Alternating Physically Based Rendering in Low-lit Areas

Itamar Kupersmidt
This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in Digital Game Development. The thesis is equivalent to 10 weeks of full time studies.

The author declare that they are the sole author of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author:
Itamar Kupersmidt
E-mail: itamar.kup@gmail.com

University advisor:
Stefan Petersson
Department of Creative technologies

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

Background. The increase in screen resolution has increased from HD to Ultra-HD during the last decade. A modern game today with Ultra-HD resolution has over eight million pixels that need to be shaded, combined with the expensive shading method Physically Based Rendering the equations needed to calculate each pixel are numerous.

Objectives. This the study aims to remove complexity from the Physically Based Rendering shading method in the form of roughness in low-lit areas. The low-lit areas will instead be rendered without the roughness attribute. By removing roughness less calculations will be performed.

Methods. To remove roughness from low-lit areas the light had to be approximated using a diffuse model. The pixel was later converted via Hue Saturation Perceived Brightness to calculate the brightness. If the pixel was under the given threshold, the pixel was shaded using a low-complexity Physically Based Rendering implementation without roughness. A user study was conducted using Unity game engine with eight participants being asked to compare different stimuli all rendered with different thresholds for darkness with a reference picture. The aim of the study was to ascertain if the stimuli without roughness had any perceivable difference from the reference.

Results. The results of the study show the majority of the participants noticing a difference when comparing the stimuli with the reference. The areas affected was not only the low-lit areas but the whole scene. The energy conversion without the roughness value made the whole scene appear darker.

Conclusions. The roughness value is an integral part of energy conversion and without it, the scene will appear much darker. While the majority of participants noticed a difference, the lowest threshold resembled the original the most.

Keywords: Physically based rendering, Roughness, Low-lit environments.
# Contents

Abstract i  

1 Introduction 1  
   1.1 Aim and Objective .................................. 2  
   1.2 Research Questions and Hypothesis .................. 3  
      1.2.1 Risk and Limitations ............................ 3  

2 Background 5  
   2.1 Video Game Lighting ................................ 5  
   2.2 Physically-Based Rendering .......................... 5  
      2.2.1 Microfacets .................................... 6  
      2.2.2 Subsurface scattering ........................... 7  
      2.2.3 Energy conservation ............................. 7  
   2.3 Brightness Calculation ............................... 7  

3 Related Work 9  

4 Method 11  
   4.1 Design ........................................ 11  
   4.2 Procedure ........................................ 12  
   4.3 Ethics ............................................ 13  
   4.4 Unity Game Engine PBR Implementation .............. 14  
   4.5 Set-Up ........................................... 15  

5 Results 17  

6 Discussion 19  

7 Conclusions and Future Work 23  

References 25
Chapter 1

Introduction

For gamers today one of the most important settings when it comes to visual quality is the resolution. The resolution represents the number of pixels shown on the screen. The higher the resolution the sharper and more detailed the image [17]. Most commonly used resolutions today are 1920x1080 full high definition (FHD), 2560x1440 quad high definition (QHD) and 3840x2160 ultra high definition (UHD) see Figure 1.1. In the console market alone we have seen a jump from 640x480 to FHD, a 6.75 times increase in pixels from the mid-’90s [13]. In 2016 both Sony and Xbox released consoles supporting UHD [20]. The actual number of UHD gamers on steam only make up 0.49% while FHD makes up for 76.47% of the players on steam. The number of people that owns an NVidia GeForce GTX 1080ti graphical processing unit (GPU) is 0.47% of the steam user database. Not many graphics cards can support a flawless UHD experience. The cost of shading over eight million pixels is more than most graphics cards can processes [3].

Figure 1.1: Different screen resolutions. acquired from [17]

Most modern games today uses the GPU to calculate the color of the pixels. With more pixels to calculate the time needed will increase. With a resolution of FHD, around two million pixels will be calculated while a UHD resolution has around eight
million. When combined with the rendering technique Physically Based Rendering (PBR) the individual shading cost per pixel will increase. PBR uses math to calculate light and color and does not scale well with higher resolutions. Unless one has a top of the line GPU a flawless gaming experience in UHD is very hard to accomplish.

Today several games and game engines, like Unreal engine and Unity support PBR and high-resolution screens [8][9]. With more game studios making the transition to PBR and more people changing to UHD screens and with the possibility of 8k resolution in the future the graphical processing demands will only increase.

With the improvements of screen resolutions and the increase of pixels the cost of rendering will only grow. To allow more people to play games with a high screen resolution, the rendering cost needs to either be lowered or the price of GPU’s need to dramatically decreased. To try and improve the rendering cost, this study will conduct an experiment to evaluate the effects of removing PBR’s roughness attribute in low-lit areas. Roughness will be removed in a way that will not affect the visually perceived quality of the image. By removing complex calculations from the PBR implementation, the calculations the GPU has to perform will be reduced and could improve the performance.

1.1 Aim and Objective

The aim of the study is to investigate at which darkness threshold PBR’s roughness attribute can be removed while having minimal effect on the graphical quality. Instead of using the standard PBR in low-lit areas, the areas qualified will be rendered without the roughness attribute. By removing roughness low-lit areas fewer calculations are performed on the GPU. The objectives of this research can be split into three,

- Research how players perceive a low-lit area.
- Create a test scene with different darkness thresholds and variations.
- Conduct an experiment on the visual perceived difference while alternating darkness thresholds.

To accomplish these steps there first has to be a screen with the same brightness settings during each experiment. By using the same screen and settings, it is possible for participants to experience the same colors and brightness. The brightness of the scene needs to be accurately measured in the same way for each scene to know when roughness should be removed from the equation.

Eight participants were invited to participate in the experiment. They were introduced to a scene rendered with full PBR and were asked to compare it with scenes rendered with different darkness thresholds. Lastly, the results of the experiment were analyzed. The result showed at which darkness threshold the participants perceived the least difference in graphical quality.
1.2 Research Questions and Hypothesis

This thesis has two relevant research questions.

- At which darkness threshold is the effect of roughness not contributing to the image?
- What is the visual perceived difference when removing Physically-Based Rendering roughness attribute, in low-lit areas?

To investigate the perceived difference when removing PBR roughness attribute in low-lit areas, it is important to find out what players perceive as a low-lit area. With help from the result of the first research question, it is then possible to remove roughness in those areas. The hypothesis for this study was that the difference when removing roughness from low-lit areas will not be noticeable. But this will only be true in areas with a low darkness threshold and will only cover a small range of pixels as seen in Figure 1.2.

![Figure 1.2: Scene rendered with PBR. The pixels under the threshold is in greyscale](image)

1.2.1 Risk and Limitations

For lack of time, the test will be conducted in Unity’s game engine. When using the Unity PBR implementation it will be hard to measure performance and make the implementation optimized. Unity does a lot of calculations and optimizations that are not clearly visible to the user. To limit the research, performance will not be measured, this study will instead focus on the how people perceive the image when alternating between two PBR implementations.

PBR includes many different methods and equations that are used when performing the shading, but this study will only limit the research to the roughness features. To better evaluate the effects of removing complexity from the PBR pipeline it is
preferred to only focus on one aspect of the pipeline instead of trying to improve multiple parts at once.

To reduce validity threats for the experiment, the stimuli shown to the eight participants were chosen in a randomized order with the exception of the first two scenes. By showing the scenes in a randomized order, counterbalancing measures were taken and should have minimal effect on the results. Each participant had to run a program that randomized the order of scenes before the experiment started.
Chapter 2

Background

This section will include relevant information for the reader to understand the experiment conducted and the reasoning behind some of the choices made.

2.1 Video Game Lighting

"A believable scene is based on the coherence and the correctness of the lightning: every object should receive lightning from its surrounding environment, reflect light with the right amount of intensity and have shadows" [16] Lights are an important aspect of how players perceive the game and the graphical quality, it can also be used to provoke reactions from the players. With lights, it is possible to capture a feeling and make the scene feel more realistic. By placing out lights in a scene the feeling of the picture can change, is the area scary or is it safe? Lights in games are something that is not remarkably obvious if done right but extremely noticeable if done wrong. Lights can be used to guide or distract players and can entice them to pay attention to an object or location.

Dark areas are an important part of a realistic environment. With the different screens and the option to adjust brightness, some games choose to go pitch black in shadows[18]. It is important to have coherent light during the development of a game. By having light units like Frostbite [16] it is easier for a light artist to create realistic light. Usually, lighting artist defines a reference light, like the sun. This reference can then be used as a numerical reference in form of intensity for the rest of the scene.

2.2 Physically-Based Rendering

In the real world, materials interacting with the light affects the appearance of the object. Based on the materials physical properties such as conductivity and absorption the final light is effected [16]. Physically based rendering is a collection of mathematical equations that calculates light based on real-world physics. Most implementations of PBR is slightly different from each other but to qualify as PBR, Bidirectional Reflectance Distribution function (BRDF) is required. BRDF defined by Fred Nicodemus [12] is a function used to calculate light reflectivity. After Nicodemus, newer models based on the original has been researched, like Cook-Torrance light model [4] and Burley’s diffuse model [2]DISNEY. Today most implementations are based on
Cook-Torrance material model and Burley’s model[16, 8]. Cook-Torrance is mostly used for Specular BRDF while Burley’s is used for diffuse BRDF.

2.2.1 Microfacets

One of the main theories in PBR is the Microfacet BRDF. Microfacet BRDF comes from the idea that no real surface is either smooth like a mirror, nor ideal Lambertian, meaning matte surface. Depending on the surface of the material on a microscopic level the light reflections and the amount of light absorbed is affected[11] The effect of roughness gives surfaces depending on the value, different reflection highlights. A rough surface has a dimmer highlight while a smooth surface has a sharper one as shown in Figure 2.1 and 2.3. The roughness value is mostly used in the diffuse calculation as seen in Figure 2.2.

![Figure 2.1: Different roughness values, from high to low. (right-left) acquired from [2]](image1)

\[
f_d = \frac{\text{baseColor}}{\pi} \left(1 + (F_{D90} - 1)(1 - \cos \theta_l)^5\right) \left(1 + (F_{D90} - 1)(1 - \cos \theta_v)^5\right)
\]

where
\[
F_{D90} = 0.5 + 2 \cos \theta_r^2 \text{roughness}
\]

![Figure 2.2: Burley’s diffuse term acquired from [16]](image2)

![Figure 2.3: Shows how light interact with the surface microfacets and effect the roughness acquired from [16]](image3)
2.2.2 Subsurface scattering

Subsurface scattering is techniques that calculate how much light is refracted after absorption. In metals, the refracted light is absorbed while in non-metals refracted light scatters until it re-emerges. The light hits the object at one point and after interacting with the material properties of the object it will exit at a different point[11].

2.2.3 Energy conservation

Energy conversion explains the limit of outgoing light. A surface can not reflect more light than received[16].

2.3 Brightness Calculation

If three different colors red, green and blue are shown with the same radiance they will not appear equally bright. According to Charles Poynton [15] of the three, the luminous efficiency function peaks in the green region of the spectrum. Green will be perceived as brightest while, the blue will be the darkest with red in the middle. Two common methods to calculate if a color is bright is Hue, Saturation, and Brightness (HSB) and Hue, Lightness and Saturation (HLS). When computing brightness with HSB and HLS the most common formulation is \((R+G+B)/3\). According to Charles Poynton, this computation conflicts badly with the properties of color vision. The result will compute yellow to be about six times more intense than blue using the same brightness value as seen in Figure 2.4.

Hue, Saturation Perceived brightness (HSP) is a lesser known method created by Darel Rex Finley [6]. The HSP formula adds three constants (.299, .587, and .144) representing the different degrees the different colors affect the brightness perception of the color.

Figure 2.4: Compares HSV, HSL and HSP, *acquired from [6]*
Chapter 3

Related Work

The area of PBR and light culling methods is a popular research topic. One of the most performance-demanding steps in real-time rendering is the light calculations. Forward+ by Carlos Andujar and Enrico Puppo is a method of rendering that supports a large number of lights [7]. The lights are culled and only the lights that contribute to the pixel are saved. Forward+ reduces memory traffic compared to deferred lighting and also removes many drawbacks of deferred rendering which restricts materials and lighting models.

Previous research of culling none contributing lights and shadows also include Bittner et al [1]. Bittner et al introduce a method for efficient construction of shadow maps by culling shadow caster which does not contribute to visible shadows. When implemented the method speeds up rendering between 1.5x-2x for actual game scenes. PBR has during the last decade started to gain popularity and has been researched and improved. Burly released a paper 2012 [2] improving on the Lambertian material model while in Sebastien Lagarde and Charles de Rousiers presentation on moving Frostbite to PBR Burly’s method was slightly modified to improve energy conservation [16].

The technique Adaptive approXimate Anti-Aliasing (AXAA) [10] increases the performance of the technique by setting the search range of pixels according to its luma contrast. Observing that is is hard to distinguish the result between narrow and wide search ranges Nah et al limit the search iteration according to luma contrast to improve performance.
Chapter 4

Method

To answer the research questions an experiment was conducted in Unity’s game engine using the free Sponza scene [5]. Eight participants were invited using word of mouth, to partake in a user study and were shown stimuli rendered with the two different PBR implementations. The user experience then gathered through a survey and was recorded for further analysis.

4.1 Design

To design the test the following requirements needed to be fulfilled as to strengthen its validity and results.

1. Easily recreated
2. Counterbalancing measures
3. Making sure the participants recognize PBR

The first requirement of being easily recreated supports the choice of using the free Sponza scene to conduct the test. The Sponza scene is easily accessible and can be acquired from [5] with PBR textures acquired from [14]. With a low threshold on participants, the results will not be as broad as if having a larger amount but is sufficient to answer the research questions.

The second requirement is required to strengthen the results and minimize deviations. The participants were exposed to four different stimuli in a randomized order for each participant, not knowing what darkness threshold is being used. Three reference scenes were also shown. The reference scenes were not shown in a random order and were made known to the participants.

To fulfill the third requirement the participants were exposed to two reference stimuli containing two variations of a lit scene as seen in Figure 4.1 and 4.2. One variation rendered with Unity’s full PBR implementation and the other rendered with the without roughness. The participants were asked to answer if they noticed any difference between the two. The stimuli were shown one after the other and not in side by side comparison. While not having a side by side comparison it will become harder for the participant to notice any difference, the scene will either have to be scaled down or only half of the scene can be shown. This will affect the overall impression of the scene and may affect the results.
4.2 Procedure

The experiment was performed in a controlled environment, with a borrowed computer and screen from Blekinge Institute of Technology as seen in Figure 4.3. Each participant was introduced to the experiment by reading a paper informing them of the aim of the study. The participants were later asked to sign a waiver confirming their participation to be voluntary. The participants were first introduced to a brightly lit scene fully rendered with Unity’s own PBR implementation and after, the same scene rendered fully with the low complexity PBR implementation. After viewing both Unity’s normal PBR rendering and low complexity PBR rendering, the participants were exposed to a low-lit scene rendered with Unity’s PBR implementation. The scene was later used as a reference point.
4.3 Ethics

The later stimuli were shown in a randomized order rendered with different darkness thresholds. The participant was given a survey to fill out after each stimuli answering if they noticed any difference from the reference. After viewing every stimulus and filling out the survey they are free to ask questions about the experiment. The survey contains forced choice questions for each of the stimuli. The question asked was if the participant noticed any difference between the stimuli shown and the reference. When the participant was done they are thanked for participating and asked verbally not to reveal the contents of the experiment.

![Figure 4.3: Picture of one of the participants participating in the study](image)

4.3 Ethics

To minimize any risk for the participants the stimuli was created based on the following rules. First no sudden camera movement. The camera used slowly flew through the scene at a slow pace and had no sudden changes in direction. Secondly, no sudden color or mesh changes. The mesh in the stimuli was static and did not move, similarly, the color and texture of each mesh did not change in any way. Thirdly the participants could quit at any time. The experiment was voluntary and if the participant chooses to quit during the experiment no explanation was needed and they were free to do so at any time.
Chapter 4. Method

4.4 Unity Game Engine PBR Implementation

Unity’s PBR implementation is derived from Disney’s work [2] seen in Figure 4.4 and based on Torrance-Sparrow micro-facet model [4]. The changes made to the already existing PBR pipeline was mostly related to the roughness value. The low complexity implementation ignores the roughness calculation and removes it out of the pipeline. By removing the roughness value in Unity’s PBR implementation six multiplications can be saved for each pixel. Disney’s diffuse uses roughness to calculate scattering of light and the amount of energy preserved by the material as seen in Figure 4.5. The roughness is also removed from the specular BRDF in both the GGX term and the Smith Joint GGXV Visibility Term. Unity converts roughness to smoothness but this does not affect the result of the final calculations for the low complexity implementation.

\[
f_d = \frac{D}{\pi}(1 + F_{90}(1 - (n \cdot l))^6)(1 + F_{90}(1 - (n \cdot v))^6) \text{ where } F_{90} = 0.5 + \cos(\theta_d)^2\alpha
\]

Figure 4.4: Roughness is the alpha variable. acquired from [16]

```cpp
half DisneyDiffuse(half NdotV, half NdotL, half LdotH, half perceptualRoughness)
{
    half fd50 = 0.5 + 2 * LdotH * LdotH * perceptualRoughness;
    // Two schlick fresnel term
    half lightScatter = (l + (fd50 - l) * Pow5(1 - NdotL));
    half viewScatter = (l + (fd50 - l) * Pow5(1 - NdotV));

    return lightScatter * viewScatter;
}
```

```cpp
half DisneyDiffuse(half NdotV, half NdotL, half LdotH,)
{
    half fd50 = 0.5 + 2 * LdotH * LdotH;
    // Two schlick fresnel term
    half lightScatter = (l + (fd50 - l) * Pow5(1 - NdotL));
    half viewScatter = (l + (fd50 - l) * Pow5(1 - NdotV));

    return lightScatter * viewScatter;
}
```

Figure 4.5: Shows the implementation for, upper with roughness and lower without roughness

```cpp
BrightnessForCurrentPixel = sqrt((Diffuse.x * 255) * Diffuse.x * 255) + 0.241 + (Diffuse.y * 255) * (Diffuse.y * 255) + 0.691 + (Diffuse.z * 255) + (Diffuse.z * 255) + 0.068;
```

Figure 4.6: Function approximating brightness base on diffuse light model
The low complexity equations ran for every pixel with brightness under the given threshold as seen in Figure 4.8 and 4.7. The color was approximated using a diffuse shading model. The result was later processed with HSP to approximate the brightness of the pixel as seen in Figure 4.6. The lower the threshold the darker the pixel has to be for it to be allowed to be rendered using the low complexity PBR implementation. HSP use the range of 0-255, in this implementation, the value was converted to 0-1 range. The thresholds used during the experiment was 0.01, 0.03, 0.05 and 0.07. These thresholds were chosen after experimenting with different ranges. If the threshold used was higher the range of the thresholds would include lit areas as well.

Figure 4.7: Shows the area without roughness in color and rest in greyscale with threshold at 0.07

4.5  Set-Up

The screen and computer used in the user study had the following specifications. Screen: Dell P2715q 27 inch 4k screen. Screen brightness 75, contrast 75 and sharpness 50. The computer specifications were Intel i7-6700k with 16 GB of RAM and Nvidia GeForce GTX 980. The computer specifications should not affect the final image shown.
Figure 4.8: Shows the area without roughness in color and rest in greyscale with threshold at 0.01
Chapter 5

Results

The result of the study can be seen in Figure 5.1 and 5.2. The first part of the study when participants were asked to compare two stimuli, one rendered with Unity’s PBR and the other, fully rendered with low complexity PBR equations can be seen in Figure 5.1. Out of eight participants, five noticed a difference while three did not see any difference.

Figure 5.1: Result of the lit comparison study
The majority of participants noticed a difference when comparing the lit stimuli. Seven participants noticed a difference with the threshold at 0.01 and 0.03 with only one person not perceiving any difference. The higher thresholds at 0.05 and 0.07 had two participants not perceiving any difference. One of the participants saw no difference in any of the pictures while another participant saw no difference in three of the comparisons. One of the participants did not notice any difference in the lit scene but when comparing the low-lit scenes the answered that they noticed a difference.

The effect of removing roughness from the low-lit areas had two major consequences. The first was the areas with high roughness value, like the symbol on the curtains, these areas were calculated without a roughness value which resulted in the symbol not reflecting light properly. This can be seen as the symbol is darker and the light is absorbed rather than reflected. Secondly, the whole scene appears much darker than the original. By removing roughness from the energy conservation equation and the sub-surface scattering the light did not travel through the scene as it should causing the scene to appear darker than the original.
Chapter 6

Discussion

The majority of participants noticed a difference between the different scenes, with only one participant not seeing any difference in any of the stimuli. This could be caused by many different reasons, the participant could have been stressed to finish faster or could have misunderstood the instructions and focused on the wrong aspects of the stimuli. The other two participants that answered no difference in one or three questions respectively could be caused by learning to spot the difference better during the experiment. The first stimuli compare a lit scene had three participants not noticing any difference. These stimuli were shown first and the possibility of the participants getting better at noticing the effect of roughness during the duration of the experiment is a possibility. The low-lit stimuli were shown in a randomized order while the lit stimuli were always shown first which can contribute to three participants not noticing any difference.

The first theory about not calculating roughness was that the only areas to be affected would be the portions of the scene with high roughness value. The biggest difference as shown in Figure 6.1-6.5 was the entire scene being darker. Roughness is used to calculate energy conversion and sub-surface scattering. The result of having incorrect energy conversion and sub-surface scattering could be the reason the scene appear darker. In the areas where roughness is removed the material seems to absorb too much light the result being, while having the same light source the scene with missing roughness values results in a darker picture the original.

Also noticeable is the lower thresholds resembling the original more than the scenes with the higher threshold, but when compared to the reference the difference is still noticeable. The first research question, At which darkness threshold is the effect of roughness not contributing to the image, can be split into two parts. The first as seen in the pictures are the areas with high roughness values like the symbol on the curtain. Second is the effect of energy conservation that makes the scene appear much darker than the reference. In the answers from the experiments, the majority of participants noticed a difference in the stimuli which can be seen as roughness always contributing. The most noticeable difference between the low complexity PBR and Unity’s PBR is the effect of the overall brightness of the scene and not the areas with high roughness values, which also answered the second research question. If roughness is removed in low-lit areas it will be perceived as even darker then originally intended. If the energy conversion and sub-surface scattering are correctly approximated the difference might be less noticeable.
Chapter 6. Discussion

The results were gathered with a user study instead of using a method like structured similarity index (SSIM) [19]. While in some cases SSIM might be preferred to avoid the human element that is prone to error it is not suitable for this study. The study focuses on the noticeable difference rather then if there is a difference. The stimuli compared to the reference contains a difference, the study researchers if this difference is noticeable in low-lit environments.

![Reference scene to be compared to, rendered with full Unity PBR](image)

Figure 6.1: Reference scene to be compared to, rendered with full Unity PBR

One of the larger limitations of the study was not accounting for energy conversion. With incorrect energy conversion, the material absorbs too much light and the scene appears much darker than intended. This could be countered by adding more light to the affected area to balance it out. The real world application for this study is to save performance. The performance saved is unknown and needs to be measured. Using HSP is not cheap and by checking each pixel the performance could be less than using the Unity’s PBR depending on the implementation. When applying this in an uncontrolled environment the settings used on the screen can affect the perceived difference. If the scene has a high gamma value the whole scene will appear brighter, this might not represent what the artist intended and an area designed as dark could appear bright instead. The survey the participants answered only asks if they perceive any difference. The survey could be more thorough and include questions about what difference they noticed, to conclude if the energy conversion was the most apparent or if the area with high roughness value more noticeable. More stimuli could be shown with more lights to get more information about the difference in low-lit areas vs lit areas to research the difference further.
Figure 6.2: Low lit scene rendered with threshold 0.001

Figure 6.3: Low lit scene rendered with threshold 0.003
Chapter 6. Discussion

Figure 6.4: Low lit scene rendered with threshold 0.005

Figure 6.5: Low lit scene rendered with threshold 0.007
Conclusions and Future Work

The question *At which darkness threshold is the effect of roughness not contributing to the image?* is something that can not be answered from the results of this study. At every darkness threshold participants in the study noticed a difference, which can be seen as roughness always contributing to the image. The second question *What is the visual perceived difference when removing Physically-based Rendering roughness attribute, in low-lit areas?* is when removing roughness from the PBR implementation there were two noticeable differences. The first was the area with high roughness value being affected and not reflecting light properly. The second effect is the scene becoming darker than the original. The missing roughness value when calculating energy conversion and sub-surface scattering will make the material absorb too much light while not reflecting it properly, creating a visual difference where the scene appears much darker.

In the future, the study needs to be more precise and research the effect of removing roughness from energy conservation, and sub-surface scattering separately. By doing this it will be easier to pinpoint what effect roughness has in the equations. The study could be extended to evaluate the difference in lit areas. This study is focused on the roughness attribute but in the future other parts of the PBR pipeline can be removed and research the effect in low-lit areas. The idea of having cheaper rendering in low-lit areas should be researched more.

This research could also benefit from testing different materials with higher or lower roughness values in different environments. It is also possible to research if using diffuse shading in low-lit areas and alternate it with PBR is noticeable. This study uses HSP to calculate what areas are qualified to be rendered using the roughness free implementation. HSP is not cheap and may create artifacts and a CPU based sorting of low-lit areas could be researched to both improve the light approximation and the performance of the implementation or improve on the already existing HSP implementations. Performance should then be measured to compare the improvement. There is a possibility of testing using low-poly meshes in dark areas and create a user study measuring the perceived difference and research if it is noticeable.
References


