A Novel Approach for Reflectance Capture

By Kasper Skou Ladefoged

AALBORG UNIVERSITY
STUDENT REPORT

VGIS 10
SEPTEMBER 2016

GROUP
1088

AALBORG UNIVERSITY
STUDENT REPORT

Department of Electronic Systems
Fredrik Bajers Vej 7
DK-9220 Aalborg Ø
http://es.aau.dk

Title:
A Novel Approach for Reflectance Capture

Theme:
Computer Vision

Project Period:
Fall/Spring Semester 2015/2016

Project Group:
1044, 10th Semester

Author:

Kasper Skou Ladefoged

Supervisors:
Claus B. Madsen

Number of Pages:
67

Date for Submission:
September 09, 2016

Abstract:

Capturing the reflectance of an reconstructed object is of use in many area of computer graphics. Currently the only way to do this is using big, one of a kind, static setup. This limits the accessibility of reconstructed objects. Creating a more novel approach with, off the shelf hardware, would increase the presence of real world object in augmented reality, virtual reality etc. This project strives to achieves this, by capturing multiple images at the same locations in order to isolate a known light source, and then reverse engineer the physical object reflectance. This creates some guidelines in the capturing process, in order to ensure all the variables are inside a given range, in order to limit errors. The precision of the results primarily depends on the precision of the calibration of the flash, and the reconstruction of the physical object. In order to limit the error, a process for calibrating a light source is presented. Furthermore the reconstruction of the physical object is done with the use of ContextCapture, as this creates very high quality reconstructions, while presenting the cameras estimated positions and rotations in relation to the reconstruction. The acceptance test shows that the error of a synthetic test, has a mean error of -0.0816. Given a range for the pixel values of 0 to 1. The shadows produced by unknown light are removed, but a series of bright spots are introduced in the real world test. This has to be investigated further, but seems very promising.

Preface

This project is written as the master thesis in Vision, Graphics and Interactive Systems at Aalborg University. It is a long thesis and is rated at 50 ECTS. The target group of this report are supervisors and future developers such as students and other interested parties. The central interest is to create a novel approach to recreate the reflectance of an reconstructed physical object without the need of highly specialised and static equipment.

A special thanks goes to my girlfriend, whom have been immensely patient and understanding doing the process.

Reading Guide

The report is structured in four parts. In each part different aspects of the project will be analysed.

I Preliminary Analysis
II Implementation
III Evaluation
IV Appendix

The Preliminary Analysis describes the problem and investigates theories used to solve it. The Implementation part deals with the actual implementation of the system while the Evaluation tests the system and concludes on the project via the predefined requirements.

Appendix

Appendices are found after the main report as part IV. This includes supporting theory that is not necessary for the overall understanding.

All figures, tables and equations are referred to by the number of the chapter they are used in, followed by a number indicating the number of figure, table or equation in the specific chapter. Hence each figure has a unique number, which is also printed at the bottom of the figure along with a caption explaining the content. The same applies to tables and equations, the latter of which have no captions. Appendices are referred to by capital letters instead of chapter numbers.

Bibliography

At the very end of the main report, a Bibliography is listed which contains all sources of information used in the report. In the Bibliography books are indicated with author, title, publisher and year. Web pages are indicated with author and title. All information sources are referred to by the authors' name and the year of publish for example (Dutré et al., 2006)

Contents

Part I

Preliminary Analysis

Chapter 1

Introduction

Today, the world of computer graphics and the real world are continuously getting more intertwined because of technological advances and the general consumers interest herein. This can be seen in the increased popularity of augmented reality games and equipment. Virtual reality has also been having a big surge in interest from consumers.

To fuse these two world together, the real world and the computer generated world, an increasing amount of solutions are present, to bring an object from the virtual world in to the real world, noticeable with 3D printing. Furthermore solution to bringing real world object in to the virtual world is increasing in need as the technology progresses. A example of doing this is 123D Catch (Autodesk, 2016). It works by having the user take a series of images around a giving object of interest. The application then recreates the object, trough structure from motion, to a 3D model, ready for use in games and other applications. There is some problem with this solution though. When recreating the object, the resulting 3D models appearance, is already affected by light from the world around the object of interest. This can create sharp and noticeable shadows on the recreated model. Taking this model in to an computer simulated environment with a new lighting setup, the model will be affected by a new lighting setup, creating new shadow etc. Because of this, the models shadows etc. will look wrong, making the model look out of place in the environment.

To solve this, the light from the real world environment has to be removed from the reconstructed model. The state of the art at the time of writing is the Lightstage.

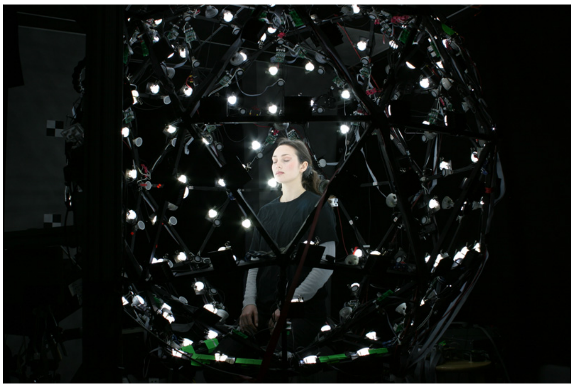


Figure 1.1: The lightstage.

- **Lightstage 5**(Alexander et al., 2010)
  Lightstage is an elaborate setup that aim at controller all light when capturing images. The setup is capable of capturing the geometry, diffuse reflectance, and the specular intensity map. Furthermore it uses a normal map, generated from the specular reflections, as these does not subsurface scatter, in order to augment the geometry created from the diffuse reflections. By doing this, smaller details are captured like skin pores etc. The main problem with this solution is that the rig is very static and cumbersome because of the aim to

control everything. This also limits the use for other individuals with out this special purpose equipment.

With this, the main problem is the static nature of the current solution.

The aim is therefore to create a method that can reconstruct the reflectance, without the need of specialised static equipment, to enable a broader audience to use physical object in three dimensional computer simulations.

To solve this problem, fundamentals of how light travels are used in order to reverse engineer this process. This enables the calculation of the original reflectance of the object. To do this, a known light source needs to be isolated in order to know the lighting conditions. This is paired with a simulation of the known light to create the illumination on the object. To do this simulations, an reconstruction of the physical object needs to be created. With this, the reflectance can be recreated as the lighting applied can be reversed.

The first step in the process is to acquire the images needed. The main purpose of these is to isolate a known light, paired with the use of recreating the physical object as an 3D model. A series of images are capture around the object, capturing two images at every position, one with the known light source turned off, and one where it is turned on. To isolate the known light source, the two images, captured from the same position, can simple be

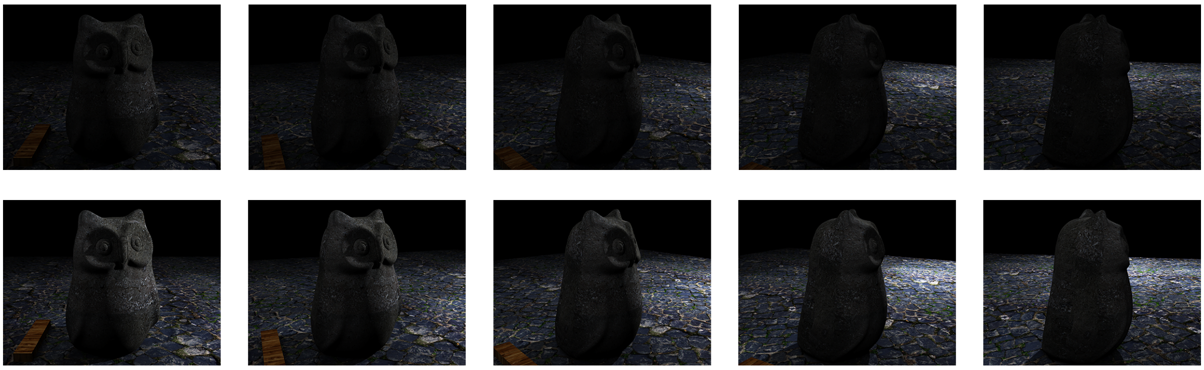


Figure 1.2: An example of the images captured.

subtracted, resulting in only the light from the known light source remaining. This is possible as light is additive, meaning that if an extra light is added to an environment, it is simply added on top of the already present light. The resulting image is in principle the same as if no other light source was present at the time of acquisition. Because of this, the same results would be capture, if an image was acquired in a completely dark room, with only the known light source.

Figure 1.3: An example of the isolation of the known light of the images above.

In order to actually have a known light, the chosen light source for the purpose has to be correlated to a simulation, in order to reverse the light added. In order to create the correlation, a surface of a known reflectance at a known distance is needed. With this, and the settings from the camera, a similar simulated setup can be created. With the simulation in place, a light source can be calibrated to the same intensity and light patter as the physical light source.

In order to progress with the calculation of the reflectance, the physical object needs to be reconstructed. This is done using structure from motions. The features on the physical object can be tracked in-between the images. These features can be triangulates and then a mesh can be estimated from these points on the object surface. The result is a reconstruction of the physical object that can be used to estimate the illumination of this object. Together with the reconstructed model, the positions and rotations of the images, or cameras, is also estimated in relation

Figure 1.4: The image that the flash is calibrated from. The big white square to the right, is a piece of paper.

to the reconstructed model. This information is used later, in order to ensure the simulated irradiance fits with the original images.

Figure 1.5: A rendered image of the reconstruction with a diffuse white material.

The next step is to simulate the objects irradiance. This is done using the reconstructed model together with the calibrated flash. The position of the camera, and other information, is exported from the reconstruction of the model. The reflectance of the model, is set to one to have a known reflectance in order to isolate the unknown reflectance. This is discussed further in section Chapter 2: Reflectance. The resulting images generated is the irradiance of the object at every pixel of the images only containing the known light.

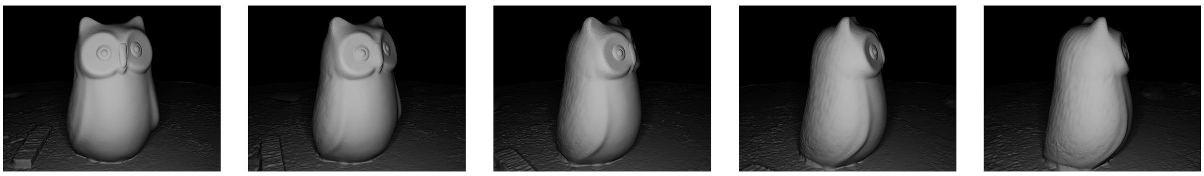


Figure 1.6: The simulated irradiance.

With the irradiance calculated, the reflectance can be recovered using the inverse of the light interacting with the object. This results in the reflectance of the object without any form of light applied. The object is ready to be used in many different applications, for example movies, games, and augmented environments.

For this solution to work properly, the two most important factors are the calibration of the light, and the reconstruction of the object. This have a big impact on the final result.

Further more the solution is limited to diffuse surfaces. This is because the main part of most objects, are the diffuse reflections. Diffuse reflections are also a requirement in order to reconstruct the physical object. Specular highlights could late be done, but is out of the scope of this project.

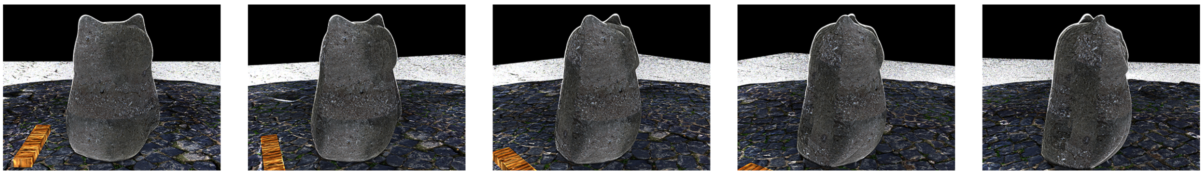


Figure 1.7: The calculated reflectance.

In order to understand the limitations and create the best results, the underlying theory needs to be understood. This will in term enable the reader to make the right decision, especially at the time of acquiring the images.

This process enables a larger group of people to capture and reconstruct 3D models of real world object with the actual appearance of the object, ready for use in other 3D applications. This approach could possible be extended to be used on phones, as there is a known camera, at a very specific position in relation to the camera, paired with the ability to control the flash on/off.

Figure 1.8: A visual render of object with different textures. The HDRI is Grace Cathedral, San Francisco.
CL is the Calculated Reflectance. Org is the original model. Dif is a diffuse surface on the reconstruction. Rec is the normal reconstruction. GT is the Ground truth.

Chapter 2

Reflectance

In order to recreate the reflectance of a given object, the way light interacts with the physical world needs to be investigated. This is the basis to enable the reconstruction and backward calculations to find a given objects reflectance. This interaction between an object and a light source can be investigated by looking into the Bidirectional Reflectance Distribution Functions, abbreviated BRDF, and supporting theory. The BRDF covers the light coming from a light source, interacting with an surface, and leaving in the direction of some receptor, in this case a camera. The outcome is given in pixel value. The step from light leaving the surface to a digital image will be expanded upon in Chapter 3: Camera Theory.

2.1 Bidirectional Reflectance Distribution Function

The first point of interest is how an object is perceived by some sort of observer. To explain this interaction the Bidirectional Reflectance Distribution Functions (BRDF) is used. The BRDF defines how light is reflected by an opaque surface. The BRDF is defined in the following way:

$$f_r(x, \Psi \rightarrow \Theta) = \frac{dL(x \rightarrow \Theta)}{dE(x \leftarrow \Psi)} \tag{2.1}$$

Where x is the surface that is being examined, Θ is the incoming light direction, Ψ is the reflected light direction, $f_r(x, \Psi \rightarrow \Theta)$ is the BRDF which is a four dimension function for each point, given in steradian (sr), $dL(x \rightarrow \Theta)$ is the radiance in direction Θ from the point x given in power per steradian per unit meter($W/m^2 \cdot sr$), and $dE(x \leftarrow \Psi)$ is the irradiance incoming at point x from the direction Ψ, given in power per unit area(W/m^2). This is illustrated in Figure 2.1. Looking at Equation 2.1 it can be seen that the BRDF is actually a factor between the incoming irradiance and the outgoing radiance from the surface x. This factor is then dictated by the BRDF of the objects surface.

Because the reconstruction of the leaving light is of interest, and the BRDF of the surface is defined as a lambertion diffuse surface, the equation is rearranged to define the leaving light instead of the BRDF. Because of the definition of irradiance, power per unit area, the sum of the energy affecting the point x is needed. This is done by integrating over the top hemisphere, in the direction the normal vector of the point x. This is notated as $\int_{\Omega x}$. The final rearranged equation to define the total reflected radiance is as follows:

$$L(x \rightarrow \Theta) = \int_{\Omega x} f_r(x, \Psi \rightarrow \Theta) dE(x \leftarrow \Psi) \tag{2.2}$$

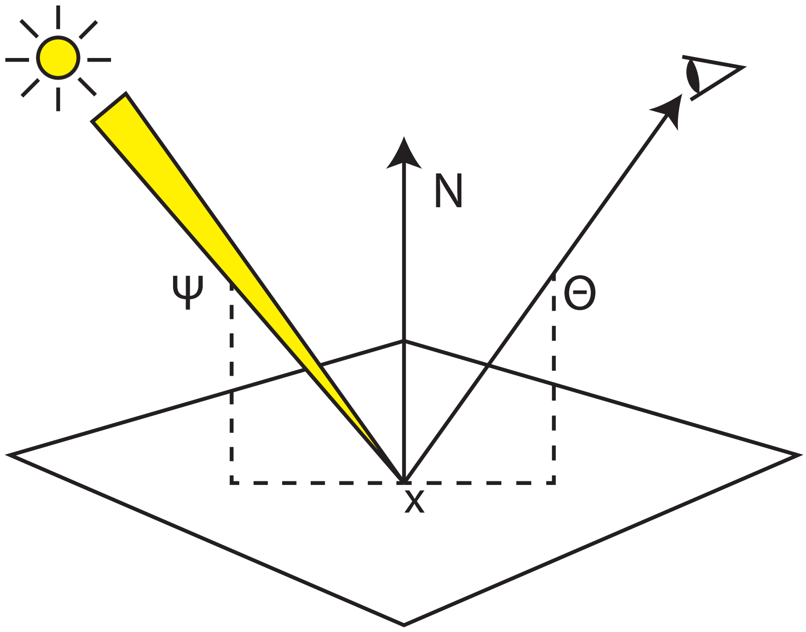


Figure 2.1: An illustration of the parameters present in the BRDF equation.

2.2 Diffuse Surface

To limit the view of the report, the BRDF of the surface is defined as an Lambertian diffuse surface. The reason for the limitation to diffuse surfaces is, that almost all material has some part of diffuse reflections. Furthermore diffuse surfaces is a requirement for image based modelling.

In order to define the BRDF of a Lambertian surface, the reflectance needs to be define. The reflectance of a point is defined by the fraction of reflected radiant energy of the incoming radiant energy. The reflectance of a surface is given in the fraction of light exiting the surface over the light incident on the surface (Dutré et al., 2006). The reflectance of a point, and the an area is define in the following way:

$$\begin{aligned} \rho &= \frac{d\Phi_r}{d\Phi_i} \qquad\qquad \rho, \rho_\Omega \in [0;1] \\ \rho_\Omega &= \frac{B}{E} \end{aligned} \tag{2.3}$$

Where ρ is the reflectance of a point. $d\Phi_r$ is the total reflected radiant energy. $d\Phi_i$ is the incident radiant energy. ρ_Ω is the reflectance of an area. B is the radiance of the area. E is the irradiance of the given area. The exiting amount of light is defined by Lambert's cosine law. The cosine law states that the bigger the angle between the light source and the surface normal, the lower the irradiance of the surface will be. This is because the same amount of light is projected at a larger area. This can be defined in the following way for the exiting light:

$$\begin{aligned} &B = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} L(x \to \Theta) \cos\Theta \sin\Theta d\Theta d\varphi \\ &\Updownarrow \\ &B = \pi \cdot L(x \to \Theta) \end{aligned} \tag{2.4}$$

The BRDF is moved outside of the integration as this becomes an constant because of the view independent nature of diffuse surfaces. Looking at Equation 2.2, the BRDF can then be redefined in therms of E giving the incident radiant power. With these changes, the definition of E is as follows:

$$\begin{aligned} L(x \to \Theta) &= f_r(x, \Psi \to \Theta) \int_{\Omega x} dE(x \leftarrow \Psi) \\ &= f_r(x, \Psi \to \Theta) \cdot E \\ &\Updownarrow \\ E &= L(x \to \Theta) / f_r(x, \Psi \to \Theta) \end{aligned} \tag{2.5}$$

With the definitions of B and E, Equation 2.3 can be expanded with these. This equation can then be reduced to a form where the reflectance is defined in terms of the BRDF and π.

$$\begin{aligned} \rho_\Omega &= \frac{B}{E} \\ &= \frac{\pi \cdot L(x \to \Theta)}{L(x \to \Theta) / f_r(x, \Psi \to \Theta)} \\ &= \pi \cdot f_r(x, \Psi \to \Theta) \end{aligned} \tag{2.6}$$

This definition of ρ_Ω can be solved for the BRDF, giving the BRDF of a Lambertian diffuse surface. The BRDF for a Lambertian surface is as follows:

$$f_r(x, \Psi \leftrightarrow \Theta) = \frac{\rho_\Omega}{\pi} \tag{2.7}$$

Where ρ_Ω is the reflectance of the area x.

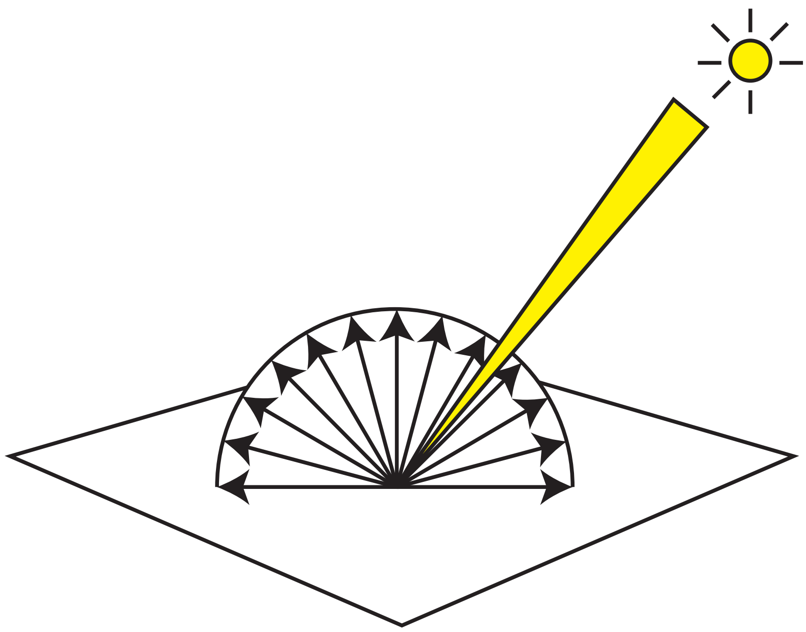


Figure 2.2: A figure illustrating the equal emittance of radians in all directions.

2.3 Pixel Value

With the BRDF describing the interaction between a surface and the incoming light, an equation from the outgoing light to a pixel value needs to be defined. The definition is described in a way to simplify the interaction with the observer, usually a camera, to the point of only a scaling factor. The pixel value, P, is defined as follows:

$$P = S \cdot L(x \to \Theta) \tag{2.8}$$

Where P is the pixel value, S is a camera scaling factor and $L(x \to \Theta)$ is the radiance in direction Θ from point x. The scaling factor is the factor that were used to scale the radiance to a value below the maximum value the camera can capture. In the camera this is controlled by the aperture, shutter speed, and ISO.

With the definition of the pixel value defined, it is expanded by substituting Equation 2.2 in to Equation 2.8 giving:

$$P = S \cdot \int_{\Omega x} f_r(x, \Psi \to \Theta) dE(x \leftarrow \Psi) \tag{2.9}$$

2.4 Reflectance Reconstruction

With the definition for a given pixel, the focus can shift towards the solution of the formulated problem. In order to reconstruct the reflectance of an object, the isolation of a known light source is necessary for later reverse engineering.

In order to isolate a known light source, the basic property of light being additive can be exploited. This creates a situation where two measurements of light can be captured, one with the unknown light and one with known and unknown light. These two measurements can then be subtracted from each other. This results in thee know light being left, Equation 2.10.

$$L(x \to \Theta)_k = L(x \to \Theta)_{k+u} - L(x \to \Theta)_u \tag{2.10}$$

Where the $L(x \to \Theta)_{k+u}$ is the image containing both the unknown environment light and the light from the know light source. And $L(x \to \Theta)_u$ is the image with only the unknown light from the environment. The subscripts for known and unknown light will be denoted as, $_k$ for known light and $_u$ for unknown light.

This creates a solution by capture two images. One where the unknown light from the environment is the only light present. The other image, captured from the exact same position, but with a known light source together with the unknown light from the environment. The image with only the unknown light can then be subtracted from the one containing both known and unknown light, and the known light is then the only light left in the resulting image. If Equation 2.10 is defined in respect to pixels, a scaling factor is added for the two pixel values, in order to account for capturing with different exposure settings. This definition of this is as follows:

$$P_k = \frac{S_u}{S_{k+u}} \cdot P_{k+u} - S_u \cdot P_u \tag{2.11}$$

Where P_{k+u} is the pixel affected by both known and unknown light. P_u is the pixel only affected by only unknown light. S_{k+u} is the scaling factor for the pixel with known and unknown light. S_u is the scaling factor for the pixel only affected by unknown light. P_k is the resulting pixel only affected by the known light.

With an isolated light source, the isolation of ρ_Ω is needed. In order to do this, two images is used, where all parameters is constant, with the exception of ρ_Ω. In an simulated image ρ_Ω is set to unity. The calibrated simulated light source is added. The image is rendered with a known radiance for each pixel is given. This isolation is defined as follows:

$$\frac{P_k}{P_s} = \frac{S \cdot \int_{\Omega x} \rho_\Omega \cdot \frac{1}{\pi} \cdot dE(x \leftarrow \Psi)}{S \cdot \int_{\Omega x} 1 \cdot \frac{1}{\pi} \cdot dE(x \leftarrow \Psi)} = \rho_\Omega \tag{2.12}$$

Where P_k is the pixel only affected by known light, P_s is the simulates image, S is the scaling factor, ρ_Ω is unknown the reflectance, and $dE(x \leftarrow \Psi)$ is the irradiance.

The equation is extended in order to include different scaling factors of the two images. These two different scaling factors can then create a factor that is multiplied to ρ_Ω. The new definition for the equations is then:

$$\frac{P_k}{P_s} = \frac{S_k \cdot \int_{\Omega x} \rho_\Omega \cdot \frac{1}{\pi} \cdot dE(x \leftarrow \Psi)}{S_s \cdot \int_{\Omega x} 1 \cdot \frac{1}{\pi} \cdot dE(x \leftarrow \Psi)} = \frac{S_k}{S_s} \cdot \rho_\Omega \tag{2.13}$$

Because the image with the known reflectance is simulated, the object needs to be reconstructed. The reconstructed model needs to be close to the real world object in order to minimize artefacts.

With a known light, and the simulation of the object with a known reflectance, it is possible to isolate and reconstruct the unknown reflectance of the object at a given pixel.

2.5 Proof Of Concept

To prove that the presented theory and proposed solution works, a synthetic test is conducted. This test aims to prove the theory and solution, but also to reveal any unforeseen problems before the more elaborate acceptance tests, using real world images, is conducted. This test should, because of the low amount of variables present, show very basic problems and errors that might be hard to locate in the later tests.

2.5.1 Test Setup

The test scene containes a simulated physical camera, two area light sources for ambient light, and a simulated flash. The simulated flash is a ring flash locked to the cameras rotation and translation. The setup can be seen in Figure 2.3.

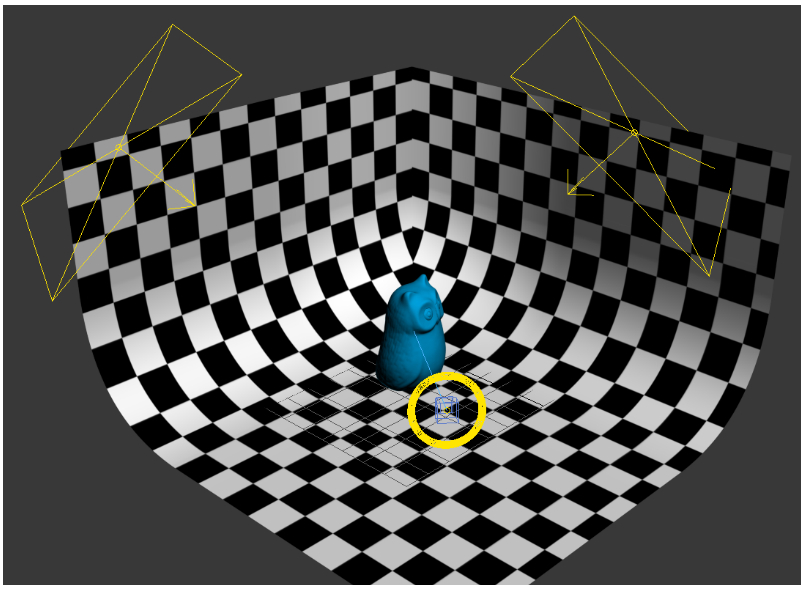


Figure 2.3: The test setup for the prof of concept test. The camera is in the centre of the ring light.

	Length	**Width**	**Radius 1**	**Radius 2**	**Intensity**(Lumen)
Ambient light 1	774.5	895.6			250
Ambient light 2	774.5	895.6			250
Flash			100	10	4000
	Focal Length	**f-number**	**Shutter Speed**(Seconds)	**ISO**	
Camera	45.5	3.2	1/25	200	

Table 2.1: The parameters of the light sources and the camera in the test scene.

2.5.2 Test Data

To test the proposed theory, the synthetic equivalents of the real world images was created. The first two images was be captured by a camera, and the last image was be simulated in an real world, use case scenario. In this test all the images was synthetically created.

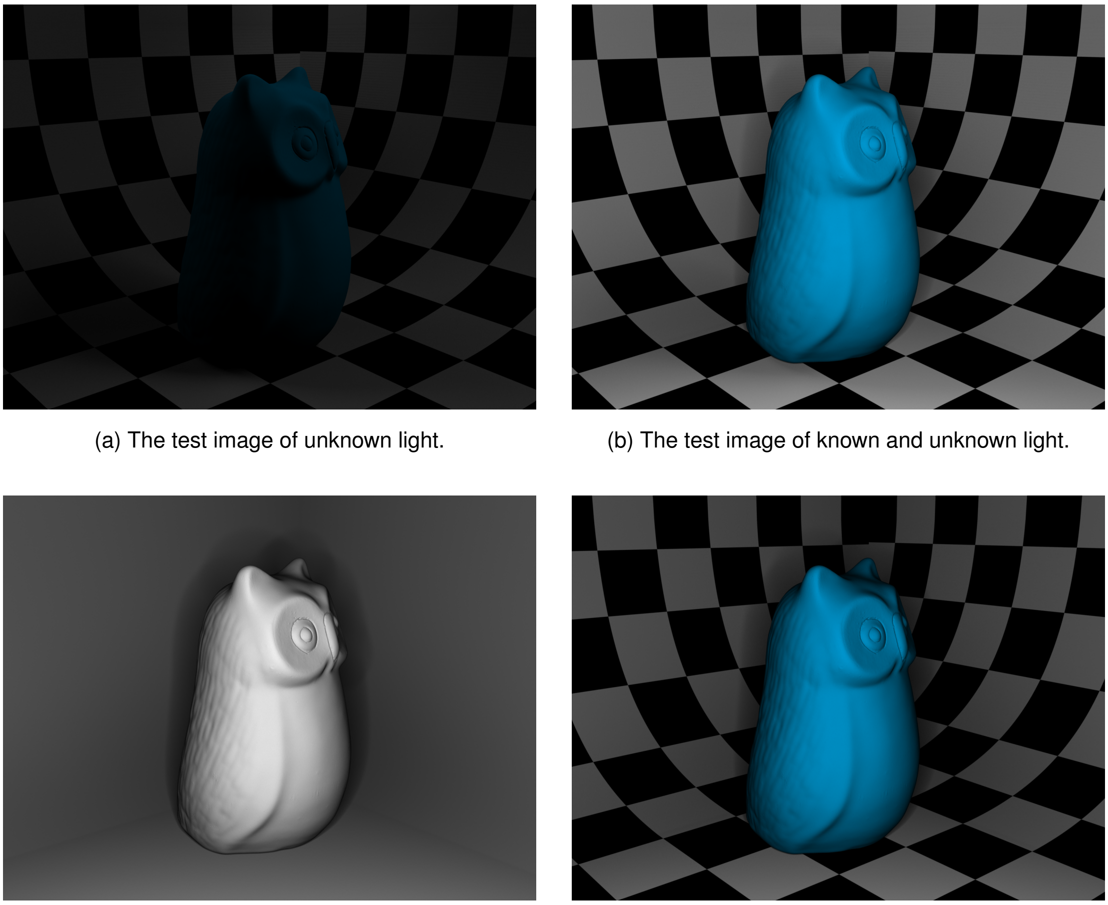
- **Unknown Light** (P_u)
  This image only contains the unknown light. This light would usually be light from the environment. This is later going to be subtracted from the image with both the known and unknown light in order to isolate the known light.

- **Know and Unknown Light** (P_{k+u})
This image contains both the known light and the unknown light. The known light would be created from a flash or similar. From this image and the image with the unknown light, the known light can be isolated.
- **Irradiance** (P_s)
These images contains the irradiance of all the pixel caused by the known light soruce. This enables the backward calculations for the reflectance. This image would be an simulated image in a real world scenario. This would be set up from the knowledge of the know light source. The diffuse reflectance for this image is set to one.

The next two images is created as ground truth of the two calculated variables being tested, the known light image, and the reflectance.

- **Known Light** (P_k)
This image is the ground truth of the isolated known light image. The error from the isolation can be tested by looking at the difference between the two images.
- **Reflectance** (ρ_Ω)
This image is the ground truth of the reflectance. The image contains the reflectance of the pixels visible from the cameras view.

The first four mentioned images for the test, can be seen in Figure 2.4.



(a) The test image of unknown light.

(b) The test image of known and unknown light.

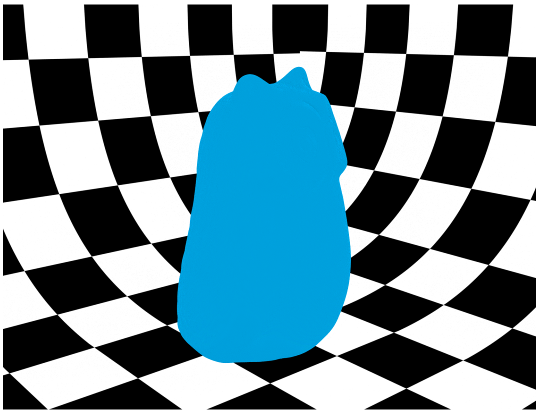
(c) The calculated irradiance image.

(d) The calculated image of known light.

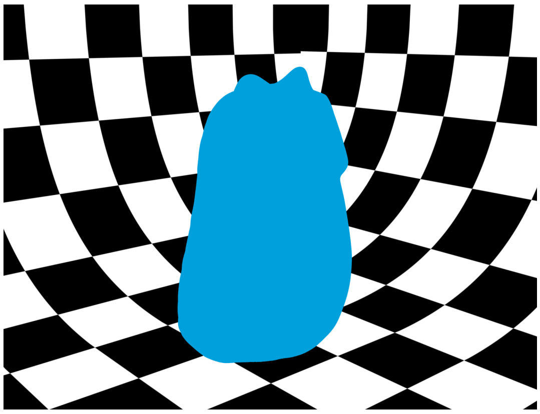
Figure 2.4: The four images used to calculate the final reflectance.

2.5.3 Test Results

Looking at the results after having calculated the reflectance using the proposed solution. By visually inspecting the resulting reflectance, no defects is present. The calculated reflectance and the ground truth of the reflectance can be seen in Figure 2.5.



(a) The calculated reflectance of the image.



(b) The ground truth of the reflectance.

Figure 2.5: The calculated reflectance (a), compared to the ground truth (b).

The visual inspection of the resulting reflectance is backed up by the calculated mean and standard deviation of the difference between the two images, in pixel values. The known light error calculations is also calculated in the same manner, comparing the calculated image with the ground truth. The results can be seen in Table 2.2. The flash only calculation had less the an integer in pixel value in error, with a mean of 0.2897, and a standard deviation of 0.4972. This would result a change from the ground truth pixel value of ± 1. Looking at the reflectance, the mean is 1.0555 and with a standard deviation of 2.1960. This is higher then the flash only reconstruction. This is due to clipping and low precision in the low light area. The low light area is the background as the exposure of the images is set for the object of interest. To show this, a clipping mask is used to separate the foreground and background. This clipping mask is an alpha mask that is generated from the scene, but with only the object visible to the camera. Doing this lowered the mean error to 0.1431 and the standard deviation to 0.8814. These results can be seen in Table 2.2.

Requirement	**Mean error**	**Standard deviation**
Flash	0.2729	0.5251
Flash object only	0.0603	0.2565
reflectance	0.9422	2.4152
reflectance object only	0.1080	0.9914

Table 2.2: The results from the proof of concept test. The values are in calculated from pixel values between 0-255.

2.5.4 Conclusion

Looking at the results, the flash only image has an error of under a single pixel value. Looking at the origin the error, it seems very likely that it is a difference in the rounding from the actual, simulated, radiance to a value in integers. This rounding of the value creates small discrepancies between the calculated and the simulated images, and thereby creates the error. This effect can be seen by comparing the object only results with the entire image. The error decreases when calculating the error for the object only. This could then be explained by the exposure being higher and thereby giving higher resolution for the actual radiance to be described.

The effect of the precision carried over to the results of the reflectance where the same effect is seen. The reflectance for the entire image again had a higher mean error compared to the the object only reflectance error.

Looking at the mean error for the object only, the values are lower then a single pixel value.

Taking the results in to account, the test proves that the proposed solution works as intended, and that it is possible to find the reflectance under conditions where the ambient light is unknown. It can also be seen that the results are reliant on the precision of the images used, and are limited to this precision. This error produced as a consequence of the precision is further investigated in Section 2.6: Precision Problem.

2.6 Precision Problem

Being limited to a precision of 256 steps, from no light to over exposure, may create errors. It is therefore needed to investigate the amount of error this can introduce and how to minimise this problem. To reduce this error, an increase in the pixel data could be done. In a use case scenario, this would be dictated by the camera. Most cameras use an 8 bit representation of the colour of a given pixel. Some cameras, for example the Canon 5Ds, uses 14 bit per pixel when shooting raw images, giving it a more precise representation. But because this is a variable that is dictated by the hardware, and can therefore not be changed, it should be investigated whether there is another way to minimise this problem.

The first method to increase the precision is to strive for the highest factor between the intensity of the known and unknown light. Increasing the precision of the known light representation. This could be done by having a room with little to no light present, and as strong a flash as possible. The second method could be to have different exposures for the unknown light image and the known and unknown light image. Then scale the images to the same point, and doing all calculations in a higher precision environment.

The setup for this test was the same as for Section 2.5: Proof Of Concept.

2.6.1 Test data

For this test, four sets of synthetic images is created. One where the flash has a low factor of known light intensity compared to the ambient light intensity, and one where the factor of known light intensity were high compared to the unknown light intensity. The reason for not investigating the use of 16 bit images is that the increase in precision would always be a gain. It would therefore always be preferable to have a higher bit depth for the images captured.

In order to calculate the error, the ground truth was created as a 16 bit image. Calculating the difference between the ground truth and the test images results in the error due to the precision difference.

In order to test if precision can be incread by capturing the unknown light image at a higher exposure in order to have more precision, an extra image with a different exposure was render in addition to the other images.

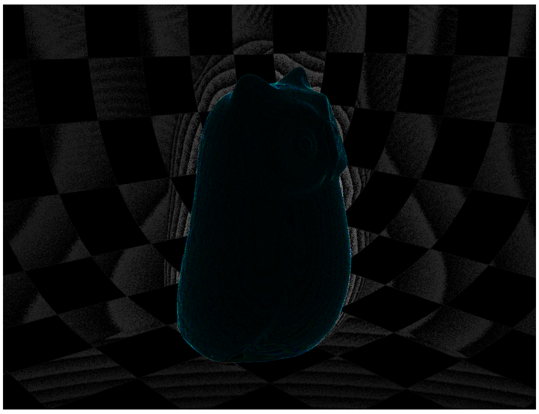
2.6.2 Test results

By visually inspecting the error images from the two test images, it is clear that the test with a small factor between the ambiance and the flash, has a higher error then the test with high factor. Closely investigating the test with a small factor, a wave patter can be observed. This wave effect is due to pixel values continuously getting further, or closes, to the ground truth and then at some point the two are equal when no rounding occurs. The biggest error is present when the biggest amount of rounding occurs. The error images from both tests can be seen in Figure 2.6.

Looking at the results for different exposure, they in general have a lower mean error, and an slight drop in standard deviation.

2.6.3 Conclusion

From the results presented, having a higher factor between the know light and the unknown light improves the result. This is due to less information getting lost as the two light sources are added together and thereby limited by the precision of the image. This can be seen in the decreased error in the calculated flash only image. The error of the calculated known light decreased by about $1/4$ in the test, and because of this decrease, the reflectance error was only around $1/10$ of the error compared to the test with low factors between the lights. The different exposure



(a) The resulting error from the ground truth with a low factor between the unknown and know light.

(b) The resulting error from the ground truth with a high factor between the unknown and know light.

Figure 2.6: The error from the high factor and low factor test. (a) is the low factor test results and (b) is the high factor test results.

		Low Flash Factor		**High Flash Factor**	
		Mean Error	**STD**	**Mean Error**	**STD**
	Flash	0.2602	1.3186	0.2719	0.5301
Same	**Flash object only**	0.0627	0.2608	0.0620	0.2679
Exposure	**reflectance**	2.2833	13.1587	1.1626	2.8791
	reflectance object only	0.2851	2.3622	0.1333	1.1902
	Flash	0.1694	1.3357	0.0372	0.4974
Different	**Flash object only**	0.0417	0.2200	0.0075	0.2008
Exposure	**reflectance**	1.4442	13.3170	0.0973	2.6719
	reflectance object only	0.1599	2.0603	-0.0272	0.9783

Table 2.3: The results from the proof of concept test. The values are in calculated from pixel values between 0-255.

showed great promise as the mean error was a little less then halved for the reflectance by doing this. The standard deviation fell slightly in general to. This also showed that calculating the flash only image precisely had a big impact on the final result, making this a priority. Further more the capturing of different exposure images for the unknown image and the known and unknown, improves the results in all aspects.

2.7 GI Problem

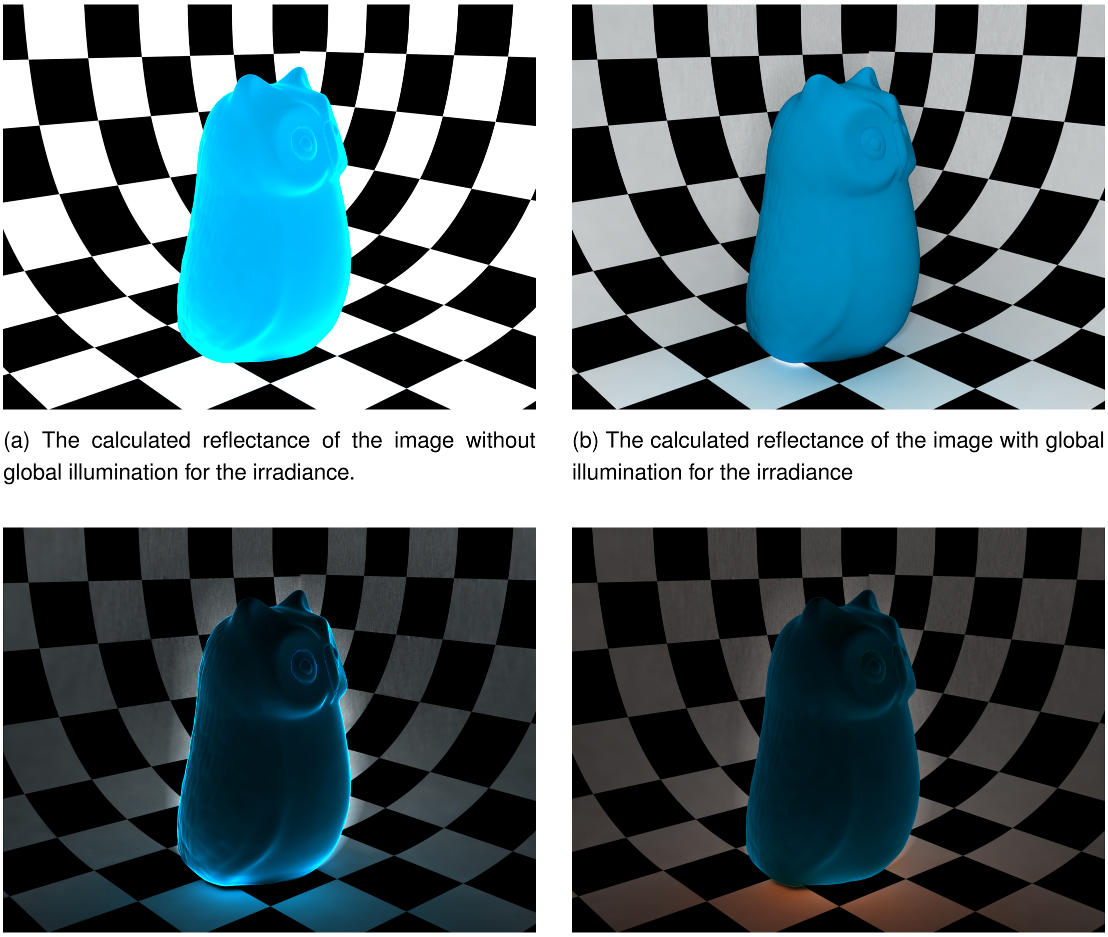
Global illumination is a natural occurring phenomenon as all surfaces gives off light and therefore add irradiance to other objects. This is problematic as the reflectance of the object is not know in advance, together with the entire environment. Because of this the global illumination can not be simulated precisely. Investigating whether or not the irradiance simulation gives the best results by simulation with global illumination or not is therefore of interest. This problem will be investigated further in this section.

To investigate this, the same test setup used for Section 2.5: Proof Of Concept is used.

2.7.1 Test Data

To test this problem, the same set of images is rendered as in Section 2.5: Proof Of Concept. This is the unknown light image, the known and unknown light image, the irradiance image, the ground truth of the known light image, and the ground truth reflectance image. This time the images is rendered with global illumination in order to investigate the error from this. The irradiance map is then rendered with and without global illumination. This way the test mimicked the images that would be captured in a real world scenario. The error of the test is calculated with the two different irradiance maps finding the individual error.

2.7.2 Test Results



(a) The calculated reflectance of the image without global illumination for the irradiance.

(b) The calculated reflectance of the image with global illumination for the irradiance

(c) The error image without global illumination for the irradiance.

(d) The error image with global illumination for the irradiance.

Figure 2.7: The results from the two test set of images. The left side is without global illumination for the irradiance. The right side is with global illumination for the irradiance.

Looking at the visual results from Figure 2.7, it can be seen that there is a noticeable difference in the error map and the resulting reflectance. This is backed up by the calculated error from these renders. The error can be seen in Table 2.4. As the object is the main focus, the error of the object is the most important factor. With an increase in mean error of 48.3% and a increase in standard deviation of 74.7% the choice must be said to have a noticeable impact on the precision of the calculation.

	GI on Irradiance		GI not on Irradiance	
Requirement	**Mean error**	**Standard deviation**	**Mean error**	**Standard deviation**
reflectance	-31.8876	32.4318	34.2450	39.4582
reflectance object only	-4.7059	13.8529	7.1200	26.0751

Table 2.4: The results from the global illumination test. The values are in calculated from pixel values between 0-255.

2.7.3 Conclusion

Comparing the irradiance with and without global illumination, it can be seen that with global illumination, the reflectance map has a smaller error. This is because that global illumination is part of the light from the flash that bounces on the other surfaces in the scene. The reason for the high error is that the reflectance is not know at the point of simulating the irradiance, making the global illumination based on the white materials applied, and thereby creating the error.

Looking at the results for the object only, it is clear that global illumination should be simulated on the irradiance map, even though the reflectance is unknown at the point of simulation.

Chapter 3

Camera Theory

In order to understand and manipulate the process from a physical object to a model, the way an images of the physical object is capture is needed. This consists of an understanding of how a physically camera works. Further more, how the different settings controls how light enters the camera, and how these are described in relation to each other.

3.1 APEX model

In order to correlate the different settings affecting the light of the resulting image, an international standard the Additive system of Photographic EXposure (APEX) system is used. This system converts the different settings in to an additive system where an increase of one, is a doubling in light. The end result of the APEX system an exposure value (EV). If the EV is the same for two different combinations of settings, the resulting image will have the same exposure.

To calculate the EV Equation 3.1 is used where A_v is the Aperture Value, T_v is the Time value, B_v is Brightness value, and S_v is the Speed value (Kerr, 2016). A nice property of Equation 3.1 is that $A_v + T_v = B_v + S_v$, this creates a situation where two of the values are constant, the two other values can then be change to correlate with some EV. The same EV can therefore be reached by changing one of the variable and offsetting the other value inversely by the same amount. Usually when calculating the EV, the Speed value and Brightness value is static, and it is therefore a matter of choosing a pair of aperture and shutter speed that satisfied the wanted EV.

$$EV = A_v + T_v = B_v + S_v \tag{3.1}$$

The aperture value is a linear model of exposure and should therefore increase by one every time the exposure is doubled. This is done by taking the base-2 logarithm of the aperture. But because the aperture is an area, the second power of the aperture is taken. The final equation is define as follows:

$$A_v = log_2(A^2) \tag{3.2}$$

To make the exposure value go up by one, the Shutter Speed needs to halve. In a camera the shutter speed is represented as $1/20$. Therefore the number actually doubles when the shutter speed is halved. This way of representing the shutter speed is used in the equation to give a value of the amount of a second the shutter is open. This gives the expected result as the value halves when the exposure is halved. The base-2 logarithm of the amount of time the shutter is open is taken to make this linear. The final equation is define as follows:

$$T_v = log_2(1/T) \tag{3.3}$$

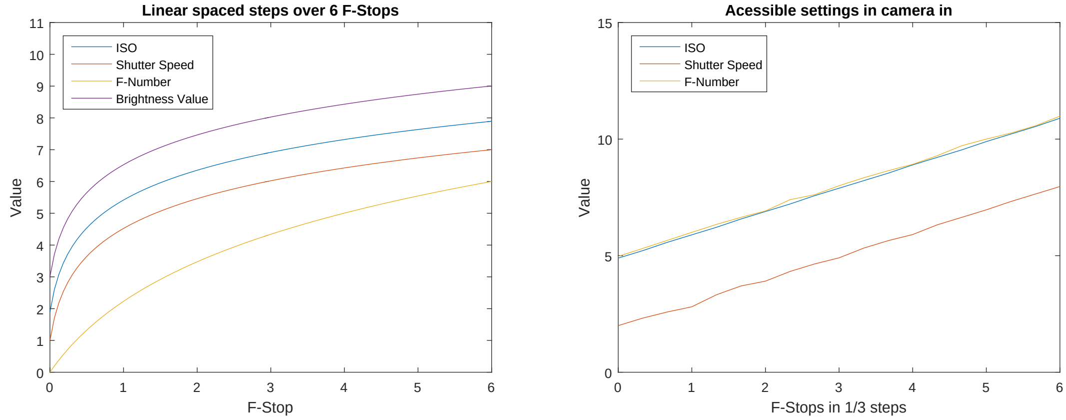
The film speed (ISO) is included by multiplying the film speed and the variable relation constant, N. The base-2 logarithm of this is then taken. N is a constant relation between the film speed and speed value. The value of this constant is $2^{-7/4}$. With this the final equation is define as follows:

$$S_v = log_2(NS_x) \tag{3.4}$$

The Brightness Value is the metered luminance of the scene. The luminance is represented in $candelas/m^2$. The K is the reflected-light meter calibration constant. This value indicates what the manufacturer of the light meter represents as correct exposure. The base-2 logarithm is taken of the value to create a linear result. With this a halving in the radiance of a light source standing in an entirely black room, lower B_v by one. The final equation is define as follows:

$$B_v = log_2(B/NK) \tag{3.5}$$

The logarithmic nature of the light settings inside the camera is then be converted to a linear scale. This is illustrated in Figure 3.1. With the calculations to EV values defined. The revers calculations can be usefull in order to define



(a) The EV values of the individual variables over 6 EV, showing the logarithmic nature. The variables are off-set by one Value for visibility.

(b) The values the camera can be set to, showing this is in fact a linear increase in exposure.

Figure 3.1: Plots of the individual EV of the four variables, and the cameras settings plotted in EV showing this is close to linear.

a scale value for an image. By taking two to the power of the EV difference of two images, the actual scaling value is calculated. The definition for this is as follows:

$$\begin{aligned} EV_{diff} &= EV_a - EV_b \\ s &= 2^{EV_{diff}} \end{aligned} \tag{3.6}$$

An example of this could be if two images was taken with two different settings. The settings was recorded, and the files should now be scaled so the values correlate to each other. If the images was captured with a change in both the shutter speed and ISO, keeping the light and aperture consistent, calculating the scale from these, the individual values needed to be calculated. Image A have a EV value of:

$$\begin{aligned} -7.2288 &= log^2(1/150) \\ 4.8939 &= log^2(2^{-7/4} \cdot 100) \\ -2.3349 &= -7.2288 + 4.8939 \end{aligned} \tag{3.7}$$

Image B have a EV value of:

$$\begin{aligned} -6.6439 &= log^2(1/100) \\ 7.8939 &= log^2(2^{-7/4} \cdot 800) \\ 1.2500 &= -6.6439 + 7.8939 \end{aligned} \tag{3.8}$$

To find the amount image A should be scaled by, the two are subtracted to find the difference in EV. The scale is then calculated by taking two to the power of the EV difference.

$$\begin{aligned} 3.5849 &= 1.2500 - -2.3349 \\ 11.9995 &= 2^{3.5849} \end{aligned} \tag{3.9}$$

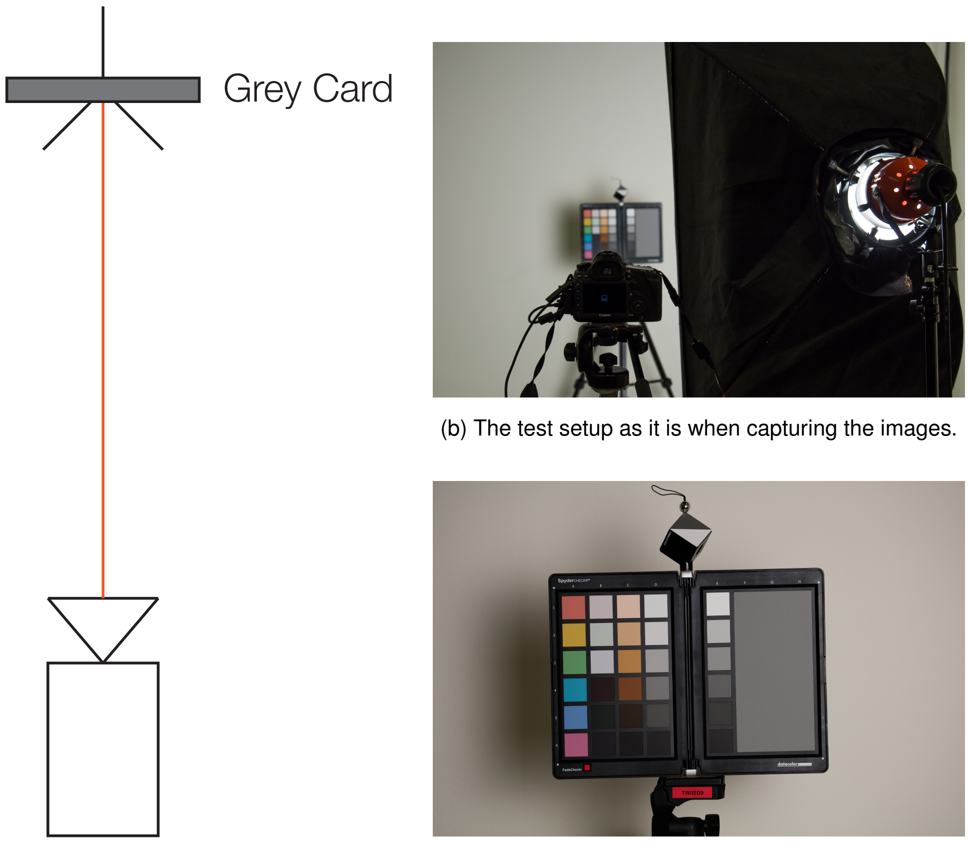
With this, image A should be multiplied by 11.9995 in order to correlate with image B.

3.2 Camera Exposure Linearity

The camera linearity test is created to test the linearity of a physical camera compared to the simulated camera. This is important as the images of the unknown and known plus unknown image should both be well exposed, and then scaled to correlate. This scaling needs to be tested as this could create artefacts and errors in the final reflectance, because of the incorrect amount of light from the known light image.

3.2.1 Test Setup

The setup for the test consists of two main objects. The camera and a grey calibration board. This setup is created both in the simulation software and in the real world. The simulated setup is used as ground truth. With these two datasets, the physical and the simulated cameras can be compared in relation to the scaling of the captured light.



(a) Illustration of the test setup.

(b) The test setup as it is when capturing the images.

(c) One of the test images used in the test.

Figure 3.2: The test setup used when capturing the test images. One of the test images is show, (c).

As the test focuses on the factor between the different exposure setting in the camera, the distance between the grey card and the camera does not matter. Neither does the amount of light in the environment, as long as the pictures being captured does not over expose on the highest EV setting, and does not under expose on the lowest EV setting. The most ideal setup is a setup where the captured image of the highest EV setting produces an image that is close to, but does not, overexposure. This would give the highest precision as the pixel value is halving every EV step.

3.2.2 Test Data

The data for the test consists of groups that spans one EV each. This is due to the pixel values decreasing by half every EV step. By making them span only one EV, the precision decrease by, the halving of the pixel value is minimized, and thereby give sufficiently high results for the validation. Furthermore any non-linearity in the sensor can be omitted. The groups for the three settings can be seen in Table 3.1. To make sure the camera is comapred to the actual values it provide, the settings was taken from the EXIF data in the test images. The values chosen for

<table>
<tr><th>EV</th><th>ISO</th><th>Shutter Speed</th><th>Aperture</th><th colspan="2">Test Group</th></tr>
<tr><td>6</td><td>6400</td><td>4</td><td>5.6</td><td>|</td><td></td></tr>
<tr><td>5 2/3</td><td>5000</td><td>5</td><td>6.3</td><td>|</td><td></td></tr>
<tr><td>5 1/3</td><td>4000</td><td>6</td><td>7.1</td><td>|</td><td></td></tr>
<tr><td>5</td><td>3200</td><td>8</td><td>8</td><td>|</td><td>|</td></tr>
<tr><td>4 2/3</td><td>2500</td><td>10</td><td>9</td><td></td><td>|</td></tr>
<tr><td>4 1/3</td><td>2000</td><td>13</td><td>10</td><td></td><td>|</td></tr>
<tr><td>4</td><td>1600</td><td>15</td><td>11</td><td>|</td><td>|</td></tr>
<tr><td>3 2/3</td><td>1250</td><td>20</td><td>13</td><td>|</td><td></td></tr>
<tr><td>3 1/3</td><td>1000</td><td>25</td><td>14</td><td>|</td><td></td></tr>
<tr><td>3</td><td>800</td><td>30</td><td>16</td><td>|</td><td>|</td></tr>
<tr><td>2 2/3</td><td>640</td><td>40</td><td>18</td><td></td><td>|</td></tr>
<tr><td>2 1/3</td><td>500</td><td>50</td><td>20</td><td></td><td>|</td></tr>
<tr><td>2</td><td>400</td><td>60</td><td>22</td><td>|</td><td>|</td></tr>
<tr><td>1 2/3</td><td>320</td><td>80</td><td>25</td><td>|</td><td></td></tr>
<tr><td>1 1/3</td><td>250</td><td>100</td><td>29</td><td>|</td><td></td></tr>
<tr><td>1</td><td>200</td><td>125</td><td>32</td><td>|</td><td>|</td></tr>
<tr><td>2/3</td><td>160</td><td>160</td><td>35</td><td></td><td>|</td></tr>
<tr><td>1/3</td><td>125</td><td>200</td><td>39</td><td></td><td>|</td></tr>
<tr><td>0</td><td>100</td><td>250</td><td>45</td><td></td><td>|</td></tr>
</table>

Table 3.1: This table shows the different groups and what value range they cover.

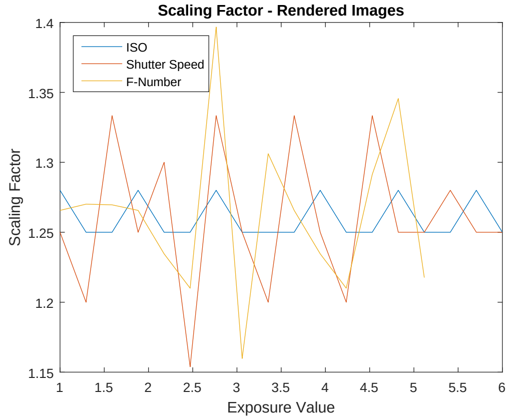
the test brackets, are based on the settings the physical camera is capable of. The actual value of these settings are the same for most cameras and follows a standard. Usually there is three steps in between each full EV increase.

These test groups is used to render the ground truth images. Observing the values, the ISO and Shutter Speed does not always half or double over one EV, this creates a situation where the ideal output from the real world camera should not be linear over one group. This is visualised by calculating the factor from the ground truth images and plotting them. This visualization can be seen in Figure 3.3a.

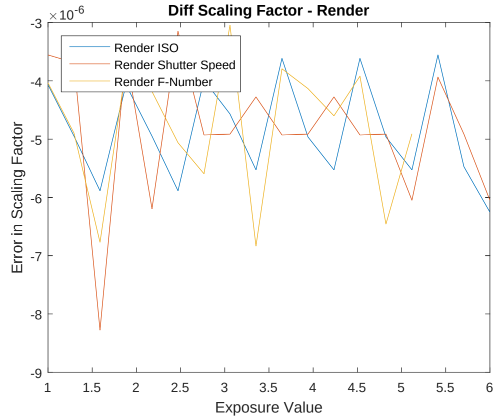
The reason for these settings not being exactly halved or doubled each time is, that they are shifted to hit easier to remember number in order to make it easier for the end user. This is partly true for the aperture, because the aperture is rounded as it never reached a perfect integer. This is because the aperture is a logarithmic function as it is the area of the blades inside the lens that gets adjusted. It can therefore not be represented in the lower settings by whole integers. When reaching over 7.1 the aperture is still rounded but to integers. The most consistent of these three values are the ISO that is precise over every full EV step.

3.2.3 Test Results

Looking at the scaling factors between the different setting steps in a physical camera and the renders, it can bee seen that there is some differences. See Figure 3.4. It varies between the three settings how well they correspond to the theoretical values. The setting that is least varying are the apeture. But the mean difference, from the ground truth, is lowest in the shutter speed. All the results can be see in Table 3.2. Looking at the simulation results, it is perfectly mimicking the theoretical calculations, only with an error that is due to precision.

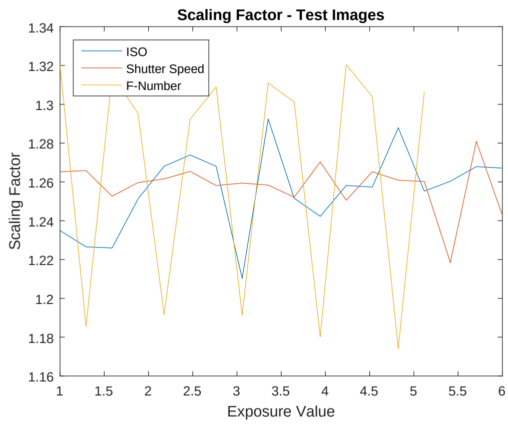


(a) The simulated cameras factors between each step.

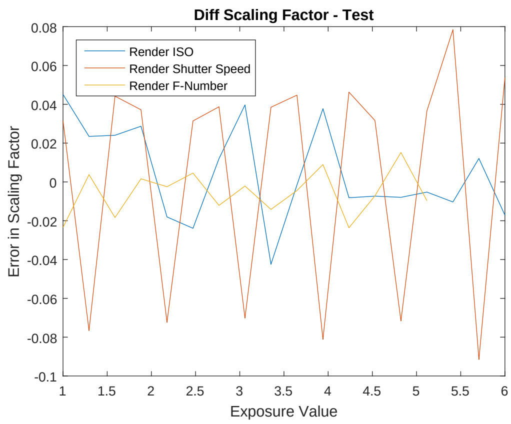


(b) The resulting error from the theoretical calculations and the simulation.

Figure 3.3: The simulated factors for between the settings steps available on the camera, and the resulting error by comparing the simulated and theoretical factors.



(a) A plot of the scaling factors for the tested setting ranges in the physical camera.



(b) A plot showing the error of every setting step compared to the theoretical perfect scaling factors.

Figure 3.4: Plots showing the results from the tested settings in the physical camera.

3.2.4 Conclusion

The test shows that the camera mostly follows the APEX model with a difference in how precise the three individual settings are related to the theoretical scaling factor. Furthermore the test proves that the simulation environment follows this model precisely, only with the limitation of the precision of the individual pixel value. The scaling factors for the three different settings, ISO, Shutter speed, and Aperture can be used as a lookup table for precise up and down scaling between images taken with different exposure settings. This are important when a precise result is wanted, and loss due to data precision are a factor, as the resulting lookup table enabled the use of different exposed images.

	Mean	**Max**	**Min**	**STD**
ISO	0.0045	0.0451	0.0016	0.0245
Shutter Speed	0.0028	0.0915	0.0314	0.0593
F-Number	-0.0056	0.0236	0.0016	0.0115

Table 3.2: The results from the test comparing the physical camera with the theoretical perfect system.

3.3 Camera Sensor Linearity

The last thing to test in regards to the behaviour of the camera, is the linearity of the sensor in the camera. This is an important aspect as this could distort the values captured in dark areas, either due to under exposure or dark colours on the object of interest. The following section will look in to a procedure to test this, and the results from this test.

The setup is similar to Section 3.2: Camera Exposure Linearity.

3.3.1 Test Data

To test the linearity of the sensor, everything, but the colour values in the picture, needs to be constant. This includes the ISO, Shutter Speed, and Aperture. To get as much data as possible, a colour chart containing 13 squares with different intensity of grey is used. This makes it possible to test the entire range of the sensor while keeping all the other variables static. The values of the different squares can be seen in Table 3.3 (Datacolor, 2016).

% Gray	RGB value
0% (White)	255,255,255
5%	242,242,242
10%	229,229,229
20%	204,204,204
30%	178,178,178
40%	153,153,153
50%	127,127,127
60%	102,102,102
70%	77,77,77
80%	51,51,51
90%	26,26,26
95%	13,13,13
100% (Black)	0,0,0

Table 3.3: The values of the 13 test squares on the data board.

The data board used for this test is the datacolor SpyderCHECKR (datacolor, n.d.).

3.3.2 Test Results

Looking at the plot of the data, it is not linear. The squares on the calibration board does fade over time, making the dark fields lighter. Looking in the low end, there is a small bend that is non consistent with the bigger curve. This is also present on the top most point.

3.3.3 Conclusion

There can be multiple reasons for the big curve that can be observed in the test data. UV light does break down colour slowly over time. This would fit with the curve and explain this phenomenon. This is further validated by the colour chart having a small square that indicates if the chart has gotten exposed to an specified amount of UV light. Looking at other studies of sensor linearity, it is highly unlikely that the sensors response is a curve as seen. The sharper bends in the very top and button of the date, fits the other studies findings. A sensor is usually not linear in the top button most parts of the spectrum (Jensen, 2015). This test could be run with a new set of colour chats, but the supporting studies show that the sensor is linear in the spectrum that is used.

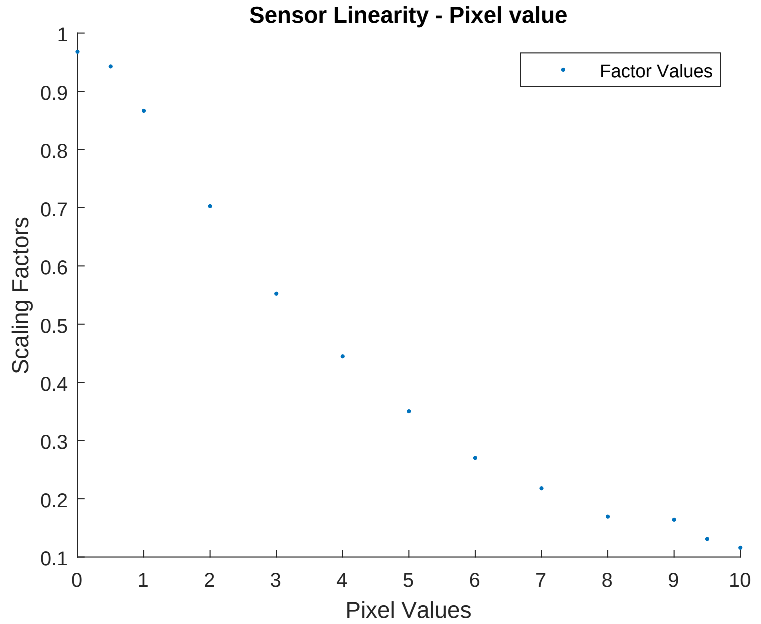


Figure 3.5: This shows the data points of the pixel values for an image of the colour chart.

3.4 Flash Testing/Calibration

The aim of the test is to validate the performance of the flash chosen for the final validation of the proposed system. The end result is a calibration of the intensity of the simulated flash to the physical flash unit. This is an important factor as this dictates the precision of the en result, and could lead to lighter or darker reflectance then what is present on the object of interest. The reason for not using the intensity presented in the data sheet for the flash is, that every flash unit is usually a little different, and therefore to achieve the best results, the individual flash unit needs to be calibrated.

3.4.1 Test Setup

The test setup consistes of a camera, a flash, and a know reflectance object place at a given distance, from a known point on the flash unit. The know reflectance object is a piece of white printer paper with a reflectance of 0.94 (Bartman et al., 1964). The setup are shown in Figure 3.6.

To enable the testing and calibration of the flash, some variables needs to be know about the test setup. These should not be changed during the calibration precess. These values are the cameras settings, the distance from a know point on the flash unit to the know reflectance object (Δ), and the intensity setting on the flash unit. Furthermore the room used for the calibration should be entirely deprived of light in order to only have the flash add light to the know reflectance object.

3.4.2 Test Data

To investigate the consistency of the flash release in terms of intensity, a series of images are captured without changing any variables. 40 pictures are captured for this purpose. To calibrate the flash, images of an know object is captured to use for the real world reference. An image for the simulation is created where the starting value for the lumination of the flash is close to what are expected. In this instance the value specified in the data sheet of the flash is used. After a correction factor is calculated a new rendered simulation is created to validate the calibration.

3.4.3 Test Results

Investigating the consistency of the flash release, it is found that the standard deviation is 0.0029. The data and standard deviation can be seen in Figure 3.7.

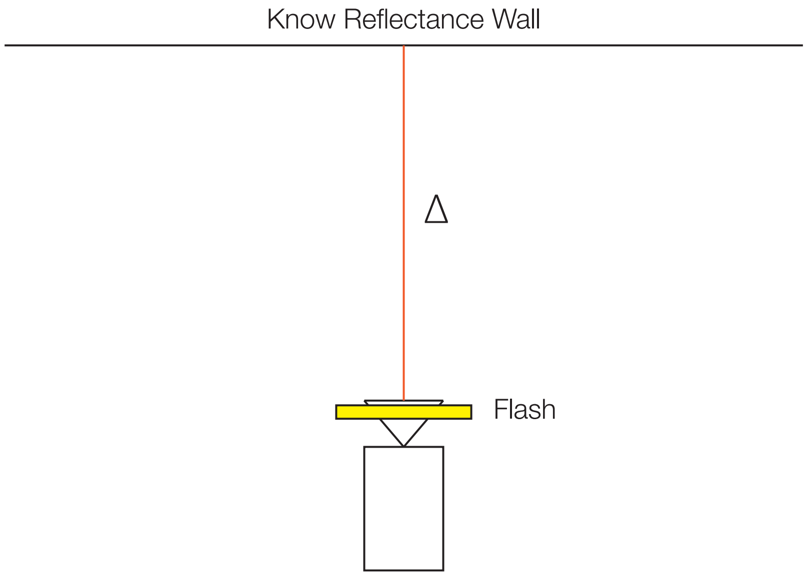


Figure 3.6: The test setup for flash testing/calibration.

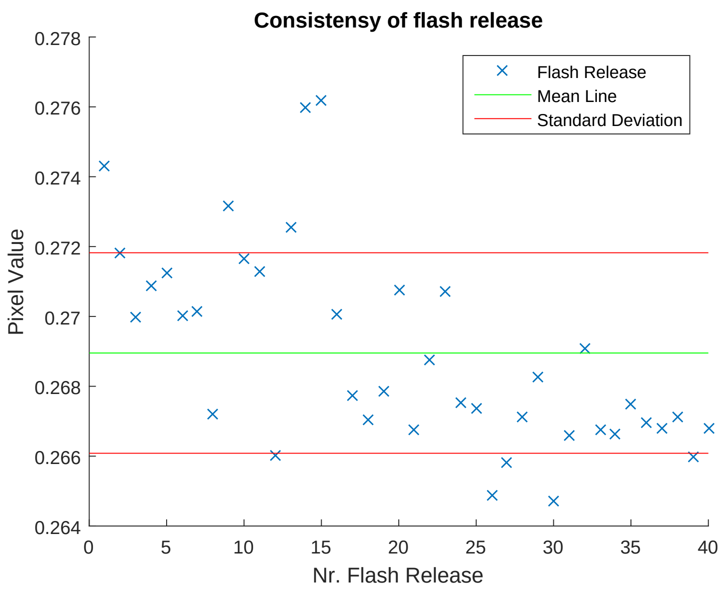


Figure 3.7: The plot is showing the consistency of the flash release. The green line is the mean value, and the red lines is the standard deviation.

Looking at the results from the flash calibration, it is found that the first guess of 50 is a good starting point. The reason for 50 is that the flash at full power should deliver 400 watt. The flash is at 1/4 power. Further, to not over expose the image, it is halved. The factor that the light source should be scaled by to correlate with the real flash, is calculated to 0.7534.

After the scaling is conducted the same procedure is repeated to validate this. The second validation scaling factor is 0.9998 with the simulated flash at 37.665 watt. The results can bee seen in Table 3.4.

3.4.4 Conclusion

With the consistency of the flash having a standard deviation of 0.0029 the flash had a high enough precision to be used in the experiment compared to some of the other error factors. That said, if some of the procedures and equipment was improved, the consistency should be kept in mind as this could become the highest contributor to

	Pixel Value	Scaling Factor	Flash Intensity (Watt)
Test Image	0.7033		
First render	0.9334	0.7534	50
Second Render	0.7035	0.9998	37.665

Table 3.4: The results from the flash calibration, showing that the calibration procedure works and gets within 0.02% of the real image.

the error of the final calculations.

Looking at the calibration process, this is working as intended and gave a precision lower then the consistency of the flash.

Chapter 4

Requirements

From the theory it can be seen that there is some general requirements when capturing images for this purpose. These requirements stem from the need for known variables further in the purposed solution. Most of these requirements are needed for the simulation of illuminance or calculating the reflectance, and are therefore very important.

There is also more general requirements in terms of what is trying to be achieved by the solution. This includes the acceptable error of the calculated reflectance together with a visual inspection of the perceived quality of the resulting model.

4.1 Limitations

To limit the size of the project to a manageable scope, given the time frame, a series of limitations is chosen. These limitations are expansions to the solution, and would extend and generally improve the solution.

- **Diffuse Surfaces Only**
  The solution is limited to only include diffuse surfaces. Almost all object have some sort of diffuse reflections. Further more diffuse surfaces are a general requirement for image based modelling due to the view independent nature of this kind of reflections. More specifically only Lambertian diffuse reflections will be investigated.

4.2 Image Capture Requirements

The requirements for the image capturing process is the specific requirements that needs to be followed when acquiring images. The reason for these is partly due to the reflectance recreation, and partly due to irradiance simulation. Some of the requirements overlap, but they all are used to ensure proper images is acquiring for the best possible end result.

The requirements for the capturing process is as follows.

- **Known Camera Settings.**
  It is required to know the cameras settings at the time of acquisition to enable the simulation of the illumination. It is also needed if the two images is captured with different settings, as one of the images would need to be scaled. This scale value is calculated from these settings.
- **Control of Camera Settings.**
  To correctly expose the pictures, total control of the cameras exposure settings is needed. This is duo to the cameras auto exposure feature might overexpose parts of the images, and thereby lose needed information of the appearance of the object.
- **Control of Flash Settings.**
  In order to have a consistent flash release, that can be calibrated with different lighting condition, the control

of the flash release is needed.

- **High Resolution Images(Size and Precision)**
  In order to have precise information about the appearance of the physical object, the size of the image and the precision of each pixel is of importance. The size of the image gives more detail about the physical object. This in term gives better reconstruction, as more features can be found and matched, together with more information about the appearance, giving a better calculated reflectance. The precision of the pixel value also increased the precision of the results of the calculated reflectance.
- **Two Images at the Same Position.**
  For the isolation of the known light source it is of utmost importance that the two images are captured at exactly the same position. This is to enable the subtraction of light on a given point and later be able to calculate the reflectance for this particular point. To enable this a solid setup of the capturing devise and an unobtrusive way of capturing the image is needed.
- **Known Light Source**
  A known light source is needed in order to simulate the radiance from the light source. The simulation dictates whether the reflectance is to bright or dark.
- **Diffuse Surface**
  A diffuse surface is needed for several reasons. First of it is a requirement for the reflectance model proposed for this solution. Further more it is a requirement for image based modelling. Very few surfaces are completely diffuse, but the more diffuse the surface, the better results is generated.
- **Consistent Ambient Lighting**
  This requirement is due to the nature of image based modelling. To enable the best results, the reconstruction needs static light to improve conditions for feature matching, and there by reconstruction.
- **Quality of the Reconstruction**
  The quality of the reconstruction is very important as this will create errors in the simulation and create artefacts on the final reflectance. This is one of the most important factors of the error introduced to the calculated reflectance.

4.3 Requirements

With the capturing requirements present, the requirements for the final result needs to be defined. These requirements is tested upon and analyse the proposed solution. There is two main requirements.

- **Remove Shadows**
  The first requirement is to remove the shadows present on the physical object. Especially shadows with sharp edges as these are very noticeable on a reconstructed model. More smooth shadows is also of interest as these also will introduce unbelievable appearances.
- **Reflectance error**
  The error in the final calculated reflectance is needed to be low, in order to create a model that will be believable in an computer generated environment. As the main purpose is the appearance of the object, it is an requirement that the perceived quality is higher then the original reconstruction. Further a calculated error of the reflectance should be no higher then 5% of the ground truth.

Part II

Implementation

Chapter 5

Method Flow

This chapter gives an overview of methods flow, from the capturing of images, to the finished reflectance, Figure 5.1 There is six modules for the method. The first model is the Image Acquisition. In this module the actual capturing

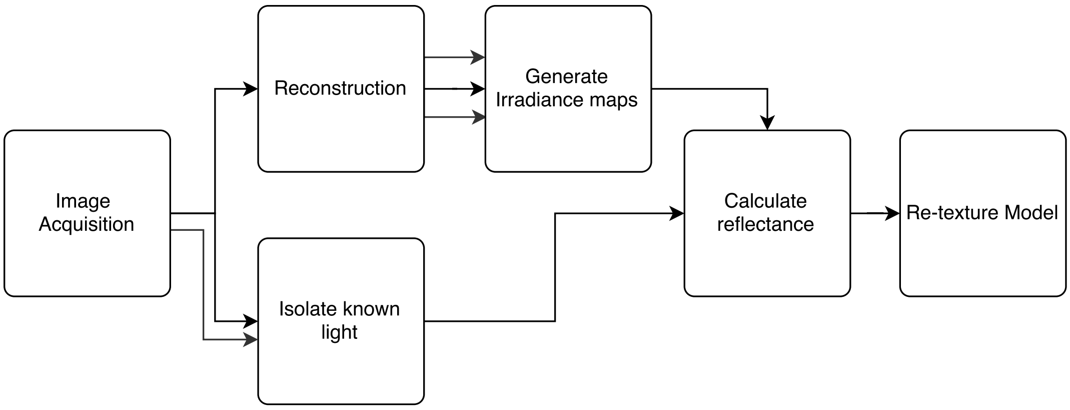


Figure 5.1: The general flow for the proposed solution.

of the images are done, and the images are prepared for further processing. The process then splits in to two parallel branches. The first branch consists of two modules. The reconstruction of the object, and the simulation of the irradiance maps are done here. The other branch is responsible for extracting the known light source from the source images. These branches then joins in order to calculate the reflectance. The last step is then to apply this to the reconstructed object.

In the following sections, each of these modules will be explained and the methods used in the report will be discussed.

Chapter 6

Image Capture

In order to acquire the images used for this method, an assortment of equipment are used. One of the main reasons for this method, is a more trivial approach without a high amount of specialised equipment. Therefore all of the equipment used to acquire the images is off the shelf hardware.

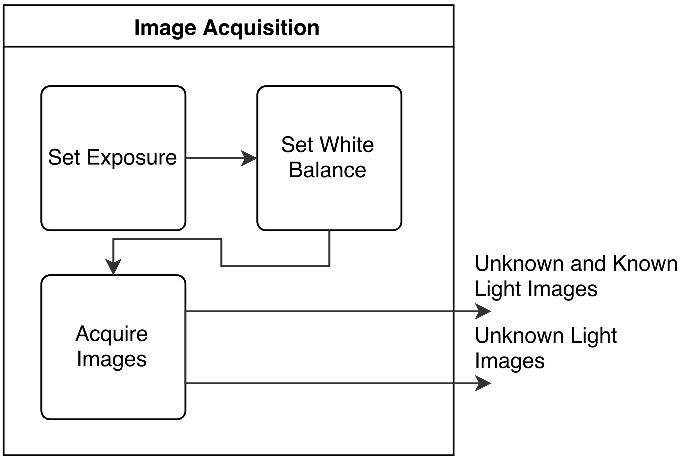


Figure 6.1: The flow for the image acquasition.

6.1 Equipment

For the method to work, some key pieces of photography equipment is needed. The first piece is the camera body.

- **Canon 5Ds (Canon, 2015)**
  This camera is chosen as it has the highest resolution of any photography camera body. The sensor in the body is an 50.6 megapixel fullframe CMOS-sensor giving a resolution of 8736 by 5856 pixels. The high resolution gives a very good basis for the reconstruction, but also gives a high amount of detail in the resulting reflectance map. Further more, the 5Ds captures 14 bit images, giving a higher precision of the pixel values in the resulting raw file. All the images is captured in raw format to retain all possible information from the camera and sensor.
- **Canon EF 70-300mm f/4.0-5.6L IS USM (Canon, 2010)**
  The lens is chosen for the optical performance, and the zoom range it provides. The one drawback with a very high resolution sensor, is that the sharpness, and general optical quality, of the lens becomes more noticeable. Further more a polarity filter is attached on the front of the lens. This is to reduce the amount of specular reflections.
- **Godex Witstro AR400 (Godox Photo Equipment CO.,LTD, n.d.)**
  The Godex Witstro AR400 is chosen for two main reasons. The flash is a ring flash. A ring flash goes around

the lens,with an opening in the middle, hence the name. This crates a light that is very uniform when the flash is triggered. The second reason, is that the flash is very powerful even though it has no need for an external power source, as it uses a battery.

Besides the photography equipment, a manfrotto tripod is used. The exact tripod used is on the weak side and needed counter weight as the equipment are very heavy. A very sturdy tripod is advised, both because of the equipments weight, but also for the stability when capturing the images. Further more small accessories is also used, but will not be mentioned further.

This equipment is the equipment needed in order to capture the images for the proposed solution. There is therefore no need for any special build equipment for the capturing process.

6.2 Defining Exposure

Getting well exposed images are paramount to the result. If the acquired image are overexposed at any point it results in the loss of data for that specific point. Further more the exposure should be set for something close to white. The reason for this is the that the simulation of the irradiance is with a completely white material. Therefore overexposure in this simulations will also lead to errors.

The simple way of doing this would be to enable over exposure warnings in the camera body. It is found, that the camera are looking at the overexposure for an 8 bit image, where the output of the camera is actually 14 bit. This limits the actual precision of the pixels considerably. Therefore a small Matlab script is developed to ensure no overexposure.

```
% Get the area of interest
fig = figure();
imshow(orgImg);
rect = uint32(getrect(fig));
close(fig);

% Calculate the area of interest max and mean values
max(max(im2double(imcrop(orgImg,rect))))
mean2(im2double(imcrop(orgImg,rect)))

% Find the entire images max value
max(max(im2double(orgImg)));

% Show a histogram of the pixel values with 100 bins
figure()
h = histogram(im2double(orgImg),100);
```

This script gets an area the user specifies, then calculates the mean and maximum RGB values. Further more it calculates an histogram of the entire image. The histogram in particular is very useful as it shows where the exposure are for the given image. The maximum value can be erroneous if a single point is very reflective and therefore have a higher value.

6.3 Image Conversion

When the images are captured, they need to be converted in to a more useful format. In the case of a canon camera, the raw files is .CR2 files. These are converted with a open source raw processor called dcraw (Coffing, n.d.). This converter is able to process the images in to 16 bit linear TIFF images through the command line. Because this is a command line tool, it is easy to batch convert entire folders etc.

Chapter 7

Reconstruction

With all the images ready, the next step is to create the reconstruction of the physical object. This is done by using a proprietary software called ContextCapture. The reasoning for choosing this software over an open source solution is that the final reconstruction is of very heigh quality, together with having all the capabilities that is needed further in the process.

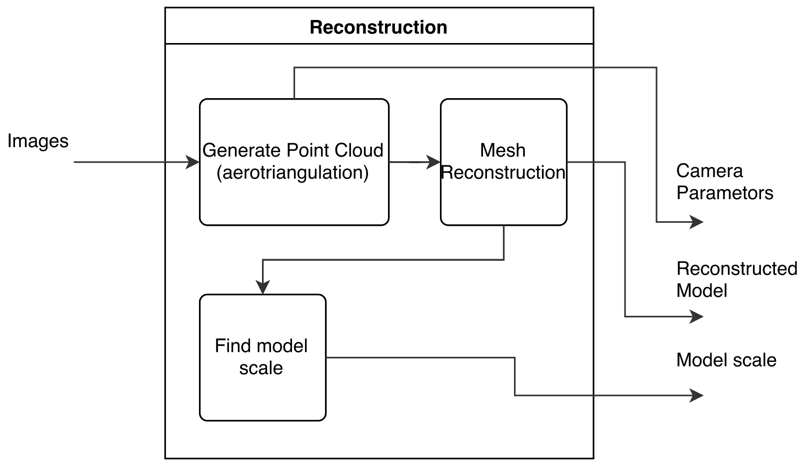


Figure 7.1: The flow of the reconstruction process.

7.1 Input Images

Following the guidelines from ContextCapture the light should stay consistent throughout the images, that said, the reconstruction from known and unknown images sometimes comes out better then the once only containing the unknown light. The reason for this is that the shadow areas produces less features to be tracked between images, and therefore gives a lower polynomial count area. This is visible in the reconstructed model where the lower polynomial count areas fits with the shadows on the texture.

7.2 Point Cloud

When the images are specified, the generation of a point cloud from these images is needed. The settings for this point cloud should be set for as good a result as possible, as the result of the reconstructed model has a large effect on the final reflectance. This step in ContextCapture is called aerotriangulation. The settings used can be seen in Table 7.1.

In this step all the information about the cameras positions and rotations are calculated and stored. This information is used later when setting up the irradiance simulation.

Keypoints density	Pair selection mode	Component construction mode
Heigh	Exhaustive	Multi-pass

Focal length	Principle point	radial distortion
Keep	Keep	Keep

Table 7.1: The settings that are not set to default in the aerotriangulation step.

7.3 Mesh Reconstruction

With the points of the surface of the physical object calculated, this cloud of points needs to be turned in to a mesh. This is done by creating a reconstruction, and set up the settings for this. In ContextCapturer the reconstructed model is smoothed. The amount of smoothing can not be reduced without a special settings file from the company responsible for ContextCapturer. With this file, the simplification can be decreased to 25%. This is used to keep as many details as possible. The amount this should be decreased to comes down to the physical object. It might be best to keep the smoothing high if the physical object only have smooth surfaces. After this, an production needs to be specified. This handles the file format that gets generated in the end. In this case the file type .obj is chosen as the texture is separated from the actual mesh file.

After the reconstruction, the 3D mesh and the texture is located in two different files and are ready for further processing.

7.4 Exported Information

The information about the cameras can be exported as an XML file. When this is done, the rotation of the cameras should be set to exported as Omega, Phi, Kappa angles. The XML file contains the Position, Rotation, and focal length of the cameras. All this information is later used to setup up the scene in relation to the reconstructed model for the irradiance simulation.

The resulting files from the construction should therefore be the Mesh(.obj file), the texture, and the XML file containing all the camera information.

7.5 Model Scaling

The reconstructed model's size is given in an arbitrary size. This is not relating to the physical object size in any way. To handle this, as the size is important because of the flash calibration, it is scaled by having something in the scene that has a known size. With a known size, a scaling factor can be calculated and used to scale up the position of the cameras, together with the scale of the reconstructed model in the irradiance simulation in order to achieve correct irradiance simulations.

$$S = \frac{\Delta_p}{\Delta_m} \tag{7.1}$$

Where S is the scale Δ_p is the measured distance between two predefined points on the physical object, and Δ_m is the measured distance between the same two points on the reconstructed mesh.

Chapter 8

Irradiance Maps Generation

With the outputs from the reconstructed model, all the information needed to generate the irradiance maps are available. A number of tools are developed to ease the scene creation, keeping most of properties that needs to be set once, out of these tools. These properties included the render settings. The final output should be the same amount of images as there is image pairs, containing the irradiance of each pixel of the model.

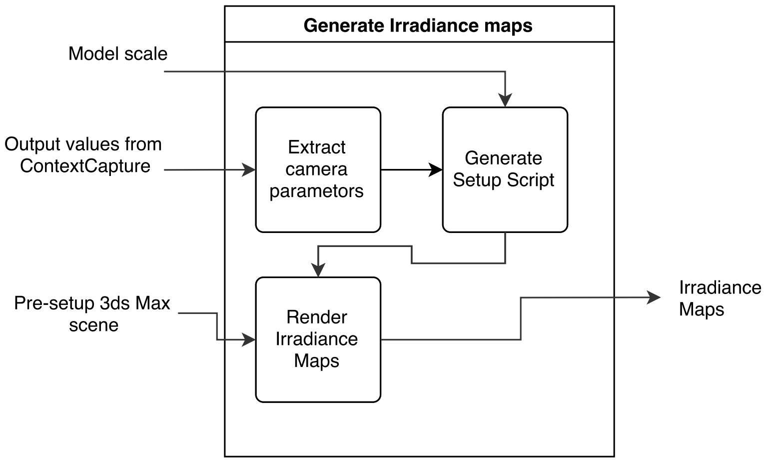


Figure 8.1: A flow diagram for the irradiance map generation.

8.1 Used Tools

To generate the irradiance maps two main tools are needed. A 3D modelling software and a rendering engine.

- **3ds Max 2016**
  3ds Max is chosen because of two main reasons. The software is a high quality 3D modelling software that is well known. Furthermore it is the software the writer of this paper had the most experience with. This could be replaced by many other 3D modelling software, though limited to the specific render engine that is chosen.
- **V-Ray**
  For the render engine some more specific capabilities was the reasoning behind the choice. The render engine needed to accept photometric parameters for light sources and cameras. The main reason that the default render engine, Mental Ray, is not used are the lack of a mesh based light source. This is needed in order to produce close to reality light output for the simulated flash. Furthermore the render engine is capable of producing photo realistic renders, making it capable of simulating light very accurately.

8.2 Setup Script

In order to set up all the cameras and flashes for the simulation, a script is used to translate from the XML file to a MAXsctript, capable of initialising and applying all the needed settings. Further more it takes the rotations and translates and converts them to 3ds Max coordinate system.

To let the script run, the base values for the camera settings need to be manually entered. This is used to set up the initial settings for the cameras. Further more the scaling factor is required in order to calculate the correct position for the cameras.

```
% Settings
fileName = 'SynthTestV4\Block_1 - AT -export.xml';
flashName = 'slaveFlash_';
focalLength = 83.6111;
sensorSize = 36;
iso = 1600;
shutterSpeed = 166;
fnumber = 32;
scalingFactor = 1;
```

A note on the focal length, ContextCapturer takes the original focal length and converts it to the focal length for a 35mm sensor. To make the irradiance maps align with the images, it is the converted focal length that is needed.

The first part of the XML file is skipped, until the first photo is reached. For every photo the script then pharses for the different parameters and gathers them in appropriate arrays.

The arrays are then sorted in the order of the input images to match the flashes and the cameras. From this information a MAXscript file is written to spawn the cameras and flashes. The script uses the object named masterFlash to clone for every flash, this object is set up in advance with the correct settings.

```
% Get the Master Flash
fprintf(file,'masterFlash = getNodeByName "masterFlash"\n');
fprintf(file,'\n');

for i = 1:length(imagePath(:,1))
% Spawn new camera
    fprintf(file,'cam = VRayPhysicalCamera name:"%s"\n',imagePath(i,:));

% Set camera parametors
    fprintf(file,'cam.focal_length = %f\n',focalLength);
    fprintf(file,'cam.film_width = %f\n',sensorSize);
    fprintf(file,'cam.shutter_speed = %f\n',shutterSpeed);
    fprintf(file,'cam.ISO = %f\n',iso);
    fprintf(file,'cam.f_number = %f\n',fnumber);
    fprintf(file,'cam.whiteBalance = color 255 255 255\n');

% Rotate camera
    fprintf(file,'cam.transform = rotateYPRMatrix %f %f %f\n',omega(i),-phi(i)+180,-kappa(i)-90);

% Set position of camera
    fprintf(file,'cam.pos = [%f,%f,%f]\n',-y(i),x(i),z(i));

% Spawn Flash
    fprintf(file,'maxOps.CloneNodes masterFlash newNodes:&flash\n');
    fprintf(file,'flash[1].name = "%s%i"\n', flashName,i);

% Rotate Flash
    fprintf(file,'flash[1].transform = rotateYPRMatrix %f %f %f\n',omega(i),-phi(i)+180,-kappa(i)
        -90);

% Position Flash
    fprintf(file,'flash[1].pos = [%f,%f,%f]\n',-y(i),x(i),z(i));

    fprintf(file,'\n');
end
```

8.3 Scene Setup

Before loading the script to spawn all the cameras and the flashes, the 3ds Max scene needs to be set up. The object that is reconstructed by ContextCapture needs to be imported. When importing .obj files in to 3ds Max there is an option to flip the z-axis. This should be turned off. Further more the scale of the imported object should be set to the same as in the setup script. The model is placed in the environment, and should not be moved. This would otherwise ruin the cameras and flashes alignment with the object. The object should be rotated 90 degrees around the z-axis, this is to align everything.

The next step is to add the master flash that will be cloned. The model should be loaded in to the scene, and then a V-ray mesh light should applied. The light source should have the intensity found from the flash calibration.

The settings of the render engine should be set to produce a high quality render, and as found, the global illumination should be turned on. The specific settings for the render is out of the scope of this explanation.

8.4 Rendering

A MAXscript is used to render the irradiance maps. The reason is that some specific actions needs to be executed in between each render. More specifically, the flashes needs to be turned on and off in relation to the camera being rendered.

The script contains some global variables specifying the name of the flashes, the cameras name, the output path, and the output file name.

```
----  SCRIPT SETTINGS  ----
global usedCameraName = "RealWorldTestRenders_FlashAmbi_";
global usedLightName = "slaveFlash_";

global out_path = sysinfo.currentdir + "\\renders\\";
global out_irradianceFileName = "SynthTest_irradianceMap_"
```

Before the rendering begins, the cameras and flashes are loaded in to an array. This is done by finding all the objects in the scene matching the name pattern. The number of cameras is used to define the amount of renders that should be created. Then all the flashes gets located in the same manor and are turned off to make sure non of them are adding light.

```
--      get number of cameras to render
cams = for o in objects where (matchPattern o.name pattern:(usedCameraName + "*") ignoreCase:on ==
     true and matchPattern o.name pattern:(usedCameraName + "*.Target") ignoreCase:on == false)
     collect o
numberOfCams = cams.count

-- Get all the flashes to turn them off
flashs = for o in objects where matchPattern o.name pattern:(usedLightName + "*") ignoreCase:on ==
     true collect o
for f in flashs do f.on = off
```

For each render pass, the camera and flash are located. The flash is turned on and the camera is set to the active render camera. The output path is defined from the global variables and then numbered. The output path is set as the rendered files name and the file is saved with a gamma of 1.0. The image is then discarded to release memory, and the light is turned off for the next render pass.

```
for i = 1 to numberOfCams do
(
        -- Get camera
        curCamName = StringStream ""
        format "%%.tif" usedCameraName (formattedPrint i format:"03d") to:curCamName
        curCam = getNodeByName (curCamName as string)

        -- Get flash
        curLightName = StringStream ""
        format "%%" usedLightName (formattedPrint i format:"d") to:curLightName
        curLight = getNodeByName (curLightName as string)

        --==== Render image =====-
```

```
        -- Turn on flash
        curLight.on = on

        -- Render Image
        if renderImage then outImage = render camera:curCam vfb:false

        -- Create the save path
        outString = StringStream "";
        format "%%%.tif" out_path out_irradianceFileName (formattedPrint i format:"03d") to:
            outString

        -- Save Image
        outImage.fileName = outString as string
        save outImage gamma:1.0
        close outImage

        curLight.on = off
)
```

It should be noted that the render engine should be set up to save the images after a render is done, as this dictates the format of the image. The locations in the script, is the location the images is actually save to.

After this, all the irradiance maps are generated and saved to the predefined location. The only part missing is to enable the reflectance calculation by isolating the known light.

Chapter 9

Isolating Known Light

With the irradiance maps calculated, the last part needed is to calculate the reflectance, of the images only contianing the known light.

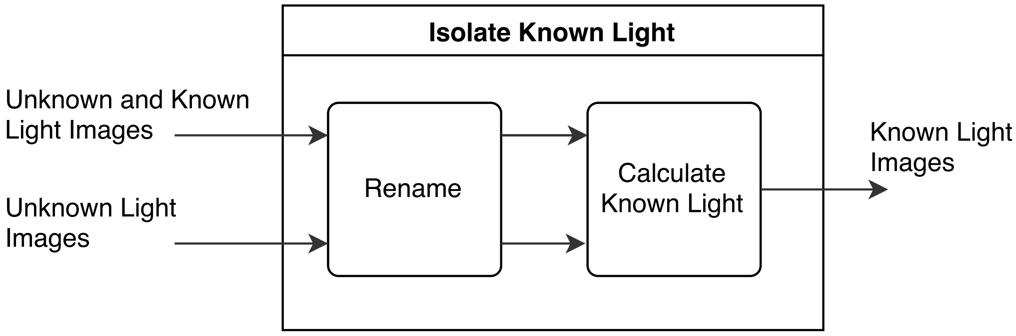


Figure 9.1: Flow diagram describing the isolation of light.

9.1 Input Images

To begin the process, the acquired images are renamed to make it easier to pair them. This give a series of images, numbered, with only the unknown light, and a series of images with unknown and known light.

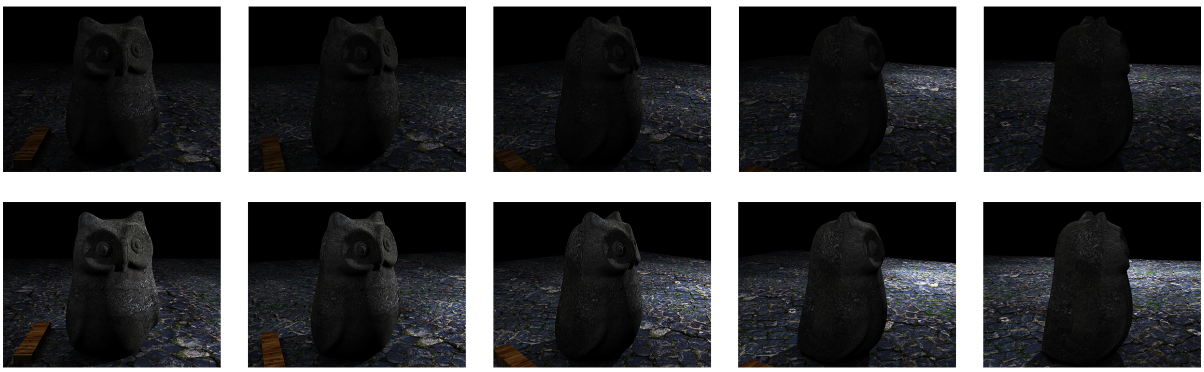


Figure 9.2: An example of the images captured.

9.2 Calculating the Known Light

To isolate the known light of all the image pairs, a small Matlab script was created. The biggest part of the script is the setup for all the paths and names. For the actual calculations, the two image are loaded, and the unknown light image is subtracted from the image containing the unknown and known light.

```
% Load images
unknown_img = im2double(imread(sprintf('%s%s%03d.%s',ambiOnlyPath,ambiOnlyName,i,imgExtension)));
knowUnknown_img = im2double(imread(sprintf('%s%s%03d.%s',flashAmbiPath,flashAmbiName,i,imgExtension
    )));

% The calculations to get the known light image
know_img = knowUnknown_img - unknown_img;
```

This is done for all the images and results in images only containing the known light. All the needed data is then ready for the calculations of the reflectance.

Chapter 10

Calculating Reflectance

With the isolation of the images only containing the known light, and the irradiance maps, all the information needed to calculate the reflectance is present.

To calculate the reflectance the image containing the known light should be divided with the irradiance map. To not loss data when doing the division, the images is loaded and then converted to double values. This is done in the same script as the isolation of the known light, it is therefore only the irradiance map that is loaded from a file.

```
% Load irradiance map
irImg = im2double(imread(sprintf('%s%s%03d.%s',irradiancePath,irradianceName,i,imgExtension)));

% Calculate reflectance
reflectance = known_img ./ irImg;
```

This is done for all the images in order to calculate the reflectance for every view of the physical object. This ables the regeneration of the texture for the reconstructed model.

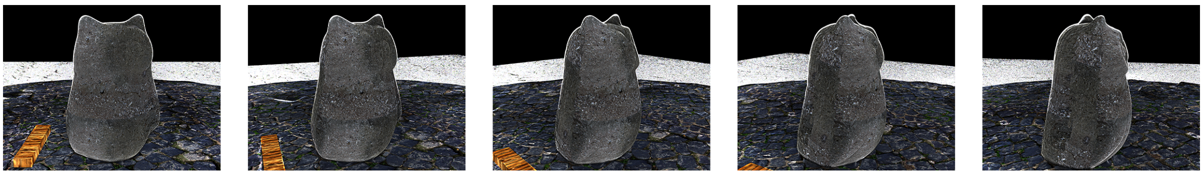


Figure 10.1: The calculated reflectance.

Chapter 11

Re-Textur Object

With the reflectance calculated, the texture for the reconstructed model needs to be generated with the reflectance images. To do this ContextCapture needs to be tricked in to believing that the texture was never created, and that the new images containing the reflectance, is the original images the reconstructed model was created from.

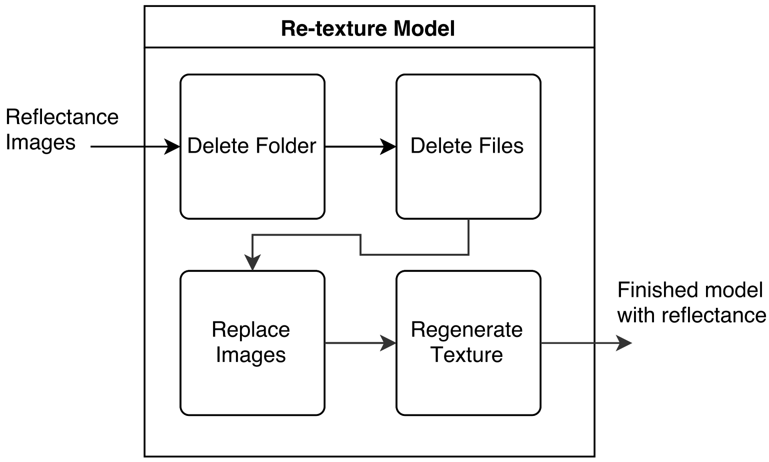


Figure 11.1: The flow of the re-texturing process.

To do this, ContextCapturer creates temporary files in the project folder in order to keep track of the progress of the reconstruction. Some of these files needs to be deleted for ContextCapture to regenerate the texture for the object, without recreating the mesh. It is specifically the folder named *images* inside the block folder containing the aerotriangulation. This entire folder should be deleted. Together with the files, *textured_geometry.a3d and textured_geometry*.jpg*, where the star indication a number as there might be more then one. These files are located inside the folder with the name of the reconstruction, where a folder called Model is located. With these files deleted ContextCapturer does not know that the texture has already been generated, and the next time a production is created, the texture will be generated again, but still from the original images.

To solve this, the original images should be replaced with the calculated reflectance images. They should be named exactly the same and be located in the same folder. These images will then be used to create the texture for the new production of the reconstructed model.

With this, the final model is created, with the calculated reflectance, ready to be used in any 3D software, real time or rendered.

Part III

Evaluation

Chapter 12

Acceptance Testing

12.1 Synthetic Test

The synthetic test is conducted in order to evaluate the proposed solution in a simulated environment before conducting a real world test. As this test is conducted in a much more controlled environment, it enables the detection of errors in the solution that might not be observed otherwise.

12.1.1 Test Setup

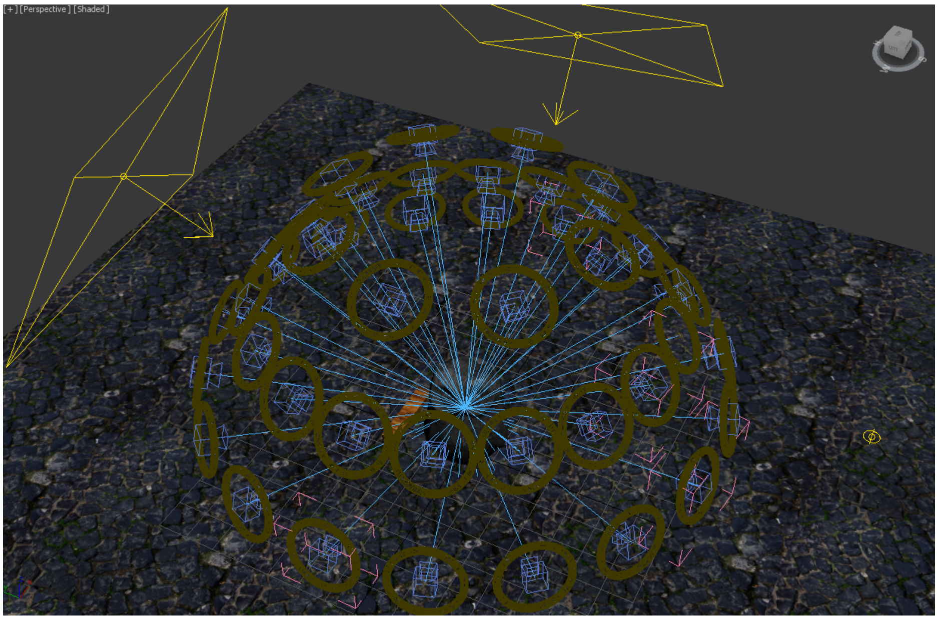


Figure 12.1: The test setup for the synthetic test

The test is set up to mimic a real world test. This means creating an environment with some ambient light that simulates the unknown light. Creating a series of camera and flash pairs to render out images from these different positions. All the flashes are clones from a master flash making them exactly the same. The object used is a scanned 3D model of a owl giving it a high amount of small details. This 3D scan has no texture, this is therefore created with a high resolution image containing a lot of small details.

12.1.2 Test Data

The test data consists of 90 images from 45 different camera positions in the simulated environment. Two images are captured from each camera position as an image with the unknown light and an image with the unknown and known light is needed. Further more the individual channel for the diffuse colour is saved separately. This channel is used as the ground truth.

Two textures is generated for the reconstructed model, one which is created from the ground truth images. And one that is created from the calculated reflectance images. These textures are created in exactly the same way, for the same model. This is possible as the areas taken from the images used to reconstruct the object is saved, and the software used can be forced to recreate this final texture, from the same regions, making these two textures comparable.

12.1.3 Test Results

The test results for the individual images are created by comparing each image with the ground truth for each camera position. The differences, and therefore the error, is then calculated. Looking at the mean over the image

Figure 12.2: An error bar plot showing the results from test of the individual images.

sequence, the mean gets closer to zero, but the interesting part is that some kind of grouping is present in the standard deviation over the sequence. Three different grouping can be observed. The reason for these groups can be the layers the cameras is located at. The cameras are spread over three circles. These circles of cameras is offset in heigh, and therefore have a steeper angle on the model. These groups can bee seen in Figure 12.2. This meant that the highest cameras covered a larger portion of the area that is able to be reconstructed. This in turn gives a smaller error as there is less of the image missing compared to the lower cameras. The lowest cameras are the first in the sequence and the highest ones are the last. The error can be seen in Table 12.1.

This might in turn not correlate to the final result on the texture as only parts of these images is going to be used to rebuild it. The areas creating a high error is the ones that could not be reconstructed and therefore would not be present in the final texture.

Looking at the error of the texture, the error is smaller then the worst images in the individual sequence. This confirmed the theory about the worst areas not getting used in the texture. That said, the results have an mean error of -0.0816 and standard deviation of 0.1156. This indicates a rather big span of the error observed. This can be seen in Figure 12.3 and Table 12.2. Visually inspecting the textures, it is seen that there is some edges of the reconstruction of the floor, not the object of interest. As the floor is something that would be removed from the model for further use of the reconstructed object, this might not be a problem in the final result.

As this method focusses on the visual perception of the object, this is also evaluated. Evaluating this is also a way to identify whether the texture of the object of interest is acceptable, and weather the main error is present in

Mean Error										
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Image 1-10	-0.0776	-0.0774	-0.0727	-0.0703	-0.0591	-0.0527	-0.0616	-0.0645	-0.0693	-0.0738
Image 11-20	-0.0837	-0.0807	-0.0800	-0.0706	-0.0533	-0.0685	-0.0675	-0.0732	-0.0421	-0.0442
Image 21-30	-0.0379	-0.0304	-0.0236	-0.0248	-0.0308	-0.0397	-0.0350	-0.0400	-0.0479	-0.0435
Image 31-40	-0.0399	-0.0345	-0.0314	-0.0338	-0.0368	-0.0399	-0.0067	-0.0079	-0.0000	-0.0064
Image 41-45	-0.0101	-0.0112	-0.0015	-0.0043	-0.0066					

Standard Deviation										
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Image 1-10	0.2261	0.2339	0.2301	0.2325	0.2258	0.2182	0.2275	0.2277	0.2265	0.2270
Image 11-20	0.2395	0.2350	0.2382	0.2337	0.2410	0.2358	0.2302	0.2328	0.1671	0.1694
Image 21-30	0.1653	0.1617	0.1614	0.1611	0.1651	0.1816	0.1636	0.1697	0.1812	0.1680
Image 31-40	0.1626	0.1654	0.1704	0.1679	0.1683	0.1678	0.0849	0.0846	0.0722	0.0809
Image 41-45	0.0883	0.0845	0.0753	0.0790	0.0849					

Table 12.1: The mean error and standard deviation from the individual images.

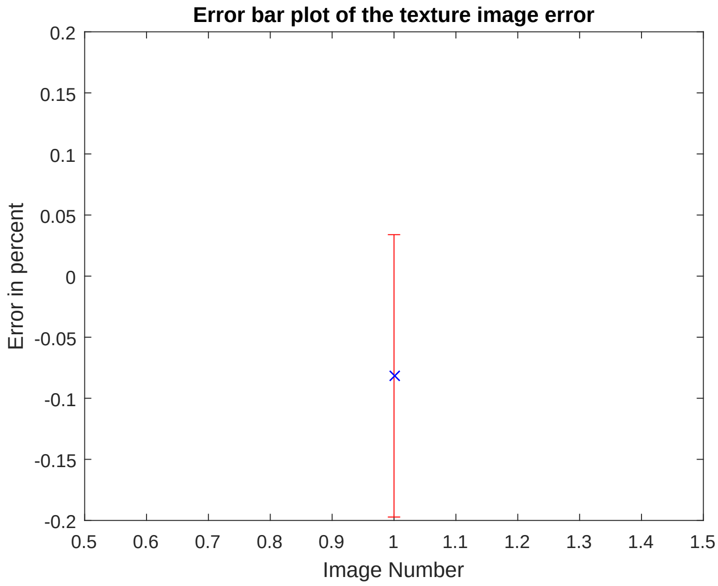


Figure 12.3: An error bar plot showing the results from test of the texture image.

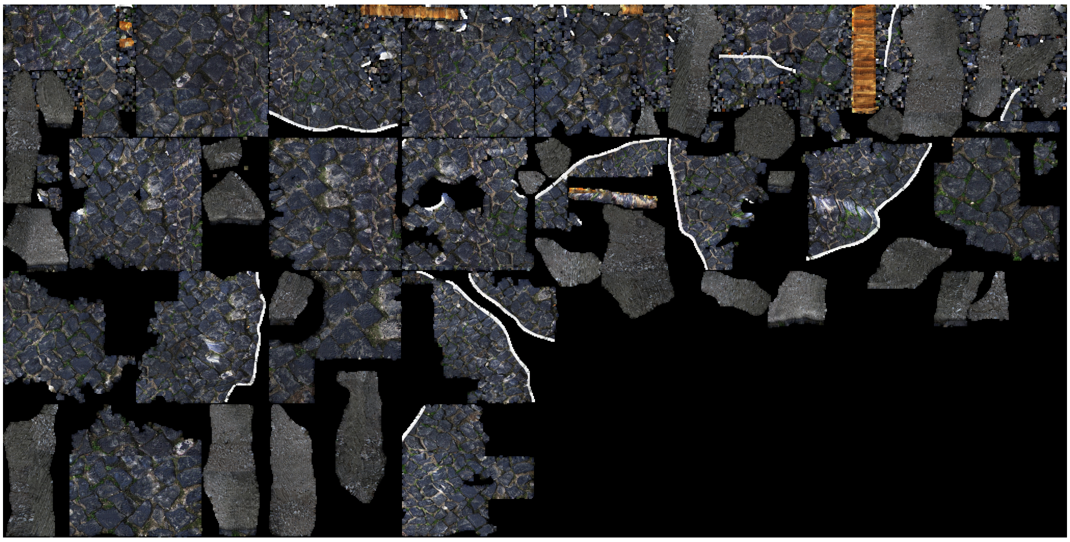
the floor of the model.

12.1.4 Test Results

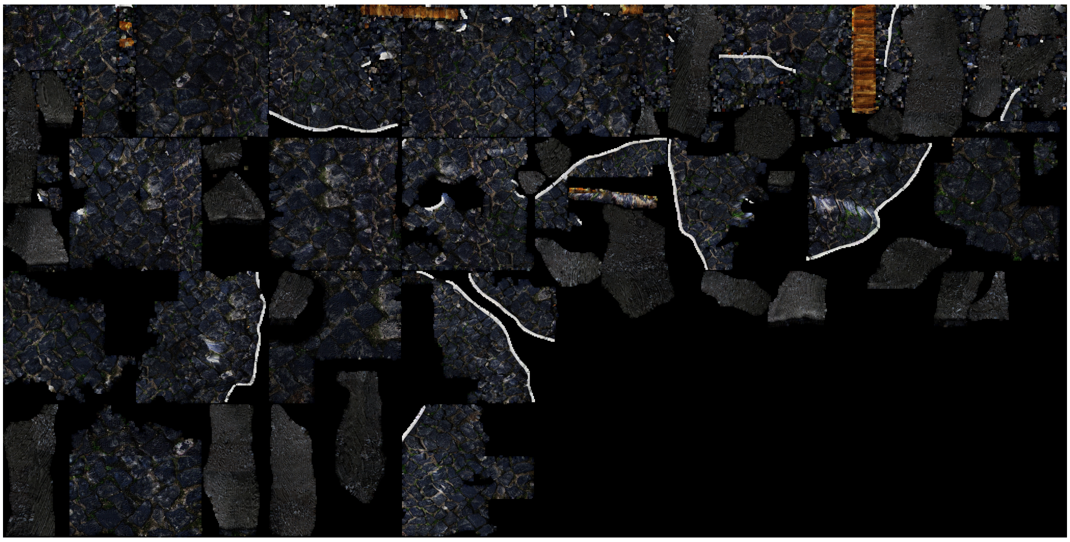
Looking at the error map from the texture, it is seen that the edges of the floor had a high error. The box with a wood texture added for unit reference, also had a high error. This supported the claim about the error stemming from other places then the object of interest.

Mean Error	Standard Deviation
-0.0816	0.1156

Table 12.2: The results from the calculated reflectance texture versus the ground truth reflectance texture.



(a) An image is the resulting texture of the calculated reflectance.



(b) An image showing the absolute error for every pixel between the calculated and ground truth texture. Notice the white strips of high error at the edges of the platform.

Figure 12.4: Showing the resulting calculated texture (a), and the difference map from the ground truth texture (b).

From the visual test results it can be observed that there is a small difference between the ground truth and the calculated reflectance. The calculated reflectance is still a lot better then the original reconstructed texture. The texture is very dark, this is due to it being created from underexposed images that only contain the ambient light. On closer inspection a dark side is also observed, this is due to the light coming from one direction and creating a shadow. This is one of the main problems by reconstructed object in general. On the calculated reflectance this shadow is not present.

12.1.5 Conclusion

From the test it is concluded that there is a significant improvement from the reconstructed texture and the calculated reflectance. This is seen when looking at the error of the calculated reflectance. For the texture only a mean error of -0.0816 with a standard deviation of 0.1156, where some of it can be due to error at the edges of the reconstructions. These errors is observed in the image containing the error per pixel. The removal of shadows from the texture are successfully with the final appearance of the object is a little darker then the ground truth.

(a) A visual render of object with different textures. The HDRI is Grace Cathedral, San Francisco.

(b) A visual render of object with different textures. The HDRI is The Uffizi Gallery, Florence,

(c) A visual render of object with different textures. The HDRI is a newer Grace Cathedral, San Francisco,

Figure 12.5: Test renders for visual inspection of the calculated reflectance compared to other textures.

12.2 Realworld Test

The real world test is conducted in order to evaluate the proposed solution in a real world scenario. This test will evaluate the final results and conclude on the problems that appear, in order to later improve the solution. As this is a real world test, there is no ground truth to test with. Therefore this test will mainly consist of a visual inspection of the results.

12.2.1 Test Setup

The test setup was created to be consistent with the precess a user would use. Though within an controlled lighting setup where three studio flashes was used to simulate the unknown lighting condition. This was done in order to create harsh shadows on the physical object, in order to showcase the removal of these. The test object was a bird bath made out of granite. The granite had small specular spots, to account for this, an polarizing filter was attached to the lens.

12.2.2 Test Data

The test data consisted of 104 positions around the object. These was roughly divided in to three different heights to capture the object from different angles. This created an image set of 208 images in total.

12.2.3 Test Results

The resulting texture of the model has a lot of bright spots. These are generally located on the lower part of the bird bath. This can clearly be seen in Figure 12.6.

Figure 12.6: The texture created from the calculated reflectance images.

These highlights are very visible in the renders where the object has been used in a simulated environment. Looking at the shadows, these are gone, and is no longer part of the texture. Further more it can be noticed that the model is still a little dark.

In order to create any form of usable reflectance images, the irradiance maps needed to be scale with a value of six.

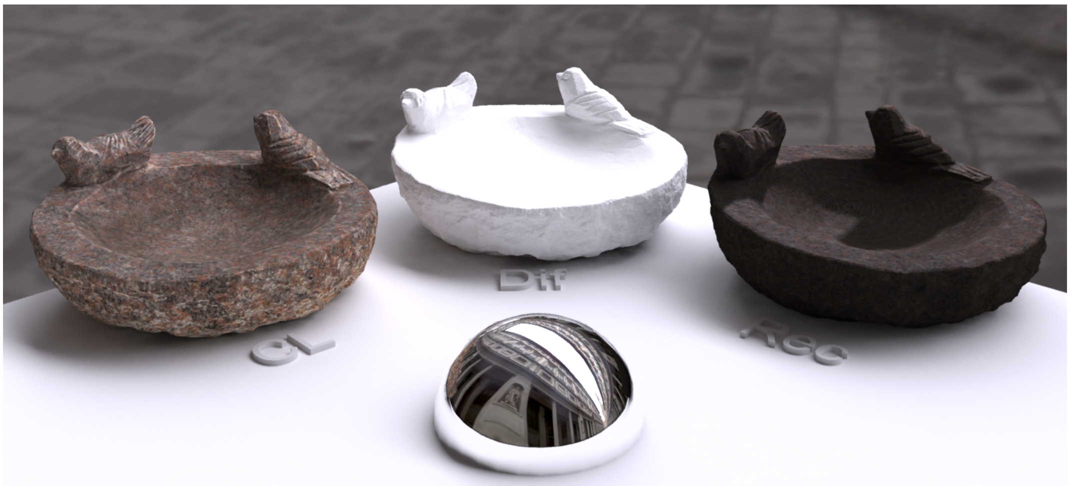
12.2.4 Conclusion

Looking at the results, there is some bright spots on the underside of the model. This could be due to several things. First of the image of the physical object and the simulated irradiance could be misaligned. This would create brighter spots, especially at points where the surface is rough. The second possibility is that there is missing light in the irradiance maps at these points. This would in therm make the spot brighter on the reflectance.

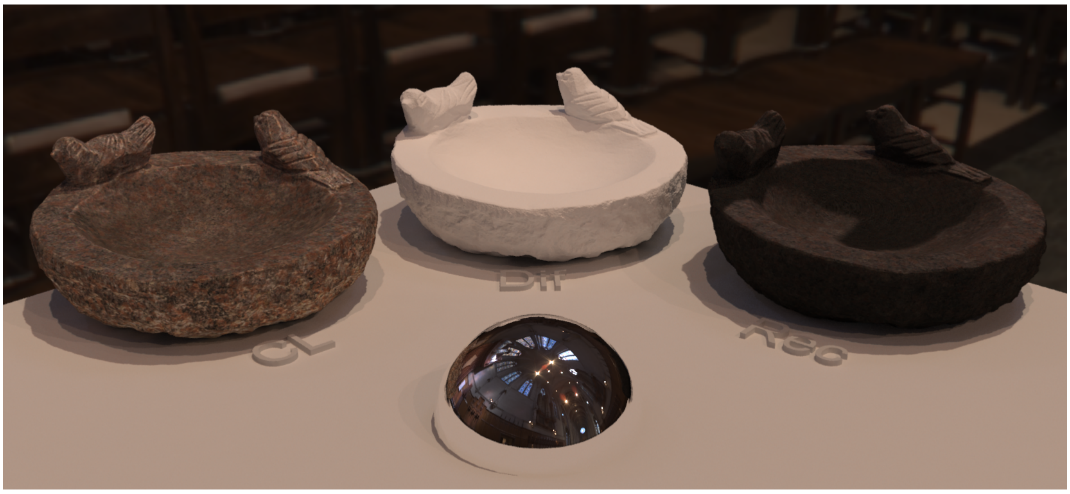
Looking at the need to scale the irradiance in order to create non-white images, this is most likely due to an error in the calibration of the light source. This should be revisited with a more robust testing method in order to reveal this problem. This problem however does not invalidate the test, as this is limited to the right intensity, but not anything else.

One of the requirements of the system was that the shadows got removed. Looking at the results, this does in fact happen. There is no more shadow on the inside of the bird bath, as there is in the original caption. This shows that the main principles work, but there is some error introduced at some point in the method.

(a) A visual render of object with different textures. The HDRI is Grace Cathedral, San Francisco.



(b) A visual render of object with different textures. The HDRI is The Uffizi Gallery, Florence,



(c) A visual render of object with different textures. The HDRI is a newer Grace Cathedral, San Francisco,

Figure 12.7: Test renders for visual inspection of the calculated reflectance compared to original texture from the reconstruction. Left is with calculated reflectance. Middle is with a diffuse white material. Right is with captured images.

Chapter 13

Conclusion

The main problem with current reflectance capture methods are the specialised and very limited equipment used. This makes reflectance capture something with few use and know about. It is therefore the goal to create a method of capturing reflectance with off the shelf hardware. This would enable users to capture the true reflectance of an reconstructed object. The object can then be used in any 3D computer graphics environment. Having recovered the reflectance, in term means that any effect of light at the time of acquisition is negated, giving a clean model.

The proposed method is limited to diffuse surfaces. This is foremost because of the requirement of structure from motion. But also because the underlying model for the calculation of the reflectance assumed an Lambertian surface. The first parts to enable reconstruction of the reflectance is, two acquired images, one with unknown light and one with known and unknown light. This enables the isolation of the known light due to light being additive.

With the acquired images, a reconstruction of the physical object can be created. In addition to the resulting model, the reconstruction also calculates the cameras position and rotation in relation to the model.

This information is used when the irradiance needs to be simulated. The irradiance of an object is the incoming light of a given surface. Having this, the inverse process of the interaction of light on the physical object can be done. To simulate the irradiance, the light source used as the known light source needs to be calibrated. This is in order to make the simulated known light correlate with the physical known light. With a calibrated light, the irradiance for each pixel of the images can be calculated.

With the isolated known light, and the simulation of the irradiance on the object from the light, the reflectance can be calculated. This results in the irradiance for each image containing the isolated known light.

To create a texture for the reconstructed model, the software generating the reconstructed model, can be tricked in to thinking it needs to create the texture. With this, the resulting model is texture with the reflectance, ready for use in a 3D environment.

The proposed system consisted of six modules, each responsible for a small step. The modules are image acquisition, light isolation, object reconstruction, illumination simulation, reflectance calculation, and re-texturing. This is pared with some translations scripts to ease the setup process of the illumination simulations. The illumination simulation is done in 3ds max with VRay in order to simulate a mesh light source for precise known light calibration.

A test is run on a proof of concept setup with very limited variables. This test showed that global illumination should be turned on, together with the observation of precision limitations in 8 bit images. This limitations of 8 bit images could be improved by having different exposures for the unknown light image, and the know and unknown light image. This made the isolation of the known light more precise, which translated in to a more precise reflectance.

The test of the camera revealed a reliable camera with a linear sensor response. The expected response for setting changes on the physical camera mostly correlated with the theoretical values. It should be noted that the camera showed rounded versions of the actual values used. The used values is stored in the image files as EXIF data.

The full synthetic test showed error at the edges of the reconstructed floor, these areas are not of interest as these are in principle not part of the object of interest. Including the floor, the mean error was -0.0816 with a standard divination of 0.1156. The shadows produced by the unknown lights, are removed with the proposed solution. This makes the resulting model more believable as now unwanted shadows or light is applied.

In the final real world test, the main purpose is to test the entire solution as the user would. The resulting model from this test had again the shadows removed. It is noted that the calibration of the flash is wrong and gave very bright results. As this is a matter of intensity the irradiance is manually scaled. This then created a reflectance with bright spots on the under side of the bird bath. This could be due to multiple problems, but mainly the reconstruction of the model. This could also be related to the flash calibration. Using the resulting model in an simulated environment, the bright spots is generally not noticeable, and the model did not seem out of place.

Chapter 14

Future Work

The proposed method has shown promising results, together with problems that needs further investigation.

The most severe problem is the results from the real world test showing a high amount of error on the underside of the test model. This is not acceptable as this might lead to a even worse looking model, then without this model. This problem is however an problem that, if solved, is the only thing major problem. to solve this, test regarding the spread of light should be created. Further more testing the reconstruction up against some ground truth, could result in the reason as the errors can come from bad reconstructions. The last possible solution to pursue, could be the alignment of the images. This might be able to be solve with calculations in order to align these images even though they was miss aligned in the first place.

The problem found in the real world test, regarding flash calibration, can have different reasons. The problem could be tested by having images from different angles and distances to the calibration target. This would give better data do test up against to make sure the movement of the camera would not be a problem. This might en turn provide an answer of to low irradiance map. Another reason could be the known target for the calibration, was not precise enough. Finding a pre calibrated target, with a more precise reflectance, could also solve this problem.

Looking past the immediate problems with the system, it is interesting to look at additional capabilities to further increase the reconstructed quality.

The main limitation of the system is that is does not capture specular reflections. This could be solved by having a polarized light source, and take one image more then currently. The image should consist of the polarization filter in a position where it does not remove specular reflections. This would result in an image with specular reflections added. Again using the principle of additive light, the two images could be subtracted, resulting in an image only containing specular reflections. This could the same way as re-texturing be mapped to the reconstructed model.

To eliminate the problems of having to calibrate the light source, a custom light source, that would fit on the hotshoe of an DSLR, could be made. This would keep the more versatile and mobile nature of this solution, while limiting a problem. This would also enable the polarization of the light source, and in general make the might more predictable.

Part IV

Appendix

Appendix A

Helmholtz Reciprosity

The next step was to substitute the Lambertian BRDF in to this generalised pixel value function. In order to do this, a principle called Helmholtz Reciprocity needed to be used. Helmholtz reciprocity defined that, if an ray of light is originating from the point A, and captured at point B, the source and observer can change place, and the result would be the same. This is illustrated in Figure A.1. An simple example of this would be a mirror, where if you can see an observer, the observer can see you.

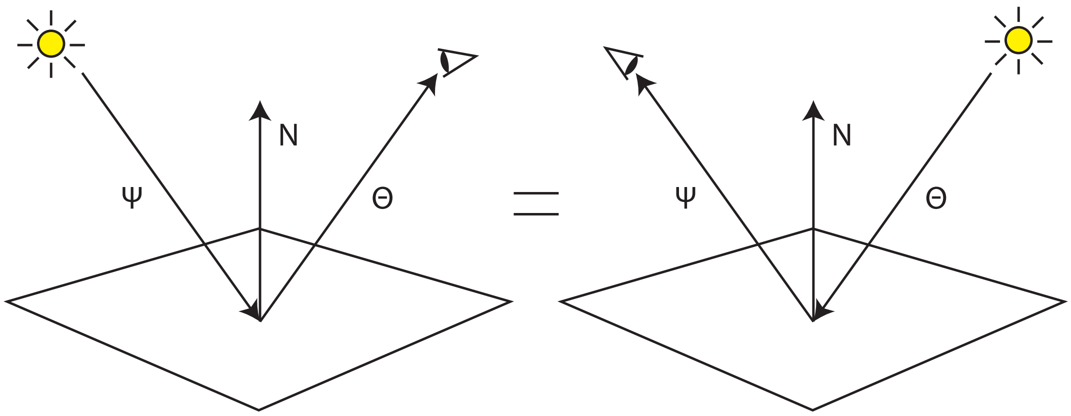


Figure A.1: The figure are showing Helmholtz reciprocity, that the light is the same if the receiver and emitter changes place.

Keeping this in mind, the notation of the BRDF could be changed from a direction dependent notation to a direction independent notation, this was defined in Equation A.1.

$$f_r(x, \Theta \to \Psi) = f_r(x, \Theta \leftrightarrow \Psi) \tag{A.1}$$

With this the pixel value equation ((Equation 2.9) became direction independent, see Equation A.2.

$$P = S \cdot \int_{\Omega x} f_r(x, \Psi \leftrightarrow \Theta) dE(x \leftarrow \Psi) \tag{A.2}$$

With this notation the Lambertian BRDF could now be substituted in to the pixel value equation (Equation 2.9).

$$P = S \cdot \int_{\Omega x} \cdot \frac{1}{\pi} \cdot dE(x \leftarrow \Psi) \tag{A.3}$$

Appendix B

Image Based Modelling

In order to create a model of a physical object, image based modelling is used. The exact theories that is used to create the model, will not be covered as a proprietary software called ContextCapturer is used. The general theory of how image based modelling works will however be covered to give a basic understanding of what happens.

The first step of images based modelling is to capture a lot of images, the more complex the object, the more images is needed. After the images are captured, the images needes to be correlated with each other. To do this, features is extracted. A feature is a point, or area, of the image that is unique. The uniqueness is important as the feature should be able to be recognised in another image.

In this field there exists different problems, for example, what if the image is darker, rotated, or it is captured further away making it smaller. This creates a necessity, in some situations, for scale and rotation invariant, and resistance to illumination, noise, etc. Further more, for the use of recreating physical object, it is also needed to handle view point changes.

This is split up in to two steps. The first step is to find these unique features. And the second is to describe these features.

To find features several methods exists.

- Harris Corner Detection (Chris Harris, 1988)
- Shi-Tomasi Coner Detection (Shi and Tomasi, 1994)
- SIFT (Scale-Invariant Feature Transform) (Lowe, 1999)
- SURF (Speeded-Up Robust Features) (Bay et al., 2006)
- FAST (Rosten et al., 2010)
- BRIEF (Binary Robust Independent Elementary Features) (Calonder et al., 2010)
- ORB (Oriented FAST and Rotated BRIEF (Rublee et al., 2011)

When the features are found, they needed to be described in a way to make them more general to enable recognition in a new image. Several methods can be used to extract features, some of these are:

- SIFT (Lowe, 1999)
- SURF (Bay et al., 2006)
- HOG (Dalal and Triggs, 2005)

With this, the features of the each image can be matched to each other. This enable the calculations of the position and rotation of the images. Further more the the matched points can be triangulated to enable the feature to be placed in 3D space.

Given enough of these points, a mesh can be generated to estimate the shape of the physical object. To generate the texture of the object, the images are projected on to the model. A method is applied to choose the best projection in the cases where multiple projections overlap.

Figure B.1: The resulting pointcloud. Each square is a matched and triangulated feature.

Figure B.2: The resulting reconstruction without texture.

The end result it then a model generated from images of a physical object, with a texture extracted from the images.

Bibliography

Alexander, O., Rogers, M., Lambeth, W., Chiang, J. Y., Ma, W. C., Wang, C. C. and Debevec, P. (2010), 'The digital emily project: Achieving a photorealistic digital actor', *IEEE Computer Graphics and Applications* **30**(4), 20–31.

Autodesk, I. (2016), '123d catch'.
URL: *http://www.123dapp.com/catch*

Bartman, F. L., Chaney, L. W. and Surh, M. T. (1964), Reflectance of kodak white paper, Technical report, The University of Michigan, College of Engineering.

Bay, H., Tuytelaars, T. and Gool, L. V. (2006), Surf: Speeded up robust features, *in* 'Proceedings of the ninth European Conference on Computer Vision'.

Calonder, M., Lepetit, V., Strecha, C. and Fua, P. (2010), Brief: Binary robust independent elementary features, *in* 'Proceedings of the 11th European Conference on Computer Vision: Part IV', ECCV'10, Springer-Verlag, Berlin, Heidelberg, pp. 778–792.
URL: *http://dl.acm.org/citation.cfm?id=1888089.1888148*

Canon (2010), 'Ef 70-300mm f/4-5.6l is usm'.
URL: *http://www.canon.dk/lenses/ef-70-300mm-f-4-5-6l-is-usm-lens/*

Canon (2015), '5ds'.
URL: *http://www.canon.dk/for_home/product_finder/cameras/digital_slr/eos_5ds/*

Chris Harris, M. S. (1988), 'A combined corner and edge detector'.

Coffing, D. (n.d.), 'dcraw'.
URL: *https://www.cybercom.net/ dcoffin/dcraw/*

Dalal, N. and Triggs, B. (2005), Histograms of oriented gradients for human detection, *in* 'Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01', CVPR '05, IEEE Computer Society, Washington, DC, USA, pp. 886–893.
URL: *http://dx.doi.org/10.1109/CVPR.2005.177*

Datacolor (2016), 'Spydercheckr color data'.
URL: *http://cdn.spyder.datacolor.eu/wp-content/uploads/datacolorproductliterature/SpyderCheckr%20Color%20Data.pdf*

datacolor (n.d.), 'Spydercheckr'.
URL: *http://spyder.datacolor.com/portfolio-view/spydercheckr/*

Dutré, P., Bala, K. and Bekaert, P. (2006), *Advanced global illumination*, second edn, A K Peters, Ltd.

Godox Photo Equipment CO.,LTD (n.d.), 'Witstro ar400'.
URL: *http://www.godox.com/EN/Products_Camera_Flash_Witstro_AR400_Powerful_Ring_Flash.html*

Jensen, T. J. (2015), *Budget Astrophotography: Imaging with Your DSLR or Webcam*, Springer.

Kerr, D. A. (2016), 'Apex—the additive system of photographic exposure'.

Lowe, D. G. (1999), Object recognition from local scale-invariant features, *in* 'Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on', Vol. 2, pp. 1150–1157 vol.2.

Rosten, E., Porter, R. and Drummond, T. (2010), 'Faster and better: A machine learning approach to corner detection', *IEEE Trans. Pattern Analysis and Machine Intelligence* **32**, 105–119.
URL: *http://lanl.arXiv.org/pdf/0810.2434*

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G. (2011), Orb: An efficient alternative to sift or surf, *in* 'Proceedings of the 2011 International Conference on Computer Vision', ICCV '11, IEEE Computer Society, Washington, DC, USA, pp. 2564–2571.
URL: *http://dx.doi.org/10.1109/ICCV.2011.6126544*

Shi, J. and Tomasi, C. (1994), Good features to track, *in* 'Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on', pp. 593–600.