Based Rendering system. [2][11][18][19])
6 Result

The general current state of Physically Based Rendering seen in the real time industry today is more powerful in many different aspects compared to the ordinary and older AD-HOC system which only uses specular and diffuse attributes. The result for this thesis project includes specifying notable benefits and drawbacks of using a Physically Based Rendering system for real time applications. These are some of the benefits worth mention:

- Art content will appear with high accuracy and consistency in different lighting conditions.
- Easy to follow physically measured real world values as reference points.
- Predictable result reduces the guesswork and tweaking for the artist.
- Creates a standard for the industry which is easy to follow – able to share content to some extent.
- Simpler material creation for the artist, due to robust but flexible parameters. Almost all combinations of parameters should be plausible [5].
- Easier to achieve photorealism with physically based material, lighting and rendering. Still works very well for stylized art, as an example Monsters University (2013) from Walt Disney Animation Studios [16].
- Troubleshooting becomes easier when the system is built on real world measurements and algorithms rather than fabricated.
- Easier to extend the system [3].

And of course there are some drawbacks with Physically Based Rendering. These are the ones most worth mentioning:

- Workflow - Some part of the workflow are more complex and time consuming, such as keeping out shadow information and highlights from the albedo and the increased amount of attributes.
- Can be hard, but possible to use it parallel with other rendering models [15].
- Some implementations are requiring more computing power and could be called less efficient.

There is no strict modern production pipeline that the entire industry uses. But there are some notable changes in the Physically Based Rendering workflow that can be found when comparing with the traditional Ad-hoc system using diffuse and specular. Both when it comes to attributes and how they should be handled and some general guidelines that most companies seem to follow, see figure 26.
Several companies in the realtime industry have changed their pipeline to support a Physical Based Rendering workflow. For example: game engines such as Unreal Engine 4, Unity 5 and Cryengine. But the list does not stop at game engines, specialized tools have appeared on the market for artists to facilitate the production with the new workflow. Companies like Quixel and Allegorithmic have entire suites for handling the Physically Based Rendering workflow which indicates a point in the right direction for the future of Physically Based Rendering. More of the development and utilization of Physically Based Rendering will be discussed in the discussion section and the conclusion.
7 Discussion

Physically Based Rendering brings a lot of benefits to the industry of real time graphics. This section will discuss the previous subjects of this thesis and bring in a personal opinion and views of the matter.

7.1 Workflow and pipeline

The workflow used in the Physically Based Rendering pipeline is different than the previous standard using specular, diffuse, normal map workflow. This is mostly due to the different attributes used in Physically Based Rendering. The workflow changes have impact on several different departments in a production including: rendering, tech, character, environment and lighting in many ways. For companies it could very well become a difficult transition to give up on the pipeline and proven production guidelines they know and commit to a new workflow. As an individual artist the changes are fairly simple to adjust to for starting creation of art content, since the basic concept are not very complex or hard to understand regarding Physically Based Rendering for real time. By using a Physically Based Rendering system it many times means that you will have to control more texture attributes, for example: albedo, normal, specular, roughness, metallic, ambient occlusion and sometimes more. This can be more time consuming, but it also allows you to have more control of the final result without coming to the point of overwhelming parameters being redundant. All attributes have their key features needed in the Physically Based Rendering system. Substance Painter is a tool developed by Allegorithmic which allows artists to paint in several channels simultaneously. This is one example of a solution to a problem which Physically Based Rendering could cause to a pipeline.

The production pipeline and workflow varies between companies and there is no strict standard. This seems quite strange at first glance, but I personally think this is logic. For some companies, the jump to a Physically Based Rendering workflow is very strenuous and they try to change as little as possible as of the previous workflow while other companies might want entire new routines and workflow. The implementation of Physically Based Rendering can also vary a lot and demand differently calibrated attributes, which could change the workflow [4][18][19]. And finally there is also a decision on how you like to work. Using more manual tools for more precise control or a more automated workflow with presets and multi functions, texturing on a flat 2D UV surface or straight on a 3D model. There are several techniques in the industry in use and it is hard to say which one is to prefer.

As a personal opinion formed from writing this thesis, I believe that the general approach of using real world measured data to drive or be used as references in computer graphics could be the ultimate way to achieve great quality in the image. In most of the cases, both in movies and games, the purpose is to create spectacular images with the ordinary visual rules of light and physics to make it believable and plausible. And this is what Physically Based Rendering delivers in a small scale to the real time graphics industry. The point is to create whatever you want, but with correct lighting conditions and with the right behaviors of the materials. Walt Disney Animation Studios movie Monsters University (2013)
is an outstanding example of stylized art with a monster theme, still following the physical laws using Physically Based Rendering. Physically Based Rendering in real time is of course far from one hundred percent accurate, and has a long way until it reaches a high level of photo realism.

7.2 Development and utilization of Physically Based Rendering

By discussing some of today’s utilization of Physically Based Rendering you can see the direction that Physically Based Rendering is heading within real time graphics regarding the further development, benefits and drawbacks. To start off, it is worth to mention once more that almost every major game engine have or will in a short time period convert to support for Physically Based Rendering, Unreal Engine 4, Stingray[26], Unity 5, Cryengine[25] and Frostbite[24] to mention some. Most of the leading game developing companies have also shifted into Physically Based Rendering systems with Sony Computer Entertainment[22], Electronic Arts[24], Ubisoft[23], Microsoft Studios – Crytek[25], Capcom, Dontnod Entertainment[20] as example. Several new games are currently using Physically Based Rendering and some already existing games are under further development to handle the Physically Based Rendering system, EVE Online to mention one game [27]. This clearly shows the strong adoption the industry is having and also the increased number of users [22]. This indicates at least for now a promising future for the industry as well, since there is a universal need for an art content standard which will come with Physically Based Rendering [22].

Tools and applications have in general become easier to master and the online community contributes to this via sharing, support, forums, indie projects and game engines as well as educations and online courses [18]. This makes it easier for individuals and companies to transition to the Physically Based Rendering workflow. Additionally, Physically Based Rendering has a more intuitive workflow than some other models. Having a large amount of users using the same model benefits everyone. With a large community, content and ideas are easily shared and the system becomes even more developed. Content produced for Physically Based Rendering can in general be shared quite smoothly depending on the different implementations. The content is supposed to have real world measured values as references which should be quite identical. With more and more applications developing support for Physically Based Rendering the user group becomes larger. And with a large amount of users the further development could be accelerated. Physically Based Rendering for real time can be further developed in different ways [3]. As an example, the computational efficiency is always a fact and since Walt Disney Animation Studios among others have shown great promise in the visual quality in pre-rendered graphics, the optimization is an essential development territory for the real time section. There is of course always room for further development when it comes to usability for the artist, and making it more user friendly. One thing that can be slightly misleading when texturing is that a white roughness map will give very dull specular reflection while a white specular in the traditional rendering system would give you are very intense specular reflection, this can feel confusing when you adjust the specular. This is something that some implementations already have a solution for with the specular and glossiness
workflow that uses a gloss map instead of a roughness map which basically is an inverted roughness map. This can be found in Marmosets Tool Bag 2 and Cryengine.

A constant bottleneck for real time graphics has always been the limit of computer power for the end user. Since the overall computer power increases for the end users all the time, the future could allow the artists with the possibilities to use more complex shaders with additional attributes to describe materials. Since metallic materials is represented far more accurate in the Physically Based Rendering system compared to the traditional Ad-hoc system, skin and more organic materials will probably be the next target of improvement. Perhaps an increased use of Separable Subsurface Scattering incorporated with the Physically Based Rendering system would allow rendering more plausible materials such as skin [28].
8 Conclusion

Physically Based Rendering is a word that is used a lot and it can be hard to understand the meaning of it. This is because of the many different implementations that exist. The corner stones of Physically Based Rendering are not the use of metallic maps or the use of a specific algorithm. Physically Based Rendering is more of a concept, where you want light to interact accurately with materials guided from real world measured material values along with using energy conservation as a feature [2] [18].

The visual richness and quality of the real time graphics using Physically Based Rendering have contributed a lot for the realism in materials and light. But it is far from achieving a quality equivalent to pre rendered graphics. Still metal materials are now having a far more accurate representation than the older rendering systems within real time graphics, can be seen in figure 21 and 22. Materials in addition to the metal are also both easier to reproduce and better represented in the final product generally. However, the quality in the final graphics are not everything. Physical Based Rendering also brings a more manageable workflow to achieve consistency, control and is providing broad versatility for the artist.
9 References

9.1 Literature


Hoffman, Naty. Crafting Physically Motivated Shading Models for Game Development


Physically Based Materials. Unreal Engine 4 Documentation. Epic Games, Inc.


MERL BRDF Database. Mitsubishi Electric Research Laboratories.


Christophe, Hery. Ryusuke, Vilemin. Pixar Animation Studios Physically Based Lighting at Pixar

Wilson, Joe. PBR Texture Conversion. Marmoset Toolbag 2.
http://www.marmoset.co/toolbag/learn/pbr-conversion (Accessed 2015-08-04)

Bouquet, Rémy. nehon. Physically Based Rendering – Part One/Two/Three.
http://jmonkeyengine.org/299803/physically-based-rendering-part-one/
http://jmonkeyengine.org/300495/physically-based-rendering-part-two/

Schulz, Nicolas. RYSE – THE TRANSITION TO PHYSICALLY BASED SHADING. Making Games.

Cooper, Adam. DDO Introduction. Quixel.
[22] Seymour, Mike. Game environments – Part A: rendering Remember Me. Fxguide


[26] Game engine and design visualization software. Stingray

[27] CCP Mankiller. PHYSICALLY BASED RENDERING AND MAKING EVE LOOK “REAL”


9.2 Figures

(Figure 7) Unreal Engine. Epic Games, Inc. 
https://de45xedrsdbp.cloudflare.net/blog/invsq01-600x300-1318205312.png (Accessed 2015-05-22)

(Figure 1) Burley, Brent. Physically-Based Shading at Disney. Slide. Walt Disney Animation Studios.