Perceptual Evaluation of Photo-Realism in Real-Time 3D Augmented Reality

Master Thesis
30-05-13

Mathias Borg
Martin M. Paprocki

Aalborg Univeristy
School of Information and Communication Technology

AALBORG UNIVERSITY

Medialogy 10th Semester
School of Information and
Communication Technology - Medialogi - Aalborg
Sofiendalsvej 9 – 11
DK-9200 Aalborg
Tlf. 99 40 24 84
Fax 99 40 97 88
www.medialogi-aalborg.dk

Titel: Perceptual Evaluation of Photo-Realism in Real-Time 3D Augmented Reality

Theme: Master thesis
Project Period: 04.02.2013 – 30.05.2013

Project Group: mta131035

Authors:

Mathias Borg

Martin M. Paprocki

Prints: 2
Pages: 86
Appendices: 8 and a CD

Synopsis:

We present a pipeline for creating photo-realism of three-dimensional augmented objects, as well as a perceptual evaluating of the virtual scenes. Most of the content is rendered in real-time, while the augmented reality solution performs constant tracking of a marker. A setup utilizing different lighting conditions is created and artefacts of the video-feed are simulated. Different parameters affecting the realism are evaluated. These are screen space artefacts, shadows, lights, highlights and geometry. The results show that silhouettes of the shadows and the geometry and highlights on specular objects are important, as well as the simulation of noise, for creating a photo-realistic augmentation. Furthermore, a side by side comparison is conducted to verify that it is possible to render a virtual object in real-time that is indistinguishable from a real object. The results showed that the virtual object is perceived as real under the best conditions. The design of the pipeline is important, to be able to virtualise the objects and the lights correctly.

Supervisor:
Claus B. Madsen

Preface

This report is written as a result of a 10th semester Medialogy master thesis by group mta131035 at Aalborg University during spring 2013.

This group, mta131035, consist of two group members, where Martin Paprocki is on the "Computer Graphics" specialisation and Mathias Borg is on the "Medialogy (general)" specialisation. This combination of specialisations is reported and approved by the study board.

Reader's Guide

This document presents both a paper and a report. We encourage the reader to read the paper first, before proceeding to the longer description of the project in the report. Additionally, the appendix gives a detailed description of selected subjects.

Square brackets containing the surname of the author(s) and the publication year are used when referencing the sources in this report. All referenced sources will be placed in alphabetical order by the author's surname in the chapter "References".

CD

The appendix CD contains the following files:

- **Documentation:** The paper, report and video documentation.
- **Source:** The source code and files for the project.
- **Experiment Material:** Data from the experiments.

Acknowledgement

Thanks to Claus B. Madsen for supervision of the project. Also, thanks to all the people participating in the experiments.

Perceptual Evaluation of Photo-Realism in Real-Time 3D Augmented Reality

Mathias Borg*
School of Information and Communication Technology
Aalborg University, Denmark

Martin M. Paprocki†
School of Information and Communication Technology
Aalborg University, Denmark

Claus B. Madsen‡
Department of Architecture, Design and Media Technology
Aalborg University, Denmark

ABSTRACT

We present a pipeline for creating photo-realism of three-dimensional augmented objects, as well as a perceptual evaluating of the virtual scenes. A setup utilizing different lighting conditions is created and artefacts of the video-feed are simulated. Different parameters affecting the realism is evaluated. These are artefacts, shadows, lights, highlights and geometry. The results show that silhouettes of the shadows and the geometry and highlights on specular objects are important, as well as the simulation of noise, for creating a photo-realistic augmentation. Furthermore, a side by side comparison is conducted to verify that it is possible to render a virtual object in real-time, which is perceived as real under the best conditions.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; H.5.1 [Multimedia Information Systems]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 INTRODUCTION

Virtual realism or photo-realism has always been a goal within 3D computer graphics (CG), where still art and the film industry have already benefited from photo-realistic rendering to integrate virtual elements and effects with a high level of realism. Augmented reality (AR) which by definition is a mix of a video-feed and virtual elements would also benefit from having the virtual visualisations reaching this level of realism. Nevertheless, several challenges still exist in the way of reaching this goal, where realistic rendering of the 3D graphics in real-time is still a future goal.

The goal of this project is to investigate whether it is possible to obtain such realism in a static environment. The purpose of the experiments is for test subjects to assess the realism of an object. The test subjects will be shown a scene with either a virtual or a real object and have to assess whether or not he or she believes it is real (see example in Figure 1).

Even today the development of photo-realism within AR could help some industries. Some examples could be the medical [3], architectural and entertainment industry, where precise replication of the real world is important and/or where aesthetic factors play a role.

It is well recognized in computer graphics that parameters such as high model complexity, accurate highlights and both low frequency shadows (soft shadows) and high frequency shadows (hard shadows) are important for realistic synthesis [9, 14]. Elhelw et al. [9] mentions the importance of context in a scene, as well as the complexity of the human visual system and how to assess what is perceived by the user. Verbal answers combined with Likert scales are often too biased, therefore, Elhelw et al. perform a gaze study using eye-tracking. The results showed that highlight and silhouettes are important. However, since the context in which the 3D object is observed, can change the perception of the parameters and their importance, it is also interesting to see which quality the objects need to be in to be considered real. Moreover, as the rendering in AR occurs in real-time the minimization of computation usage is a requirement, hence a guidance for which quality of the different parameters to use would be beneficial.

An overview of the framework will be described in the next section. Afterwards, the experiment setup and procedure is described in section 3 and 4, while the results are presented in section 5, followed by the discussion and conclusion.

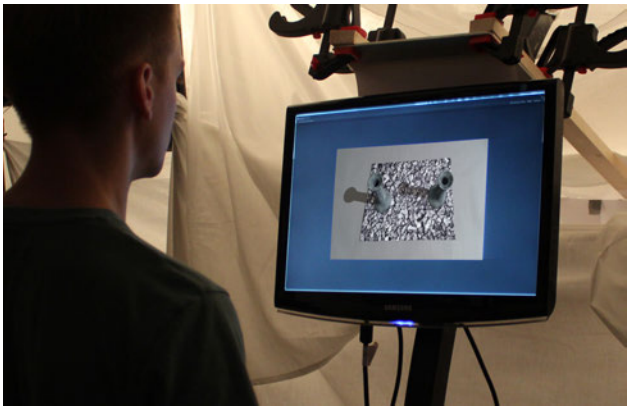


Figure 1: In the experiment test subjects assess virtual objects and compare them with real objects in an AR context. The scenes are rendered in real-time and artefacts of the camera are considered to integrate an object in the video-feed such that it is indistinguishable from a real object.

2 FRAMEWORK

In order to investigate whether it is possible to obtain realism in AR, a controlled setup is needed to be able to isolate the parameters for the experiment. This setup should utilize the ability of watching the scene from different perspectives. To obtain a correct perspective, from which the virtual objects are rendered, marker based tracking is used. Additionally, test objects are needed, both in a physical and a virtual form. These should have different shapes and materials, to be able to evaluate the geometry and the shading. Two objects are chosen for the experiments, which can be seen in Figure 2. One way to transfer the physical objects into virtual objects is to reconstruct the objects from multiple images or by scanning. This way, a mesh and a texture can easily be generated. To illuminate the objects lights are needed. A common way to achieve realistic lighting given a static environment is to use an environmental map [7, 1]. Lastly,

*e-mail: mborg08@student.aau.dk
†e-mail: mpapro09@student.aau.dk
‡e-mail: cbm@create.aau.dk

Figure 2: A photograph of the diffuse candleholder and the specular toy elephant chosen for the experiments.

it is important to address artefacts in relation to the rendering and the web-camera to integrate a virtual object into a video-feed [11]. Therefore, some of the most important artefacts will be addressed.

3 Experiment Setup

Five lights with diffusers (three 65 $\times$ 65 cm and two 53 $\times$ 53 cm) are set up in a circle with a radius of 1.5 meters and with a distance to each other of 65 degrees (see Figure 3). In the centre is a table on which the marker to be tracked is placed. The five lights are located one meter higher than the table and points upwards with a 45 degree angle to reflect the light in the white ceiling. A spot light is located higher than the ambient lights to minimize the length of the high frequency shadows from the objects, such that they are visible in the field of view of the camera. The whole setup is covered by white sheets to enhance the ambient illumination of the scene and to visually shield off the test scene from the test subjects.

To prevent the real object and the virtual object from occluding each other the angle of the positions from which the web-camera is capturing the scene must be restricted to 90 degrees. Additionally, the camera must be directed at the centre of the scene at all time. To ensure this, a metal arm is installed into the celling above the table which is able to rotate 90 degrees. However, this restrict the users' freedom of movement, since only one rotational axis is used and only one distance from the web-camera to the object on the table is available. The web-camera is positioned closer to the centre of the scene to be able to see the details in the objects (see Figure 3).

To ensure real-time rendering, the screen space effects and tracking is performed on a desktop PC. The following specifications are given for the hardware used in the setup:

1. A Logitech C920 Pro web-camera, which features HD video recording in 1080p.
2. A 22" Samsung SyncMaster 226BW monitor, which has a resolution of 1680 by 1050 pixels and a contrast of 700:1.
3. A PC with an Intel i5 CPU, an AMD 6970 HD 512MB RAM graphics card and 6 GB RAM.

For the execution of the 3D rendering and the marker based tracking Unity is used in combination with Qualcomm's AR solution Vuforia. A 540p resolution is used for the tracking, as well as for displaying the scenes, because of the limitations of the camera in relation to real-time execution.

3.1 Test Objects

Instead of modelling the objects virtual replicas were generated using Autodesk's 123D Catch, as it otherwise requires much manual

Figure 3: Top: An image of the setup. The test subjects is only able watch the monitor and not the scene. Furthermore, they are able to rotate the metal arm on which the monitor is attached. Bottom: An image of the scene that the web-camera captures. The web-camera is positioned closer to the centre to be able to track the marker and to be able to see details in the objects.

work. Furthermore, an investigation of an automated process could show to be advantageous. The replicas were generated through a capture of around 20–40 images per object taken from 360 degrees. Thereafter, the program reconstructs a 3D object from the images and outputs the mesh and the corresponding UV texture. Overall, the process is difficult because contract features have to be added to the toy elephant and much manual refinement is required. However, the quality of the 3D object is acceptable, especially given the cost of such a scanning of objects into a virtual space.

Two low-poly versions were created using the scanned objects as references, where contour modelling could be performed on top of the scanned reference object. This was achieved in Autodesk 3ds Max. The final result of the toy elephant can be seen in Figure 4. A reflective object was initially included, but preliminary tests showed that the quality of it was too poor to be included in the experiments.

3.2 Light Generation

In order to acquire an environment map, from which the lights will be generated, the setup needs to be captured. Five photographs are taken with a fish-eye lens covering 180 degrees of view. The camera is placed at the position where the objects are presumed to be placed on the marker, such that the surfaces of the virtual objects receive the correct light given in the environment. Moreover, the photographs are taken with nine different exposures ranging from 1/2000 of a second to 30 seconds, all with a aperture-stop of 8.

Figure 4: From left to right: The specular toy elephant in two low complexity versions, as well as the scanned original. Bottom row shows the same objects in wire frame.

Also, the process is repeated for both light conditions; ambient lighting only and ambient lighting with the spot light turned on. For the ambient setup only 7 exposures were needed (1/125 to 30 seconds).

The raw image files are imported into the program Panoweaver 8 Professional Edition and then stitched into a latitude and longitude environment map for each exposure. Thereafter, the environment maps are merged into a high dynamic range (HDR) image using all the exposure levels. This was achieved with Adobe Photoshop CS5.

To acquire the lights with the correct intensity, colour-temperature and distribution the HDR environment map is imported into HDR Shop [15]. Here a plug-in using the median cut algorithm is used [8]. Median cut generates the lights in accordance with the intensity distribution in the environment map, and exports them to a text file. For use in Unity a custom script is written to read the exported text file correctly.

Median cut divides the energy because the image is interpreted as areas of light. This approach is good for ambient light scenes, since each light radiation from a given area of a surface is correctly represented. In contrast, if a spot light is presented by a low number of lights, the little area of pixels representing the spot light will be split up into several light with less intensity. Moreover, they will be positioned apart from each other (see example in Figure 5). This result in a displacement from the actual spot position, which might influence the shading of the virtual objects. Instead, the spotlight was masked out from the environment map and lights were generated from this modified map. The spot light was then manually added to the scene and the intensity was matched with the physical spot light.

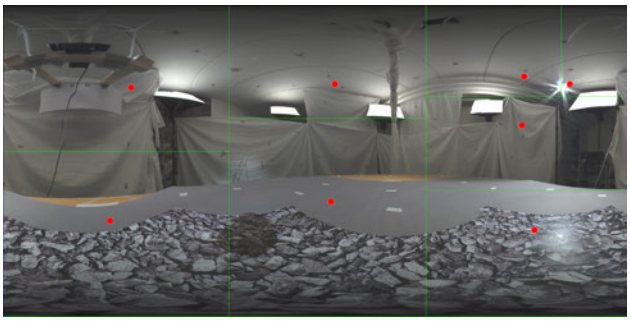


Figure 5: Example of how the spot light will be split up into two lights using the median cut algorithm. The red dots represents the generated lights.

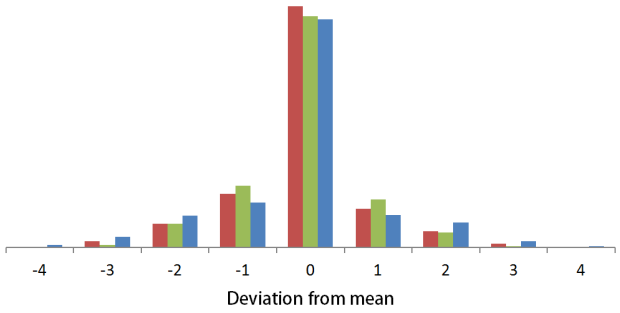


Figure 6: The pixel deviation from a mean calculated per pixel, where the column height indicate number or occurrences. The red, green and blue channel are shown correspondingly. The graph is based on data from 50 images with resolutions of 960 $\times$ 540 pixels, all capturing a gray paper.

3.3 Integration of the Virtual Objects

In order to integrate the augmented object as if it was a part of the video-feed, some artefacts have to be replicated and applied to the virtual object [11, 12, 10]. One of those is noise, which can be interpreted as a deviation from a "true" pixel value. Therefore noise is measured by capturing a sequence of images [6]. The mean between the individual pixels in these images is assumed to be the "true" pixel value. From this mean the deviation is considered to be the noise. The deviation sampled from 50 images can be seen in Figure 6.

The noise samples do not account for the correlation between the RGB channels, therefore a covariance matrix is calculated, which addresses the noise variance and covariance in relation to the channels. With a Cholesky decomposing of the matrix, the random samples from the three channels can be transformed into correlated samples [2, 16]. The correlated samples are randomly sampled for each pixel and saved in a texture, which is used by a screen-space shader that only adds the noise texture to the virtual object in the scene. The noise texture is repeated, and offset randomly for each frame in x- and y-directions, so the noise is not static. Moreover, anti-aliasing (AA) is used on the entire screen space. Because AA is applied on screen space it will create a bit of blur and smooth out the silhouettes of the 3D objects.

As the colours on the virtual object are noticeable different from the colours of the real objects in the video-feed a colour correction is needed. To balance the texture colour of the 3D objects to the colour of the real objects seen through the web-camera, an implementation of colour matching was implemented. The implementation uses histogram matching and requires a region of interest (ROI) of the source image and a target texture. A summed histogram is created for the source ROI and the target texture. For a given pixel value in the target texture the number of occurrences is found in the histogram. For the given number of occurrences a pixel value is found in the histogram of the source ROI. Now the pixel value of the target texture can be mapped to the pixel value of the source ROI. The RGB channels were converted to HSV and each channel was histogram matched as this resulted in the most satisfying colour correction.

Internal test showed that the quality of the method was not acceptable and it was realised that further corrections were needed to match the colours more exactly. The main problem is that if the texture was matched to a region of an image of the real object, the texture of the virtual object would gain double light — both from the lights implied in the image of the real object used for colour matching and from the shading of the lights in the virtual scene. Instead, the colour and intensity of the texture was matched manu-

ally by perceptually modifying the ambient colour of the materials. This way, a plausible simulation of the real surface was created, yet not in a correct way.

4 Experiment Design

In the original experiment design the users were able to move the web-camera as wanted to see the scene from different perspectives. However, a preliminary test showed that the tracking of the marker was not stable enough resulting in noticeable jittering. As this would compromise the purpose of the experiment, the scene was displayed from three pre-determined positions with approximately 20 degrees of disparity. By locking the position of the web-camera (and the virtual camera) to pre-defined positions, no jittering could be observed. However, it removed the element of a changing perspective. The test subject had to keep a distance of 60 cm from the screen to keep the basis consistent between the trials. The scene was visible for 4 seconds at each position, before the test subject had to assess. This procedure was repeated for each trial.

Before proceeding to the actual trials some mental calibration scenes were shown to the test subjects. These scenes contained all the information about the lighting, the environment, the objects and the quality of the video-feed. This ensured that the test subjects knew what to expect in the scene and how the experiment would be conducted.

The first experiment intended to identify the thresholds or necessity of certain parameters. At first, the effect of artefacts (noise and anti-aliasing) was evaluated, as a lack of it might make it possible to identify the virtual objects. Given both a spot and an ambient light setup, both low and high frequency shadows could be present in the scene. The high frequency shadow was evaluated both as rendered in real-time and as pre-rendered (baked into a semi-transparent ground plane, so the underlay was visible). The low frequency shadows were always baked, since they were a product of global illumination given 1024 lights, approximating the real light distribution in the scene. The number of lights needed to shade the objects (2 to 16 lights) was also evaluated, which was always performed in real-time. The number of lights is suppose to determine how accurate the light distribution should be to in order to have a realistic shading on the objects. This is important since it is difficult to generate lights from the surrounding environment and the minimization of the light calculations could be beneficial. Moreover, the lack of highlight wws also evaluated to assess the importance in AR solutions. Lastly, in order to confirm that model complexity is important and to see how important smooth silhouettes are in an AR context, the two low polygon models and the scanned model were evaluated.

Only the specular object was used for evaluating the artefacts, the shadows and shading, the highlights and the geometry. Each test subject would see all of the scenes once (5 for the first test, 18 for the second, 3 for the third and 8 for the last), which resulted in 34 assessments in all. The real object was used as control to verify the realism of the representation of the physical scene.

The second experiment aimed at revealing the possibility of making a virtual object that could be perceived as real under the most difficult condition, that is when the virtual object is compared side by side to the real object and the test subjects are allowed to watch the scene for as long time and as many positions as wanted. The side by side comparison scenes were illuminated by the spot light. In this experiment the test subjects watched a scene with the diffuse objects side by side and a scene with the specular objects side by side.

The sequence of the scenes was randomised in both experiments, as well as the position of the objects was switched randomly in the side by side comparison.

5 Results

The virtual objects are considered indistinguishable from the real objects if the ratio of answers approaches random chance, that is when test subjects are just guessing. The probability for random chance is 50 % given at least 100 observations [13]. However, for smaller sample sizes it might not even be possible to get a significant result with the best possible data [4]. Therefore, another probability of 19 % is suggested to compensate for smaller samples sizes [5] and relates to the commonly used threshold of people guessing incorrectly at least 25 % of the times.

The critical number i_c of answers to significantly reject the null hypothesis that people can determine whether the object is virtual or real can be calculated by the probability mass function for binomial distributions:

$$i_c(n,p) = \min\left\{ i \;\middle|\; \sum_{j=i}^{n} \frac{n!}{j!\,(n-j)!}\, p^j\,(1-p)^{(n-j)} < 0.05 \right\}$$

where $p = 0.19$ and n is the sample size. The number of assessments that a virtual object is real have to exceed the critical value i_c for an object to be perceived as real in a statistically significant manner.

5.1 Evaluation of Parameters

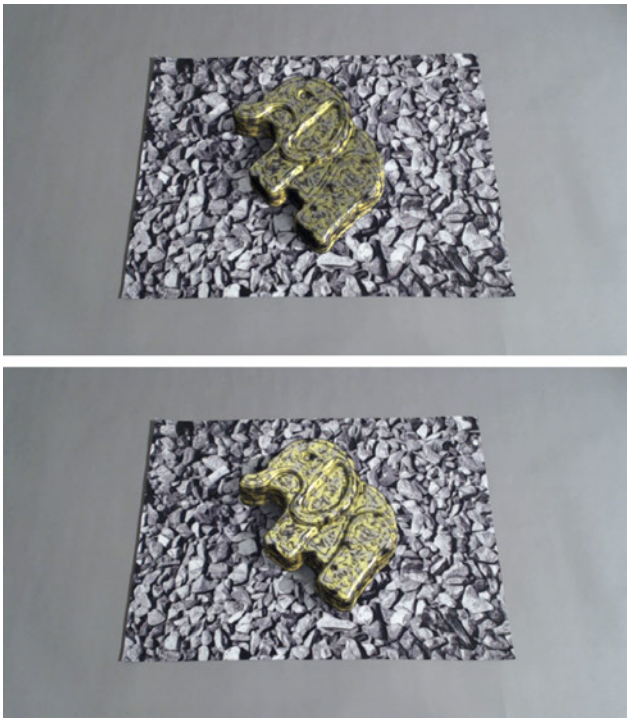


Figure 7: Top: Image from scene with specular elephant shaded by two lights and with pre-rendered shadows. Bottom: Image from scene with specular elephant shaded by 16 lights and with pre-rendered shadows.

The experiment was conducted with 16 test subjects in the age of 21 to 30 years — one woman and 15 men. All had normal or corrected-to-normal vision and most subjects were experienced with 3D computer graphics and augmented reality. The critical value for an object to be perceived as real given 16 test subjects is 7.

The simulated artefacts of the camera and the rendering is evaluated to verify its importance in augmented scenes. However, none

of the scenes was perceived as real, as can be seen in Table 1. Only the real object was assessed as real.

The number of lights needed to create a perceptually realistic shading is evaluated in combination with different methods for creating shadows (see example in Figure 7). For the spot light environment two different methods for creating high frequency shadows is used; one pre-rendered and one in real-time. Both of these include pre-rendered low frequency shadows. For the ambient lighting environment pre-rendered low frequency shadows are evaluated, as well as a lower limit without any shadows. The results for these combination can be seen in Table 2. All of the scenes for the spot light environment are perceived as real. This means that just 2 lights can be used to shade the object and both the pre-rendered and real-time high frequency shadow can be used when wanting to create photo-realistic objects. On the other hand, for the ambient lighting environment only the object with pre-rendered low frequency shadows shaded with 4 lights and the object without shadows shaded with 2 lights are significantly perceived as real.

The necessity of highlights on specular objects is evaluated and the results can be seen in Table 3. The results show that only the object with specular highlights was perceived as real.

Lastly, the quality of the geometry of the object is evaluated. The quality in this context relates to the number of polygons that the object consist of. The quality is evaluated in both lighting conditions to assess whether or not the light has an influence. The results can be seen in Table 4 where the high-polygon model is perceived as real in both lighting conditions. This does not apply for the object consisting of the medium amount of polygons as it is only perceived as real in the spot light environment. The low-polygon model is not perceived as real for any of **the** two lighting conditions.

Table 1: Number of answers out of 16 that a scene was real when evaluating camera and rendering artefacts. The scenes are shown in ambient lighting. Results in bold exceed the critical value of 7 that a scene is significantly perceived as real.

	Noise	No noise	Real
Anti-aliasing	5	4	**9**
No anti-aliasing	5	5	

Table 2: Number of answers out of 16 that a scene was real when evaluating the number of lights to create a perceptual correct shading and when evaluating different methods for creating shadows in a spot light and an ambient light setup. Results in bold exceed the critical value of 7 that a scene is significantly perceived as real.

SPOT LIGHT	Number of lights				
	2	4	8	16	Real
Baked high and low frequency shadows	**9**	**10**	**13**	**11**	**14**
Real-time high frequency shadows and baked low frequency shadows	**7**	**10**	**7**	**9**	
AMBIENT LIGHT	Number of lights				
	2	4	8	16	Real
Baked low frequency shadows	6	**8**	3	4	**9**
No shadows	**7**	3	4	4	

Table 3: Number of answers out of 16 that a scene was real when evaluating highlights on a specular object. Results in bold exceed the critical value of 7 that a scene is significantly perceived as real.

	Specular highlight	No specular highlight	Real
SPOT LIGHT	**9**	3	**9**

Table 4: Number of answers out of 16 that a scene was real when evaluating the model quality. Results in bold exceed the critical value of 7 that a scene is significantly perceived as real.

	461 polygons	1028 polygons	52387 polygons	Real
SPOT LIGHT	5	**8**	**11**	**15**
AMBIENT LIGHT	2	2	**7**	**9**

5.2 Side by Side Comparison

This experiment was conducted with 15 test subjects between the age of 21 and 27, where one woman and 14 men participated. All had normal or corrected-to-normal vision and most test subjects were familiar with 3D computer graphics and augmented reality. The critical value for an object to be significantly perceived as real given 15 test subjects is 6.

Example of the scenes that test subjects are watching can be seen in Figure 8. The results of the side by side comparison can be seen in Table 5, where it can be noted that only the diffuse virtual object is significantly perceived as real when compared directly to a real object.

Table 5: Number of incorrect answers out of 15 for the side by side comparison of the two objects. Results in bold exceed the critical value of 6 that a scene is significantly perceived as real.

	Diffuse	Specular
SPOT LIGHT	**6**	0

6 Discussion

Even though the evaluation of artefacts did not prove any results, the generation of artefacts is still considered important to integrate a virtual object into a scene. We believe that virtual objects would otherwise look uncanny. This is supported by preliminary tests which showed that test subjects was able to pinpoint virtual objects solely based on the missing noise. However, the lack of noise was first noticed after a while, when the test subjects had gotten familiar with the scene. This indicates that noise is a subtle effect which must be evaluated over several trials.

When evaluating the shading and the shadows all of the objects in the spot light were perceived as real. This means that it is possible to use only two lights for creating ambient lighting is necessary when a strong spot light is present in the scene. Additionally, there is no need of pre-rendering high frequency shadows as they can be rendered in real-time, as long as they are of sufficient quality. Especially the edges must be of good quality as test subjects in particular were looking at the silhouettes of the shadows to determine

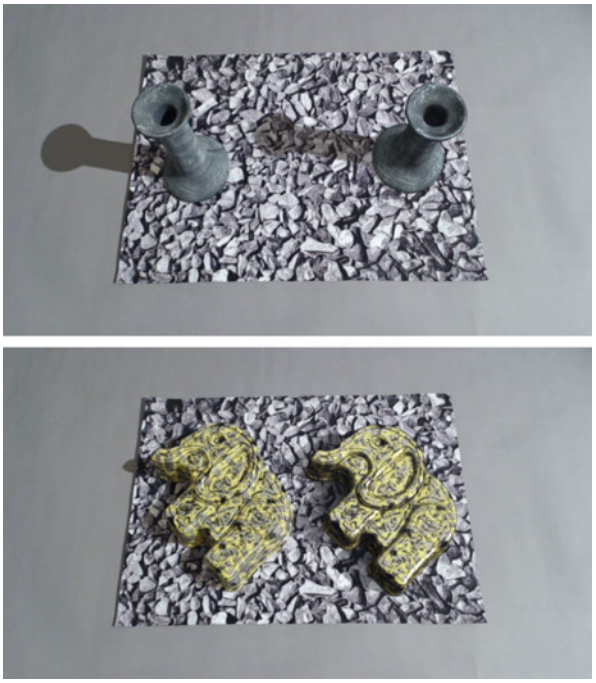


Figure 8: Image of the side by side comparison. Top: the real diffuse object to the left and the virtual diffuse object to the right. Bottom: the real specular object to the left and the virtual specular object to the right.

the realism. For the scenes with only ambient lighting just two were perceived as real, two with few lights for shading the object. One reason might be that people are not use to watching scenes without any noticeable shadows, and therefore assess them as virtual, even though they are actually real. Otherwise, the ambient scenes might not have been set up and adjusted appropriately to be able to match the real one.

Also highlights are important on specular objects, as they would otherwise look uncanny. This of course depends on the fact that the test subject knows about the material of the object, as it might have given another result if the object was believed to be completely diffuse.

The knowledge of the context is very important when conducting a perceptual experiment evaluating photo-realism. Therefore, mental calibration is suggested to compensate for bias in relation to the environment, the lighting, the objects and the quality of the video-feed. Without this knowledge the test subjects will have no basis for evaluating the objects displayed on the screen. Preliminary tests showed that if no basis is available test subjects will assess more objects as being virtual, even though they might be real, because the test subjects might not be familiar with the object used or the artefacts of the camera makes the scene look unnatural.

One way to avoid double lighting on texture of the virtual objects would be to calculate the intensity and colour of the virtual lights hitting each point on the mesh, assumed that the intensities and colours of the virtual lights are adjusted to the corresponding physical light. Then this UV map with baked lights could be subtracted from the UV texture of the object. This would remove the double lighting and only leave the albedo. However, as a perfect match of intensity and colour between the physical and virtual light is difficult to obtain this option was skipped due to time and resource limitations.

When creating such an experiment setup the most difficult task is to capture the physical setup and convert it to a virtual — and maintaining the right units throughout the pipeline. In most cases, the majority of the pipeline has to be redone if a step fails. Therefore, it is crucial to have a clearly defined setup and approach of how to capture it. In the best case, no changes are applied to the setup and hardware when capturing the environment and the objects.

As long as marker based tracking is not stable enough to be unnoticed the freedom of movement has to be restricted. Optionally, the tracking might be more stable on the expense of the frame-rate. Otherwise, another tracking method can be used.

7 Conclusion

A setup has been created to evaluate the visual realism of augmented objects, which takes into consideration the environment and the artefacts of the video-feed. Results show that highlights are important for the perception of realism, as well as silhouettes of objects and shadows. Furthermore, it is shown that real-time shadows can be of sufficient quality to enhance the perception of reality. Additionally, preliminary tests show that noise is an important factor to integrate a virtual object. Lastly, it is proven that it is possible to render an augmented object in real-time (besides pre-rendered ambient shadows) which cannot be distinguished from a real object, even when compared side by side.

8 Future Work

It would be of great interest to create a common way to capture the environment and maintain it throughout the pipeline. With such guidelines it would be easier to quickly set up a photo-realistic scene, which can be used in an application.

More research is suggested evaluating other parameters, for instance colour bleeding and a larger variety of materials and shapes. Movement and animation, as well as context, could also be interesting. With moving objects the influence of motion blur could be evaluated. Also, the attention to an object would presumably be different. When evaluating context different sceneries and their influence on the objects could be evaluated.

References

[1] K. Agusanto, L. Li, Z. Chuangui, and N. W. Sing. Photorealistic rendering for augmented reality using environment illumination. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 208 – 216. IEEE Computer Society, 2003.

[2] L. A. Apiolaza. Simulating data following a given covariance structure. `http://www.quantumforest.com/2011/10/simulating-data-following-a-given-covariance-structure/`, 2011. Last seen: 13-03-2013.

[3] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments 6*, 4:355–385, 1997.

[4] R. H. Bojesen. Statistical methodology for sensory discrimination tests and its implementation in sensR, 2013.

[5] M. Borg, S. Johansen, D. Thomsen, and M. Kraus. Practical implementation of a graphics turing test. In *Advances in Visual Computing*, volume 7432 of *Lecture Notes in Computer Science*, pages 305–313. Springer Berlin Heidelberg, 2012.

[6] A. C. Bovik. *Handbook of Image and Video Processing*. Elsevier, 2nd edition, 2005.

[7] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 189 – 198. ACM, 1998.

[8] P. Debevec. A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2005 Posters*, SIGGRAPH '05. ACM, 2005.

[9] M. Elhelw, M. Nicholaou, A. Chung, G. Yang, and M. S. Atkins. A gaze-based study for investigating the perception of visual realism in simulated scenes. *ACM Transactions on Applied Perception*, 5(1):3:1 – 3:20, 2008.

[10] J. Fischer, D. Bartz, and W. Straßer. Enhanced visual realism by incorporating camera image effects. In *ISMAR '06*, Proceedings of the 5th IEEE/ACM International Symposium on Mixed and Augmented Reality, pages 205 – 208, 2006.

[11] G. Klein and D. W. Murray. Compositing for small cameras. In *ISMAR '08*, Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, pages 57 – 60, 2008.

[12] G. Klein and D. W. Murray. Simulating low-cost cameras for augmented reality compositing. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):369 – 380, 2010.

[13] S. P. McKee, S. A. Klein, and D. Y. Teller. Statistical properties of forced-choice psychometric functions: Implications of probit analysis. *Perception & Psychophysics*, 37(4):786–298, 1985.

[14] P. Rademacher, J. Lengyel, E. Cutrell, and T. Whitted. Measuring the perception of visual realism in images. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 235–248. Springer-Verlag, 2001.

[15] USC Institute for Creative Technologies. Hdr shop. `http://www.hdrshop.com/`, 2013. Last seen: 28-05-2013.

[16] T. van den Berg. Generating correlated random numbers. `http://www.sitmo.com/article/generating-correlated-random-numbers/`, 2012. Last seen: 13-03-2013.

Contents

Danish Summary — Dansk resumé

Projektet "*Perceptual Evaluation of Photo-Realism in Real-Time 3D Augmented Reality*" beskriver en opsætning hvordan perceptionen af 3D augmenterede objekter er evalueret. Målet er at opnå fotorealisme således at der ikke kan ses forskel mellem et virtuelt augmenteret objekt og det virkeligt objekt. Augmenteret realitet (AR) er grafik projekteret på en skræm, hvor et 3D objektet er vist oven på den videosekvens som kameraet filmer. For at et program kan placerer 3D objektet korrekt i scenen skal en markør være til stede. På den måde vil programmet finde ud af hvordan markøren er vist i den filmede videosekvens og derudfra kunne udregne hvordan 3D objektet skal placeres (inklusiv rotationen, skaleringen og projektionen af den).

For at opnå fotorealisme er der en rækker aspekter der skal tilses. Først og fremmeste skal de virkelige parametre konverteres til en virtuel repræsentation. Her bliver objekter skannet ind ved hjælp af programmet 123D Catch fra Autodesk. Dette bliver gjort ved at fotografere et givet objekt fra adskillige vinkler, hvorefter programmet rekonstruerer et 3D objekt ud fra billederne.

Lyset skal også reproduceres i det virtuelle rum. Derfor bliver testopsætningen også fotograferet og et panorama-billede kan bruges til at finde frem til lys-fordelingen, lys-temperaturen, samt lys-intensiteten.

Kun givet lys og objekter er det stadig ikke muligt at opnå fotorealistiske objekter da kameraet der filmer scenen også skaber artefakter. Her er der især lagt meget vægt på at genskabe støjen korrekt. Støjen bliver først indhentet og isoleret fra en sekvens af billeder, for så at blive genskabt henover det virtuelle objekt i scenen. Ydermere er der tilføjet anti-aliseringen for at integrere grafikker bedre sammen med video sekvensen afspillet i baggrunden.

Efter objekter, lys og kamera-artefakter er blevet genskabt i det virtuelle rum kan eksperimenterne udføres. Ved det første eksperiment blev flere parametre evalueret. Disse parameter var:

- Med og uden støj og anti-aliseringen
- Bløde og hårde skygger, samt skyggelægning
- Forskellige kvaliteter af 3D objektet
- Med og uden lys-højdepunkter

I dette eksperiment blev objekterne vist en ad gangen, hvor der så blev skiftet mellem virtuelle og virkelige objekter. Testpersonerne skulle så vurdere hvorvidt objektet var rigtigt eller virtuelt.

Resultatet på eksperimentet var at testpersonerne sÃ¥ en del af de virtuelle objekter som virkelige, men at silhuetterne på objeketet og skyggerne skulle være af høj kvalitet, samt lys-højdepunkterne skulle være til stedet, for at de virtuelle objekter blev vurdet som virkelige. Ydermere viste pilottests at støj også er meget vigtigt at inkludere.

Det andet eksperiment præsenterede to objektet for testpersonen, side om side. Af de to objekter var det ene virtuelt. Her skulle testpersonen så vurdere hvilket af de to objekter der var det virkelige. Resultaterne viste at et objekt godt kan bestå en test side om side.

Chapter 1

Introduction

Augmented reality (AR) is a display of mixed reality [Milgram et al., 1995], where an augmented virtual element is added to the displayed reality, essentially adding information to the view of the displayed image that normally would not exist.

1.1 Application Areas

Adding more information to a view can assist the user in various ways. Azuma and colleagues [Azuma, 1997; Azuma et al., 2001] mention the following main application areas for AR: medical visualisation, maintenance and repair, annotation, robot path planning, entertainment, and military aircraft navigation and targeting. Regarding the medical visualisation the augmented 3D graphics can be used in relation to surgery training as well as aiding the surgeons performing non-invasive operations [Liu et al., 2003]. Given some 3D-datasets the augmented graphics would give an "x-ray" vision into the body of the patient, which would be performed in real-time. Azuma et al. also gives examples of how architects could use AR to get a sense of how a building would look like in a given environment. Overlaid 3D graphics can also assist in visualising structures in difficult environments where visual cues are limited, such as it is on space-stations, under water and when fog is present. Elhelw et al. [2008] notes that photo-realism can be essential when wanting the perceiver to achieve high visual reasoning, which for example can be essential when performing a surgery. Photo-realism can also be important in relation to aesthetic aspects in for instance 3D AR games.

1.2 Games and Interactive Systems

Augmented reality has been defined by Leino and colleagues as *"real-time interaction and 3D registration between the real and virtual world"* [Leino et al., 2008] which often are elements in games and other interactive entertainment systems. Some of these systems have been found to strive for realistic representations of graphics, such as the SONY's Head-Mounted Display (HMD) show case [Nakamura, 2012] that augments video sequences in stereo graphics, where the aim is higher immersive systems. The problem with such a system is that a pre-shot video sequence is not suited for real-time interaction, especially because the environment and position of the viewers head has to be predefined. On the other hand it reaches a very high quality of the content. For 3D real-time interaction a 3D game engine could be considered, even though it

might not be able to reach as high a realism. Some of the most commonly used AR interactive 3D solutions are:

- ARToolkit; is a relative easy C-library to use for creating AR on mobile devices.
- DroidAR; is a framework for Android which features location and marker based tracking.
- metaio; is a SDK for 3D tracking.
- Vuforia; is a SDK for AR to mobile devices. It has the possibility to be integrated with Unity.

All of these solutions have the option of exporting to mobile devices, which makes them attractive since they can reach a wider range of people. Capability of real-time rendering has advanced a lot, but it is still not explored if it can represent photo-realism. Hardware limitations does not permit — not in real-time at least — global illumination execution, but some approximations are given. For example in Unity the Beast system is available which can bake the global illumination into light map textures. Additionally, real-time techniques are already given in Unity such as motion blur, blur, anti-aliasing (AA), shadow projection (only for one directional light) and more.

1.3 Synthesis of Photorealism

One of the main aspects to consider when synthesising a realistic scene is to look at how the light and illumination is represented. Commonly used models and techniques often consider a surface point orientation (the normal) relative to a given light, another point on the same surface or another point on a different surface, which gives the possibility to imitate reflections, amount of illumination, self shadowing and more, on a material. These are models and techniques such as Phong, Blinn-Phong, Lambert and object/surface based ambient occlusion, etcetera. These are executable in real-time since they include relative few variables and consider lights to behave in a very simplistic manner. In order to work with a more varied representation of the light an umbrella term that includes various behaviours of light is to be considered, namely global illumination. Global illumination represent both direct and in-direct illumination, meaning it considers light coming directly from the light source but also the ray bouncing from surface to surface. This requires a lookup of the illumination state of the scene multiple times [Watt and Watt, 1992], which prevents is from being rendered in real-time.

While considering synthesizing a photo-realistic scene a trade-off have to be assessed between a simplistic interpretation of light (which can be run in real-time) and a more correct interpretation (which cannot be run in real-time) [van Dam, 2009]. The ideal situation would be to deliver scene that it so complex that it reaches a level which is perceived in the real world [Chiu and Shirley, 1994]. However, today's hardware limits execution of such complexity, often rough simulations have to suffice. To minimize the complexity of a 3D object an imitation of the complexity can

be achieved by applying a detailed texture to it [Catmull, 1974], combining it with a bump map [Blinn, 1978] and maybe even perform tessellation at runtime [Boubekeur and Alexa, 2008]. Moreover, much focus have also been directed at how to interpret 3D frustums in 2D projections and preform effects such as blur, motion blur, screen space ambient occlusion, depth of field and more. Such approaches often needs information stored about normals and depth positions, sometimes using multiple buffered frames in high screen resolution which often requires a lot of storage capability.

1.4 Previous Work

The quality of various 3D aspects and their perception in relation to realism have been investigated before. Some factors related to realism were investigated by Elhelw et al. [2008], namely the light reflections (specular highlights), 3D surface details, depth visibility, 2D texture details and edges/silhouettes. With an eye-tracker it was shown that much attention was paid to specular highlights and object details on the edges/silhouettes, which was somewhat contradicting with what the test subjects though they were looking at.

Rademacher et al. [2001] have also investigated which factors affect the perception of realism. The evaluate shadow softness, surface smoothness, number of objects, number of light sources and object shapes. They show that only a few light sources is needed to create photo-realism, as well as detailed object surfaces. Furthermore, the objects need to be of sufficient quality.

Both studies compared computer generated imagery with photographs.

Klein and Murray [Klein and Murray, 2008, 2010] mentions the following technical challenges when creating an augmented reality scene with the goal of creating an illusion of a object being a part of the real environment.

1. Accurate tracking
2. Occlusions between real and virtual objects
3. Lighting and shadows should match the real world
4. Quality of the rendered image should match the video feed quality

Here it should be noted that Klein and Murray investigate item 4 and presents a concrete pipeline for how a given camera works, which should help in the task of creating a render that makes the objects blend into the video feed taking, into account its imperfections. The camera information flow is described as followed:

1. Lens effects - includes barrel distortion, image softness and vignetting
2. Bayer mask - includes colour crosstalk and anti-aliasing

3. Image sensor - includes motion blur and image warping given motion and noise
4. In-camera processing - can vary, but is usually sharpening, exposure, Bayer filter and colour-space conversion (see next item)
5. Colour-space conversion - for example conversion to YUV-411, where luminance information per pixel is given

Here Klein and Murray mentions the usage of OpenGL and GLSL moreover noting the importance of using pre-multiplied alpha for correct alpha blending and compositing. Their final processing and blended is done through eight steps, some of them being; radial distortion, subsampling and colour mixing (desaturation), Gaussian blur, motion blur and UV-scale. The paper did not evaluate the end result in relation to human perception.

Another study conducted by Fischer et al. [2006] is also addressing camera image effects. They particularly look into noise, anti-aliasing and motion blur.

1.4.1 Human Visual Perception and 3D Graphics

As summarized by Elhelw et al. [2008] the human visual system (HVS) is sensitive to many kind of variations and many models representing different features of the HVS have been investigated [Daly, 1993; Lubin, 1995]. But many models represent only a part of the HVS, therefore Elhelw and colleges mentions that in the end the graphics are made to be perceived by humans and that images quality and the model integrity should be assessed with human judgement. Yee et al. [2001] looked into the quality of global illumination and its computational performance in relation to human tolerance of errors. This also addresses the balance previous mentioned between hardware requirements and imitation of the real world behaviour of light and materials. Many psychophysical studies assess a very specific feature in the complex HVS. Take for example Biederman's *recognition-by-components* [Biederman, 1987, 2000], a theory which proposes how a 2D projection of the 3D environment we live in is perceived in relation to 3D object interpretation. Here Elhelw et al. [2008] notes that assessing the HVS in a more coherent manner is difficult, since there are too many dimension to account for. Moreover, questionnaire based evaluations are not enough when wanting to investigate specific rendering features and different understanding of visual realism. For instance due to subjective scale end-points and because recall of memory can result in less correct data.

1.5 Merging Reality and Virtuality

Presumably, none have jet explored the limits of realistic 3D elements and effects in an AR context, while the film and art industry have merged 3D into their scenes for years. There could be various reasons for the lack of interest of integrating the 3D content into a live video feed and trying to make it look real. First of all, both the AR technology and high-end hardware on

mobile devices are relative new. Moreover, there is a limited interest in AR from the consumers [Kilde]. The limited interest from the consumers could be a result of the required marker, the amount of space needed around the marker, as well as the required solid hardware specifications. Even though some of these problems are addressed by industry and academia, factors like a good clear video feed, stable tracking [Wagner et al., 2010] and object occlusion within the video feed [Shah et al., 2012] are difficult problems to overcome. There are still many obstacles to overcome before the integration of 3D into a real-time video feed will get close to the level seen in films and still art. The first step is to test whether or not now-a-days render techniques even have the capability to seamlessly integrate 3D in real-time with a background video feed.

1.6 Summary

AR is used across various industries and real looking objects could prove useful in relation of effective work and training, as well as pleasing aesthetic needs. There is a wide understanding that there are many perceptual dimensions and technical challenges to address and that it is difficult to investigate an augmented reality scene where all features are included. It is therefore easier to investigate one aspect related to perception and/or technical challenges. This have done, but little have been done to see if an augmented reality scene, seen as one coherent entity, would pass a test of realism and see weather the augmented 3D graphics are indistinguishable from the displayed real content.

1.7 Project Description

In this project an evaluation of 3D content in varying qualities will be performed in an real-time augmented reality setup, to see whether it is possible to create a representation of a real object in 3D that is perceptually indistinguishable. The goal is to assess the key aspects that are required to create a realistic looking scene, given a video feed from a web-camera displayed on a PC monitor, to create the necessary content and to utilize rendering and image-processing methods. Regarding the scope of the project the following should be realised as a minimum:

1. Create different AR scenes using a real-time framework commonly used
    - Creation of 3D objects
        - Create materials that match to the surfaces of the real objects
    - Usage of different lighting setups and techniques
        - High and low frequency shadows (hard and soft shadows)
        - Real-time lighting
        - Baked lighting

- Must run in real-time

2. Address essential camera artefact
3. Design and perform perceptual evaluation of realism across all parameters and some combinations thereof.

In this project "real-time" is defined as minimum around 30 frame per second and augmented reality is understood as a real-time rendering of 3D graphics augmented in a real-time video feed utilizing marker based tracking, where the position of the camera is undergoing a constant movement. Also it is assumed that the used standard rendering models given in the graphical framework are assessed in relation to the HVS and specific features thereof, such that they deliver an end result with a degree of perceptual correctness.

1.7.1 Problem Formulation

How can a 3D object be created and integrated in a real-time AR application such that it is indistinguishable from a real object in the video-feed content? Given a known static environment, which parameters contribute to the perceptual photo-realism of an augmented object?

1. Can high resemblance between the 3D content and the video-feed content be reached?
2. Which parameters contribute to photo-realism?
3. Are there any perceptual factors to be observed and considered?
4. Synthesise a pipeline that can be used to achieve photo-realistic rendering of 3D content in AR.

Chapter 2

Framework

To answer the questions of the project a framework is needed, which delimits the scope of the project. The choices behind the limitations of the scope is addressed and the main structure of the experimental setup is reviewed.

2.1 Scenes

When choosing the scene that has to be displayed for the evaluation some general aspects have to be considered, first of all the context, in which the objects have to be seen. Hasic et al. [2010] mentions that besides the consideration of the rendering quality the environment or background can also play an important role. A simple as possible background was chosen for the project so that no attention bias and context consideration would play a role. If a everyday scene including other object were to be created, a risk that a form of side by side assessment would arise during object evaluation, where the shading, shadows, colour temperature and other factors could be considered in detail by the human test subjects. In real life usage of an AR application the attention level between details to other objects and its comparison would properly not be as high as in a test situation where one is to assess whether the object is real or not. To be sure not to create a biased baseline, for when the objects are perceived as real, the only thing in the scene, besides the object itself, will be a part of a table and the marker on which the object will be placed on. This will limit the object and shadow occlusion aspect. Furthermore, no movement or animations will be displayed either. This simplified scene should also minimise the constrains given by a context, for which object can be used with the environment.

Regarding the light in the scenes two types of setups have been considered interesting. One that produces an ambient lighting (light coming from everywhere) and one the produces a hard shadow (light coming from one point or direction). The reason is that scenes normally consist of some ambient lighting and one or more hard shadows (or at least soft shadows). The intensity of the ambient lights should be such that a spot can be turned on without making the scene too illuminated.

2.2 Game Engine

The chosen game engine for the project is Unity. Unity provides easy building of scenes through the UI and gives the possibility of high-level scripting as well as some low-level programming options. Moreover Unity represents what the present day's game engines provide of rendering features and effects.

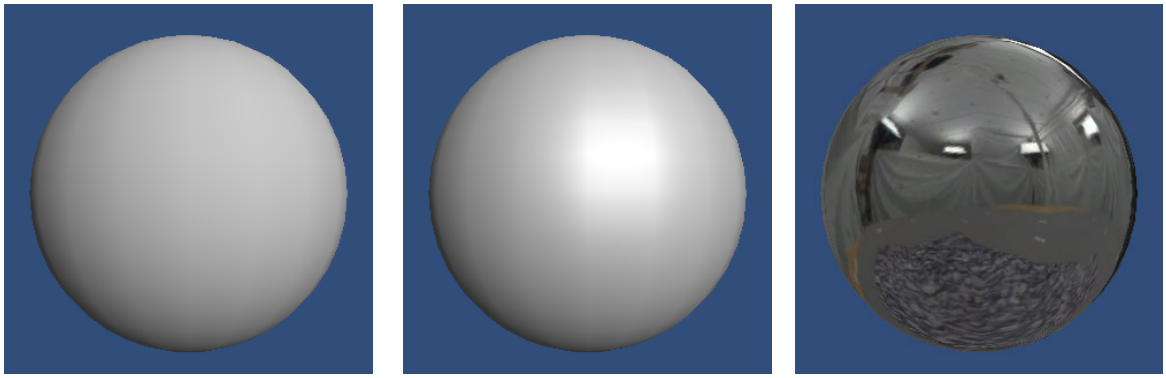
Unity provides a relative easy accessibility to setup a scene, including object placement, material assignment, lighting setup and easy light baking options with the Beast system [Autodesk, 2013]. Qualcomm's AR SDK solution, called Vuforia, can be used with Unity. Vuforia's AR functions in Unity can be executed and debugged in the editor without the need of creating a build. All these aspects of Unity gives an easy development and testing process and saves time and resources, where the ARToolKit and the Java Vuforia library does not provide such immediate access to project assets and functions. On the other hand, Unity with the Vuforia package restricts one to a given framework to work within, for instance it has some limitations such as light calculations and number of real-time shadows.

2.3 Parameters

Some limitations were performed during initial decisions in relation to the scene, context and game engine mentioned in the previous section, but a further delimitation is needed. A list of the considered parameters is given in Appendix A on page 87. Considering the physical setup the following parameters were chosen:

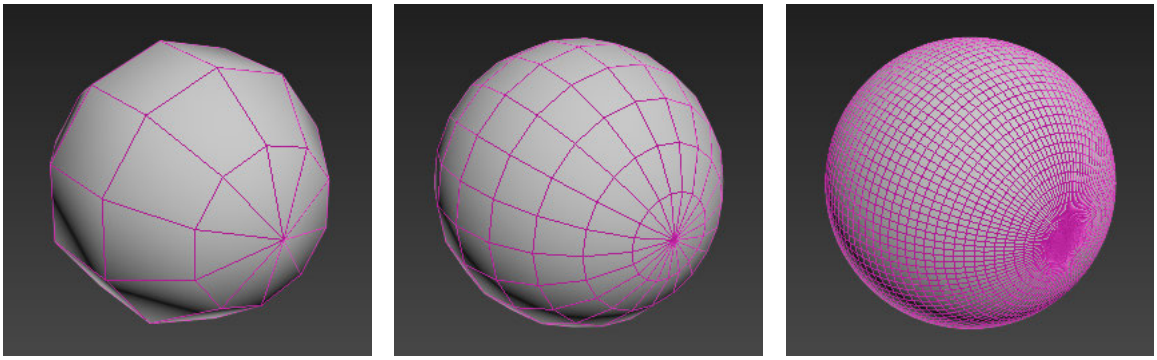
- Physical scene:
  - Objects
    - Material — Diffuse, specular and reflective
    - Complexity — Three levels in relation to the polygon count
  - Lighting (static)
    - Light dispersion — Ambient and spot

The material desired to be evaluated are diffuse, specular and full reflective (see example in Figure 2.1). This way a range of shadings will be evaluated in relation to human perception and the importance of highlights. Based on the work of Elhelw et al. [2008] the aspect of silhouette smoothness is considered interesting. Therefore the three objects should be created with a very low polygon count, a low polygon count and a very high polygon count, essentially representing three levels of geometry complexity for each object (see example in Figure 2.2). Additionally, two light setups are desired to be able to cast low and high frequency shadows (see example in Figure 2.3). These represent the extremes of shadow edge sharpness that can be achieved.



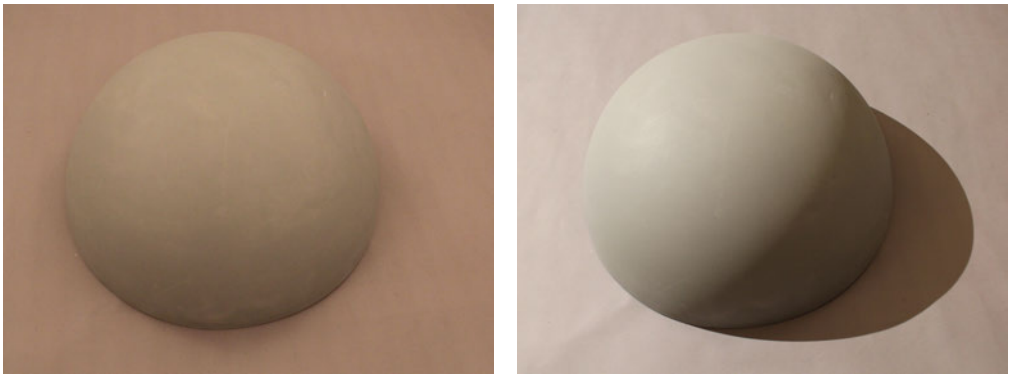
(a) Diffuse object (b) Specular object (c) Reflective object

Figure 2.1: Virtual examples of the three materials.



(a) Very low polygon object (b) Low polygon object (c) High polygon object

Figure 2.2: Examples of the quality of the mesh.



(a) Ambient light (b) Spot light

Figure 2.3: Examples of the lighting.

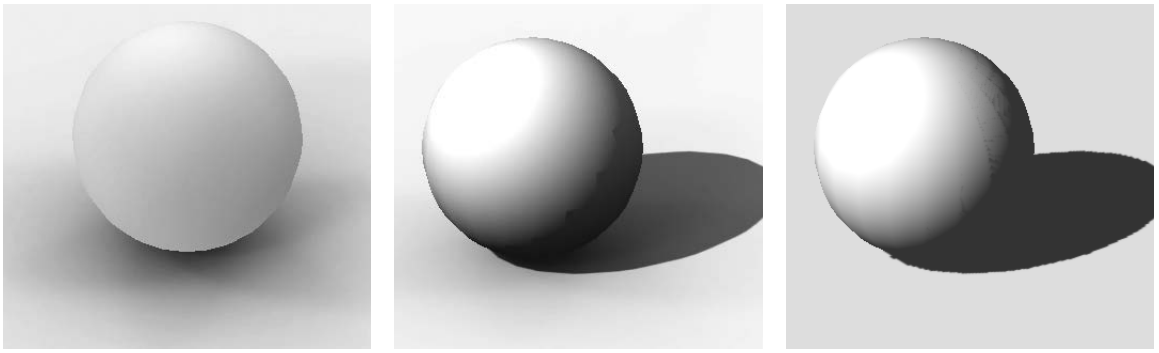
The occlusion of objects and shadows is not evaluated in this project as this is a problem of its own, confer the technical challenges mentioned by Klein and Murray [2008, 2010]. Furthermore, movement in the scene is eliminated as this would touch another topic of research, namely

realistic animation and physics. Also, the context will change when an object is in motion [Hasic et al., 2010].

For the rendering of the virtual scene the parameters chosen to be implemented and evaluated are:

- Rendering:
  - Shadows
    - High frequency shadows — Baked and real-time
    - Low frequency shadows — Baked only
  - Materials
    - Diffuse
    - Specular — With and without highlights
    - Full reflection — Using a cube map
  - Lights
    - Shading with 2 to 16 lights — Ambient and spot light

The high frequency shadows are shadows given by the spot light in either the virtual or real case. For the virtual case the high frequency shadows can be cast by a directional light in real-time or by any light set to cast shadows in Unity's global illumination system, Beast (see examples in Figure 2.5).



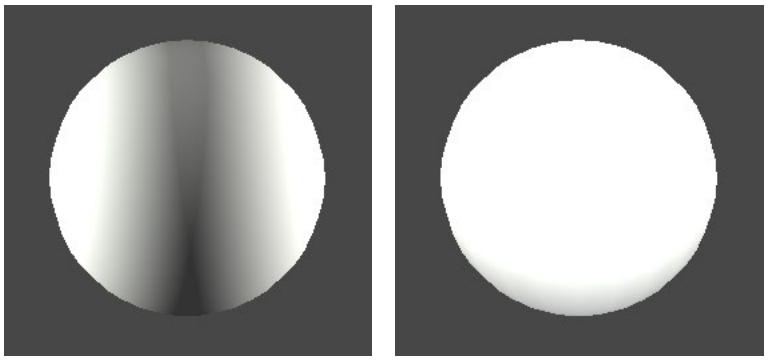
(a) Baked low frequency shadow (b) Baked high frequency shadow (and low frequency shadows) (c) Real-time high frequency shadow

Figure 2.4: Examples of the shadows.

For the reproduction of the lights and the reflections a environment map has to be created. This would require to capture the environment either by stitching photographs or by photographs of a chrome ball. With this approach it should be possible to reproduced the illumination correctly from the resulting high dynamic range (HDR) environment map. The global illumination system in Beast will only be used to produce shadows, so aspects as colour bleeding and shading on the

objects will not be included. Colour bleeding can presumably be omitted since it is considered a very subtle effect and might not be decisive for synthesising photo-realism.

The lights in the scene that shade the object in real-time should be reproduced in relation to the HDR environment map. So the light temperature and intensity and distribution of light is addressed. The number of lights is 2 to 16, where 2 is hypothesised not to be enough and 16 light should be sufficient (see example in Figure 2.5). These assumptions are based on previous work, see [Rademacher et al., 2001; Madsen and Laursen, 2007].



(a) Shading with 2 lights (b) Shading with 16 lights

Figure 2.5: Examples of the number of lights needed to shade an object.

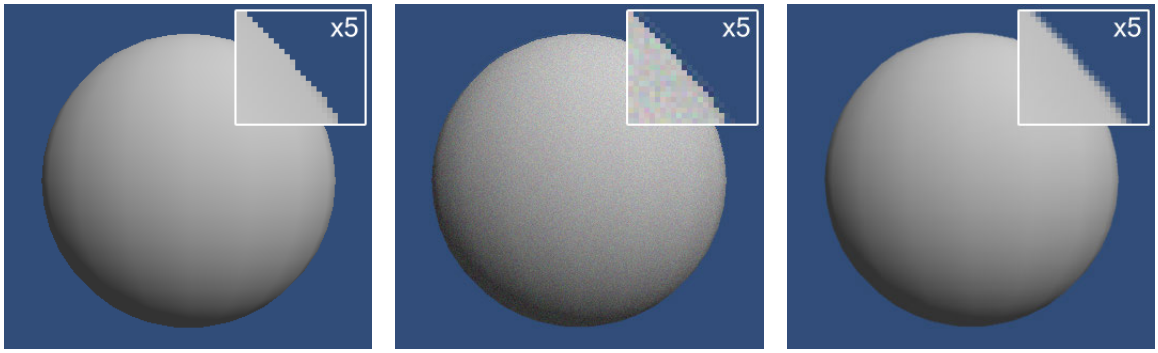
The parameters chosen for the project that are related to the camera is:

- Camera:
  - Noise
  - Colour-correction
  - Anti-aliasing

Moreover some camera artefacts should be reproduced. Noise is a perceptually easy thing to notice since it is a constant changing contrast or illumination which the HVS is sensitive to [Henning, 1988]. Therefore it is important to reproduce the noise given by the camera sensor (see example in Figure 2.6).

Colour temperature between what is seen on the video feed, the real object and the virtual object is also something that could vary a lot, since both the texture colour, the web-camera, the SLR camera (that makes the HDR environment map and eventually the light in the scene) and Unity all do some kind of their own colour interpretation which probably is displaced between each other. Moreover, a amount of blur should be added to the virtual object, since the web-camera have a depth-of-field.

Anti-aliasing is also a method what could be advantageous since it would hide the hard edge after the rasterazation, which presumably would be easily notable given it is related to the silhouette (see example in Figure 2.6).



(a) Without anti-aliasing and noise (b) With noise (c) With anti-aliasing

Figure 2.6: Examples of artefacts.

As AR is to be used in constant movement, motion blur should also be considered. Moreover distortion could be considered, but since most of the image is from the real video-feed and the distortion is most noticeable in the corners, the vignette and projection distortion will not be accounted for, since the virtual object is placed in the centre of the image.

During the evaluation, control group scenes with the real objects will be used, plus combinations of the various parameters applied in different scenes as well.

2.4 Summary

The following diagram present an overview of the mentioned parameters and the implementation methods that are planned. Moreover, the work flow and expected dependencies are depictured in Figure 2.7.

With the choices taken described in this section a small scale framework is defined (given that there are numerous parameter to include). Here both rendering, scene and light setup, as well as camera factors are addressed, which will be run by Unity with Vuforia's AR solution. Moreover the scene will be simple so the consideration of the object in a certain context is limited.

Even though a rather extensive demarcation is made, and presumably a lot of factors were not even considered in the first place, there are still many aspects to see to. The aspects of the chosen framework should account for both the HVS (including, cognitive psychology and perception), computer graphics and image processing to synthesis photorealism. This is important when not only a specific feature is evaluated but an overall coherent approach is taken. This coherent approach is considerably easier given a known static environment which will be used in this project.

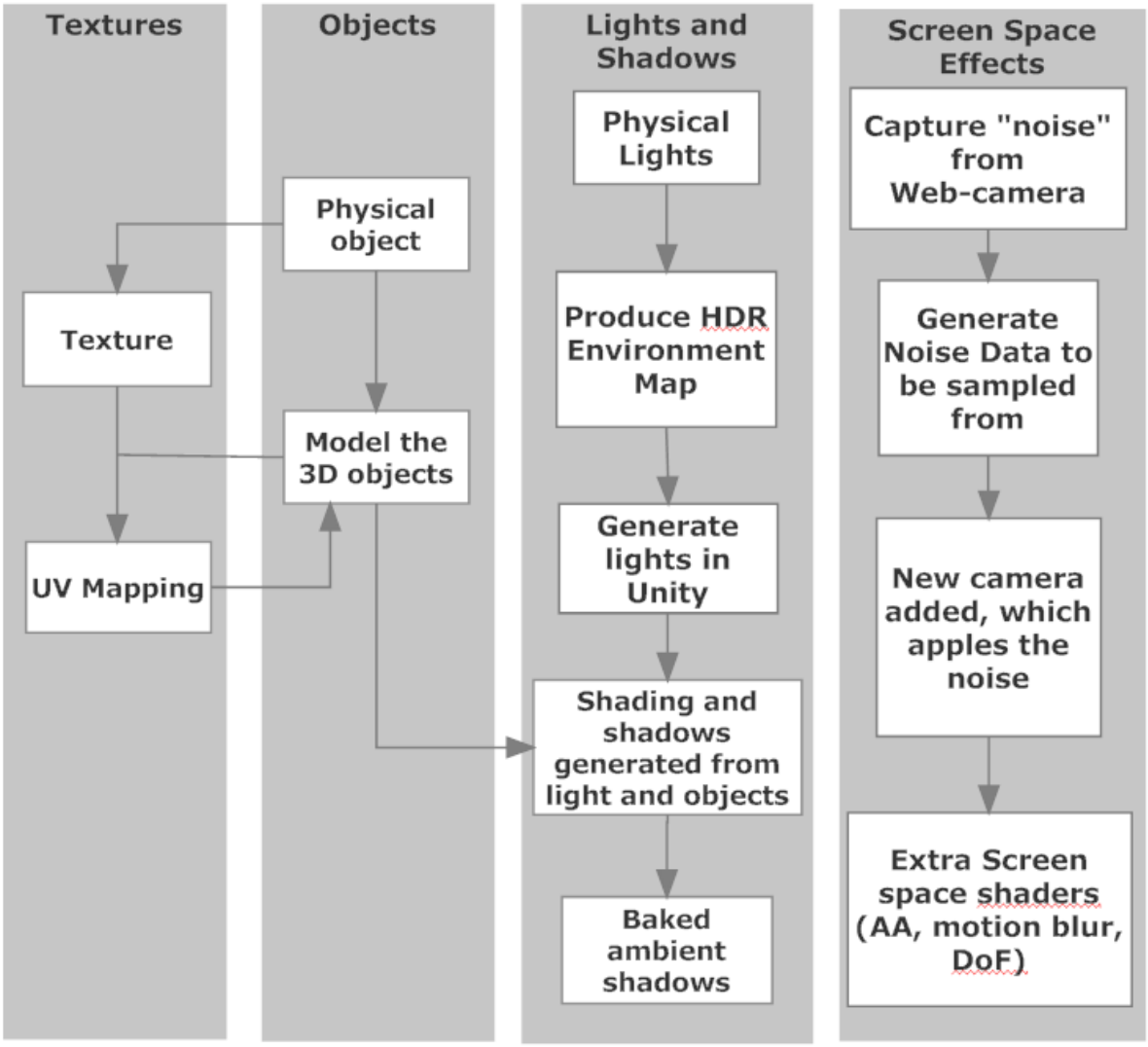


Figure 2.7: Bright boxes represent the planned implementations.

Chapter 3

Methods

To evaluate the parameters chosen a setup needs to be created, the objects and the environment must be captured and imported into the virtual scene, and artefacts must be simulated to integrate the objects into a scene. These steps will be reviewed in this chapter.

3.1 Setup

A setup is needed to evaluate the influence of different parameters. To be able to generalize the results to different scenes, two different light setups are used; an ambient light setup and a spot light setup. Also, the test subjects should be able to view the scene from different angles, hence the position of the camera and the monitor should be able to move.

To achieve ambient lighting a lot of lamps with diffuse filters can be used. The more lamps, the more ambient lighting. To achieve the spot light a single spot with a high intensity can be added. As the test subjects should also be able to change the perspective (without the lamps occluding the field of view) the lamps should move with the web-camera. This however would affect the high frequency shadow of the spot light, as it would always fall in the same direction independent on the perspective from which the scene is viewed. Therefore, it is more appropriate to keep the position of the lamps static. This also means that the lamps cannot be positioned everywhere, and therefore might not create a "perfect" ambient light. In the best case the test subjects should be able to hold the monitor and web-camera, and have the possibility to walk round the marker is wanted. However, to ensure that the tracking is never lost, the web-camera should be restricted to always point towards the centre of the scene. This should be achieved with a rick. Also, the test subjects should not be able to see the real scene (only through the video-feed), therefore the test subjects' view should also be restricted. A setup which takes into account all of these considerations is described.

Five lights with diffusers (three 65×65 cm and two 53×53 cm) are set up in a circle with a radius of 1.5 meters and with a distance to each other of 65 degrees (see Figure 3.1). In the centre is a table on which a marker is placed. The five lights are located one meter higher than the table and points upwards with a 45 degree angle to reflect the light in the white ceiling. A spot light is located between two of the ambient lights and is located 40 cm higher than the ambient lights to minimize the length of the shadows from the objects, such that they are visible in the field of view of the camera. The whole setup is covered by white sheets to enhance the ambient lighting of the scene. The ambient lighting setup illuminates with approximately 100 lux, while

the ambient together with the spot light illuminates the scene with approximately 300 lux at the centre of the setup. This creates a balance between the two light settings, which makes sure that the spot light casts a visible high frequency shadow, even with the ambient lights turned on. To avoid the real object and the virtual object from occluding each other the field of view of the user must be restricted to 90 degrees (see Figure 3.1). The scene could be seen from above to avoid occlusion, however this is not a natural augmented reality viewing situation, and the depth in the scene is limited. Moreover, occluding shadows can also be avoided by restricting the field of view and by creating a small cast shadow, by placing the lights higher than the objects. The setup is furthermore set up to avoid:

1. Shadows from the test subjects appearing in the scene
2. Light shining directly into the camera
3. Excessive motion blur

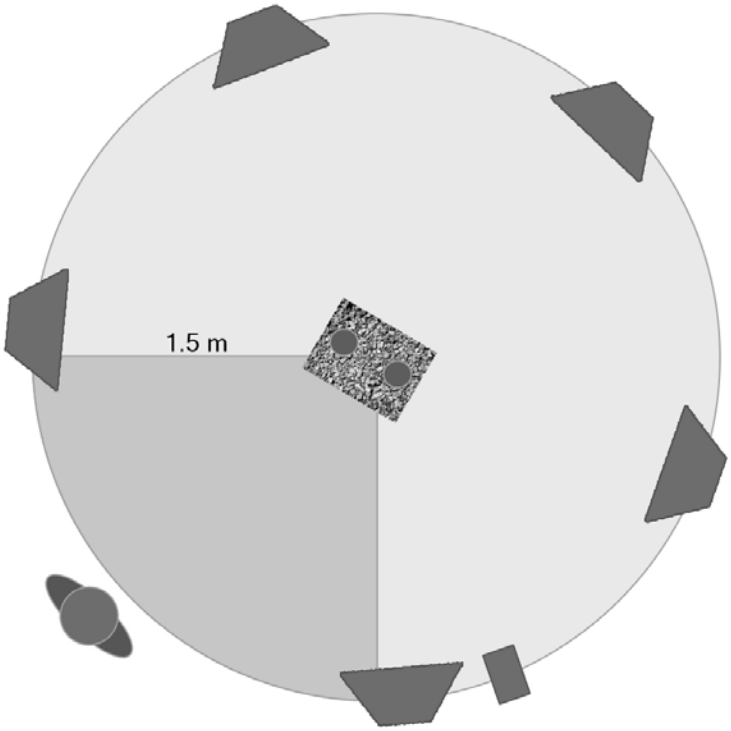


Figure 3.1: An overview of the setup. The lamps are displaced 65 degrees from each other and the spot light is 20 degrees from the closest lamp. The user can watch the scene from a maximum of 90 degrees.

The first point can be avoided by placing the lights out of the field of view area. The second point can be avoided by viewing the scene slightly from above, while the lamps also are placed above the scene. The third point can be limited by attaching the monitor and web-camera to a

rotating arm, which has a maximum speed of rotation. This is achieved by a wheel on the end of the arm touching the ground. This also limits vibrations of the arm. By attaching the monitor and web-camera to an arm fixed to rotate around an vertical axis the freedom of movement gets limited. However, the fixed degree of freedom will ensure a more stable (and non-losable) tracking and will prevent test subjects from getting tired in their arms if they where to hold the screen themselves. This arm can also be used to create the field of view area, if limited to rotate only 90 degrees.

The metal arm is mounted in the ceiling and spans 1.5 meters in radius. On the arm the monitor and the web-camera are attached. Because the web-camera has a static lens, it has to be physically moved to zoom in on the scene, such that the details are clear. To do this a sub-rack is used (see Figure 3.2). The test subject will then be able to move the monitor and the web-camera accordingly. To cover the scene, sheets are attached to the arm, such that the test subjects are only able to view the monitor (see Figure 3.3). Furthermore, the sheets prevent the user of observing the rest of scene, which also should minimize the knowledge of the context in which the test subject observes the object.

Figure 3.2: An image of the scene. The web-camera is positioned closer to the centre by a sub-rack to capture the scene in details.

Both the real and the virtual object is placed on the custom made and printed A3 marker. The fact that both types of the objects are placed on the marker should avoid hinting the user about which object is bound to the marker. Additionally, a large marker will ensure better tracking capabilities and make it possible to preform a side by side comparison, where both objects can be placed on the marker. To insure real-time rendering while handling a video-feed and perform tracking a desktop PC is used. The following specifications are given for the hardware:

- A Logitech C920 Pro web-camera is used, which features HD video recording in 1080p.
- A 22" Samsung SyncMaster 226BW monitor is used to display the stream. The monitor has a resolution of 1680×1050 and a contrast of 700:1.
- A PC with an Intel i5 CPU, an AMD 6970 HD 512MB ram graphics card and 6 GB RAM.

Figure 3.3: An image of the outer setup. The test subjects is only able watch the monitor and not the scene. Furthermore, they are able to rotate the metal arm on which the monitor is attached.

3.2 Vuforia

Vuforia provides marker based AR where a custom image can be utilized as the marked. As the video feed is utilized in unity, the Vuforia algorithm will analyse the feed with the trackable marker and position the camera according to the perspective.

Vuforia creates three cameras within Unity during runtime, the first one holding the video feed data (see Appendix B on page 89 for more information about the data flow between the camera).

An orthographic “background” camera renders a plane which have the video feed applied as a texture. The last camera, called ARCamera (which is the main camera of the scene), renders the game content on top of the background camera render. With those cameras and a plane representing the marker used within the scene the Vuforia solution preforms its tracking algorithm, which is not assessed in this project.

Stability of the tracking is crucial, since imperfect tracking can result in an observable misplacement of the objects. Moreover, as this displacement would happen per frame the result would be jittering of the virtual object, which again would result in a biased evaluation in relation to the importance of the different parameters. Therefore, it is important that the marker provides as many contrast features as possible such that the tracking is as stable as possible. Vuforia provides a web-service which assesses ones custom chosen image for the marker in relation to the tracking capability. The chosen marker for this project is evaluated as having the best quality from the web-service. The marker and a more detailed description of the quality of the marker can be seen in Appendix B on page 89.

3.3 Test Objects

As mentioned in the previous section the objects have to differ to be able to evaluate as many parameters as possible. For the physical objects the following aspects were to be evaluated:

- Reflection level
- Shape
- Texture
- Colour

In relation to the parameters chosen, the reflection level is required to range from diffuse to very reflective. Regarding the remaining items the shape has to be replicated in a virtual version so it should be relatively simple in its structure. The texture and colour also have to be replicated so it should be abstract and simple with not too many different colours, since it could prove to be difficult to preform a seamless UV-mapping and colour-temperature correction.

A diffuse candle holder with an abstract texture with small features was chosen to represent the diffuse reflection level. A chrome candle holder was chosen to represent the highest reflectance level, which do not have a texture other than the mirrored environment, while a yellow specular toy elephant was chosen as the object in between. See the objects in Figure 3.4.

Figure 3.4: The three objects chosen for the experiments.

3.3.1 Capturing and Modeling

Instead of modelling the objects a virtual copy was created using Autodesk's 123D Catch, as it requires much manual work and an investigation of an automated process could show to be advantageous. This was done through a capture of around 20–40 images per object from 360 degrees around it. Thereafter, the program stitch them together to a 3D object by structure-from-motion. However, the application had some limitations such as:

- The object must be diffuse
- The object must be lit up uniformly
- The object must not occlude itself too much
- The object must be opaque
- The object must have a characteristic texture to be able to be tracked

To optimize the tracking of an object a marker was laid below it. That way, the stitching had more reference points if the object had a low-contrast texture. The diffuse candle holder was created in this way, with addition hole patching and smoothing in Autodesk 3ds Max. Because the elephant is specular and uniformly coloured the application would not be able to generate any reference points on the object from the captured images. Therefore, the elephant was painted and taped in with dark tape such that it had a high-contrast texture (cf. Figure 3.4). To limit the highlights of the light when taking an image from another angle, the diffuse lights illuminating the scene was moved around. Since the chrome candle holder is fully reflective it was created by spline modelling using the lathe tool in Autodesk 3ds Max.

Furthermore, two low-polygon versions of the objects had to be created in order to evaluate what influence the object complexity parameter has. The two low-poly versions were created using the scanned objects in from 123D Catch as references, where contour modelling could be performed on top of the scanned reference object. This was also done in Autodesk 3ds Max.

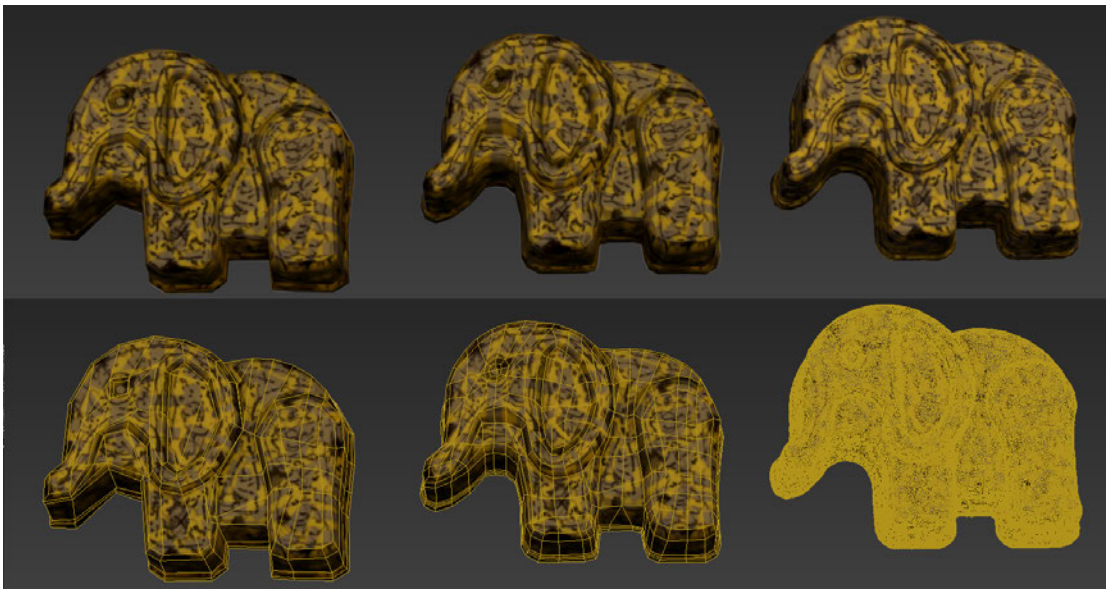


Figure 3.5: From left to right: The specular toy elephant in two low complexity versions (461 and 1028 polygons), as well as the scanned original (52387 polygons). Bottom row shows the same objects in wire frame.

This process of importing the objects in the virtual reality is presumably an important step. Since the human test subject might assess the objects in detail the object must be truly replicated. The process used with 123D Catch yielded an acceptable result, whereas if the replication were to be modelled by hand the process and result could show to be insufficient.

3.4 Simulation of Camera Artefacts

To integrate the objects into the video-feed artefacts of the web-camera must be addressed, and will be reviewed in the following sections.

3.4.1 Noise Estimation and Generation

One artefact produced by the web-camera is noise, which is presumed to be important since the small intensity changes covering the content is something the HVS is sensitive to [Henning, 1988]. So, a certain lack of noise over the virtual object could make the virtual objects look uncanny. Noise can come in different ways as unwanted grains in the image [Quantum Scientific

Imaging, 2008]. In many cases the noise is just assumed to be Gaussian distributed [Fischer et al., 2006; Klein and Murray, 2008].

Before assuming a (possibly mistaken) specific distribution of noise, the noise can be measured by capturing a sequence of images [Bovik, 2005]. The mean between the pixels in these images is assumed to be close to the real pixel value. From this mean the deviation is considered to be the noise.

A test with 10 captured images was implemented to specify the distribution of the noise. These images captured a grey paper. Each pixel value (R, G and B) was compared to a mean of all the pixels in all the images for the specific channel. However, the distribution does not follow a continuous pattern and the deviation from the mean is relatively large due to uneven lighting of the grey paper.

To overcome this incorrect deviation, in the second approach, the pixel values should be compared to a mean for each pixel across the number of images rather than comparing each pixel to the mean of all the pixels in the images. The histogram for this approach can be seen in Figure 3.6-bottom. This approach gives a more realistic view of the noise.

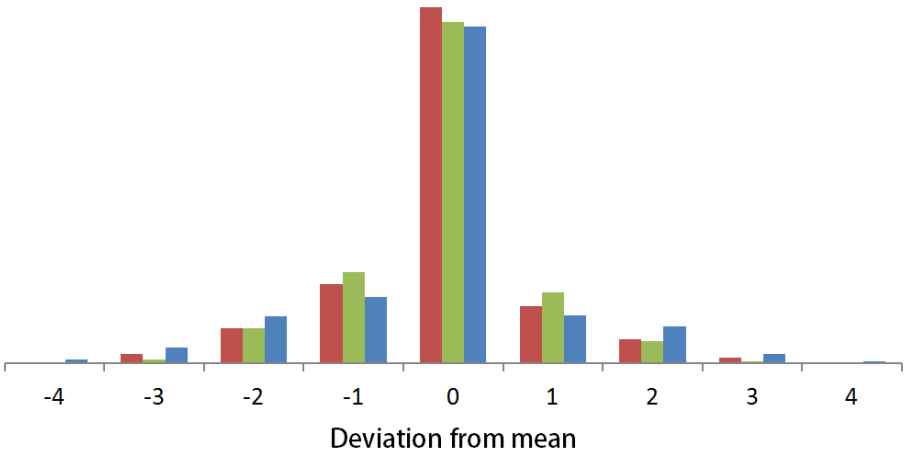


Figure 3.6: Example of a histogram of the deviation from the mean.

For some cameras the noise is separated in tiles (quadratic group of pixels), in which the noise is interpolated. This is presumably due to the compression of the data, which happens internally in the web-camera. To compensate for this a 3×3 pixel neighbourhood is used to create a tile. A 50 % weighting between the centre pixel and a neighbour is used for the neighbouring pixels. This gives a smoother transition between the pixels in the tile. A tile of 5×5 pixels was also tried. For the outer edge pixels, a 25 % weighting was used between an edge pixel and the centre. This gave an even smoother transition. However, this tile approach was not necessary for the final experiment setup.

A Gaussian function can be estimated from the histogram, but as the data is already sampled (and thereby follow the true distribution of the noise) the histogram can be used directly. To be able to sample random noise directly the data is summed to a cumulated histogram. By sampling

a random value on the second axis, the noise value can be found (see Figure 3.7). This can be done for each channel.

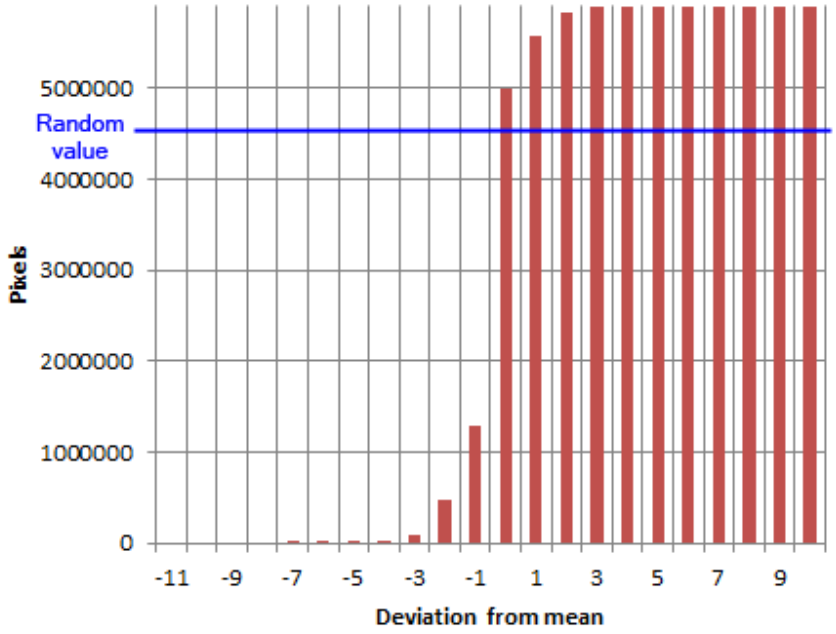


Figure 3.7: A random value on the y-axis (i.e. a random number of pixels out of the total amount of pixels) can be mapped to a given deviation on the x-axis. An example of a random value on the second axis is shown, which results in a deviation of 0, as it intersects with that column.

This approach does not take into account the relation between the three channels, as there might be a correlation. For instance, if a deviation is positive for one channel, the deviation of the other two channels should also be positive (in cases where the deviation is not close to zero), and within the range of the sampled noise values. This would limit additional intensity changes where one channel is more dominating than the other two channels, which perceptually would be more likely to be noticed.

To take into account such a relationship a covariance matrix is calculated, which addresses the noise variance within the channels and the covariance between the channels. By a Cholesky decomposing of the matrix, the random samples from the three channels can be transformed into correlated samples [van den Berg, 2012; Apiolaza, 2011]. If one considers the RGB noise-pixel-values as a vector the Cholesky decomposing matrix is multiplied with these values and transforms them to a fitting representation as if the channels were correlated. The random correlated samples are sampled for each pixel and saved in a texture (see Figure 3.8).

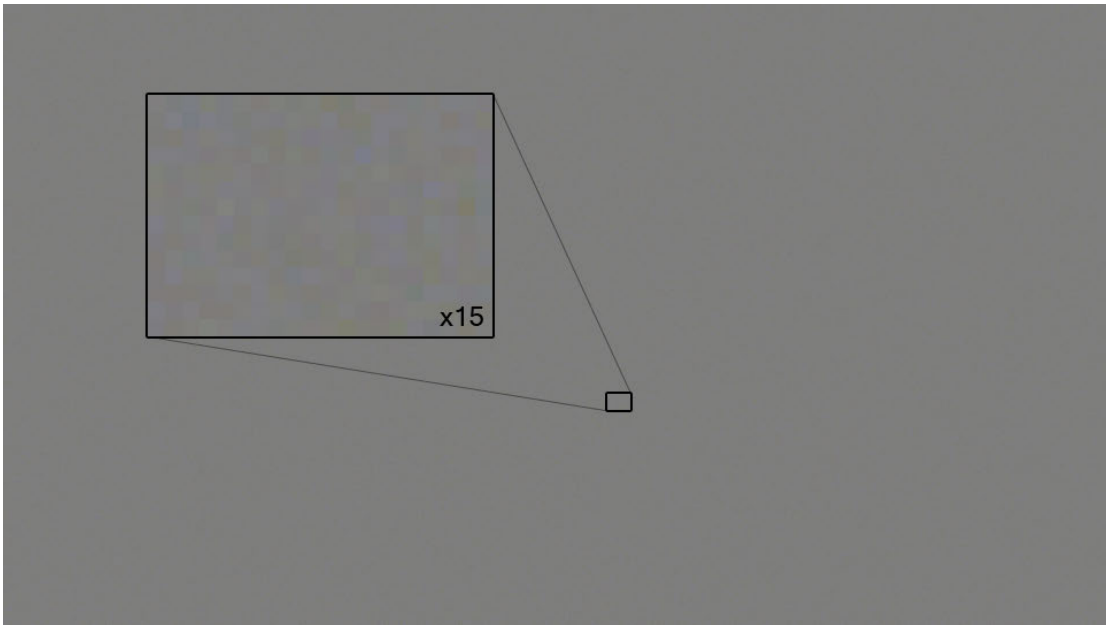


Figure 3.8: The resulting noise texture. The deviation is saved around the value 128 to ensure not to exceed the range between 0 and 255, since the deviation can be both positive and negative.

The whole process of the noise sampling and generation can also be seen in 3.9.

The noise texture is used by a screen-space shader that adds the noise texture to the scene. The noise texture is repeated, and offset randomly in x- and y-directions for each frame. Using a pre-calculated noise texture is much faster than randomly generating the deviation for each pixel in real-time. One drawback with this approach is however that the pixel intensity of the video-feed is not considered, even though the deviation of the noise depends thereof [Bovik, 2005; Klein and Murray, 2010]. The grey reference paper is used as the noise estimation reference, as it is the mean between a totally black input and a totally white input.

For the experiment 50 grey images was used as noise estimation references. The covariance matrix for the data and the Cholesky decomposition can be seen in below:

$$Cov = \begin{bmatrix} 1.44 & 0.33 & 0.00 \\ 0.33 & 1.34 & 0.46 \\ 0.00 & 0.46 & 4.35 \end{bmatrix} \tag{3.1}$$

$$Cholesky = \begin{bmatrix} 1.20 & - & - \\ 0.27 & 1.12 & - \\ 0.00 & 0.40 & 2.05 \end{bmatrix} \tag{3.2}$$

As can be seen from the matrices (3.1 and 3.2), the blue channel is more sensitive to noise that the red and green. This corresponds to previous studies, which suggest that digital colour cameras

are more sensitive in the blue channel [Sigernes et al., 2009].

For more information about noise generation in Unity, see Appendix C on page 93.

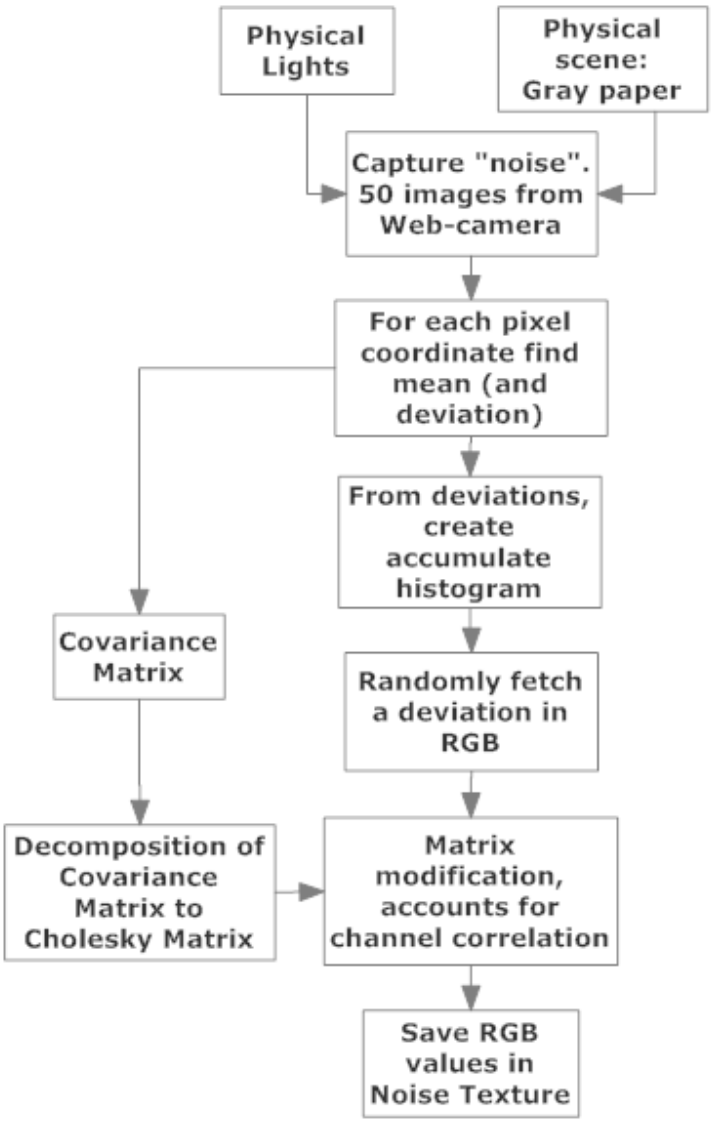


Figure 3.9: An overview of the noise sampling and generation.

3.4.2 Cameras and Cut-Out

As described, noise is generated and applied to a outputted texture from the noise generation algorithm. This texture had to be assigned only for the augmented object. The following issues have to be taken into consideration.

1. The noise should only apply to the object and not the rest of the screen-space.
2. Apply the noise in relation to the given fragment colour currently rendered.

3. Apply the noise texture UV's in screen-space coordinates rather than in model coordinates, since the noise should not wrap around the object.
4. Apply some movement to the noise so it is not static.
5. Address the limitation of Vuforia, which prevents the use of some camera functions and screen space effects.

In order to isolate the objects that require applied noise, a second camera (the forth in total during runtime, see Appendix B on page 89) is added to the scene. This camera is called *Main Camera* and it is a child of *ARcamera*, which is the camera provided by Vuforia to track the marker. Because including the Vuforia package does present some limitations in relation to *flag* options, such as *clear depth* as well as some screen space effects, a custom solution can be necessary. This is what the new camera is used for, such that a cut-out function is possible. The new camera is isolated to only render objects in the *NoiseON* layer, which is specified through Unity. It has a high depth value, such that it always renders on top of the background camera in the scene. The ARCamera does not render the *NoiseON* layer. That gives the ability to make a cut-out camera shader, and apply the noise on the objects (on the *NoiseON* layer) that are not being cut away. That way the noise is only applied on the virtual objects specified and not on the rest of the rendered scene as for example the background video feed. If this was the case noise would be accumulated over the video feed resulting in degraded quality, less of the intended data being shown to the user and an uneven distribution of the noise, essentially pointing out the virtual objects.

With a custom shader noise is added to the rendered image. This is done by taking the noise texture outputted from the noise generation algorithm (see Figure 3.8) and use it as a input parameter for the material which uses the custom shader. The second input parameter is the screen-rendered data (using OnRenderImage function).

In the fragment program the generated noise on the screen image is obtained by subtracting the colour values in the noise texture from grey (i.e. the pixel value 128), which corresponds to the value 0.5 for the red, green and blue channels. The remaining difference is added to the value of the screen texture pixel which makes up the outputted fragment (see the following code-snippet). Now the final rendered image is the original rendering of the objects with an overlay of the noise sampled earlier.

```
FragColor =
half4( mainTex.r+(0.5f−noiseTex.r) , mainTex.g+(0.5f−noiseTex.g) , mainTex.b+(0.5f−noiseTex.b) );
```

It should be mentioned that the access to the rendered screen is obtained by the *OnRenderImage* function given by Unity. As Unity renders the image the function allows temporary access (for the duration of the frame) to the frame buffer and an alteration can be done through a shader. Afterwards the temporary data can be released and the frame buffer is send to the display.

Another problem is the UV mapping. Normally, as a vertex program runs through the vertices it happens in the model coordinate system and defines the UV corresponding to them. In that case

the objects have their own material on them and the main texture is wrapped properly around the object. However, the camera shader should map the UVs to the screen space, while leaving the vertex position projection as usually. The following five lines in the vertex program project the texture area of the object coordinates to the projection plane:

```
    //ClipPos.w is -z, the inverse distance to the camera
    //Mapping of the vertex position to screen coordinates x and y (NDC)
screenSpacePos.x = ClipPos.x / ClipPos.w;
screenSpacePos.y = ClipPos.y / ClipPos.w;
    //Map the screen space range to UV range (0 to 1)
o.uv.x = (0.5f*screenSpacePos.x)+ 0.5f;
o.uv.y = (0.5f*screenSpacePos.y)+ 0.5f;
```

The whole process is summarized in the diagram in Figure 3.10.

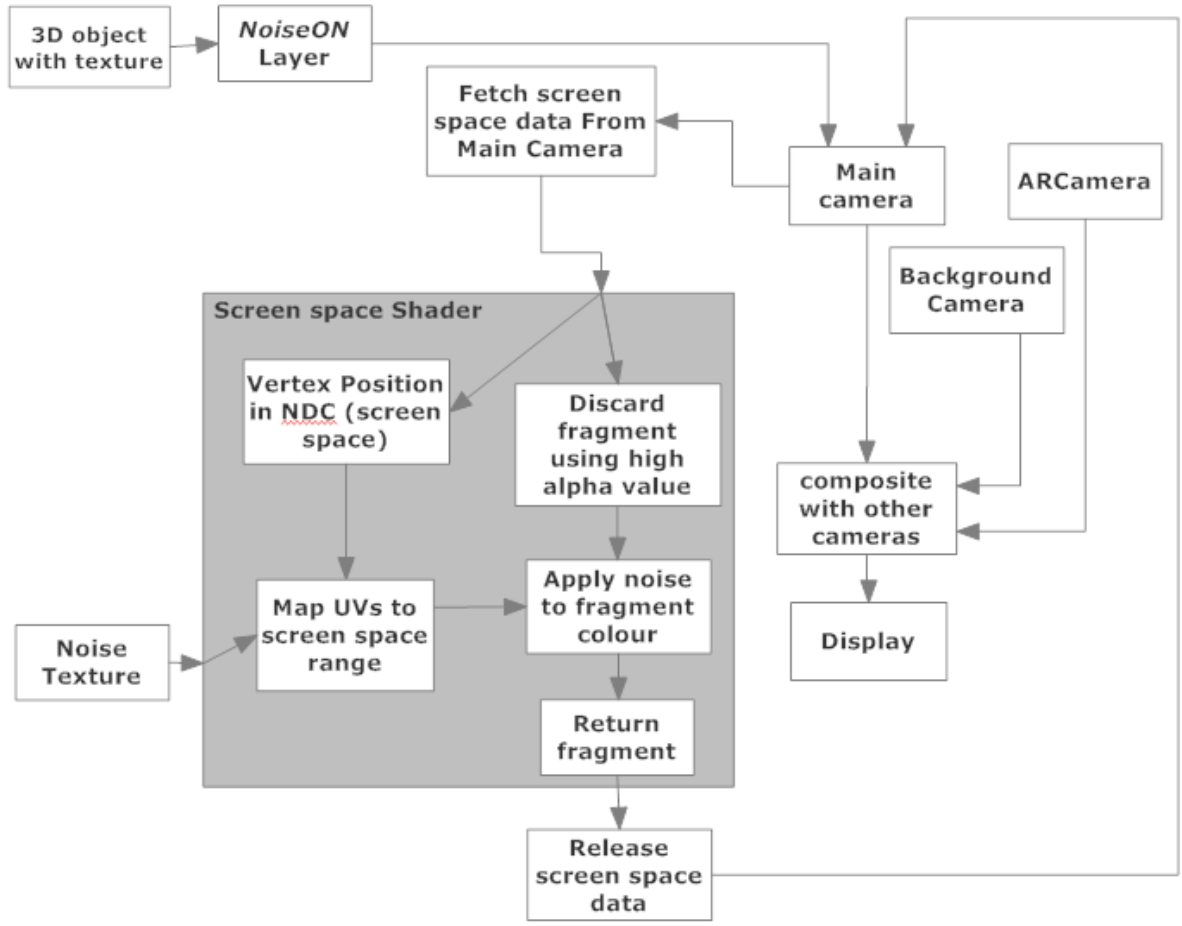


Figure 3.10: Describtion of the screen space shader (called modelViewShaderTransparent.shader) its inputs and output.

Moreover, a displacement is added to the UVs so the noise is not acting as the still image overlay. The displacement is done by adding a random value to the UVs of the noise texture.

3.4.3 Colour Correction of Texture

Because the captured texture that was generated through 123D Catch is not matching the colours of the surface on the real object — seen through the web-camera — a colour-correction is needed. If the white balance and colour temperature is too different from what would otherwise appear in the scene this misplacement could be noticed by the test subject, even if the experiment is not a side by side comparison. To balance the texture colour of the 3D objects to the representation of the real objects through the camera, two different implementations of colour matching were implemented.

The first approach is based on Lam and Fung [2009] and resembles a simple equalisation method, where a region of interest (ROI) of a web-camera image of the texture from the real object is used as a source reference. The mean and standard deviation for the three colour channels are calculated. Then a ROI from the texture of the 3D object is used as the target reference, where the contrast is modified for each channel such that the standard deviation is the same as in the source reference. The contrast is modified by multiplying each pixel value with a constant. This affects the mean of the three channels, therefore, each pixel is translated to match the source reference. This is done by adding or subtracting the difference between the means for each channel. The contrast multiplier and the translation is then applied to the texture for the virtual object that needs to be colour matched.

Because the texture from the elephant consist of yellow plastic and grey tape, a division of the image into two parts might improve the result. Two colours from the source reference and two

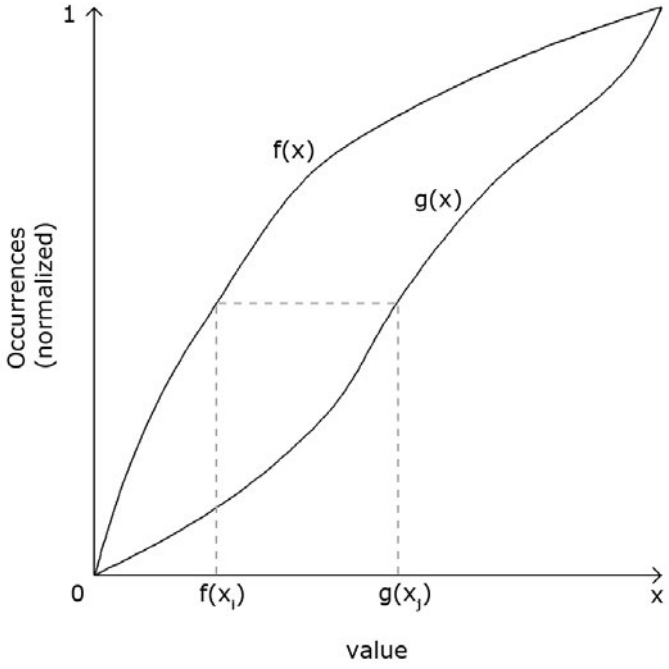


Figure 3.11: For a given pixel value x_i, the number of occurrences are found for the cumulative histogram f for the target image. The number of occurrences are then found in the cumulative histogram h for the source image, and the new output value, x_j can be found.

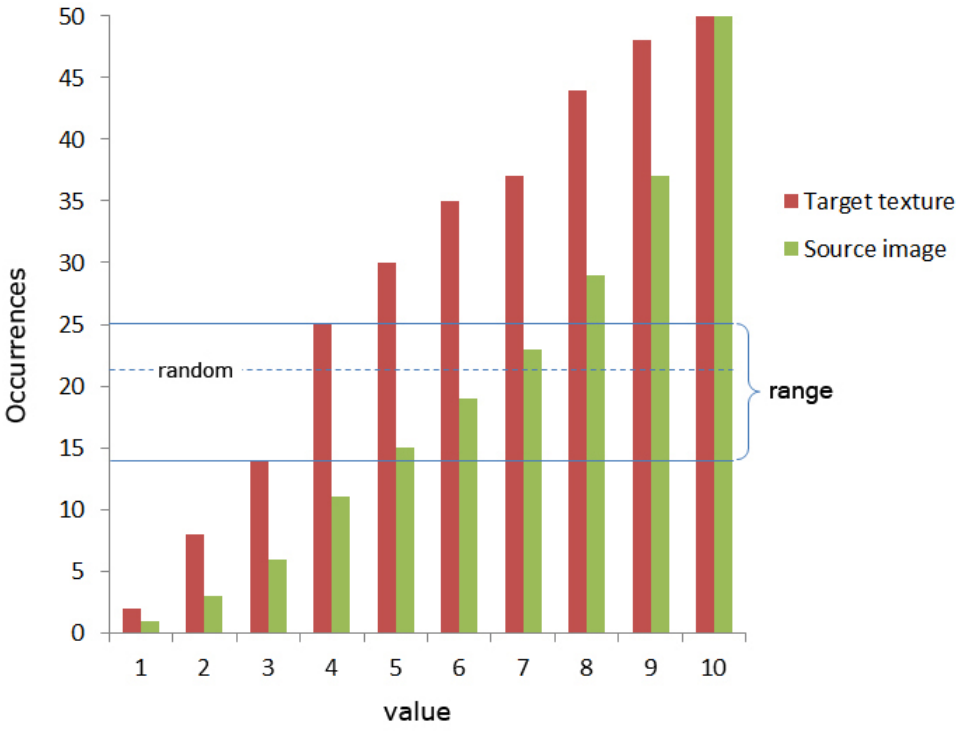


Figure 3.12: Some pixel values can be matched to more than one value. In this example the value 4 can be mapped to 5, 6, 7 and 8 (see blue interval). To overcome this problem a random value of occurrences are found between the number of occurrences for a given pixel value (in this example 4) and the number of occurrences for a pixel value lower (i.e. 3). Because this is a cumulative histogram the probability for the random mapping will be distributed according to the actual probability (the probability for mapping 4 to 6 is higher than mapping 4 to 5, as more occurrences are present for the value 6 in the interval than for 5). In the example the random value (the dotted line) will be mapped to the value 7.

colours from the target image is chosen to represent the general colour of the yellow plastic and the general colour of the grey tape. The image is then divided into two by the nearest-neighbour method (i.e. the lowest distance from the given pixel value to one of the two chosen colours). The method above is then applied to each of the two divisions.

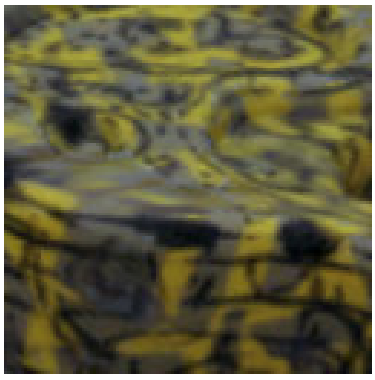
The second approach uses histogram matching. This implementation takes in a source ROI and a target texture. A summed histogram is created for the source ROI and the target texture. As the target texture might be larger (or smaller), the summed histogram for the source ROI is scaled to match the number of pixels in the target image. For a given pixel value in the target texture the number of occurrences is found in the histogram. For the given number of occurrences a pixel value is found in the histogram of the source ROI. Now the pixel value of the target texture can be mapped to the pixel value of the source ROI (see example in Figure 3.11). The histogram matching is conducted for each RGB channel.

Because the cumulative histogram consist of discontinuous values, the same number of occurrences in both histograms will rarely occur. Therefore, a range is used between the number of occurrences for a given pixel value and the number of occurrences for the pixel value below the

given pixel value (see Figure 3.12). A consequence of this is that a pixel value can be mapped to several values depending on the given position in the range. Therefore, a random value is chosen within the range. Because it is a cumulative histogram the random value will result in a correct probability of a given pixel value within the distribution of pixel values of the source ROI. In other words, the more occurrences of a pixel value, the more probable it is that it is mapped to.

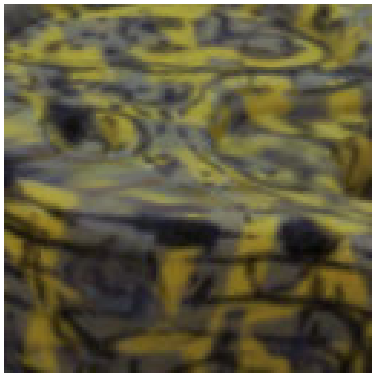
(a) Source ROI

(b) Target ROI



(c) With standard deviation and mean correction

(d) With standard deviation and mean correction for image divided by colour swatch



(e) With histogram matching in RGB-space

(f) With histogram matching in HSV-space

(g) With histogram matching in Lab-space

Figure 3.13: Example of different methods for colour matching.

Colour matching was tried in three different models of colour-spaces. First, an implementation of the RGB model was used. Secondly, in order to have a comparison a HSV model was implemented [Moeslund, 2009]. The RGB values were converted to HSV and the histogram was matched in HSV space. This gave a better result when assessing the elephant texture. Some parts of the texture (for instance the tape and plastic) are very different from each other. This results in a problematic distribution of the values in the histograms, which result in an incorrect mapping of the hue. To avoid this, it is possible to clamp the range of hues. In other words, unwanted colour mappings can be removed.

In order to try out a perceptually uniform colour-space [Reinhard et al., 2002] the CIE Lab-space (D65 illuminant) was implemented [EasyRGB, 2013], but the result in comparison to the HSV-space correction was not better. Furthermore, because the target texture consist of the UV-map and a uniform background colour, a background colour picker is implemented to separate the background colour from the texture colours. In this way only the relevant parts (i.e. the UV-mapped parts) are taken into consideration when matching the histogram (more information in Appendix C on page 93).

An example of the implementations can be seen in Figure 3.13

During the first iterations the HSV-space was the most satisfying colour-correction, but after internal test it was realised that further corrections were probably needed to match the colours more exactly.

3.5 Light Setup and Illumination Conditions

One of the key factors within photo-realistic computer graphics is the lighting, since it is what makes the scene visible and illuminates it in a specific way. Giving the goal of the project, it is importance to replicate the illumination as it is distributed in the physical setup. In the following, an overview of important illumination factors are presented followed by a description of the pipeline used to create the lighting conditions.

3.5.1 Global Illumination

Global illumination is an umbrella term for many illumination methods implemented in algorithms which illuminate the scene. It describes both directional light and in-directional light given emitted light, reflection, scattering, refraction and more. The overall illumination is often controlled by radiance and light bounce parameters in a rendering program. The result is a scene illumination that approximates the reality which sometimes uses the law of physics to mimic the light behaviour. This can give the following computer graphic effects; reflections, refractions, colour bleeding, shadows, ambient occlusion shadows and more [Brooker, 2008].

Given the complexity of such a comprehensive global illumination algorithm it cannot be used for real-time applications. Instead, either rough approximations are used or a pre-computed

baking of the lights and effects onto a texture is done. For example, ambient occlusion can be implemented both in real-time in relation to the objects vertices, in real-time in screen-space, pre-computed by ray-tracing and baked (given as a part of a baked global illumination) or as a combination of a real-time and baked solution. Each solution have its own advantages and disadvantages, often related to good performance but less quality/reality or vice versa [Eriksen et al., 2012]. Given the goals of the project, it would be useful to explore the various methods, their balance between performance and level of realism and decide which solutions would be the most profitable given the test setup and the target platform, namely augmented reality.

3.5.2 Acquiring Environmental Maps

The first step to simulate the real lighting conditions in Unity is to "capture the real world". Five photographs were therefore taken with a fish-eye lens covering 180 degrees for view, three taken horizontally around the point of view, one upwards and one downwards. The camera was placed at the position where the objects were presumed to be placed on the marker, such that the reflective shader, as well as the other shaders, receive the correct light and reflect the environment correctly. The photographs are saved in both raw (cr2 extension). Moreover, the photographs are taken in 9 different exposures ranging from 1/2000 of a second to 30 second with a aperture-stop of 8, resulting in 9 images per direction. Only every second exposure level is used. Also the process is repeated for both light conditions given in the setup; ambient lighting only and ambient lighting with the spot light turned on. For the ambient setup only 7 exposures were needed (1/125 to 30 seconds).

The raw image files (converted to dng format) are imported into the program Panoweaver 8 Professional Edition and the stitching process can take place, resulting in a latitude and longitude environment map for each exposure. Thereafter, the environment maps are merged into a high dynamic range image using 9 exposures for the spot setup and 7 exposures for the ambient setup (see example of map in Figure 3.15). This is done with the program Adobe Photoshop CS5, but due to restrictions only a 16 second exposure is allowed as the upper bound. Therefore, only eight out of the nine exposure levels from the environment map with the spot was used. At this point two finished HDR maps are given, ready for calculating the lighting condition. Unfortunately, the spot was so powerful that even at an exposure level of 1/2000 the pixels are reaching the maximum value on all RGB channels. This essentially results in an incorrect representation of the range of the scene lighting as the spot is too intense relative to the environment. This means that the scene probably will result in a darker illumination, biasing the perception of the virtual object. Due to limited hardware and time the photographs were not retaken.

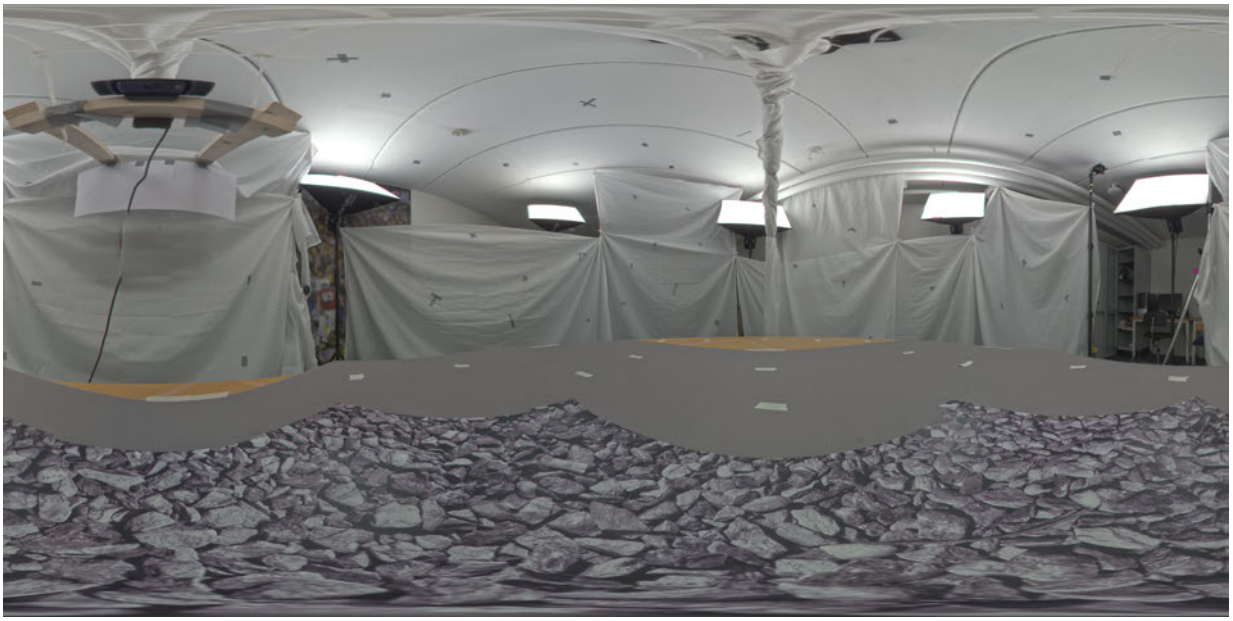


Figure 3.14: The ambient environment map used (converted to 8-bit/channel).

To create the environmental map for the scene (which should be a cube map) the HDR maps are converted to cube maps in 8-bit in Panoweaver 8. Because the environmental map is captured inside-out and the reflective objects is watched outside-in, the environmental map needs to be mirrored. These mirrored environmental maps are then used as cube maps in Unity.

3.5.3 Median Cut

The HDR environment map is imported into the program HDR Shop. HDR Shop works with images in floating point and handles high dynamic range spaces [USC Institute for Creative Technologies, 2013]. Moreover, various plug-ins are available, where median cut was of main interest, since it generates lights according to the light energy distribution across the image. The lights generated are exported into a text files format, which can be used within Autodesk 3ds Max and Maya. For use in Unity a custom script is written to read the text file correctly (see more information about importing th lights in Unity in Appendix C on page 93). Initial tests with different HDR maps were promising, as both light intensity and temperature were reproduce nicely within Unity, but as a HDR image from the project setup was used some problems occurred.

As described by Debevec [2005] the median cut algorithm works subdividing the first region (the whole image) along the longest dimension where the energy is divided evenly. This dividing is continuously iterated until the the number of specified lights is reached, resulting in multiply regions across the image (see Figure 3.15). In each subdivision (region of energy) a light source in positioned given by the centroid of the intensities in the region. The light source intensity and colour is the sum of the pixels within the region.

Since the aspiration of the project is to work with as few lights as possible and a very strong

spot light the algorithm present some problems. The light energy is divided and the image is interpreted as areas of light. This approach is good for ambient light scenes, since each light radiation from a give area of a surface is correctly represented. In contrast, if a spot light is presented by a low number of lights, the little area of pixels representing the spot light will be split up and moreover the centroid will be chosen according to all of the light energies of the current region. This results in a displacement from the spot position and its distribution of energy. The pixel values of the spot is five times stronger than any other light present in the HDR image. Scaled to a 8-bit space the spot have a value of 255 while the other light are around 50, most of the pixels in the image have the values of 1 to 5.

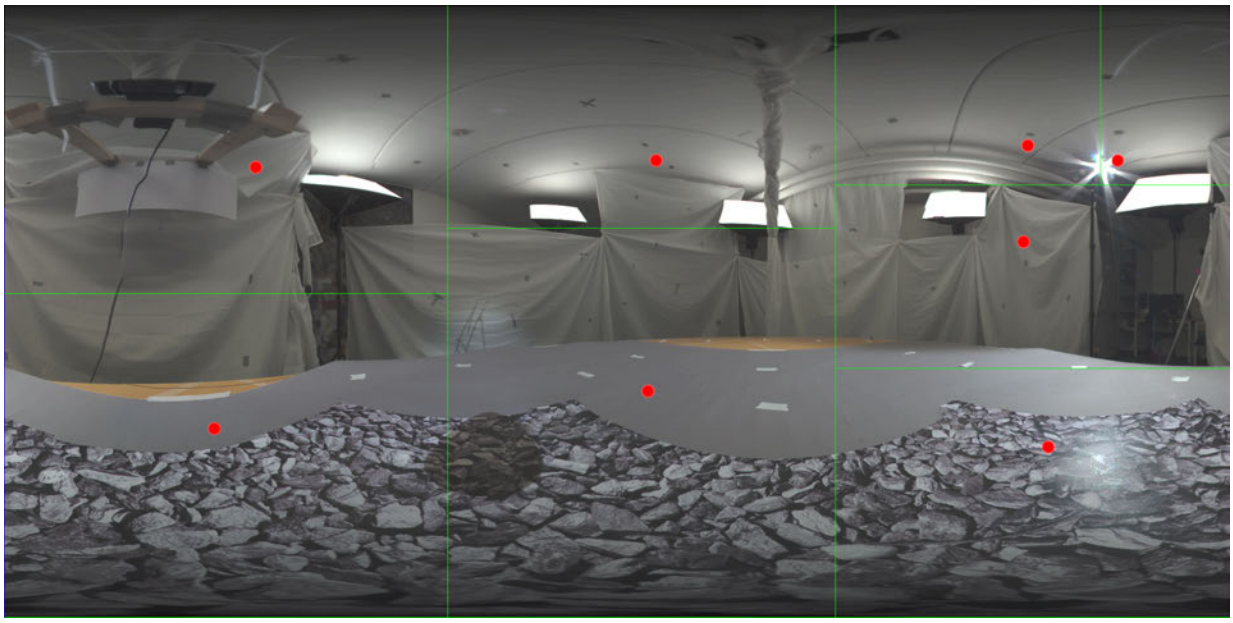


Figure 3.15: The spot environment map in HDR, where the spot is divided into several lights which represent a wide area of the environment map. Red dots represent the light positions.

To overcome this problem the following options were considered:

1. Mask out the spot light, compute the lights and finally manually add the spot in Unity at the right position with a proportional difference in light intensity as measured in the physical setup

2. Find the light source(s) computed that are closes to the spot light on the image and multiply the intensity of those lights such that the intensity is proportional to the one measured in the physical setup

3. Use a different light generator, like the “light gen plugin” [Cohen and Debevec, 2013] which represents the spot light more correctly, but still far from fully correct

4. Compute a much higher number of lights and somehow merge them into fewer lights in Unity

5. Make the spot light area N times bigger so that the centroids and regions are more depended on that area

6. Mask out the spotlight and generate the lights. Also generate lights for an image with only the spot light. Put all the lights together in the scene

The different options have their advantages and disadvantages. Number 2 have the disadvantage that the spot light will not be interpreted as a spot light but more as a dispersed area light. Moreover it is hard to account for the intensity from the whole region since the region(s) are much bigger than just around the spot light, thus representing a much more than just the spot.

Number 3: Since the plugin "light gen plugin" still does not give the right intensity the problem is eventually the same as with the median cut. The lights are represented somewhat correct position-wise, which is an improvement in relation to the spot light, which is represented by only one source light. However, the spot light is only double as intense as the rest of the lights, where it should be 5 times as intense (given from the HDR map). This is not close enough to accept as the solution. Moreover, the ambient areas are not as correctly represented as with the median cut algorithm.

Number 4 could be an eventual solution, since if the light number is high enough to create a cluster around the spot, it is represented correct. The big obstacle here is the develop a method to merge all the lights, which is considered to be a big workload given the projects limited resources. This could be done manually or with a nearest neighbour algorithm.

In solution 5 the area should be unreasonable big given the few lights present.

In number 6 it is not fully known how the intensities are computed and distributed across the regions the relations between the two intensities within the images could the incompatible. Moreover, it is not known if the lights would be centred within the spot. This method is not tested.

Number 1 was the option chosen, since the median cut calculates the ambient light nicely, while there is little workload putting in the spotlight with the correct intensity.

Step by step the pipeline can be described as shown in the list:

1. Take photograph of the scene

2. Stitch together environment maps for each exposure levels

3. Combine the environment maps into one HDR image

4. Generate light data through median cut

5. Import the lights into the rendering or game engine

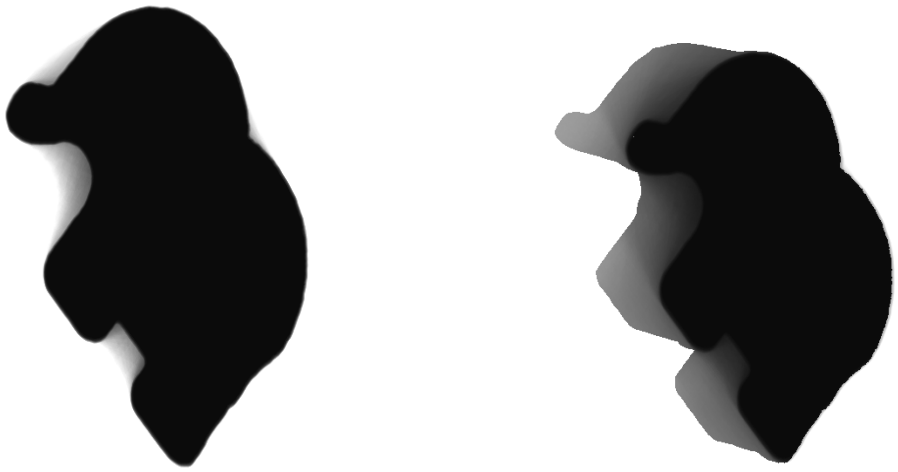
3.5.4 Pre-Rendering

Unity have some limitations in relation to lighting and shadow generation. The minimum intensity calculated is above the third decimal point (i.e. ≥ 0.01), so when generating above 100 lights with an intensity of 1, some of the light will not be accounted for. This limits the number of lights that can be used to shade the object.

Furthermore, a shadow can only be generated from one directional light in real-time. Therefore, the ambient shadows of the spot setup must be pre-rendered.

The light maps are rendered using 1024 lights (plus an extra directional light for the spot setup) in Beast with zero bounces. Beast is able to handle floating points numbers and allows a rendering with a high amount of lights. See additional information in Appendix D on page 96.

The light map is a .exr format image where the light of the scene gives the intensity and light to the map, and therefore the shadow will be represented by dark pixels in the map. Because the shadow should be used as an overlay on the live-feed, the bright pixels should be transparent and the dark pixels should remain dark. This is achieved in Adobe Photoshop by using the inverted light map as a mask on a black background (see result in Figure 3.16). The semi-transparent map is exported to Unity and used as a texture on a plane below the particular object.



(a) Modified light map of ambient lighting (b) Modified light map of spot lighting

Figure 3.16: Two modified light maps for the specular object.

The shadows are added to the live-feed using a blending based on the alpha channel. To integrate the shadows into the scene correctly the difference in intensity ratio between a shadow-point and a non-shadow point should be multiplied with the live-feed [Jacobs et al., 2005]. Due to a complex data flow between the cameras and time restrictions this problem was not investigated in this project. To compensate for the incorrect shadows, the intensity and colour of the shadows were adjusted and assessed perceptually.

3.6 Summary

Many aspect have been addressed during the planing and implementation of the test setup. It is believed that the most important implementations are done and useful methods are used. A summary is given below:

1. Stable, physical setup, that limits vibration and assures that the marker is within the view of the web-camera
2. A light setup that enables an easy switch between the creation of high frequency shadows and low frequency shadows
3. Vuforia chosen as the AR software, including the assessment of optimal tracking
4. Selection of objects that represents different parameters
5. Replication of the objects into the virtual space with a relative high precision (including low complexity models)
6. Sampling of noise and replication of it with RGB channel correlation
7. Mapping and application of noise by utilizing screen space operations
8. Preforming colour-corrections for the textures applied to the objects, where both perceptual-uniform and not perceptual-uniform methods were investigated
9. Creation of HDR environment map and thereby creating lights with the correct light temperature, distribution and intensity
10. Using the Beast system to create the virtual high frequency and low frequency shadows

Generally speaking the main obstacle is to reproduce the parameters from the real world in the virtual space. The most important item to get right are assessed to be the geometry, light (including shadows), colour-correction and noise. The geometry and lights are things that required detailed and manual work by making photographs (or scan in the objects otherwise). There are some automated steps but there are many steps to overcome if one of them fails — a result is often to repeat the whole pipeline. The noise generation and colour-correction are methods that can be automated and have been done so. The script that generates the noise is written in such a way that the noise can be sampled through the web-camera and applied to the noise texture at any moment. The colour-correction needs two textures and produces the resulting correction. Both the script run within Unity and a simple GUI is given (see Appendix C on page 93).

Moreover, AA is used on the entire screen space. Because AA is applied on screen space it will create a bit of blur and smooth out the silhouettes of the 3D objects (more information about the AA in Appendix E on page 97).

To perform such a project much time, hardware and physical space is required it is therefore important to have a clearly defined pipeline and automate as many step as possible, since many of those steps are reused both during the main implementation and while readjusting details.

Chapter 4

Final Test Scene

With the physical setup ready and the different implementations written, they all have to be assembled into the final setup. However, some changes have to be made to make the different implementations compatible with each other. Also, the pipeline have to be altered such that a final scene can be created. The final pipeline can be seen in Figure 4.1.

4.1 Setup

As planned, a setup with ambient lights and with spot lights has been created where an metallic arm can be rotated. On this arm a monitor and a web-camera is attached. The web-camera is moved closer to the centre of rotation to be able to see the objects from a shorter distance. The vibrations of the arm has been limited by attaching a wheel to the arm and making it move on a plastic underlay. This also limits fast rotations, hence motion blur. Therefore, motion blur was not implemented in the scene.

To ensure a real-time tracking the resolution of the 1080p web-camera, the resolution was lowered by a factor two to 920×540 pixels. This trade-off between stability of the tracking and frame rate was considered the most appropriate. Furthermore, the camera was positioned with an angled such that the marker was viewed more from the top (rather than from the side) to ensure a stable tracking but without removing the sensation of depth. To further stabilise the tracking different restrictions of the web camera were implemented (see Appendix F on page 98), but internal tests showed that restricting the virtual camera had unwanted side-effects. Additionally, pilot tests showed that when watching the scene from a fixed position the jittering was more noticeable then when the perspective was constantly changing. For this reason the arm was decided to be moved constantly from one side to another in the experiments.

Another problem is that the calculated field of view of the camera in Vuforia is slightly wrong, resulting in a wrong position of the virtual camera relative to the physical camera. This again results in a wrong perspective projection of the virtual objects. However, the difference is considered unnoticeable as long as the virtual object is not positioned on top of the real object. It is unknown whether or not the test subject will experience an uncanny effect, which could influence the overall perceptual evaluation.

4.2 Lights

Two environmental maps are stitched in HDR format for each light setup. However, the range of exposures are not enough to capture the whole range of intensities from the spot light. Therefore, the spot was masked out of the HDR map and the lights calculated from the modified map. Then a spot light was later added manually in Unity and the intensity was matched by measuring the difference in illumination between the ambient light and the ambient light plus the spot. This was achieved by measuring the illuminance in lux at the position where the objects would be placed in the physical scene. The difference was found to be approximately 1:2, taking the ratio between all the lights in the scene and the spot. However, because we incorrectly focused on the intensity ratio given by the HDR map this value was not addressed. This resulted in the fact that the intensity of the spot was adjusted by perceptual comparisons between the virtual and the real objects.

In addition, because the unit of the lights to achieve ambient lighting generated from median cut and the unit of the light calculation in Unity was not compatible, the intensity of the lights was scaled by perceptual comparison to match the general illumination of the scene.

4.3 Colour Correction

Because the intensity of the lights was set arbitrarily, the colour matching of the UV texture could not be achieved automatically, thus the colour correction script created was not sufficient. Furthermore, if the texture was matched to a region of an image of the real object, the texture of the virtual object would gain double light — both from the lights implied in the image of the real object and from the shading of the lights in the scene.

One way to avoid this would be to calculate the intensity and colour of the virtual lights hitting each point on the mesh (assumed that the intensities and colours of the virtual lights are adjusted to the corresponding physical light). Then the UV texture of the object should be divided by the UV map with baked lights. This would remove the double lighting and only leave the albedo. However, as a perfect match of intensity and colour between the physical and virtual light is difficult to obtain this option was skipped due to time limitations. Instead the colour and intensity of the texture was matched manually, by perceptually modifying the ambient colour factor of the material of the objects. In this way a plausible simulation of the ground texture was created, yet not in a correct way.

An overview of the final process can be seen in Figure 4.2.

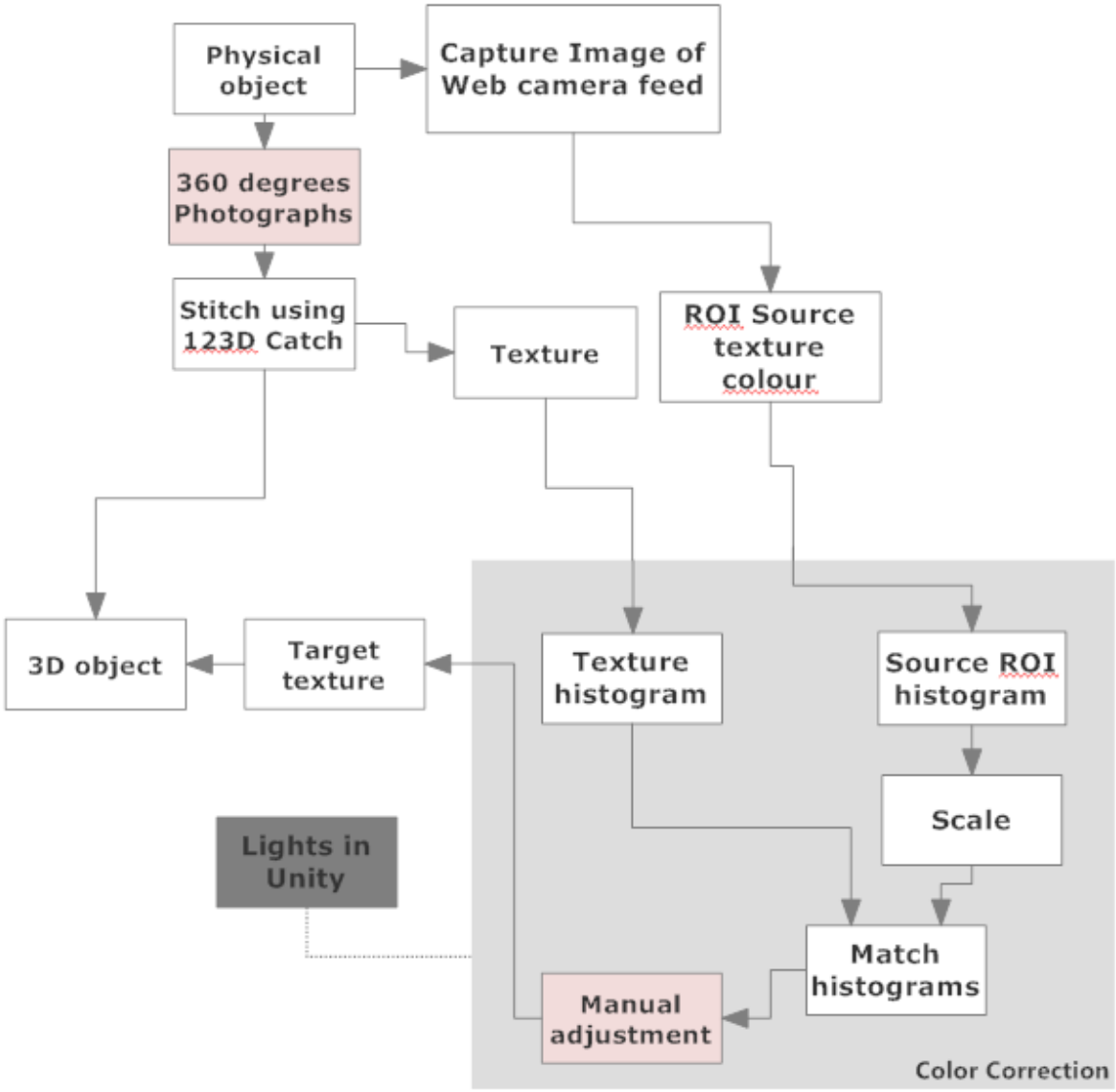


Figure 4.2: White boxes represent the implemented items and the light red boxes represent the faulty or inadequate implementations, while the dark grey shows the items that are not implemented but should have been considered.

4.4 Artefacts

The noise is estimated and correlated per pixel by looking at a grey paper in the scene with ambient lighting. A cut-out shader makes sure that the noise is only applied on the virtual objects (as well as their shadows). This however affects the AA which then is applied on the whole screen space instead of being applied on the virtual objects only. Moreover some artefacts that were not planed for occurred, which were a result of the screen-space cut-out performed on the *Main Camera*. The artefacts are given as the AA creates an semi-transparent edge around

the silhouette of the augmented object, which is party cut away by the cut-out function, which discards everything under a certain alpha value. This threshold can be controlled but as the AA blends the object and background colour (of the camera) the background colour can get expressed too much creating a border around the object. The best compromise between AA and the appearance of borders had been chosen.

4.5 Rendering and Setup

The ambient shadows and the spot light shadows are rendered using Beast with 1024 lights. Because the intensity of the spot is set arbitrarily the resulting light maps for the spot setup are modified such that the shadows are similar to the real shadows. This is achieved by changing the gamma of the light maps to make the shadows darker. Using a gamma correction preserves the intensity end-points (black and white) while the mid-tones are adjusted. A gamma of 3 is used.

To render the high frequency shadows and the low frequency shadows into the scene, a plane is used which simply uses an *unlit transparent* shader and displays the pre-baked shadows given by Beast. Additionally, the colour of the plane can be changed to tint the baked shadow to match how a shadow would look in the scene. This tint was not achieved with baking only as the light parameters did not correspond to the real light. Therefore the lights was matched manually. In the cases where the high frequency shadows are rendered in real-time another transparent cut-out shader is used which cuts away areas besides the projected shadows [Noisecrime, 2010]. The colour of the shadow plane for the real-time shadow cast is also manually adjusted. Both planes are included on the *NoiseOn* layer.

4.6 Summary

A setup with the aspects that are assumed to be most important is implemented. Some of the automated process had to be manually adjusted due to incompatible implementation and time restrictions. Furthermore, it was noted that the jittering was noticeable, so an experiment procedure was created, which limited it. Unfortunately, no time was left to implement correct integration of shadows or to redo the colour correction and light generation process. This would probably have increased the photo-realism. However, with the manual adjustments it is assumed that test subjects will not be able to distinguish the virtual objects from the real ones.

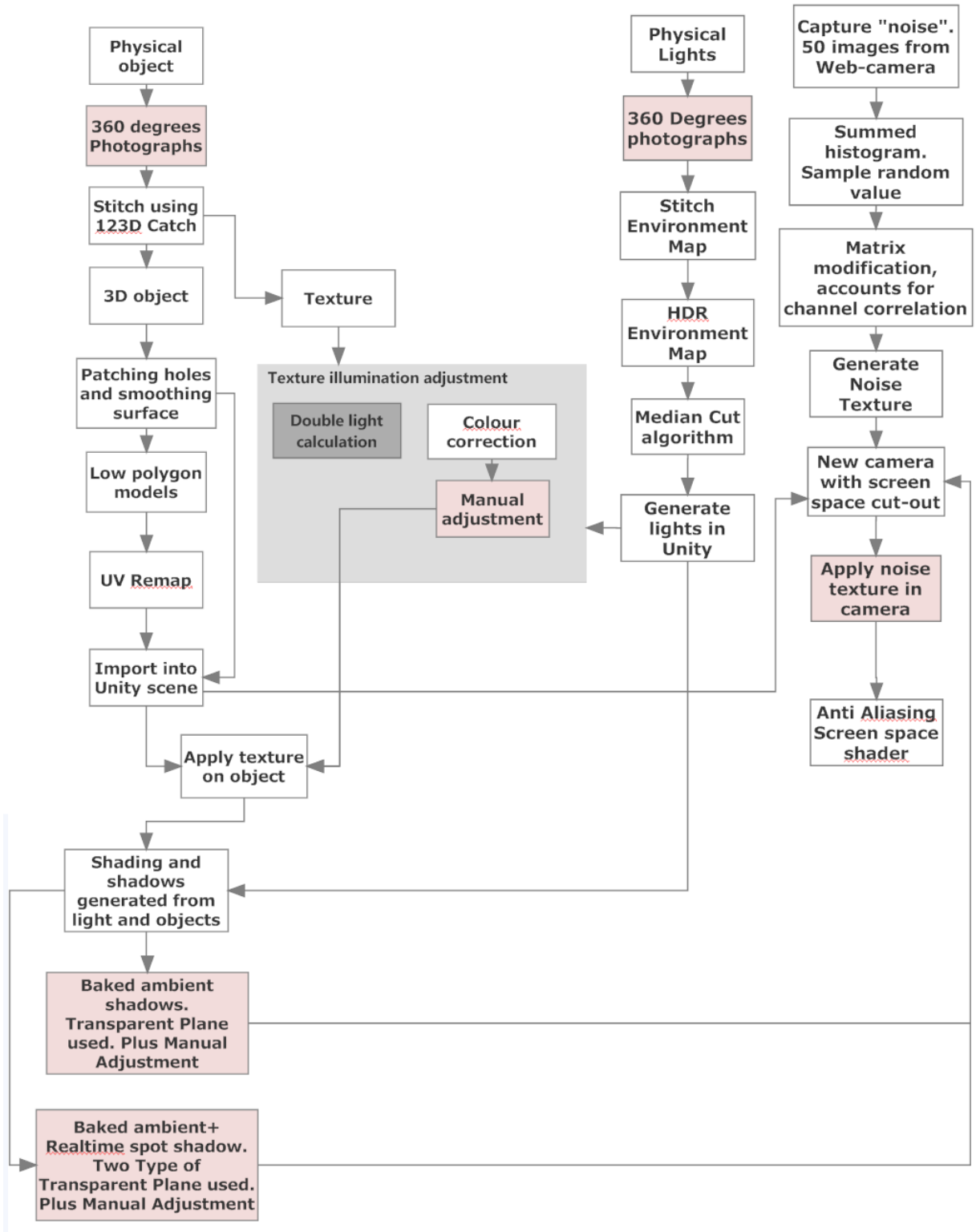


Figure 4.1: White and red boxes represent implementations, where red indicates a faulty or inadequate implementation. Dark grey means a non-implemented item that should have been used. Light grey indicates a grouping.

Chapter 5

Experiments

With the physical and virtual setup ready, experiments can be conducted to investigate the defined problem definition. The same underlying hypothesis applies for all the experiments, namely that observers cannot determine that an object in a scene is virtual.

The virtual objects are considered integrated in the scene if the ratio of answers approaches random chance, that is when test subjects are just guessing. The critical value for a scene to be significantly perceived as real can be calculated by the probability mass function for binomial distributions. The probability value used for random chance would be 0.5 given at least 100 observations [McKee et al., 1985], however, as the sample size is below 100 it might not even be possible to get a significant result with the best possible data [Bojesen, 2013]. Therefore, another probability value, p, of 0.19 is suggested to compensate for smaller samples sizes [Borg et al., 2012] and relates to the commonly used threshold of 25% — i.e. between the best case, where people are guessing (50% perceived as real) and the worst case, where people can tell the difference (0% perceived as real). The formula would then be (for at confidence interval of 95%):

$$i_c(n, p_{null}) = \min \left\{ i \;\middle|\; \sum_{j=i}^{n} \frac{n!}{j!\,(n-j)!} \left(p_{null}\right)^j \left(1 - p_{null}\right)^{(n-j)} < 0.05 \right\} \tag{5.1}$$

where $p_{null} = 0.19$ and n is the sample size. The number of guesses that a virtual object is actually real then have to exceed the critical value i_c calculated.

For instance if 100 test subjects and the probability value of 0.19 is used, a minimum of 26 of these test subjects must be that a virtual object is real, for that virtual object to be significantly perceived as real. This relates to the commonly used threshold of 25%, where at least 25% out of 100 test subjects should assess an object as real. For smaller samples sizes of for instance 16 and the probability value of 0.19, more than 25% of the test subjects must assess a virtual objects to be real. In Figure 5.1 the sum of the columns marked in red must not exceed 0.05. If the probability for 6 test subjects to identify a scene as virtual was added also, the total amount would exceed 0.05, and it would not be possible to significantly reject that test subjects could identify the virtual object. In other words, 7 out of 16 test subjects (i.e. 44%) must assess an object as real to reject the null hypothesis that people can identify a virtual object.

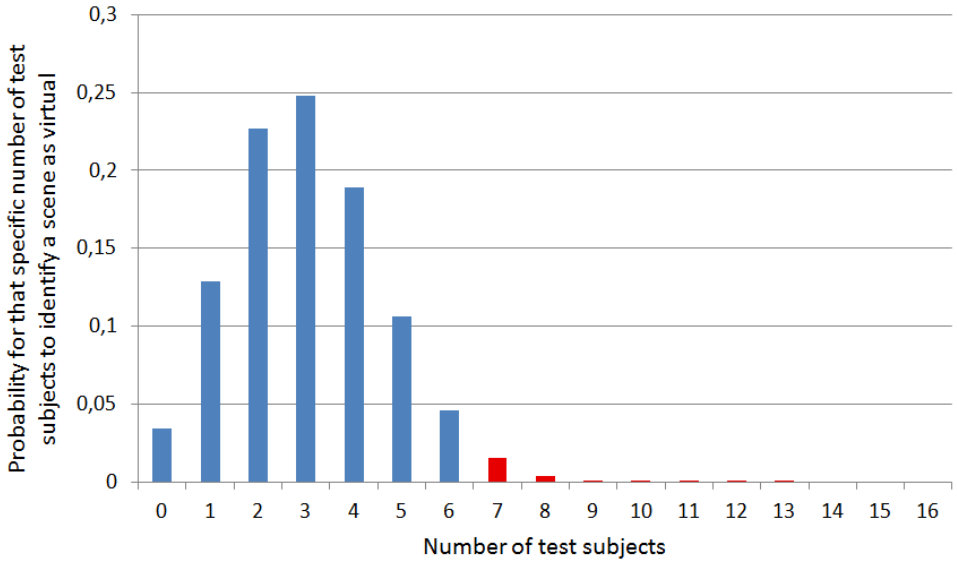


Figure 5.1: The probability for a specific number of test subjects to correctly identify a virtual object, given a p-value of 0.19. The sum of the red column must not exceed 0.05, given by the chosen confidence level.

Using this approach where the test subject have to assess whether the object is real or not (true/-false) eliminates the bias that could occur using a Likert scale where the understanding of the end points could be displaced. Moreover, the test subject will assess right after each display minimizing memory bias.

The demographic questionnaire used for all the experiments can be found in Appendix G on page 101.

5.1 Experiment 1 — Evaluation of Perceived Realism

The goal of this experiment is to see whether or not it is possible to integrate a 3D object into a scene such that the users believe it is a part of the scene. In other words, such that it is indistinguishable from the real object. The test subject is presented with either a virtual object or the real object through the monitor. The diffuse candle holder, specular elephant and the chrome candle holder will be displayed in the two lighting conditions, and so will the real objects.

When the test subject enters the laboratory he or she is asked to fill out a demographic questionnaire and read the procedure of the experiment (see Appendix G on page 101). The test subject is told to stay behind the marked line and not lean towards it. This makes sure that the test subjects have the same distance of approximately 80 cm from the monitor. Additionally, this distance is assessed to minimize the influence of jittering, without compromising the details in the scene. Furthermore, the test subject is told to move by walking with the rotating arm (hence, the monitor) as it moves from one side to another and back in around 10 seconds. This one "oscillation" makes sure that each test subject gets a consistent amount of time and the field-of-view is con-

stantly changing over a range of 60 degrees. This is consisting with the project definition of AR, namely that the position of the camera is under constant movement.

The test subject is now shown the scene, after which he or she is asked to judge whether or not it is a real object or a virtual object. When the test subject has assessed whether or not he or she believes the object displayed on the monitor is real or not, he or she is asked to wait outside. When entering again the test subject is presented with another light setup and/or another object — virtual or real — and have to assess. The sequence of the displayed scenes is randomised.

The evaluated scenes can be seen in Table 5.1.

5.1.1 Results

For the experiment 15 test subjects in the age of 22 to 28 years participated — 14 men and 1 woman. All had normal or corrected-to-normal vision and was familiar with 3D computer graphics and augmented reality (see Figure 5.2). The number of assessments that an object in a specific scene was real can be seen in Table 5.1. From the Equation 5.1 the critical value can be calculated for 15 test subjects to be 6.

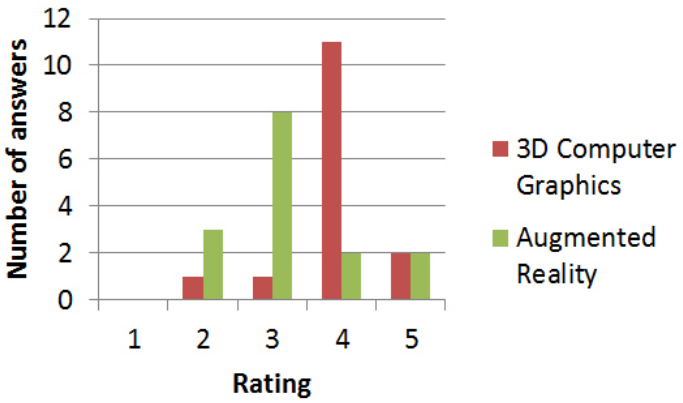


Figure 5.2: The users' experience with of 3D computer graphics and augmented reality. The rating 1 corresponds to "No experience" and 5 corresponds to "Very experienced".

In Table 5.1 the scenes that are significantly perceived as real are marked in bold. As can be seen only the specular object in ambient light is perceived as real. Furthermore, some of the real scenes were not perceived as real: the diffuse object in spot light and the specular object in ambient light.

Table 5.1: Number of assessments that an object in a scene is real. The probability for the test subjects to correctly identify the virtual object is shown in the parenthesis. Results in bold are perceived as real with $p < 0.05$. For instance, for the diffuse object the real object shown in ambient lighting is perceived as real, as 6 test subject have assessed it so. On the other hand, for the virtual representation shown under the same conditions only 3 test subject have assess it as real. As this does not reach the critical value of 6 no significant result is found.

	Diffuse		Specular		Reflective	
	Virtual	Real	Virtual	Real	Virtual	Real
Ambient light	3 (0.5635)	**6 (0.0490)**	**7 (0.0137)**	3 (0.5635)	1 (0.9576)	**8 (0.0030)**
Spot light	4 (0.3146)	2 (0.8085)	3 (0.5635)	**11 (0.000)**	0 (1.000)	**12 (0.000)**

5.1.2 Discussion and Improvements

One reason why only one scene was perceived as real was that the tracking was not stable enough. Test subjects were able to determine which scenes was virtual due to the jittering. Because the specular object was the flattest object this would not suffer as much from the jittering. Oppositely, the high reflective object would also constantly jitter resulting in a constant glittering caused by the reflection of the details in the marker of the cube map.

Because the jittering was such an influencing factor on the result, this should be limited to get reliable results. One solution is to lock the tracking (i.e. the virtual camera position) when the user is watching the scene, and therefore also the physical position of the rotating arm, which totally eliminates movement. However, the user should be able to see the scene from different angles, but if the tracking is locked the perspective will not be updated when the arm is rotated. Therefore, the user should not be able to see the scene until the arm is in a new position. While the perspective is moved the tracking should be enabled to track the marker and position the object correctly into the scene. Because there will be no jittering with this procedure, the users is allowed to get 20 cm closer to the monitor. This is assessed to be a natural viewing distance to the scene.

One reason why the real objects was not perceived as real scenes was that test subjects lacked some context. Namely, the light used to illuminate the scene, how the scene would be captured by the web-camera and a reference to the quality of a virtual versus real model. This lack of knowledge resulted in the fact that test subjects tended to assess the scene as virtual when in doubt. This can be seen by the fact that around 2/3 answers were on scenes being virtual and 13/15 started off by assessing the first scene as virtual. Given the ratio between real and virtual scenes it would be expected that the assessments would be fifty-fifty.

A way to change this is to mentally calibrate the test subjects. In other words, show them how an object looks captured by the web-camera in both lighting conditions. By displaying a real object and the virtual reproduction (not one of the objects used in the experiment) the user will be familiar with the scene and its lighting. Furthermore, the test subject get an idea of the camera

settings, quality and artefacts.

5.2 Experiment 2 — Evaluation of Parameters

With the problems from Experiment 1 addressed other tests can be conducted to determine in which cases some of the applied effects can be lowered to a minimum at which people still cannot distinguish a scene from reality — or in some cases whether the effect can be turned totally off. The parameters that are evaluated are the artefacts of the camera and the rendering, the lighting (including shadows) and its shading, as well as the quality of the geometry of the objects.

The purpose of the first experiment was to evaluate whether or not the objects, and which one, could pass as real given the highest quality achieved in the project. But since the real object was not even considered real the experiment is assessed to be somewhat unreliable and a continuation to the second experiment is done with some assumptions. In accordance with the experiment conducted in Section 5.1 the specular elephant is chosen for further study. This object is chosen as it is the only one which is significantly perceived as a real object, even though it only applies for ambient lighting. However, as it is of interest also to have the spot light condition, which produces the high frequency shadows and a higher illumination, it is assumed that the specular objects in general is the most appropriate based on the answers from the previous experiment — especially with the improvements made in relation to the procedure from the last experiment. The purpose of the first experiment was to evaluate whether or not the objects, and which one, could pass as real given the highest quality achieved in the project. But since the real object was not even considered real the test is assessed to be somewhat unreliable and a continuation to the second experiment is done with some assumptions. In accordance with the experiment conducted in Section 5.1 the specular elephant is chosen for further study. This object is chosen as it is the only one which is significantly perceived as a real object, even though it only applies for ambient lighting. However, as it is of interest also to have the spot light condition, which produces the high frequency shadows and a higher illumination, it is assumed that the specular objects in general is the most appropriate based on the answers from the previous experiment — especially with the improvements made in relation to the procedure from the last experiment.

The procedure is as follows: The test subjects enters the laboratory and is told to read about the procedure and to fill out a demographic questionnaire (see Appendix G on page 101). The test subject is told that he or she have to determine whether or not the object in the scene is real. Furthermore, the test subject is told that he or she will watch the scene from three pre-determined positions (with approximately 20 degree between them) for 4 seconds. The four seconds are chosen as a sufficient time to get an overview of the scene, limiting time to get into smaller details in the scene. Also, the test subjects are told to stay behind the marked line at the floor, which has a distance to the monitor of approximately 60 centimetres. Between the three pre-determined positions the screen will go black so the tracking can position the camera before it is locked again as the black overlay is turned off. This way the jittering related to the tracking

is eliminated, which allows the test subject to stand at a closer distance to the screen.

Before beginning the actual judgement of the scenes the test subject is shown another scene with a real object, and is told that is a real object. Then the test subject is shown a virtual replica with the best settings possible and told that it is a virtual object laid over the video-feed. Lastly, the test subject is shown another virtual replica – this time just with no parameters set to high quality or completely turned off. All three scenes are shown both in ambient lighting and with a spot light. By showing these six scenes, the test subjects know how the camera feed and the physical scene looks, as well as how a virtual object can look and what they can expect of context, objects and lighting, as well as the overall quality. The mental calibration is conducted on the diffuse candle holder.

Prior to conducting the experiment pilot testing showed that the generated noise had to be re-sampled, as it was not “strong” enough. This was apparently very important since the three test subjects participating in the pilot tests, all — after being shown the scenes a few times — began to assess the objects as virtual or real based solely on the amount of noise.

For all of the four tests in the experiment 16 test subjects in the age of 21 to 30 years participated — 1 woman and 15 men. All had normal or corrected-to-normal vision and most subjects were experienced with 3D computer graphics and augmented reality (see Figure 5.3). The critical value for 16 test subjects is 7. The sequence of the scenes was randomised.

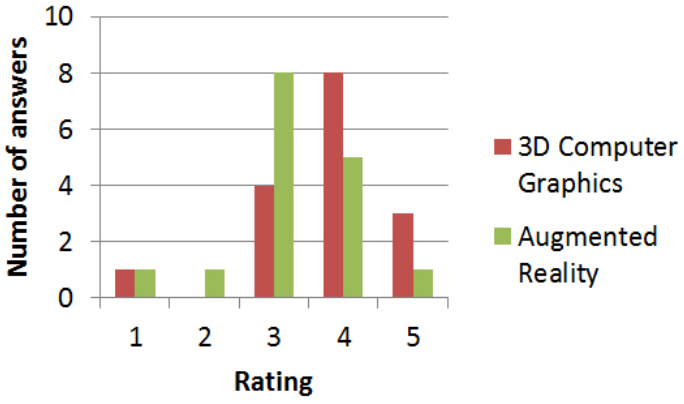
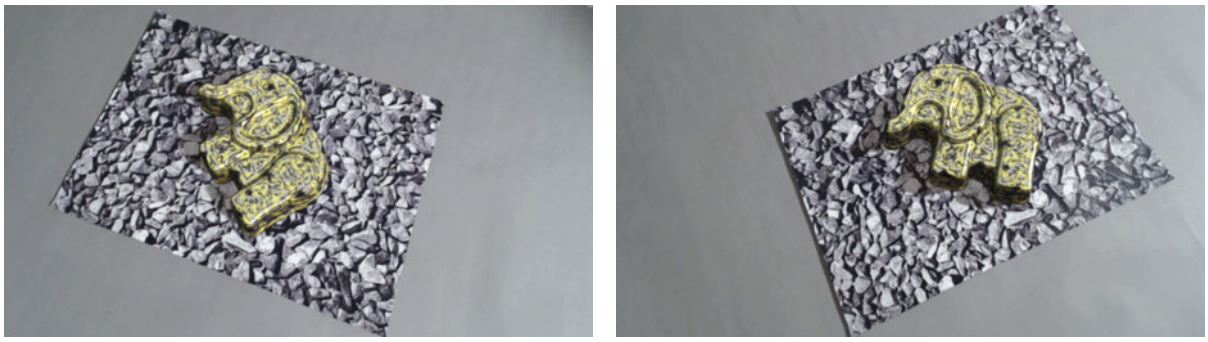


Figure 5.3: The users’ experience with of 3D computer graphics and augmented reality. The rating 1 corresponds to “No experience" and 5 corresponds to “Very experienced”.

Some of the scenes reoccur three times across the four tests, which means that these scenes are actually assessed three times for each test subject. This gives a total of 48 assessments of these same scenes, given 16 test subjects. This also means that an probability for three times as many samples can be calculated, which results in stronger power of the statistic. The result of the probability for these combined scenes is noted in the results for each of the test, however, they are not included in the judgement of the results as they do not have the same basis, namely the sample size.

Each test subject will see each of the scenes given in the tables once, for the four tests. All in all,

the test subjects will assess 34 scenes (an example of what the test subject will see can be seen in Figure 5.4). More examples of what the test subjects will see in the four tests can be found in Appendix H on page 105.



(a) Perspective from right (b) Perspective from left

Figure 5.4: An example of what the test subjects will see. The two extremes of the angle is shown.

5.2.1 Evaluation of Artefacts

The purpose of the first test is to evaluate the effect of the camera artefacts and the rendering artefacts. In other words, the test determines whether or not people notice if the 3D model is rendered without any of the two artefacts: noise and anti-aliasing.

All of the scenes seen in Table 5.2 are shown in an ambient lighting condition. The table contains the results of the test, where the scenes that are significantly perceived as real are marked in bold.

Table 5.2: Parameters and results for evaluating camera and rendering artefacts. Results in bold are perceived as real with $p < 0.05$. The real object is used as control.

	Noise	No noise	Control
Anti-aliasing	5 (0.1727)[a]	4 (0.3619)	**9 (0.0010)**[b]
No anti-aliasing	5 (0.1727)	5 (0.1727)	

[a] For a calculation of the probability for this scene across all the four tests (16 out of 48) the probability is 0.0132.

[b] For a calculation of the probability for this scene across all the four tests (27 out of 48) the probability is <0.0001.

As can be seen, there seem to be no noticeable difference between having noise and anti-aliasing enabled, as none of the virtual objects in the scenes are perceived as real. One reason could be that test subjects does not notice the small differences, but focus on getting an overview. If more time were given to observe the scene or the same scene was shown multiply times the results could have been different since some parameters may require more time to observe. Furthermore, as the scene with the best change of being perceived as real (i.e. the one with noise and anti-aliasing) is

considered virtual it cannot be concluded that the noise and anti-aliasing improves or decreases the perception of realism.

5.2.2 Evaluation of Shadows and Shading

The purpose of this test is to evaluate the number of lights needed to create a shading that looks similar to that of a real object. The lower limit to create an ambient shading is assumed to be 2 lights, while 16 lights should be enough to create a realistic shading [Madsen and Laursen, 2007]. For the scenes with the spot light setup an extra directional light is added as the spot (hence the number is actually n+1 for those scenes). Furthermore, two different methods for creating high frequency shadows are evaluated, namely a real-time shadow and a baked shadow. Lastly, it is determined whether or not an low frequency shadows is actually noticeable, therefore scenes without any shadows are also evaluated.

The results can be seen in Table 5.3, where the ones that are perceived as real are marked in bold.

Table 5.3: Results for the number of lights to create a perceptual correct shading and for different methods to create shadows in a spot light and an ambient light setup. Results in bold are perceived as real with $p < 0.05$. The real object is used as control.

SPOT LIGHT	Number of lights				
	2	4	8	16	Control
Baked high and low frequency shadows	**9 (0.0010)**	**10 (<0.0001)**	**13 (<0.0001)**	**11 (<0.0001)**	**14 (<0.0001)**[c]
Real-time high frequency shadows and baked low frequency shadows	**7 (0.0204)**	**10 (<0.0001)**	**7 (0.0204)**	**9 (0.0010)**[a]	
AMBIENT LIGHT	Number of lights				
	2	4	8	16	Control
Baked low frequency shadows	6 (0.0662)	**8 (0.0051)**	3 (0.6101)	4 (0.3619)[b]	**9 (0.0010)**[d]
No shadows	**7 (0.0204)**	3 (0.6101)	4 (0.3619)	4 (0.3619)	

[a] For a calculation of the probability for this scene across all the four tests (29 out of 48) the probability is <0.0001.

[b] For a calculation of the probability for this scene across all the four tests (16 out of 48) the probability is 0.0132.

[c] For a calculation of the probability for this scene across all the four tests (38 out of 48) the probability is <0.0001.

[d] For a calculation of the probability for this scene across all the four tests (27 out of 48) the probability is <0.0001.

As can be seen, all of the scenes illuminated by a spot light is perceived as real, even with

only two lights for shading. However, test subjects used more time to decide for more lights. No difference is to be found between the real-time high frequency shadow and the baked high frequency shadow. On the other hand, most of the objects shown in ambient lighting only was not perceived as real. Only the scene with baked low frequency shadows and 4 lights for shading, as well as the scene with no shadows and 2 lights for shading are considered real.

One reason why the scene with ambient lighting based on few lights are perceived as real might be that people are not used to see only ambient lighting, as there usually is some kind of noticeable high frequency shadow. Therefore, they might not believe that the scene is real. However, as the control is in fact perceived as real, the quality of the ambient scene is not good enough to be perceived as real — even for the best settings.

No considerable difference is found between the scene with baked low-frequency shadow and the scene without shadows, which might indicate that low-frequency shadows are not necessary — at least for this particular setup.

5.2.3 Evaluation of Specular Highlights

The test has focus on the presence or absence of specular highlights. It is determined whether or not people notice that a specular object does not have specular highlights. The test is evaluated with the spot light setup, and the real-time shadows are used in combination with the baked ambient shadows. The results can be seen in Table 5.4, where the scenes that are perceived as real are highlighted in bold.

Table 5.4: Results for a specular object with and without specular highlights. Results in bold are perceived as real with $p < 0.05$. The real object is used as control.

	Specular highlight	No specular highlight	Control
SPOT LIGHT	**9 (0.0010)**[a]	3 (0.6101)	**9(0.0010)**[b]

[a] For a calculation of the probability for this scene across all the four tests (29 out of 48) the probability is <0.0001.

[b] For a calculation of the probability for this scene across all the four tests (38 out of 48) the probability is <0.0001.

From the results it can be seen that specular highlights is indeed necessary to create a photo realistic object with a specular material. This also complements previous research [Elhelw et al., 2008].

5.2.4 Evaluation of Model Quality

Lastly, the model quality is evaluated, to see whether or not it is possible to decrease the number of polygons for an object to a certain degree, such that it still looks realistic. Three different

object geometries are used, one with very few polygons (461), one with few polygons (1028) and one with many polygons (52387). These three geometries are shown both in ambient light (with baked ambient shadows) and in spot light (with real-time shadows and baked ambient shadows). The results can be seen in Table 5.5, where the scenes that are perceived as real are highlighted in bold font.

Table 5.5: Results for a evaluation of model quality, given in number of polygons. Results in bold are perceived as real with $p < 0.05$. The real object is used as control.

	Very low-poly	Low-poly	High-poly	Control
SPOT LIGHT	5 (0.1727)	**8 (0.0051)**	**11 (<0.0001)**	**15 (<0.0001)**[a]
AMBIENT LIGHT	2 (0.8368)	2 (0.8368)	**7 (0.0204)**	**9 (0.0010)**[b]

[a] For a calculation of the probability for this scene across all the four tests (38 out of 48) the probability is <0.0001.

[b] For a calculation of the probability for this scene across all the four tests (27 out of 48) the probability is <0.0001.

From the results it can be seen that the models to some extend need to be in high quality. Furthermore, it can be seen that the quality of the geometry depends on the lighting, maybe because the rough mesh is more visible in some kind of lighting or because the two light version vary in quality where it is observed that the ambient indeed is a poorer representation of the real object. All of these results of course depends on the object and its shape.

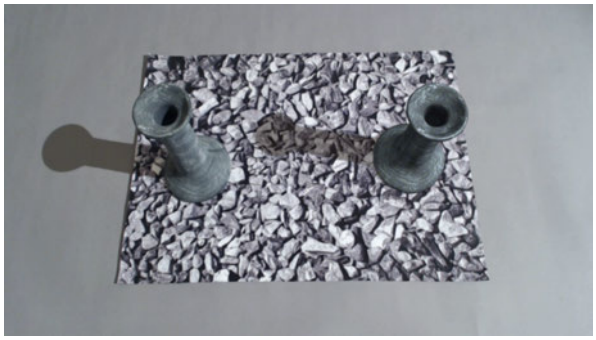
5.2.5 Summary

Four parameter tests are conducted, where no conclusion can be given on the effect of noise and anti-aliasing. However, for the shadows it seems that it is sufficient to cast high frequency shadows in real-time, without no need for pre-rendering. Pre-rendered low frequency shadows were very subtle in this setup. Purely based on the results it could be deduced that the low frequency shadows could be left out. However, this is not what would be expected, and could related to the aspect that displaying a scene only once with the lack of an effect is not enough to fully evaluate its importance.

Few lights seem to be sufficient for the shading, which is explainable for the spot light scenes because of the dominating illumination from the spot. However, it is a peculiar result for the scenes with ambient lighting given only two or four lights. It can also be concluded that specular highlights are important for photo realistic rendering. Lastly, the quality of the mesh of the object need to be of a sufficiently high quality not to be categorized as virtual.

5.3 Experiment 3 — Side by Side Comparison

As some of the scenes in the previous experiment are perceived as real it might be possible to increase the strength of the validity of the results. Instead of limiting the user to assess whether or not the object in the scene is real without any context and for a limited amount of time and angles, the virtual object will be compared against the real object in the same scene, for as long as the test subject wants and from every angle the test subject wants. This is considered the ultimate conditions for the test subject to pinpoint the real object, given the restrictions of the tracking and the setup. The possible scenes that a test subject can be shown can be seen in Figure 5.5).



(a) Side by side comparison of diffuse object — real to the left

(b) Side by side comparison of diffuse object — real to the right

(c) Side by side comparison of specular object — real to the left

(d) Side by side comparison of specular object — real to the right

Figure 5.5: Side by side comparisons of the diffuse candle holder and the specular toy elephant.

For this experiment the best scene with the specular object is chosen from the previous experiment — that is the scene with spot light, eight lights and baked high frequency shadows. Furthermore, as the first experiment was biased due to jittering and missing mental calibration also the scene with the diffuse candle holder in spot light with real-time shadows are evaluated. The scene with the reflective candle holder is considered too far from realistic by pilot testing to be included in the experiment.

When the test subject enters the laboratory he or she is asked to read a note about the procedure and to fill out a demographic questionnaire. The test subject is told that he or she shall assess

which of the two objects in the scene is real and that he or she can use as much time as needed and from as many angles as needed. As in Experiment 2 a mental calibration was performed, only this time with the reflective candle holder.

The virtual object is significantly seen as real if the test subjects guesses wrong enough times, hence approaching random chance. The sequence of scenes and position of the objects was randomised.

5.3.1 Results

Fifteen test subjects between the age of 21 and 27 participated in the experiment, one woman and 14 men. All had normal or corrected-to-normal vision and most people were familiar with 3D computer graphics and augmented reality (see Figure 5.6). The critical value for 15 test subjects is 6 and the results of the experiment can be seen in Table 5.6.

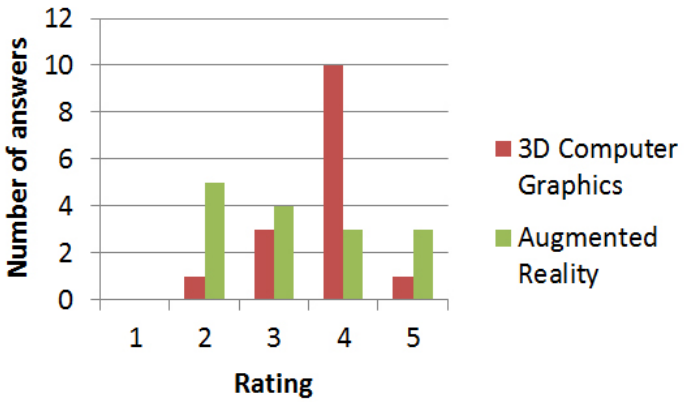


Figure 5.6: The users' experience with of 3D computer graphics and augmented reality. The rating 1 corresponds to "No experience" and 5 corresponds to "Very experienced".

Table 5.6: Results for the side by side comparison of the two objects. The number of incorrect answers are noted as well as the probability. Results in bold are perceived as real with $p < 0.05$.

	Diffuse	Specular
SPOT LIGHT	**6 (0.04903)**	0 (1.000)

As can be seen, the diffuse object is significantly perceived as real, though just barely. This does not apply for the specular object, however, as none of the test subjects chose the virtual object to be real. One of many reasons for this could be that the manually adjusted highlights were to different from the highlights seen on the real object. On comment is that the diffuse object was only perceived as real in one of the two positions; the one where the shadow did not fall outside the marker (see the scenes in Figure 5.5). This might indicate that the shadow is an important factor when evaluating the realism. This was also mentioned by the majority of the test subjects.

Chapter 6

Discussion

A setup has been created and different implementations have been written to make an evaluation of a virtual augmented object in a video-feed. However, some aspects could have been approached differently. The methods and implementations will be reviewed, as well as an overview of the whole pipeline will be given.

The physical setup has been created to be able to evaluate the scene. However, the freedom of movement have been limited to only one rotational axis. Furthermore, the rotation angle in the experiments have been within approximately 60 degrees. A setup with more degrees of freedom would require a more advanced setup, making sure that the marker is always in the field of view.

Given more freedom of degree would require a solution for when the marker is out of sight. One solution could be to have a border covering a part of the video feed, so that the tracking can take place before the virtual scene gets visible in the field of view. Another option would be to make the marker big enough and only use the centre of it to render a virtual scene, so the tracking can take place before the virtual object gets visible.

In the last two experiments the position was locked, which removed the need of a moveable arm. This was due to unstable tracking, as it was still not stable enough and resulted in jittering. Locking the position contradicts with the definition of AR stated for the project. Nevertheless, the video feed was maintained in real-time and the scene could still be seen from multiple angles. All in all, the setup performed acceptable given the goal which concentrated more on the importance of the parameter evaluations.

The web-camera should be able to record in high definition, but this limits the frame-rate. To get even higher resolutions in real-time a fire-wire or USB3 camera could be used. This could increase the stability of the tracking at the expense of the processing time. Another problem with the web-camera is the lens. Because it is a build-in lens the focus range is limited, making the scene a bit unclear.

Also the marker used in the project could have been different. The picture of stones used as the marker could have an influence on the result, as it is the only context in the scene besides the objects that have to be evaluated. The stones could for example be associated with stones seen at a beach shoreline or in a garden, where the virtual high frequency shadows does not present themselves the way they would in reality. Thought the content is only on paper a subtle influence of this context could influence the assessment of shadows on the marker. Another image with a more flat motive could be used as marker. Additionally, a 3D marker could be used to get more stable tracking. Though, a consequence of this would be to place the object without occluding

the marker.

The Unity game engine is chosen for the project as it provides an easy setup of the experimental scenes. However, the framework is locked to certain solutions, which might be more open (and thereby have the possibility to be more correct) in other AR solutions. On problem in Unity was for instance to access the camera parameters, which probably could be accessed in ARToolKit where the field-of-view could have been investigated.

The objects chosen for the experiments was chosen because they were able to be scanned or easily modelled. Furthermore, they had to have the necessary features given by the chosen parameters. But even though the objects could be scanned (after decorating the specular elephant) and fitted into the scene the objects were not familiar to the test subjects. Therefore, it should be recommended to use everyday objects, that people are familiar with.

For instance one of the subtle artefact is the result of the unknown internal processes and settings of the web-camera, which makes it hard to simulate the artefacts and mechanisms in the virtual scene. The lack of knowledge affects the calculation of the field-of-view made internally in Vuforia, which results in a wrong position and perspective of the virtual camera.

Also, as long as the tracking is not completely stable it should be recommended to use objects that are not very high, as the model will jitter more the further it gets from the marker.

When wanting high-poly objects a 3D scanner would be preferable over a software implementation as 123D Catch, as the stitching process is not yet good enough, as well as the tracking of features on objects that are not diffuse and with a abstract texture. With a more complex scanner it would be possible to evaluate more complex models, yielding more realistic scenes in terms of content. Moreover, while photographing the object the raw-format should have been used instead of jpeg format, so that a wider range of intensities could by included and no form of compression would by performed. This could have affected the efficiency of the implemented colour-correction methods.

The screen tilt angle could also have an influence on the perception of the virtual object. Since people were of different height they also look at the screen from different angles. Viewing the screen from different angles change the representation of colours and intensity, where it is observed that even small differences in appearance between the virtual and the real object is intensified. It is assessed that the change is too subtle to affect the result significantly, but future setups should account for this problem.

A lot of methods for creating details in material also exist, such as normal and bump maps, or gloss maps for specular objects. Only the gloss map was addressed in this project, but it was not compatible with the cut-out screen space shader, as the alpha value in the gloss map would become transparent and not just limit the highlights. Moreover, the glossiness of the specular object is set manually to determine the highlights. Since there are no detailed bumps on the surface of the objects the lack of a bump map is assumed to have limited effect. It is hard to assess the exact directions of the light reflections and thereby replicate a normal map. Moreover, it is assumed that the generated normals from the scanning would yield the right highlights

corresponding to the real object.

The colour matching proved to be more difficult than expected. The colour-matching implementations did not automatically adjust the colour of the textures, because the albedo was unknown for the textures. Some methods to subtract the light from a texture could be used. In the best case the objects should be captured (with 123D Catch) in completely ambient white light, such that only the intensity was unknown (and not the colour). Another problem was that some objects consisted of different colours and materials, which made the colour matching more difficult.

In general the transfer of lights and their colour should be attended with care, as two incompatible interpretations of the light and colour will result in a manual adjustment. If more steps are involved, the light intensity and colour can end up being very different from what was given in the beginning.

Some artefacts were also simulated. The noise was sampled on a grey paper and the sampled noise was applied to a noise texture. This is of course not correct, as the deviation of the noise depends on the intensity of the pixels in the video feed. For instance the noise is more visible in the mid-tone areas. A function depending of the pixel intensity in the video feed could have been used to generate the noise for that specific intensity. This had an effect on the experiment as the noise was more noticeable on the diffuse object than the specular object. Another aspect that was not considered was the relation between neighbouring pixels. Nevertheless, the relation between the channels was considered, as the HVS is sensitive to intensities. The uncorrelated RGB channels could yield too much perceptual noise.

Due to the cut-out shader used to render the noise on the objects only, the anti-aliasing was applied on the whole screen space, which of course would have been preferred to be only on the virtual objects. The reason for this is that other screen space methods in combination with the custom screen space shader affected the image in an very undesirable way (such as arbitrary rotations, ignored camera rendering order and accumulation of the virtual objects across multiply frames). This problem was never solved since many cameras in the framework made it hard to work with. The consequence of this was a small blur around high contrast edges, but it did not lower the tracking performance, since the AA shader used was executed after the tracking was performed.

The generated HDR environmental map was used to generate the lights. It would be preferable if the whole process of light generation was automated. Some implementation have been created to calculate the light in real-time, however this was out of the scope of this project. Because the SLR camera was not able to capture the range of intensities in the scene the spot was masked out — the spot should be able to be represented in the HDR map. As a consequence, a spot was inserted manually and the intensity was adjusted.

The cube map generated from the environment map was of course static and did therefore not take into account when the arm rotated inside the setup as the monitor was moved. However, as the reflective object was not considered real due to other reasons this did not have any influence. Additionally, the size of the rotating arm was small enough not to influence the lighting.

Two different methods of creating shadows was evaluated: real-time and pre-rendered. For the real-time shadows a directional light was used, as this was the only option in Unity. A more correct light could have been used (for instance a point light or area light) to create penumbra. Furthermore a more complex shader could be used to render shadows from more than one light source, on the expense of the processing time. The real-time shadow is cast on a semi-transparent (and cut-out) plane and the intensity and colour of the shadow is set manually. This should of course be determined by the light sources.

The pre-rendered light map is modified in Adobe Photoshop because the shadow is not dark enough. One reason might be that the intensity of the manually inserted spot did not match the true ratio between the physical lights. Also the colour of the pre-rendered shadow was manually adjusted. Common to both shadow methods is that they are added to the scene using alpha blending, not multiplied as it should be [Jacobs et al., 2005; Porter and Duff, 1984].

A general note is that the quality of the web-camera determines the compromise between having to generate multiple artefacts, given a low quality camera versus having to match a well represented object, given a high quality camera. If a low-quality web-camera is used, more artefacts have to be simulated but the level of realism displayed on the screen is of poor quality. On the other hand, if a "perfect" camera is used with a high-performance computer, the artefacts would we more subtle and the realism display is of high quality. Given such a good quality of the video feed the requirement of the methods used would be proportionally high and harder to implement. In contrast, a low quality feed is easier to address though the used methods. Additional, given a high quality web-camera the low amount of artefacts produced could also mean that less artefact have to be reproduced. This could eventually yield the perfect pipeline and many automated/real-time computations.

Different procedures have been conducted across the various experiments. The first experiment was mentioned to be biased since even the real object did not pass. This emphasises the importance of mental calibration so that the test subject have a baseline to relate to. Moreover the tracking was not good enough even though various resolutions, frame-rates, settings and camera positions where tried. This indicates that the tracking solution nowadays or the hardware currently running the software is not good enough when wanting to reach the level of realism. It should be noted, that the tracking consumes approximately 80 % of the CPU in the given setup. With more processing power and/or better tracking solutions the first experiment could have been more reliable. Because of the missing mental calibration, the unstable tracking and the result not agreeing with what could be expected and intuitively assessed, the experiment is seen as unreliable and used mostly used for guidance on which object to chose. In general, an important lesson was learned which helped designing the subsequent experiments.

Regarding mental calibration, no baseline of the context for the user is needed in everyday AR applications, as the user would be located within the environment. Mental calibration would not be needed since the user would do a direct comparison between the real world and the content on the display, and would more easily address objects not fitting a given context. Moreover, the degradation of the video feed would be represented on the display. That way the alienation of

what is seen on the display would be limited and the degradation level would be known. Thus, the surrounding could address the mental calibration problem itself in everyday use of AR.

Even though the test procedure improved from Experiment 1 the evaluation of the results from Experiment 2 showed some peculiarities. The peculiarities mainly regard the fact that the ambient object was perceived as real only a few times, while the spot object was perceived as real all the times — the only change was the light from the spot. One reason might be that the high frequency shadow from the spot light helps to ground the object so that it does not seem to float above the marker, whereas the low frequency shadows are more subtle.

The recreations of the virtual objects are a bit off in shading and colour-temperature. This is especially true for the ambient version, so another reason could simply be due to poorly recreation. Disregarding the peculiarities and the failures of the ambient object the experiment also gave good results. Looking into the results as well as the statements and reactions of the test subjects, it is clear that shadows (mostly high-frequency), shape smoothness and highlight correctness are the most important factors to address when wanting to create photo-realistic augmented objects.

Since pilot test subjects could distinguish all the objects based solely on noise, it is considered important. The interesting aspect is that the first few objects were not distinguished consequently. This indicates that the parameter “artefact” first reveals itself after a few trials, which again could indicate that in order to notice the effect of a parameter, it have to be shown several times and not only once within an experiment. The memory-bias, time of exposure or other HVS aspect must have played a role.

Experiment 3 did confirm that a virtual object can actually be seen as real even when the corresponding real object is right beside it. This is promising since it indicates that the parameters chosen (from the object that was perceived as real) and their combination is not degrading photorealism. Ergo, the parameters could be researched further with a level of confidence that they actually do promote photo-realism.

Another interesting aspect is that for an object to be perceived as real the shadow of the virtual object may not exceed the marker. As the shadow falls outside on the grey underlay the object is never considered to be real. It is not certain why this is so important, it could also be that when the shadow of the real object fall onto the grey underlay it is more prone not to be perceived as real, than when it is within the marker. One theory is that the imperfections of the virtual shadow is unnoticeable when falling onto the abstract marker, while noticeable when falling onto the grey uniform underlay. This emphasises the importance of the quality of the high frequency shadows and their underlay.

Another peculiarity is that in the biased Experiment 1 the specular object shown in ambient lighting is the only one perceived as real. Afterwards in Experiment 2, the specular object shown in spot light is perceived as real, while the best scene from Experiment 1 was not. Lastly, the diffuse object is the only object to be perceived as real when compared side by side. This could indicate that Experiment 1 was more misleading that thought at first when choosing the specular elephant. A consequence of this could be that a more suiting object, namely the diffuse candle holder, should have been included into Experiment 2 to give more clear thresholds for when the

parameters start to degrade photorealism.

An optimized setup would have a camera that both can be used to photograph the environment map and the objects.

Regarding the colour correction and double lighting, this would help minimize the difference between the virtual and the real object. Given full freedom of movement would be complicated since the marker should still be shielded off. Here a HMD could be used, but this would defeat the purpose of using a common game engine designed to work with mobile devises. If future wearable computer-glasses can include a GPU, it could be interesting to design a setup given this technology. It would also be beneficial if the marker could be replace with a tracking of contrast features given in the table.

The future product — which presumably would be the ultimate implementation — of realistic augmentation is considered as follows. Considering the advances within AR with wearable glasses [Google Inc., 2013] and even contact lenses with displayable graphics [Lingley et al., 2011] the scene would always be shown (or "rendered") perfectly, since the "screen" would to transparent. This would not allow the comparison of the real environment and a degraded video feed since no video feed would be given. Moreover, the glasses/contact lenses (the "screen") are more integrated and close to the person, unlike a mobile device with a screen that easily can be moved aside. This would force the augmentation to be at a very high level of realism which essentially is what this project is taking a step towards. Until such products are developed and accessible, an experiment with the closest technology given would be with a HMD. It could be interesting to see whether or not the test objects chosen for this project (which passed) would still be perceived as real without any mental calibration given. If not, the objects should be replaced with everyday objects since the objects chosen would in that case not fit the context.

6.1 Conclusion

During this project the three 3D objects have been integrated in a real-time AR application. This was done using Unity with Vuforia's AR solution. One of the objects — the diffuse candle holder — was indistinguishable from a real object in the video-feed. The object integration part requires a solid transfer of geometry and light factors. Here Autodesk 123D Catch can be sufficient but is not regarded as a solid solution. Regarding the light many steps have to be overcome and it is important to find a common unit for illuminations across the whole pipeline or else problems with both colour-correction, temperature, shading, intensity, low and high frequency shadows will occur. That being said, is it very difficult to design a pipeline which will not include manual adjustment of values, where visual perception is the only guideline, even in a known static environment where all parameters in theory can be measured.

Several parameters have been included in the experiments where these can be included to enhance photo-realism:

- Noise

- Highlight on specular objects
- High polygon count of the objects

Moreover, the appearance of shadows and a good tracking of the marker is important. It is assessed that the change of intensity is something the HVS is sensitive to, which is why the noise and the changes in silhouettes both on the shadows and the geometry are addressed by so many test subjects.

A general problem was that many small subtle (and presumably unnoticeable) artefacts and inconsistencies are present. These could together create a uncanny or subconscious influence which could tip the balance of which object to asses as real.

Overall the project is answering its questions and positive results were obtained from experiment 2 and 3. Though, it was difficult to implement it is shown that it is possible to create a virtual object indistinguishable from a real one.

6.2 Future Work

The project does present interesting areas, where future work can be done. A pipeline where a common illumination unit is given and maintained throughout the pipeline is something to be addressed, since it would essentially eliminate the majority of the manual adjustments presented in the project.

Every parameters should have been shown more than once since a form of adaption to the scene with a given parameter combination is present. There are also other parameters that could have been evaluated, namely movement and animations. Given animations a main parameter would be the importance of motion blur. Also the context and attention to the object would presumably be different when it moves. Another interesting parameters could be the importance of context in a scene, where different scenarios could be evaluated.

Looking further into the future, improved and smaller hardware could present the opportunity of integrating 3D AR into glasses and contact lenses. As the world is looked at directly through the glasses full photorealism is required for an object to been fully integrated into the scene. With this and other projects taking the first step towards such a product, it is hard to say which parameters will be essential. But if the advances in rendering technique and hardware progresses as until now, this should be possible in the future.

It could also have been interesting to evaluate the passed parameters in combination with each other to see if some combinations are more profitable then others. Moreover, the trade-off between the photo-realism and the performance requirement could have be evaluated such that a guideline for which parameters or combinations thereof would be the most profitable. This could help choosing the right parameters for, for instance mobile devices with limited processing power, where realism is strived for.

It could therefore be interesting to compile the project to a mobile devise to see if the processing of the useful parameter are too requiring and to see if humans would perceive the object differently given another type of interaction, freedom of movement and screen size.

References

Apiolaza, L. A. (2011). Simulating data following a given covariance structure. http://www.quantumforest.com/2011/10/simulating-data-following-a-given-covariance-structure/. Last seen: 13-03-2013.

Autodesk (2013). Realistic lighting for realistic games. http://gameware.autodesk.com/beast. Last seen: 24-05-2013.

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. (2001). Recent advances in augmented reality. IEEE Comput. Graph. Appl., 21(6):34 – 47.

Azuma, R. T. (1997). A survey of augmented reality. Presence: Teleoperators and Virtual Environments 6, 4:355–385.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. Psychological Review, 94:115 – 147.

Biederman, I. (2000). Recognizing depth-rotated objects: a review of recent research and theory. Spatial Vision, 13(2–3):241 – 253.

Blinn, J. F. (1978). Simulation of wrinkled surfaces. SIGGRAPH Comput. Graph., 12(3):286 – 292.

Bojesen, R. H. (2013). Statistical methodology for sensory discrimination tests and its implementation in sensR.

Borg, M., Johansen, S., Thomsen, D., and Kraus, M. (2012). Practical implementation of a graphics turing test. In Advances in Visual Computing, volume 7432 of Lecture Notes in Computer Science, pages 305 – 313. Springer Berlin Heidelberg.

Boubekeur, T. and Alexa, M. (2008). Phong tessellation. In ACM SIGGRAPH Asia 2008 papers, SIGGRAPH Asia '08, pages 141:1–141:5. ACM.

Bovik, A. C. (2005). Handbook of Image and Video Processing. Elsevier, 2nd edition.

Brooker, D. (2008). Essential CG Lighting Techniques With 3ds Max. Focal Press, 3rd edition.

Catmull, E. E. (1974). A subdivision algorithm for computer display of curved surfaces. PhD thesis, University of Utah.

Chiu, K. and Shirley, P. (1994). Rendering, complexity, and perception. In In Proceedings of 5th Eurographics Rendering Workshop, pages 21 – 33. SpringerWein press.

Cohen, J. and Debevec, P. (2013). Lightgen plugin. http://gl.ict.usc.edu/HDRShop/lightgen/. Last seen: 26-05-2013.

Daly, S. (1993). The visible differences predictor: an algorithm for the assessment of image fidelity. In Watson, A. B., editor, Digital images and human vision, pages 179 – 206. MIT Press.

Debevec, P. (2005). A median cut algorithm for light probe sampling. In ACM SIGGRAPH 2005 Posters, SIGGRAPH '05. ACM.

EasyRGB (2013). Color conversion math and formulas. http://www.easyrgb.com/index.php?X=MATH. Last seen: 08-04-2013.

Elhelw, M., Nicholaou, M., Chung, A., Yang, G., and Atkins, M. S. (2008). A gaze-based study for investigating the perception of visual realism in simulated scenes. ACM Transactions on Applied Perception, 5(1):3:1 – 3:20.

Eriksen, S. D., Hatting, J. T., Laursen, J. C. M., and Paprocki, M. (2012). Designing and implementation of tit-for-tat as winning strategy and ambient occlusion rendering in a computer game. http://paprocki.dk/Files/8sem.pdf. Last seen: 26-05-2013.

Fischer, J., Bartz, D., and Straßer, W. (2006). Enhanced visual realism by incorporating camera image effects. In ISMAR '06, Proceedings of the 5th IEEE/ACM International Symposium on Mixed and Augmented Reality, pages 205 – 208.

Google Inc. (2013). Welcome to a world through glass. http://www.google.com/glass/start/what-it-does/. Last seen: 28-05-2013.

Hasic, J., Chalmers, A., and Sikudova, E. (2010). Perceptually guided high-fidelity rendering exploiting movement bias in visual attention. ACM Trans. Appl. Percept., 8(1):6:1 – 6:19.

Henning, G. B. (1988). Spatial-frequency tuning as a function of temporal frequency and stimulus motion. Optical Society of America, 5(8):1362 – 1373.

Jacobs, K., Nahmias, J.-D., Angus, C., Reche, A., Loscos, C., and Steed, A. (2005). Automatic generation of consistent shadows for augmented reality. In Proceedings of Graphics Interface 2005, GI '05, pages 113 – 120. Canadian Human-Computer Communications Society.

Klein, G. and Murray, D. W. (2008). Compositing for small cameras. In ISMAR '08, Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, pages 57 – 60.

Klein, G. and Murray, D. W. (2010). Simulating low-cost cameras for augmented reality compositing. IEEE Transactions on Visualization and Computer Graphics, 16(3):369 – 380.

Lam, E. Y. and Fung, G. S. K. (2009). Automatic white balancing in digital photography. http://www.eee.hku.hk/optima/pub/misc/0800_SSI.pdf. Last seen: 28-05-2013.

Leino, O., Wirman, H., and Fernandez, A. (2008). Extending Experiences: Structure, Analysis and Design of Computer Game Player Experience. Lapland University Press.

Lingley, A. R., Ali, M., Liao, Y., Mirjalili, R., Klonner, M., Sopanen, M., Suihkonen, S., Shen, T., Otis, B. P., Lipsanen, H., and Parviz, B. A. (2011). A single-pixel wireless contact lens display. Journal of Micromechanics and Microengineering.

Liu, A., Tendick, F., Cleary, K., and Kaufmann, C. (2003). A survey of surgical simulation: applications, technology, and education. Presence: Teleoper. Virtual Environ., 12(6):599 – 614.

Lottes, T. (2009). Fxaa. Technical report, nVidia.

Lubin, J. (1995). A visual discrimination model for imaging system design and evaluation. In Peli, E., editor, Vision Models for Target Detection and Recognition, pages 245 – 283. World Scientific.

Madsen, C. B. and Laursen, R. E. (2007). Performance comparison of techniques for approximating image-based lighting by directional light sources. In Image Analysis, volume 4522 of Lecture Notes in Computer Science, pages 888 – 897. Springer Berlin Heidelberg.

McKee, S. P., Klein, S. A., and Teller, D. Y. (1985). Statistical properties of forced-choice psychometric functions: Implications of probit analysis. Perception & Psychophysics, 37(4):786–298.

Milgram, P., Takemura, H., Utsumi, A., and Kishiro, F. (1995). Augmented reality: a class of displays on the reality-virtuality continuum. Proceedings of Telemanipulator and Telepresence Technologies, 2351:282 – 292.

Moeslund, T. B. (2009). Image and Video Processing. Aalborg University, 2nd edition.

Nakamura, T. (2012). I've seen the future of virtual reality, and it is terrifying. http://kotaku.com/5945181/ive-seen-the-future-of-virtual-reality-and-it-is-terrifying. Last seen: 16-05-2013.

Noisecrime (2010). shadow on the plane, everything else to be transparent? how to? http://forum.unity3d.com/threads/72400-shadow-on-the-plane-everything-else-to-be-transparent-how-to. Last seen: 28-05-2013.

Porter, T. and Duff, T. (1984). Compositing digital images. Association for Computing Machinery, 18:253 – 259.

Qualcomm Austria Research Center GmbH (2013). Trackable base class. https://developer.vuforia.com/resources/dev-guide/trackable-base-class. Last seen: 28-05-2013.

Quantum Scientific Imaging (2008). Understanding ccd read noise. http://www.qsimaging.com/ccd_noise.html. Last seen: 13-03-2013.

Rademacher, P., Lengyel, J., Cutrell, E., and Whitted, T. (2001). Measuring the perception of visual realism in images. In Proceedings of the 12th Eurographics Workshop on Rendering Techniques, pages 235–248. Springer-Verlag.

Reinhard, E., Adhikhmin, M., Gooch, B., and Shirley, P. (2002). Color transfer between images. Computer Graphics and Applications, IEEE, 21(5):34 – 41.

Shah, M., Arshad, H., and Sulaiman, R. (2012). Occlusion in augmented reality. In Information Science and Digital Content Technology (ICIDT), 2012 8th International Conference on, volume 2, pages 372 – 378.

Sigernes, F., Dyrland, M., Peters, N., Lorentzen, D. A., Svenøe, T., Heia, K., Chernouss, S., Deehr, C. S., and Kosch, M. (2009). The absolute sensitivity of digital colour cameras. Optics Express, 17(22).

USC Institute for Creative Technologies (2013). What is hdr shop? http://www.hdrshop.com/. Last seen: 26-05-2013.

van Dam, A. (2009). Realism in computer graphics. http://orca.st.usm.edu/~jchen/courses/graphics/lectures/Photorealism.pdf. Last seen: 24-05-2013.

van den Berg, T. (2012). Generating correlated random numbers. http://www.sitmo.com/article/generating-correlated-random-numbers/. Last seen: 13-03-2013.

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2010). Real-time detection and tracking for augmented reality on mobile phones. Visualization and Computer Graphics, IEEE Transactions on, 16(3):355 –3 68.

Watt, A. and Watt, M. (1992). Advanced Animation and Rendering Techniques. Workingham:Addison Wesley.

Yee, H., Pattanaik, S., and Greenberg, D. P. (2001). Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. ACM Trans. Graph., 20(1):39 – 65.

Appendix A — Parameters

Some of the considered parameters for evaluation. The ones that are stroked out was not used in the project.

Rendering:

- Shadows
  - Quality
  - ~~Direction~~
  - Real-time vs. pre-rendered
  - Ambient occlusion
  - Number of lights needed
- Polygons
  - Count
- Lighting
  - ~~Advanced~~ vs. simple shaders
  - Colour
  - Number of lights
- Texture
  - ~~Resolution~~
  - Complexity
    - ...of pattern
    - ...of colour
    - ...of depth
  - Colour
  - ~~Normal/bump map~~
    - ~~...resolution~~

~~...strength~~

- Material
  - Specularness
  - Reflectiveness
  - ~~Level of physical correctness~~
- Environment map
  - ~~Dynamic~~ vs. static
  - ~~Mirrored~~
- Colour bleeding

Scene:

- ~~Animation~~
- Geometry
- ~~High~~ vs. low complexity
- ~~Much~~ vs. little context
- Materials
- Lighting
- ~~Occlusion~~

Camera:

- Noise
- Blur from defocus
- Motion blur
- ~~Distortion~~
- ~~Colour-space~~
- White balance

Appendix B — Cameras in Vuforia

The class *Trackable* holds a reference dataset of points given by the custom predefined marker. With those points a corresponding arrangement of points in the camera feed can be found. Often the marker has to be in a front of the camera to being with. After a match of points is found a *TrackableResult* instance is created. Now the algorithm searches for points in relation to their last known position of a given match and a matrix is created describing the marker's orientation and position in 3D in relation to the camera [Qualcomm Austria Research Center GmbH, 2013]. A high markers quality is essential for stable tracking and good performance. As the custom markers are created on the Vuforia's website, a rating is given from zero stars to five stars, five being the best. The rating relates to three main factors:

- Amount of feature
- Feature distribution
- Local contrast

Besides that the features should not create a repetitive pattern. An example of a feature is described as a corner on a square, so a square gives four features, while a for example a circle would not give any features. A marker should have as many features as possible, the features should be distributed evenly throughout the whole image and the contrast between the details (features) in the image should be high. The marker used in this project is presented Figure 6.1.

As seen in the Figure 6.1 and 6.2 above the marker used have a lot of feature, evenly distributed and with a lot of local contrast between the features. Moreover Vuforia suggest inspecting the overall contrast by looking at the histogram of the image. The histogram for the marker used in this project looks as follows.

Figure 6.1: The marker used in the project with alot of feature, evenly distributed and with a lot of local contrast between the feature.

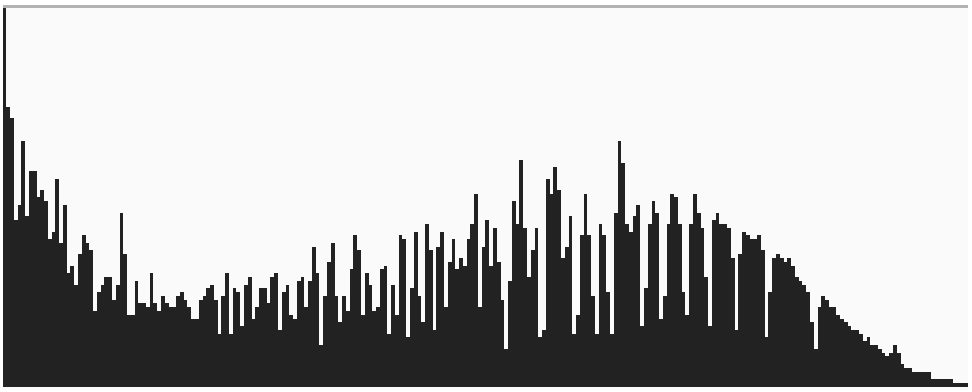


Figure 6.3: Pixel value from 0 to 255 from the grey scale image used as marker.

As seen in Figure 6.3 the values are relatively evenly distributed. Hence there should be good local contrast and likely many features to be found. The marker used in the project can be found in the appendix on page 89.

Cameras and Data Flow

Vuforia replaces the *Main Camera* in Unity with an *ARCamera*. Besides that at runtime the *Texture Buffer* with the camera feed is present. This buffer is a child of a *TextureBufferCamera*

Figure 6.2: Marker for printout.

that only applies the *Texture buffer* and is set to an orthographic projection. This camera is presumably only a camera feed holder(buffer), since the camera object itself is disabled and does not render the camera feed.

A second camera, the *background camera*, is created at runtime and does also have a child object, a plane attached. The plane have the camera feed applied as a material. This plane is present in the scene and is placed in front of its parent camera. The texture given by the material on the plane (the camera feed) is rendered by the *background camera* and is applied as the background for the scene, essential rendered by additional cameras, as for example the *AR camera*. The sequence in which to render the different cameras is defined by the depth property on the camera objects. So, given Vuforia AR solution at runtime the following three cameras are present:

- ARCamera (Depth 1)
- BackgroundCamera (Depth -2)
- TextureBufferCamera (camera is disabled)

Loosely summarizing the *textureBufferCamera* holds the camera feed data where the *BackgroundCamera* actually renders it to the scene and lastly the *ARCamera* renders any 3D content applied in the scene on top of the background feed.

Appendix C — Unity Scripts

Colour-Correction

The histogram matching script requires two textures, a source region (a region of interest from the video-feed) and a target texture (the texture that is going to be matched). These can be chosen with a GUI (see Figure 6.4). Additionally, a background colour can be picked to remove the uniform colour of the background, such that only the UV map is taken into consideration.

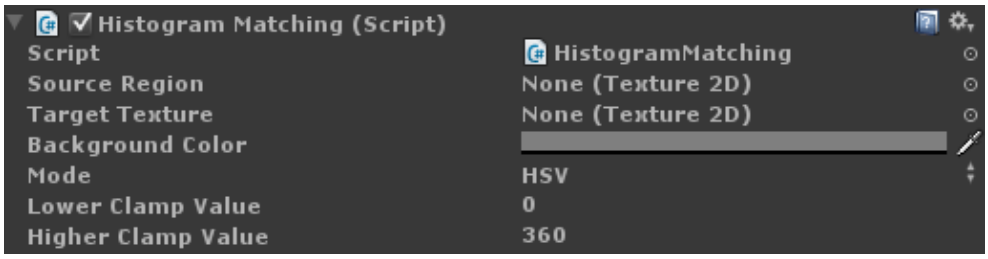


Figure 6.4: The GUI of the colour-matching script.

Histogram matching is available in four colour-spaces: RGB, Luma, LAB and HSV (see Figure 6.5. To limit the range of hues when using the HSV colour-space, a range can be specified by a lower and an upper clamp value.

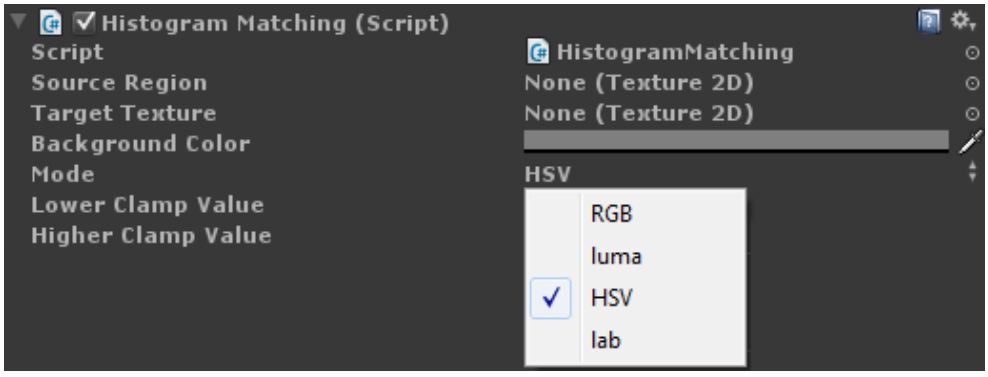


Figure 6.5: Different colour-spaces can be chosen.

The result is saved as a .png file in the same folder as the target texture.

Noise Sampling

To sample the noise, only the “Take Sample Imgs”-script needs to be enabled. A resulting texture size an be chosen, and the script can be run (see Figure 6.6).

The “Take Sample Imgs”-script automatically enables the “Histogram”-script, which buffers a specified number of images (in this case 50) and calculates the noise from these images, samples the random noise and saves it as a noise texture. The texture is automatically added to a material and used for the simulation of noise.

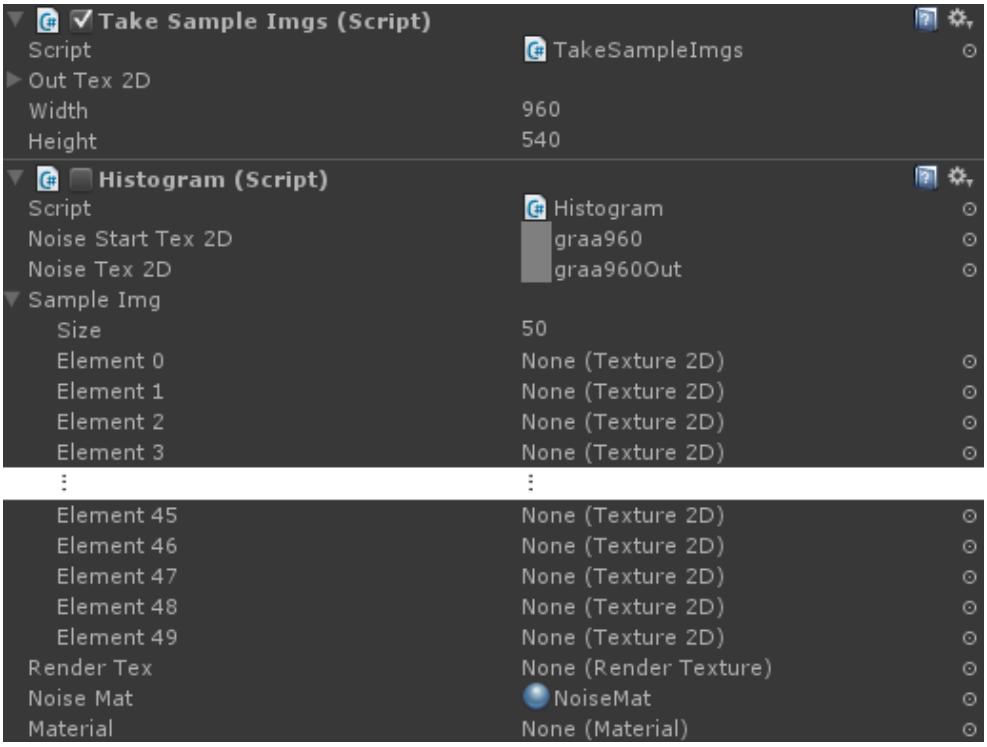


Figure 6.6: GUI for sampling the noise and generating the noise texture.

Note that for the script to work, the “Don’t use for Play Mode” check-box in Vuforia’s “Web Cam Behaviour” needs to be off, to prevent the web-camera to be initiated twice.

Light Generation

To generate the lights from the file output of the median cut algorithm in HDR Shop, the name of the file needs to be specified (see Figure 6.7). If the check-box “Generate Lights” is on, a group of directional lights will be generated according to the specifications in the .ms file. If the check-box is off, the group can be specified and a intensity multiplier can be set, which multiplies all the lights in the group with a specified value.

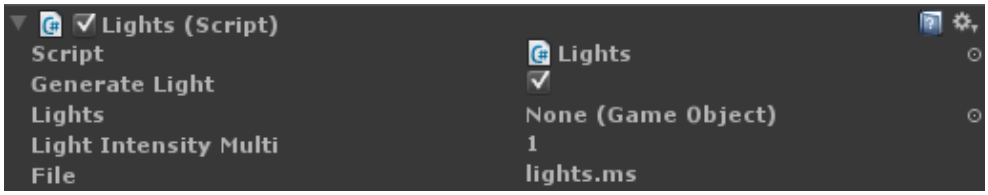


Figure 6.7: GUI for generating and modifying the lights from a .ms file.

Appendix D — Global Illumination Settings in Beast

Only game objects marked as static will be included in the calculation of the light map. A single lightmap is chosen, as there is not much depth in the scene to be rendered, because only the object is present in the scene (see Figure 6.8).

No bounces are chosen, which mean that only the direct lighting is taken into consideration. But as the indirect light from the environment is already given in from the environment map, only the indirect lighting from rays hitting the object is not taken into consideration. This excludes effects such as colour bleeding.

The ambient occlusion setting is not applied as the ambient shadows will be calculated from the ambient lights generated from the environment map.

All in all the settings are set to the lowest possible to be able to render the light maps quickly.

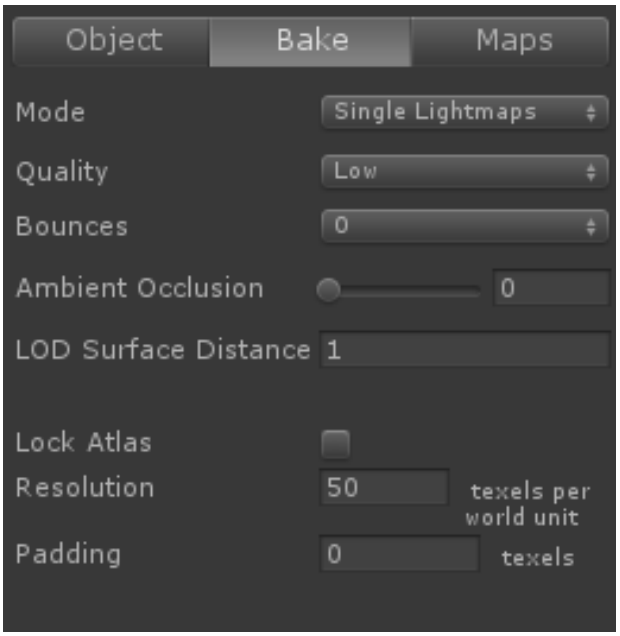


Figure 6.8: GUI for Unity's global illumination system Beast.

Appendix E — Anti-Aliasing

To compensate for unwanted spatial aliasing artefacts the built-in multi-sampled anti-aliasing (MSAA) in Unity can be used to create smoother edges. MSAA uses super sampling, where multiple points are sampled within a pixel. These sub-samples are then considered when rendering the pixels.

However, this anti-aliasing method is not applicable with the Vuforia setup in Unity. Therefore, the fast approximate anti-aliasing (FXAA) method is used, which is a post-processing effect on the screen-space. This method has several adjustable parameters, but comes with five presets (0–4), where preset 3 is considered the best compromise between quality and performance [Lottes, 2009]. The FXAA works by looking at local contrasts in luminance to find edges. This is achieved by looking at the center pixel and the four neighbouring corner pixels and finding the two which has the minimum and maximum luminance. If the difference between the two is below a threshold the value of the center pixel is returned. Otherwise, the direction is determined by the luminance difference. This direction is used to sample pixel in the calculated direction and blending it together. Either two samples if the given pixel is very dark or bright, or four samples in other cases.

This screen space AA has the consequence that the edges are blurred on the entire scene (that being the virtual object and the video-feed), instead of only on the virtual object.

Appendix F — Limiting Jittering

To reduce jittering some of the axes can be locked. In Unity the global coordinate system is set up, such that x and z creates the horizontal plane and y is in the up-direction. Because the physical camera is locked to the rotation of the arm, the virtual camera in Unity should also be limited to follow this path. This is easily done by locking the rotation around the x- and z-axis, by constraining the rotations to a static value. This means that the virtual camera can only rotate around a vertical axis, just like the physical camera.

Next is to constrain the position of the virtual camera. As the physical camera only rotates around the y-axis, the y-position can be locked by restricting its value to a given height. This leaves two unconstrained degrees of freedom. The distance from the physical camera to the centre of rotation is always the same, therefore the virtual camera should follow this constraint too. Given the radius, r, of the rotating arm and the y-position of the camera, the equation of a circle can be used:

$$r^2 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2$$

where coordinate y and radius r are given. To map the coordinate to the circular path given by the equation the intersection between the line created by the position and the equation for the circle is used (see Figure 6.9). A parametric equation is used, where t is the free parameter:

$$r^2 = (x \times t - x_0)^2 + (y \times t - y_0)^2 + (z \times t - z_0)^2$$

In the given setup the origin of the circle is located in coordinate (0,0,13.5) and the radius is 42 (given that the coordinate system is in cm), which means that the equation can be written as:

$$42^2 = (x \times t)^2 + (y \times t)^2 + (z \times t - 13.5)^2$$

Substituting the only unknown variable t:

$$t = \frac{28.5263 * (\sqrt{x^2 + y^2 + 1.25837 \times z^2} + 0.508303z)}{\sqrt{x^2 + y^2 + z^2}}$$

The variable t is then multiplied with the x- and z-coordinate to constrain the position of the virtual camera to always follow the path of a circle.

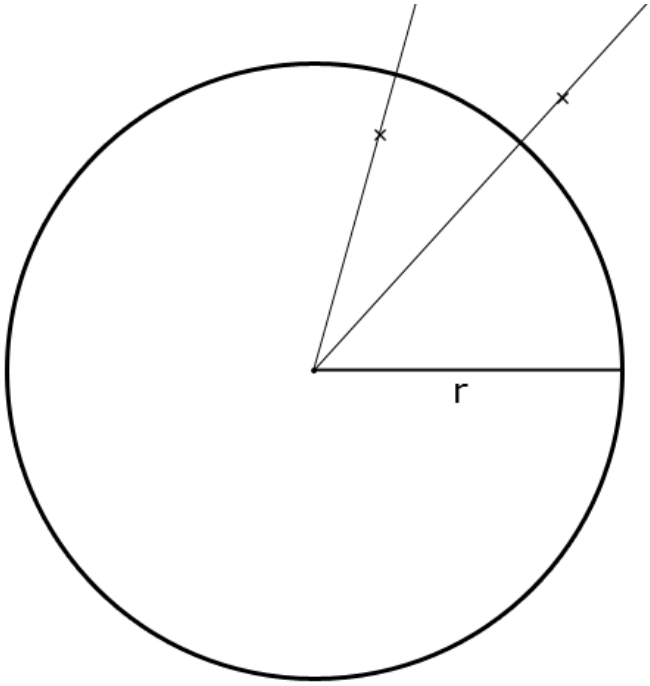


Figure 6.9: A representation of the equation of the circle in the xz-plane. Two examples of points that are not on the circle can be seen. Following a line through the origin and the point a new point on the circle can be found, which meets the requirements of a given radius.

As the maximum rotation is 90 degrees, the x- and z-positions are always kept positive, hence no special consideration is made for the cases where x and z are negative.

Because the virtual camera is constantly moving, the projection onto the circle will also move, as if the physical arm was moving slightly from one side to another. A damping as well as a threshold was tried, which removed the jitter when the arm was not moving. However, when moving the arm the virtual objects would jump from one position to another because of this limitation, instead of moving fluently with the marker. Therefore, this approach was not used.

Another approach was also implemented which moved the virtual object together with the virtual camera. However, as the position of the object is determined by the position of the virtual marker, and the marker is static, the virtual object will appear to float. Then, if the marker is moved with the virtual camera, both will begin to accelerate because the camera position is set according to the marker. In other words, for each rendering loop the camera will be positioned according to

the marker and then the marker will be moved according to the virtual camera. Therefore, this approach was discarded.

To increase the stability without constraining the camera, the marker is lit up evenly and without reflections, and the angle between the web-camera and the marker approaches orthogonal without removing the sensation of depth in the scene (the marker is seen from approximately 70 degrees from a horizontal plane) to make each corner of the marker in focus.

Appendix G — Procedures and Questionnaire

Gender: Man ☐ Woman ☐ **Age:** _____

Occupation: _____

Do you have any visual impairment today? Yes ☐ No ☐

How experienced are you with 3D computer graphics?

☐ ☐ ☐ ☐ ☐

No experience Very experienced

How experienced are you with augmented reality?

☐ ☐ ☐ ☐ ☐

No experience Very experienced

Demographic questionnaire.

Welcome, and thanks for participating in the test!

On the monitor a scene will be presented, in which an object is placed. You have to decide whether or not the object is real or not.

The monitor will be moving from one side to another and we would like you to follow it, as long as you stay behind the line marked on the floor (you are not permitted to lean towards the screen). You will have approximately 10 seconds to watch the scene before deciding whether or not the object is real or not.

After guessing, you will be guided outside to wait until you are called inside again. Then you will have to evaluate a new scene with another object. This procedure will be repeated 12 times in all.

Procedure of experiment 1 — evaluation of perceived realism.

Welcome, and thanks for participating in the test!

On the monitor a scene will be presented, in which a toy elephant is placed. The toy elephant will be shown either in an ambient light or with a spot light. You have to decide whether or not it is real or not.

We will ask you to move the monitor to the positions marked on the floor, where a scene will be displayed (otherwise the display is black). You will have 4 seconds at each position to watch the scene before deciding whether or not the object is real. Please stay behind the line marked on the floor (you are not permitted to lean towards the screen).

After guessing, you will be guided outside to wait until you are called inside again. Then you will have to evaluate a new scene with another object. This procedure will be repeated several times. The scenes can contain only virtual objects, only real objects or anything in between.

Before we begin, you will be shown some examples of the procedure with another object.

Procedure of experiment 2 — evaluation of parameters.

Welcome, and thanks for participating in the test!

On the monitor a scene will be presented, in which two toy elephants are placed. The one will be real and the other one will be virtual. You have to decide which of the two is real. The toy elephants will be shown in a spot light.

You are allowed to move the monitor to the positions from which you want to see the scene. When a position is chosen the scene will be displayed, otherwise the display is black. You will get all the time you need at all the positions you want to watch the scene before deciding which object is real. Please stay behind the line marked on the floor (you are not permitted to lean towards the screen).

After guessing, you will be guided outside to wait until you are called inside again. Then you will have to evaluate a new scene, the same way as the first one.

Before we begin, you will be shown some examples of the procedure with another object.

Procedure of experiment 3 — side by side comparison.

Appendix H — Examples of Scenes

Some of the scenes that the test subjects watch is shown.

(a) Real

(b) With noise

(c) Without noise

Figure 6.10: Examples of a scene with and without noise.

(a) Shading with 2 lights and baked shadow

(b) Shading with 2 lights and real-time shadow

(c) Shading with 16 lights and baked shadow

(d) Shading with 16 lights and real-time shadow

(e) Shading with 2 lights and baked shadow

(f) Shading with 2 lights and no shadow

(g) Shading with 16 lights and baked shadow

(h) Shading with 16 lights and no shadow

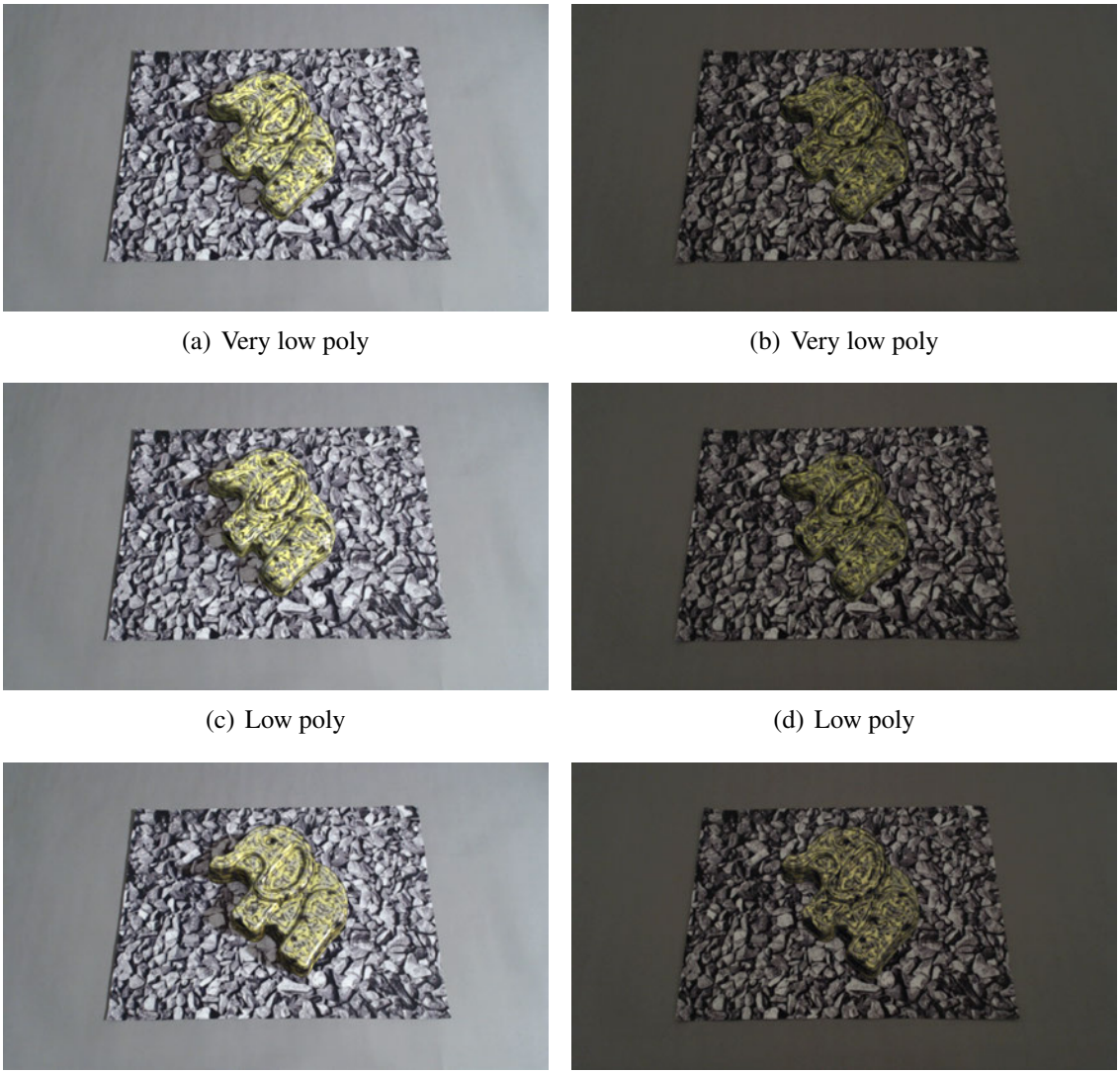
Figure 6.11: Examples of a scene evaluating number of lights to create shading as well as shadows.

(a) Real

(b) With highlights

(c) Without highlights

Figure 6.12: Examples of a scene with and without highlights.



(a) Very low poly (b) Very low poly

(c) Low poly (d) Low poly

(e) High poly (f) High poly

Figure 6.13: Examples of a scene object of different polygon counts.