SIMPLIFYING REAL-TIME LIGHT SOURCE TRACKING AND SHADOW GENERATION FOR AUGMENTED REALITY USING ARTOOLKIT

Bjarne Fisker Jensen, Jacob Sloth Laursen, Jacob Boesen Madsen & Thomas Wisbech Pedersen
09gr731 {bjen07, jlau07, jmad07, tped07}(at)imi.aau.dk
Supervisor: David Meredith
Medialogy 7th semester, Institute of Media Technology, Aalborg University

INTRODUCTION

The usage of augmented reality as a media platform, especially for handheld devices, has the potential to be of very practical use, i.e. user manuals, advertising, information visualization etc. The current state of the art, has no general and ease-of-use implementation of shadow tracking. A technique must be appropriated to human beings and their perception so that the application is seamlessly integrated with the environment. The project is focused on the shadow aspect of a virtual scene and not the constituents of it. Shadows are important cues for the human perception and is therefore of great interest [1]. Establishing criteria for the degree of detail in shadows is necessary for them to be indistinguishable from actual shadows. The criteria established are used as requirement in a lightweight application for tracking and rendering shadows in real-time using ARtoolkit and OpenGL.

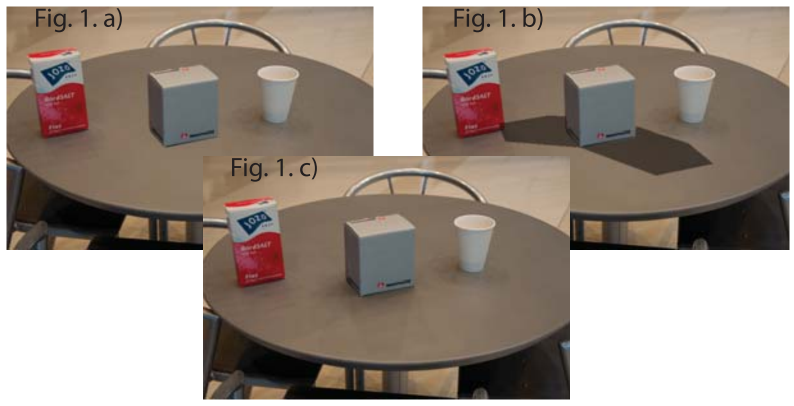


Fig 1 shows the visual difference between compositions of a) AR with no shadows, b) AR with 2 light sources, c) AR with 64 light sources.

METHODS AND MATERIALS

Fig. 2 Light sources can be tracked by an analysis of the environment map if it is present in the image.

Several user tests have been conducted:

- A rank-order test of composited images rated according to credibility
- A more detailed dose-response relationship between, complexity of shadows and their credibility.

Using high dynamic range images, a reflective sphere and the median cut algorithm [2] to generate still images with [1, 2, 4,..., 256] light sources. Casting shadows on a planar surface simplifies the implementation, using one marker to establish a base plane for all shadows to be cast. Kanbara [3] showed results that a real-time light-tracking algorithm is achievable by means of ARtoolkit and a black reflective sphere, placed on a marker. This approach generates an environment map that can be analyzed due to the variation of pixel values of the sphere, that processed by a median cut algorithm detects a user-defined number of light sources in a given environment.

IMPLEMENTATION AND RESULTS

The initial rank order test shows difficulties recognizing the difference between real and augmented shadows with 64 or more shadows. Reference objects are crucial in distinguishing between the credibility of shadows. The dose-response test shows that the credibility of a shadow increases with complexity; however there is no absolute convergence where all test subjects agree upon a credible composition or the real image, see figure 4.

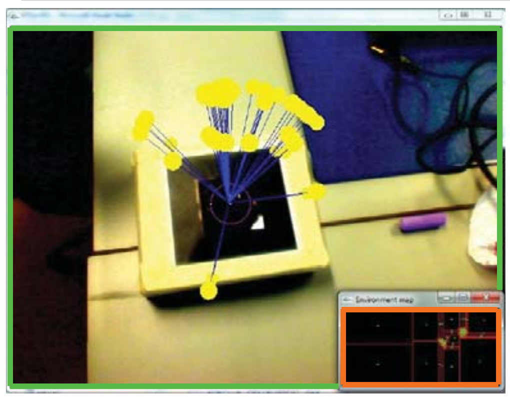


Fig. 3 Estimation of light sources using a reflective sphere and the median cut algorithm.

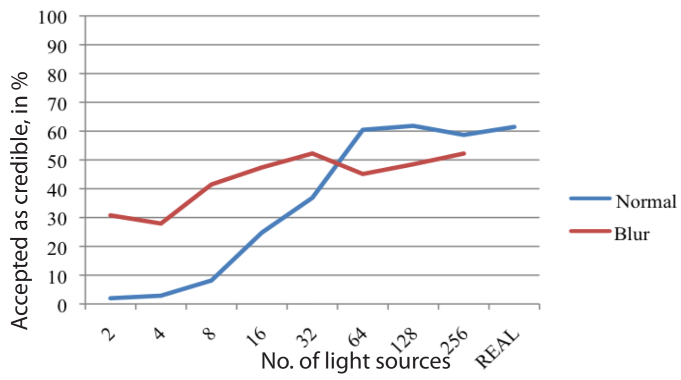


Fig. 4 Test subjects evaluation of images with different lighting, varying the light source parameter only.

DISCUSSION

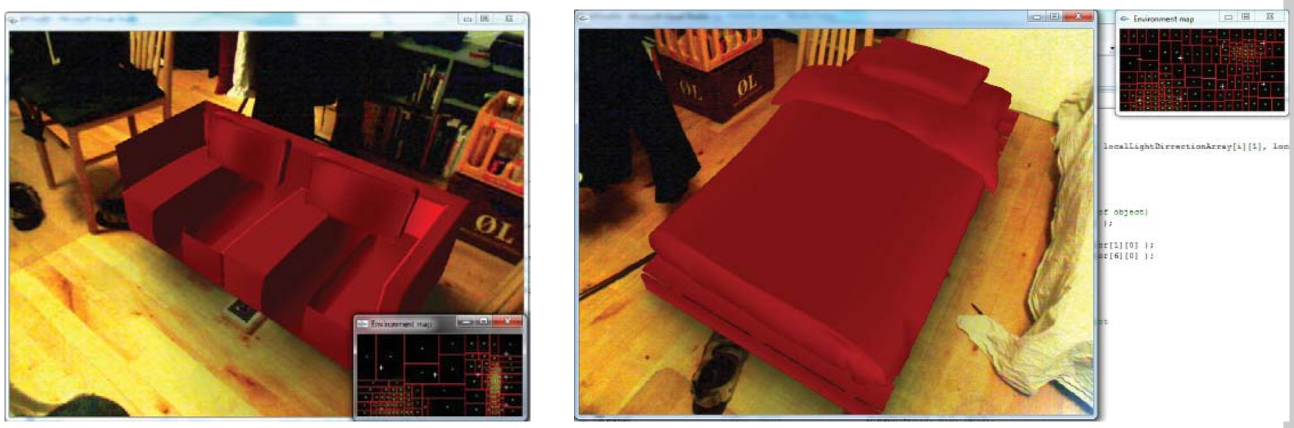


Fig. 4 Screenshots from an augmented reality IKEA planner prototype, where users can try out furniture virtually at home before buying and assembling the product.

The results sugests that it is possible to create simplified shadows in augmented reality that is credible to human perception. The dose-response curve can be used as a reference for the human perception of shadows and the complexity of the generated images. The reflective sphere cannot be packed and distributed on the internet so other options are possible for distributing the implementation. However it proved to be very easy to find reflective spheres with even surfaces, a do-it-yourself guide could get developers to achieve similar results with a little effort.

REFERENCES

[[1] Colin Ware. Information Visualization, Second Edition: Perception for Design (Interactive Technologies), Morgan Kaufmann, 2004, 2. Ed, 266-270.
[2] Erik Reinhard.. High dynamic range imaging : acquisition, display, and image-based lighting. , CA: Elsevier : Morgan Kaufmann Publishers. Morgan Kaufmann series in computer graphics. 2006, San Francisco. xviii, 502 p.
[3] Masayuki Kanbara, Naokazu Yokoya. Real-time estimation of light source environment for photorealistic augmented reality, 17th International Conference on Pattern Recognition (ICPR'04), 2004, 2, 911-914

AALBORG UNIVERSITY

AALBORG UNIVERSITET

Department of Media Technology

Medialogy, 7th Semester

Title: Simplifying Real-Time Light Source Tracking and Credible Shadow Generation for Augmented Reality using ARToolkit

Project Period: 03.09.2009 – 21.12.2009

Semester Theme: Mixed Reality Spaces

Supervisors: David Meredith

Projectgroup no.: 09am731

Members:

Bjarne Fisker Jensen

Jacob Sloth Laursen

Jacob Boesen Madsen

Thomas Wisbech Pedersen

Abstract:

This project presents an interactive vision based real-time system for estimating light source positions and generating credible shadows for augmented reality.

The implementation uses ARToolkit as a basis for geometric tracking, and a reflective sphere for tracking light sources present in the environment.

The project seeks to generate perceptually credible shadows.

User testing was conducted in order to determine the minimum criteria for the credibility of the shadows. Testing showed that 64 shadows are sufficient and indistinguishable from more complex compositions and a real image. It was clear that users need a reference object to distinguish between real and virtual shadows.

The implementation and performance are tested using a consumer-grade web-camera and a regular laptop computer. The implementation can perform real-time with 256 generated shadows.

Simplifying Real-Time Light Source Tracking and Credible Shadow Generation for Augmented Reality using ARToolkit

Bjarne Fisker Jensen, Jacob Sloth Laursen, Jacob Boesen Madsen and Thomas Wisbech Pedersen

Institute for Media Technology, Aalborg University

ABSTRACT

This paper presents an interactive vision based real-time system for estimating light source positions and generating credible shadows for augmented reality. The implementation uses ARToolkit as a basis for geometric tracking, and a reflective sphere for tracking light sources in the environment. The paper seeks to generate perceptually credible shadows. User testing was conducted in order to determine the minimum criteria for the credibility of the shadows. User testing showed that 64 shadows are sufficient and indistinguishable from more complex compositions and a real image. It was clear that users need a reference object to distinguish between real and virtual shadows.

The implementation and performance are tested using a consumer-grade web-camera and a regular laptop computer. The implementation can perform real-time with 256 generated shadows.

KEYWORDS: Image Based Lighting, Human Perception, Augmented Reality.

INDEX TERMS: D.2.6 [Programming Environments]: Interactive Environments; H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities;

1 INTRODUCTION

Augmented reality is a term for interactive and real-time visualization of virtual objects in a real scene [1]. In the real world, if an object is lit, it will cast a shadow. Despite shadows being important for easier three dimensional perceptions and the spatial relationship amongst objects [2], shadows are not always used in virtual or augmented settings.

Shadows most often reveal information of the scene, such as spatial relationships and revealing shapes of objects [2], [3]. To reconstruct a realistic scene in a virtual space is a complex task, in terms of parameters influencing the lighting and shading of a room. These parameters are i.e. the light sources, the luminance and the placement of objects, the geometry of the environment and the structural composition of the environment. People have a precise and well developed understanding of the underlying implications of shadows [1].

In this project, the research has only found Kanbara et al. [4] with a lightweight real-time shadow generating system, even though the volume of research in the field of shadow detection and generation is quite large and extensive. Therefore we seek to make a lightweight implementation for estimating light source positioning and credible shadow generation.

User testing will determine the minimum criteria for the credibility of the shadows, and so will the feasibility of algorithmically adding of visual effects that will be used to reduce the number of light sources necessary to generate credibility in the augmented shadows.

The project seeks implement an augmented reality system capable of tracking light sources using state of the art techniques from the scientific community. The implementation should focus on an easy-to-use approach and should make use of cheap and simple tools and equipments.

2 RELATED WORKS

A lot of research has been devoted to find illumination in the environment and to generate shadows based on this information for augmented reality. An often occurring trend in this research, is the need for a lot of different prerequisites, i.e. known geometry [5], [4], pre-computed environment map [5], stereo-camera setup [6], [7], etc.

Yan [7] proposed a method for estimating illumination parameters in an environment using two cameras to gather information of two lambertian spheres, each visible to the light source. This method generates parameters for a single point light source and ambient light.

Kanbara et al. [4] estimates the light source environment in real-time using a mirror ball for photometric registration. This method uses a generated light source map from the mirror ball and estimates the eight brightest light sources.

Supan et al. [6] uses an approach which tracks a mirror ball and generates a reflective environment map from the mirror ball information, as well as a diffuse irradiance map. Shadows are created by placing a reasonable amount of light sources in the scene that gets their intensity and color information from the environment map.

Based on Kanbara et al. [4], the method in this paper aims to create a real-time application for estimating light source positions and generate perceptually correct shadows using only general tools available to people, to stay in the spirit of ARToolkit.

The implementation should create shadows that are credible to the human perception with regards to placement, softness and length of the shadows. Thus, it is necessary to examine people's perception of shadows. The information gathered from testing will be compared to Nakano et al. [8], which states that 32-128 shadows as the acceptable threshold, above which people cannot distinguish between virtual and real shadows.

Bjarne Fisker Jensen, Jacob Sloth Laursen, Jacob Boesen Madsen and Thomas Wisbech Pedersen are masters' students at Medialogy, Aalborg University. E-mail: {bjen07, jlau07, jmad07, twpe07}@imi.aau.dk

3 METHODS

3.1 Implementation of the system

The implementation should track and process a known marker object and generate a user-defined number of light source positions, using the median cut algorithm [9]. This limits the implementation to use 2^n shadows, due to the median cut algorithm, dividing each area into two at each iteration.

OpenGL [10] is used for rendering the graphics to the screen, as it is the default renderer for ARToolkit [11].

In order to estimate light source positions, a modified ARToolkit marker is used with a reflective sphere at its center. The marker is tracked for position in space and the sphere has information of the light sources in the environment. An example of the setup can be seen in Fig. 1.

Fig. 1. The setup used for estimating light source position using a laptop computer and a standard consumer web-camera

The pipeline of the system is showed in Fig. 2. This serves as a basis for the implementation of the system.

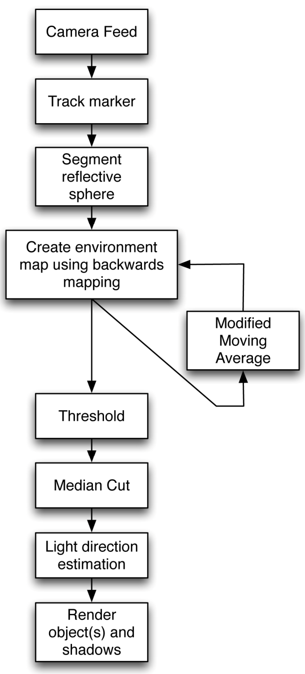


Fig. 2. Pipeline of the system

The light tracking marker consists of a table tennis ball painted with a glossy black paint that can be bought in most convenience stores.

Different colored mirror balls were analyzed to find the best color suited for light tracking without a dynamic range image. An example of these can be seen in Fig. 3.

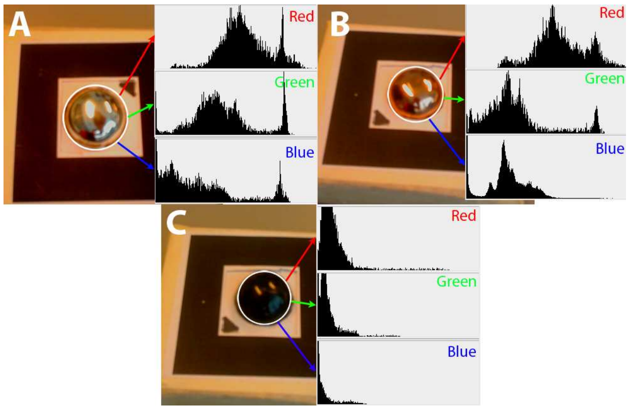


Fig. 3. Analysis of different colored spheres, to find optimal sphere coloring for non-HDRI

A black colored sphere with high gloss was selected, as it shifts less intense light to the lower end of the RGB scale, resulting in more data on higher intensity lights, making it better to track light sources.

As Kanbara et al. demonstrated in [4] the reflective sphere can be segmented , by projecting the center of the sphere and a point on the periphery onto the image plane. This can be done using the model-view matrix supplied by ARToolkit and the known relation between the marker and the reflective sphere. This method provides the center and the radius of the circle representing the sphere onto the image plane. The direction of a given reflection $\vec{r}_{reflection}(1, \theta, \varphi)$ can be calculated using the radius, rMax, of the circle in image coordinates, using the pixel coordinate P in the segmented circle. This is done by assuming that the camera is positioned at an infinite distance along the positive z-axis, thus making all camera rays parallel. The formulas (1) and (2) are used. An illustration of the approach can be seen in Fig. 4.

$$\theta = 2 * \sin^{-1} \frac{\sqrt{Px^2 + Py^2}}{rMax} \qquad , \theta \in [0; \pi] \tag{1}$$

$$\varphi = \mathrm{atan2}(\mathrm{Py}, \mathrm{Px}) \qquad , \varphi \in [-\pi; \pi] \tag{2}$$

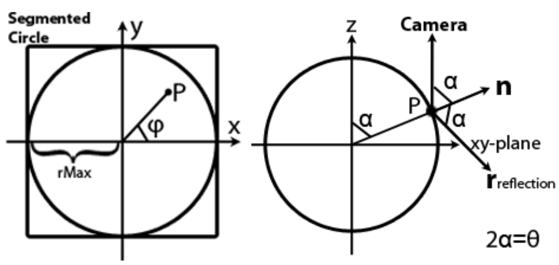


Fig. 4 Illustration of the approach for finding spherical coordinates from 2D representation of a sphere

An environment map with aspect ratio 2:1, of the reflections in the sphere is created using backwards mapping. By applying a modified moving average [12], [13] on the environment map, any sudden changes can be smoothed more or less depending on the sample size without raising computation time significantly. To acquire light directions, the median cut algorithm [9] is applied to the environment map. This is used to estimate 8 OpenGL light source positions and 2^n light source directions for shadow generation. The environment map and light source directions generated using median cut can be seen in Fig. 5.

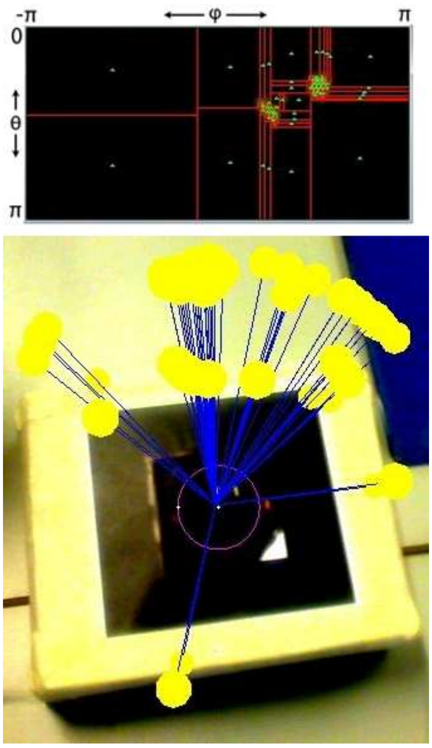


Fig. 5. Environment map of the black reflective sphere, with estimated light source directions, generated from the median cut algorithm.

3.2 Establishing criteria for realistic virtual shadows

A threshold for credibility of perceived virtual shadows as real ones must be established through testing. The test will seek to find whether it is possible to enhance virtual shadows by applying visual effects. It is assumed that an effective shadow generation algorithm can be produced, by finding an optimal compromise between realism and performance. The hypotheses for the test are:

- It is possible to create computer generated virtual shadows, that the user would rank as credible as real shadows.
- It is possible to find a lower bound of light source positions necessary to make the shadows believable to the user.
- By blurring the virtual shadows, it is possible to lower the bound of necessary light source positions.

From the hypotheses, two main problems for the test can be composed:

1. Determine the number of light source positions necessary to generate sufficient shadows.
2. Determine whether the use of visual effects can enhance credibility of virtual shadows.

The test is composed of still images taken in three different real environments with various light settings. Some of the shadows on the images were removed using Photoshop [14] and new virtual shadows were instead created and superimposed by using a HDR environment map created from a reflective sphere and the median cut algorithm [9] to calculate light positions. In each setting 10 images were composed with shadows generated from $[0, 2^1, 2^2, \ldots, 2^8]$ light sources, with and without a boxblur [15] applied, and 1 image with genuine shadows were used as a control image. The three different setting are shown in Fig. 6, each with 256 light sources. In B and C where three objects are present, only the middle object has virtual shadows. This gives a paper-cube, a book and a box setting.

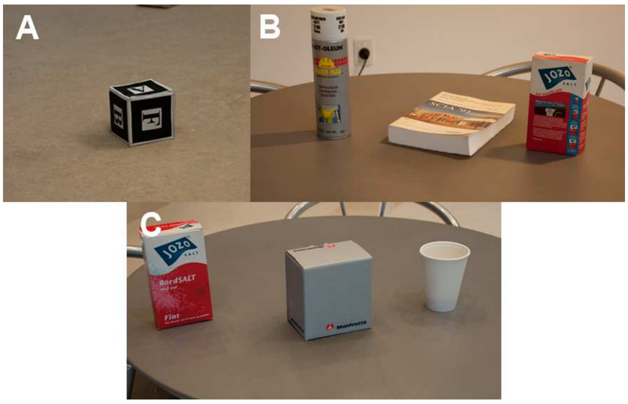


Fig. 6. Example images from the 3 different settings for use in the test. All with superimposed virtual shadows created with 256 light sources. In a) and b) only the middle objects have virtual shadows

The images are used to compare the imitated shadows against the real image and against each other, in order to find a boundary for credible shadows. The test will consist of two test stages:

1. An initial rank-order test to rank credibility of composited images. This first test act as a pilot test, to find any errors that might emerge from the images and test setting. Candidates from the initial test are used in a subsequent final test.
2. A second final test is a dose-response scenario [16] determining the percentage of the test population that judges an image as real or imitated. This test uses candidates from the first initial test.

4 Experiments

4.1 Results from shadow criteria test

The initial rank order test was performed on 13 participants, who were set to rank each image from least credible to most credible.

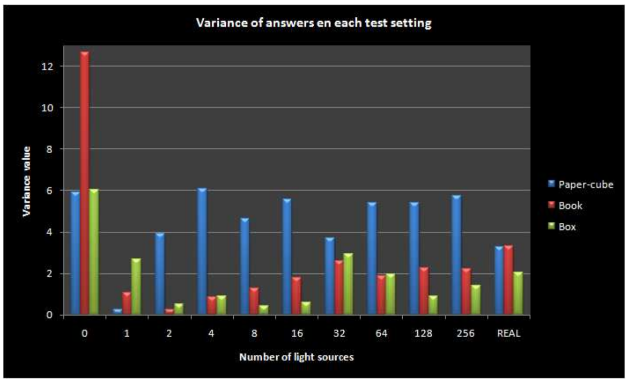


Fig. 7. Rank-order test and the plotted variance for the answers of each image in the different test settings

The diagram of Fig. 7 indicates two observations that are essential for our final dose-response test. First it indicates that an object that casts no shadows at all is very hard for people to rank. Second, the whole test setting with the paper-cube seems to confuse people as they have very scattered opinions on the ranking of each image. This could indicate that people it difficult evaluating shadows in images where no real shadows are present as a reference. Based on the results from the first test, the first test setting with no real shadows as reference together with the images with zero shadows is discarded for the final dose-response test.

The final dose-response test was performed as an online questionnaire and was posted on Facebook [17] and on the forum of Nordic-T [18]. This is done in order to reach a broad number of different people. This of course, could influence the results because both sites targets specific users, but overall it should give a fairly good understanding of how people perceive virtual shadows. 500 people took part in the test. To prevent the participant from studying an image for too long and not getting accustomed to the task, only 4 images from each of the two setting were shown, for a duration of 10 seconds. The participant then had to make a choice whether he would rate the shadow as obviously imitated or if he would not have noticed anything wrong. Also the participant had the possibility of submitting a “don’t know” answer, instead of choosing between credible and imitated.

In the diagram of Fig. 8, the results from the final dose-response test can be seen. Some data have been discarded after a thorough analysis. The data that have been discarded are that of participants that have taken a very long time to submit their answers. As the test is about finding the intuitive credibility criteria from test subjects, it makes sense that subjects, who answered did not know whether the shadows was imitated or real, is not seen as an obvious imitated shadow. As a consequence this data is plotted as if it was equal to a credible answer from the participant.

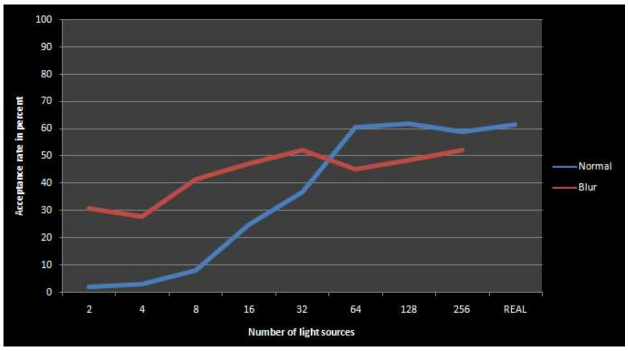


Fig. 8. Dose-response relationship between number of light sources and acceptance rate

The first apparent result is that generally people are very skeptic in an online questionnaire about shadows, as the maximum acceptance rate, even for real images, are just above 60 pct. This is in contrast to the initial first test, where many participants rated images with about 64 shadows as images that could have been real. But if we hold on to the fact that the real image indeed is real, then the test shows that 64 shadows or more should be enough to replicate a real scene. Also the visual effects seem to have a very positive effect on images with a low number of light sources. But as we pass 64 light sources or higher, the blur effect seems to have the opposite effect and just make the image less credible. On the other hand, when 64 light sources are enough to simulate a real image, it would not be necessary to use the blur effect anyway.

4.2 Test of the implementation

4.2.1 Performance test

The performance test was conducted using a Dell XPS M1530 laptop computer, using windows operating system running on a Core 2 Duo T7500 2.2 Ghz with 2.0 GB RAM, a NVIDIA GeForce GO 8600M GT graphics card with 256 MB RAM, and a web-camera with an 800x600 pixels resolution at maximum 15 frames per second, using a 20 stacks and 20 slices sphere as the virtual test object. The results can be seen in Table 1.

Table 1. Frame rate result of performance test. Camera is limited to 15 fps, which limited the performance

Nr. of shadows	Frame rate (fps)
0	15.0
8	15.0
16	15.0
32	15.0
64	15.0
128	15.0
256	15.0
512	12.1
1028	8.8
2048	5.4

The test results are limited due to the web-camera. In order to limit the need for special equipment, a consumer-grade web-camera has been used [19], which is limited to 15 fps with 800x600 resolution.

4.2.2 Qualitative user test

The final implementation has been evaluated by peer students at the department. Interested parties have volunteered to see and review the results.

The interview form was based on an unstructured interview [20]. The test subjects were presented with a few videos from the implementation, using 2-3 light sources. What we sought to investigate was whether direction, length, softness and general impression of the generated shadows lived up to the standards that they would expect.

The results are positive, based on feedback from peers. The direction and length of the shadows seems to be coherent with expectations. There are however some problems with the number of shadows generated, especially from a limited number of real world light sources. This is a problem due to the nature of the median cut algorithm, when estimating a high number of light source positions. The presence of reference real objects in the scene seems to make it harder to imitate the perception of shadows. Different shadow cast techniques could improve the implementation even further.

5 Conclusion

In this paper, a method for light source estimation has been presented, which estimate the positions of light sources in the environment and generate credible shadows of a virtual object augmented into the scene, in real-time. The system uses a reflective sphere for tracking light source direction in the environment, which is mathematically converted to an environment map. The median cut algorithm [9] is performed to find light source directions.

Virtual objects can then be placed in the scene using modified ARToolkit markers with shadows generated from the estimated light source directions.

Our implementation has succeeded in generating credible shadows with correct placement, and the implementation is interactive with frame-rate limited by the current web-camera.

The initial rank-order test confirmed our hypothesis which states that it is possible to imitate shadows using a number of light sources to create virtual ones. It also confirmed that an increasing number of shadows will increase perceptual realism, up until virtual and real shadows no longer can be distinguished from each other.

The test showed an acceptance rate of about 60% for a real image in the tests, which also is the level for 64 and more virtual shadows. This indicated that using 64 or more shadows in this implementation will generate shadows credible to human perception. This is within the range specified by Nakano et al. [8].

The qualitative interview with peers, who had seen a demonstration of the implementation, showed that the implementation had the desired general impression that was sought.

6 Further development

The estimation of light source positions can be improved by using a high dynamic range camera and getting high dynamic range information of the reflective sphere. This will lead to more precise light source estimation. The coloring of the light should also be considered, as the current implementation only use RGB values for determining intensity through median cut, and not coloring. This information could be used to generate color-bleeding as well.

The current shadows of the scene should be tracked to avoid drawing shadows into shadows already in the scene, making the appearance look fake [21].

The type of shadows should be considered as well. Having the option to render projective shadows, shadow mapping and shadow volumes should be an option to allow users the ease of generating shadows suited for their application.

Acknowledgements

This paper is a semester project from 7th semester Medialogy, Institute of Media Technology, Aalborg University.

The authors would like to thank Claus B. Madsen for his assistance which helped improve the project.

References

[1] C. B. Madsen, "The Importance of Shadows in Augmented Reality," p. 4, 2003.

[2] N. Sugano, *et al.*, "The effects of shadow representation of virtual objects in augmented reality," *Second Ieee and Acm International Symposium on Mixed and Augmented Reality, Proceedings*, pp. 76-83
363, 2003.

[3] J. M. Hasenfratz, *et al.*, "A survey of real-time soft shadows algorithms," *Computer Graphics Forum*, vol. 22, pp. 753-774, Dec 2003.

[4] M. Kanbara and N. Yokoya, "Real-time estimation of light source environment for photorealistic augmented reality," *Proceedings of the 17th International Conference on Pattern Recognition, Vol 2*, pp. 911-914
1048, 2004.

[5] M. S. Andersen, *et al.*, "Estimation of dynamic light changes in outdoor scenes without the use of calibration objects," *18th International Conference on Pattern Recognition, Vol 4, Proceedings*, pp. 91-94
955, 2006.

[6] P. Supan, *et al.*, "Image Based Shadowing in Real-Time Augmented Reality," *The International Journal of Virtual Reality*, vol. 3, pp. 1-7, September 2006.

[7] F. Yan, "Estimation of light source environment for illumination consistency of augmented reality," *Cisp 2008: First International Congress on Image and Signal Processing, Vol 3, Proceedings*, pp. 771-775
815, 2008.

[8] G. Nakano, *et al.*, "Generating Perceptually-Correct Shadows for Mixed Reality," *7th Ieee International Symposium on Mixed and Augmented Reality 2008, Proceedings*, pp. 173-174
211, 2008.

[9] P. Debevec, "A Median Cut Algorithm for Light Probe Sampling," presented at the International Conference on Computer Graphics and Interactive Techniques, Los Angeles, California, 2005.

[10] 20. December). *OpenGL*. Available: http://www.opengl.org/

[11] P. Lamb. (1999, 20. December). *ARToolKit* Available: http://www.hitl.washington.edu/artoolkit/

[12] P. J. Kaufman, *Trading systems and methods*, 3rd ed. New York ; Chichester: Wiley, 1998.

[13] 20. December). *Modified Moving Average*. Available: http://en.wikipedia.org/wiki/Moving_average

[14] (2009, 12. Dec). *Adobe*. Available: http://www.adobe.com/products/photoshop/photoshop/

[15] 12. Dec). *Wikipedia*. Available: http://en.wikipedia.org/wiki/Box_blur

[16] A. S. Kane. 12. Dec). *DOSE RESPONSE CONCEPTS*. Available: http://aquaticpath.umd.edu/appliedtox/dose-response.pdf

[17] (2004, 12. Dec). *Facebook*. Available: www.facebook.com

[18] 12. Dec). *Nordic-T*. Available: http://nordic-t.org/

[19] *Logitech C905*. Available: http://www.logitech.com/index.cfm/webcam_communications/webcams/devices/5868&cl=dk,da

[20] H. Sharp, *et al.*, *Interaction design : beyond human-computer interaction*, 2nd ed. Chichester ; Hoboken, NJ: Wiley, 2007.

[21] K. Jacobs, *et al.*, "Automatic generation of consistent shadows for augmented reality," presented at the Proceedings of Graphics Interface 2005, Victoria, British Columbia, 2005.

Grp 731

Worksheets

Simplifying Real-Time Light Source Tracking and Credible Shadow Generation for Augmented Reality using ARToolkit

Grp 731
21-12-2009

1 Readers guide.

This collection of worksheets serves as guidance and in-depth description of the approach, motivation and implementation described in the paper. The worksheets will use APA 5th notations when referring to material and sources.

Any references to the appendix refer to the included DVD. The appendix consists of a data table.

2 Introduction

Shadows serve a huge contribution to a visual composition and can serve as a mediator of information of underlying objects, its material composition, the geometry of the object and its surrounding geometry (Hasenfratz, Lapierre, Holzschuch, & Sillion, 2003). There is a considerable amount of research in the detection and reconstruction of shadows that all have generated specific solutions for pre-rendering purposes, virtual reality or augmented reality, but usually they have limitations so strict that they cannot be successfully implemented into real time systems due to their complexity or their limited applicability due to restraints from necessary prerequisites, etc.

Shading techniques is a field of research that is focused around adding visual cues to object, to make virtual images more credible to the human senses. Shading focuses on realism, but to actually reconstruct a real world image the complexity of shading alone would not be computable if all known phenomena had to be reproduced and simulated. The simplification is of an infinite complex structure like that of shading that is present in our natural surroundings. To replicate this phenomenon in mixed space reality is a task that possibly requires a huge amount of information from the surroundings.

To reconstruct a realistic scene in a virtual space is a complex task, in terms of parameters influencing the lighting and shading of a room. These parameters are i.e. the light sources, the luminance and the placement of objects, the geometry of the environment, the structural composition of the environment, to mention a few. Human beings have a precise and well developed understanding of the underlying implications of shadows (Claus B. Madsen, 2003). This results in an ability to describe objects only through the composition of the shadows. Computationally this is a very complex composition of parameters and simplified models that achieve similar results to reality have been applied to achieve believable results without all the information required for a complete simulation. Because augmented reality implementations depend on fast update rates, the amount of computation required should be low

In this project the research has found no lightweight real-time shadow generating API, even though the volume of research in the field of shadow detection and generation is quite large and extensible. Therefore we seek to make a lightweight API, packaged with all necessities that serves its purpose and leaves the developers with the overhead needed to make their applications. User testing will determine the minimum criteria for the credibility of the shadows, and so will the feasibility of algorithmically adding of visual effects that will be used to reduce the number of light sources necessary to generate credibility in the augmented shadows.

2.1 Innovation and scientific research

A lot of research is very specific within very strict limitations and results can be hard if not impossible to replicate outside these restrictions. This gap between research and innovation, where people actually start using the technologies can be quite a challenge in itself. There is a substantial workload involved in the process in making findings from research papers into applicable solutions for 3rd parties, The interest from a broad variety of innovators requires that the research is boiled into a useful, general, easily distributed and applicable format or package. The diffusion of innovation by Everett M. Rogers, states according to the review by Les Robinson (Robinson, 2009) it is not the people who change but the innovations themselves that do. If that is the case, the sheer packaging of the implementation and its functionality can be the determining factor whether the technology is embraced by early adopters or not, and its second implication is that there is a lot technology out there never making it out of the realm of scientific researchers and innovators, because the implementation is too hard to do for the early adopters. If Everett M. Rogers is right there is a lot of work to be done to achieve this transition from innovators to early adopters as shown in Figure 2.1, so that the impact of a scientific discovery can achieve its full potential. This work will primarily consist of knowing and understanding the current State of The Art (SOTA) and making it applicable for the early adaptors. Therefore the target audience for the API will be the early adopters. The credibility criteria will still be using the general population, with intact vision, as its target audience.

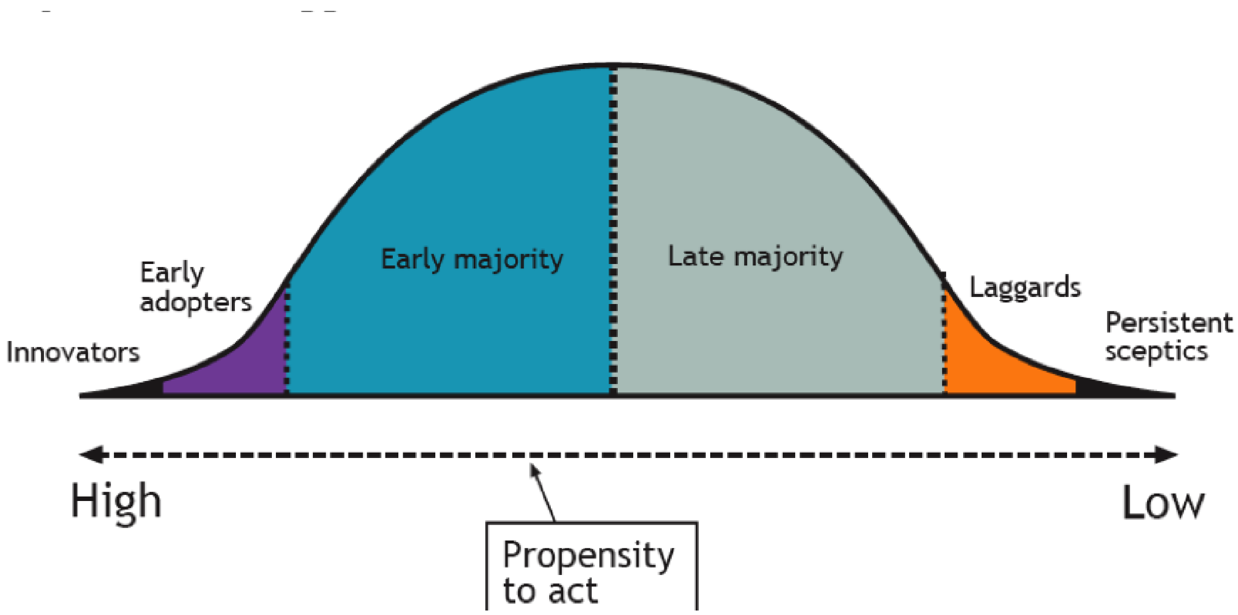


Figure 2.1 The diffusion of innovation by Everett M. Rogers decribes the appropriation of innovations by society.

2.2 Purpose of the project

The project is solely focused on the shadow aspect of a virtual scene, not the constituents of it. Therefore the project will use the ARToolkit markers to define geometrical pose and position, and will experiment with techniques that can successfully track light sources and create a basis for creating humanly convincing shadows. It is also relevant for the project to make a package that can be useful for other segments of users than innovators/developers/scientists. According to Everett M. Rogers this could prove useful to push the innovation from the realm of innovators, into the early adaptors where the propensity to act increases. This could help in the appropriation and popularization of augmented reality for all sorts of purposes, virtual shadows and of course improve the realism of real-time rendered augmented environments. The different techniques for creating shadows will be tested for their ability to convince human beings with intact eyesight.

3 The Definition of Augmented Reality

Augmented reality(AR) is based on a technique of using a livefeed from a camera where objects are superimposed on top of the image as an extra layer of information. The task is to make the layers blend so that the user receives additional information than what is present in the real environment. This additional information can be anything depending on the application as a whole.

The use of augmented reality in real-time is widespread for developers and the early adaptors as described in the introduction. Augmented reality is a tool to superimpose virtual and imaginary objects into a physical scene. The intermediary medium is however of great interest, because it is the limiting factor for augmented reality applications in general. Modern PDA's, Head Mounted Displays (HMD) and mobile devices, today possess the power to run the algorithms necessary, and is therefore a much more applicable and intuitive platform to implement augmented reality. Freeing the user from the cabling and artificial environment that most stationary implementations needs. These capabilities has only come to show lately with practical implementations like the wikitude API for googles android phones ("WIKITUDE," 2009). The tracking method is based on gps and a compass sensor which is implemented in some models of smartphones. The approach is interesting but there is no way for it to tell anything about light sources in the environment. The only valid approach at hand seems to be an image processing approach which is implemented and distributed in ie. ARtoolkit.

The issues that arises in augmented reality is the credibility of the composed image, if there is no additional cues about the objects ie shadows, color bleeding, selfshadowing the concept can become artificial and synthetic for human beings.

Augmented Reality (AR) and computer-generated imagery (CGI) are very common definitions. CGI is the process of generating and animating elements in a computer to be composited into a scene as if the elements were present in the scene. Some movies are comprised entirely of computer generated imagery, with no live action photography whatsoever ("Digital rebellion," 2009). It is well known that CGI is used by the movie industry to enhance the experience of movies. For example by creating environments, characters, effects, etc. that wouldn't otherwise be possible to create, hard, or just more expensive to achieve in the real world. Production examples of used CGI are of course animated movies like Wall-E or Up ("Pixar Animation Studios," 2009). Another example where CGI are composited into a real scene to improve story or believability is Cloverfield ("Cloverfield," 2007), which depict a monster attack in New York from the perspective of a small group of people. One thing CGI is used to create in the film is the monster which is quite convincing, and it probably would have been very hard to achieve the same results without the use of CGI. As such, when crafted correctly, CGI helps create realistic worlds for the viewer that otherwise would have been impossible to make, and CGI are essential for many of today's movie productions.

Figure 3.1 Posters of the movies Wall-E and Cloverfield, two productions where CGI is heavily used.

Augmented Reality deals with the same problem area of CGI, namely to composite computer generated elements into or onto a real scene to extend reality, that is, the reality is augmented. But where CGI normally is done as a post production process, AR requires to be real-time and interactive by definition (Claus B. Madsen, 2003). As such AR is better suited for entertainment, games, simulation, etc. because it can implicate the user and supplements the world in real-time, i.e. the user sees real environment combined with virtual objects. A related technology is Virtual Reality (VR) which replaces the real world with a completely synthetic world that the user can interact with (Milgram & Kishino, 1994). The virtual world may or may not mimic the properties of a real world environment, either existing or fictional. In contrast, a strictly real-world environment clearly must be constrained by the laws of physics. With this knowledge, a span between reality and virtuality can be created, known as the *Reality-Virtuality Continuum*, as shown in Figure 3.2.

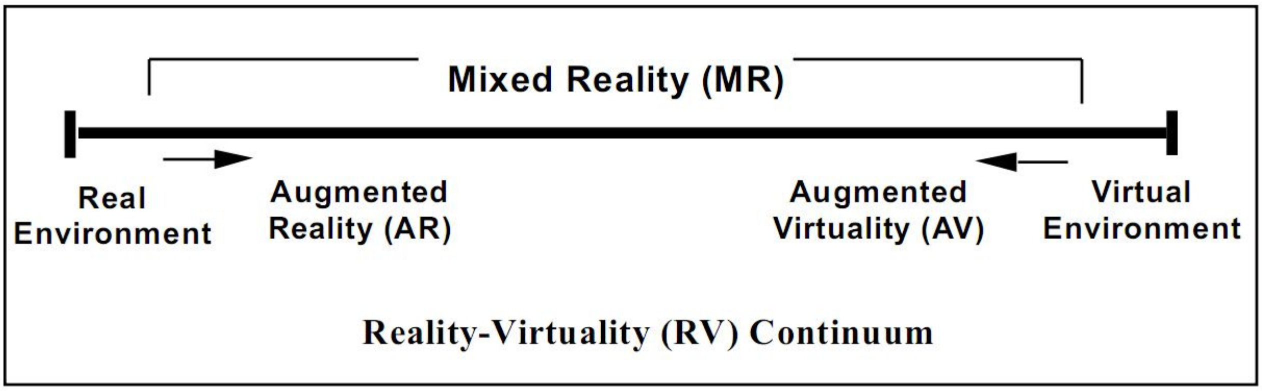


Figure 3.2 Reality-Virtuality (RV) Continuum. Show the span between a real environment and a virtual environment (Milgram & Kishino, 1994).

At the left most side of the scale of the continuum it defines any environment consisting solely of real objects, and includes whatever might be observed when viewing a real-world scene either directly in person, or through some kind of a window, or via some sort of a display. At the right most side of the span it defines environments consisting solely of virtual objects, examples of which would include conventional computer graphic simulations, either monitor-based or immersive. Within this framework it is straightforward to define a generic Mixed Reality (MR) environment as one in which real world and virtual world objects are presented together within a single display. (Milgram & Kishino, 1994)

As it is hard to define when the definition of augmented reality becomes a definition of augmented virtuality, the continuum is not sufficient for some purposes. The question is if the environment being observed is principally real, with added computer generated enhancements, or it is the surrounding environment that is principally virtual, but augmented through the use of real imaging data. Therefore Milgram and Kishino do extent the continuum with a three dimensional taxonomy. This taxonomy however, is outside the scope for this project, and will not be described further.

3.1 Prerequisite for realistic CGI and Mixed Reality

There are various aspects to consider when CGI have to look credible in a movie. Though it is much harder to get credible mixed reality the same aspects apply for getting virtual elements in mixed reality to look as if they were real. It should be mentioned, that while in many cases it would be obvious to want a virtual setting that blends with a real setting, it might not always be an important factor. It entirely depends on the application at hand. Below is listed some different aspects to make virtual objects look credible in a real environment:

- Context
- Perspective
- Geometry
- Textures
- Shading
- Shadows

Context

A virtual object, placed in a real environment, which is totally out of context, would definitely pick up a users suspicion. For example, a virtual full body armored knight placed in a modern kitchen would seem odd and make the setting less likely to happen. But maybe if the knight was part of a virtual game, where a knight is transported to the future, the context would change and the setting might seem much more plausible. Another conflict in context could be an oversized object which doesn’t correlate to the scene space.

Geometry

An erroneous geometry of a virtual object could course the object to look edgy or the object consists of geometry that would not fit a real object. This of course would affect the reliability of the virtual object when augmented on a real scene. For example a ball which is supposed to be a sphere but maybe rather looks like a polygon with many faces, would lower the credibility.

Textures

The texture of a virtual object is of course of importance. A 3d model of a stone wall that is textured with purely red, would of course never look like a stone wall to a user. The wall or at least the stones must be textured with some kind of texture to get the looks of stones. In Figure 3.3 an example of a tank with random texturing and good texturing is shown.

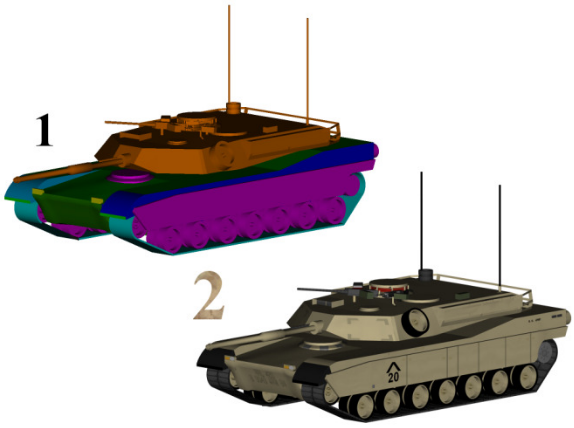


Figure 3.3 Example of texture mapping of an object. In 1 faces of the 3d object have been assigned random colors. In 2 the tank have been textured with a metallic texture to give its tank look ("Texture mapping," Wikipidia).

Perspective

Another important requirement is the perspective of the virtual object has to match the perspective of the real world which it is incorporated to. In Figure 3.4 it is shown how a virtual object can be placed with a wrong perspective in the real environment. In Figure 3.4a the perspective of the green box doesn't match the environment while in Figure 3.4b it does match.

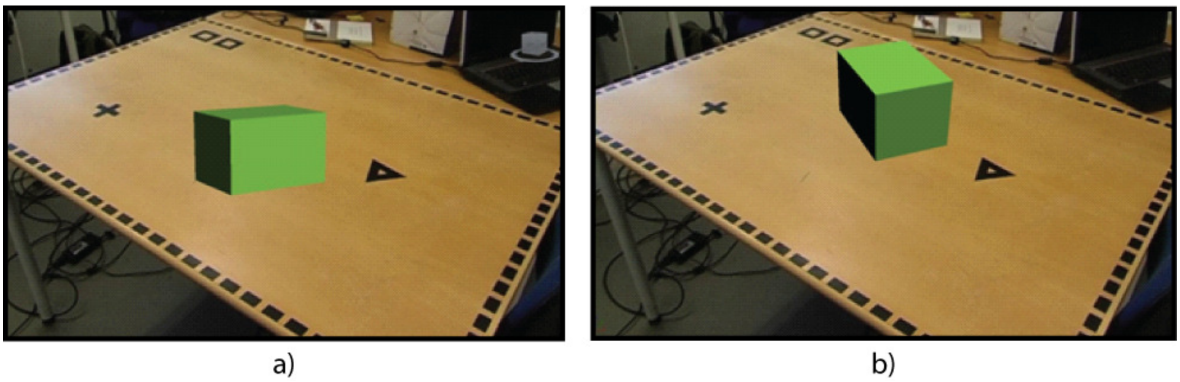


Figure 3.4 Illustrates how a wrong perspective applied to a virtual object can lower credibility. A) Perspective of the green box does not match the real environment. B) Perspective match the real environment. (Ekelund, Larsen, Laursen, Nielsen, & Pedersen, 2008)

When matching CGIs perspective in a movie, a special but tiresome technique is used, called matchmoving (Dobbert, 2005). Automatic matchmoving done by a computer, analyses the footage to find special feature points that are easy for the computer to extract. Based on a theory called photogrammetry the computer can reverse the original projection done by the camera, going from a set of 2d images to 3d space, and thereby calculate the perspective. This perspective can then be applied to a virtual object. This however, is computationally very expensive and therefore is not suited for augmented reality. A recent project though, has been published in this field, which uses a so called PTAM-algorithm that allow this to be done real-time with some limitations in accuracy (Klein & Murray, 2007). In augmented reality a tracking device with known geometry is normally used to calculate the perspective the camera records of the scene. ARToolkit (Lamb, 1999) is a very known software library used to such a task.

Shading

Shading of an object is the effect of change in local surface orientation which will cause a gradual change in reflection of light. Shading information alone is a powerful cue to depth in an object (Matlin & Foley, 1997, p. 178). As a real environment always contain depth, it is vital for virtual objects to share the same amount of depth. Objects that are not shaded or lack proper shading will look flat and dull. For example in an evenly colored object, one would not be able to distinguish each faces from each other if no shading is applied. This is well illustrated by examining the three boxes in Figure 3.5.

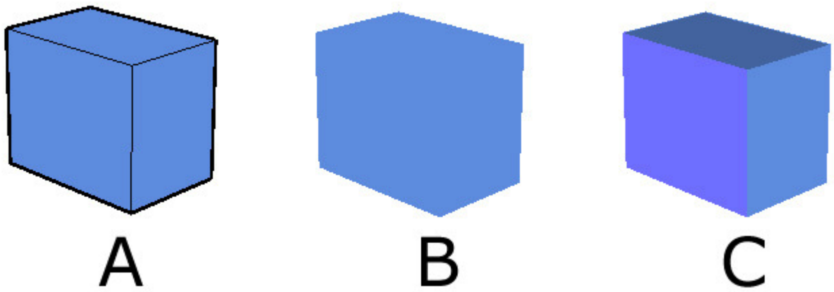


Figure 3.5 Rendered image of a box. A) This image has no shading on its faces, but uses edge line to separate the faces. B) This is the same image as in a), but with edge lines remove. C) The same image rendered with shading of the faces to alter the colors of the 3 faces based on their angle to the light sources

Shadows

Together with shading, shadows cast by an object are of very importance when adding cues about depth to an object and the scene. Shadows also give information about the shape of the object, the shape of the background surface and the spatial arrangement of the object relative to the background and other objects (Mamassian, Knill, & Kersten, 1998). By looking at the rendered image in Figure 3.6, it is shown how shadows can add depth and realism in a scene.

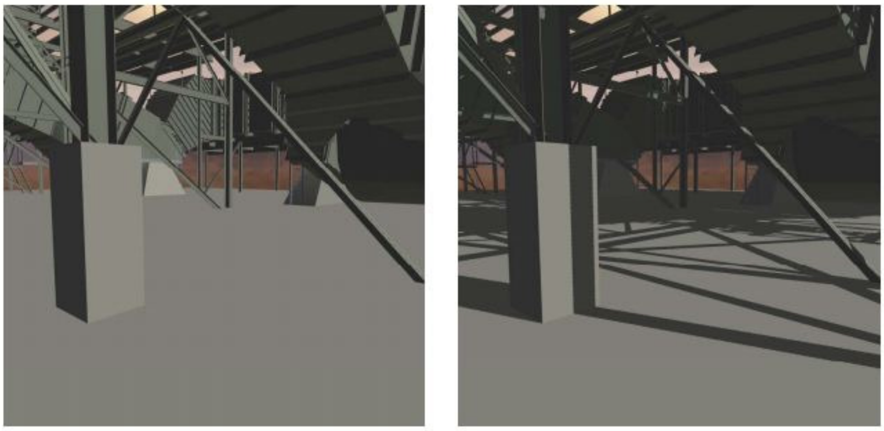


Figure 3.6 A scene rendered with and without shadows cast by objects. The shadows gives clues about depth, contact points etc.

3.1.1 The shading and shadow issue in augmented reality

In the above section different aspects for creating credible virtual objects in a real environment have been highlighted. When creating augmented reality applications, many of these aspects are up to the developer to decide, how important they are to the application at hand. What is important to notice is that some of the aspects can be handled before the application reaches its user, and is not dependent on the environment where it is used. That goes for context, geometry and textures. The developer has full control over these aspects, and it is entirely op to him to choose how believable these should appear on the user. Of course each aspect is still dependent on the skill, current knowledge and the desired performance of the application.

The perspective in augmented reality application on the other hand, is much more dependent on the usage and the environment where it is used, and the perspective is the key feature for making it possible to create such applications. Normally a maker with known geometry is used to track and calculate the perspective of the real scene. A widespread API that is used to build augmented reality applications is the software library ARToolkit (Lamb, 1999). This API enables the calculation of the perspective of a scene by using a quadratic black and white maker that can be printed by a normal printer for home use. This makes it pretty easy for a normal developer to create AR application, and use ARToolkit for the perspective calculation.

This brings us down to the actual problem when developing AR application. Shading and shadowing are very much dependent on the environment where the application is used, and there is current no easy way to extract and calculate the lighting condition of the real scene. Although shading and shadows can be assigned and calculated as a static setup when the application is written, this is not very suitable for AR applications; Either one have to oppress shadows to a level where they don't draw much of attention but still provide some depth, or the shadows will only work in very specific environments.

When making CGI for movie production the lighting condition in an environment is calculated for each scene. Two things make it difficult to transfer methodologies from movie production to realistic interactive AR. First of all lighting and shadow casting from virtual objects is manually tuned and optimized when rendering a scene for a film. Secondly the computer graphics techniques used for rendering film suitable for real-time performance. (Claus B. Madsen, 2003).

This outlines an obvious problem when developing augmented reality applications; the lack of a proper way to calculate lighting condition in the used environment, greatly reduces the credibility of virtual augmented objects for unknown environments.

3.1.2 The importance shadows and shading

This section is based on (Mamassian, et al., 1998), (Kersten, Mamassian, & Knill, 1994), (Claus B. Madsen, 2003).

As described, shadows in a scene have several cues for the perceptual understanding. Shadows provide clues about depth, scene lighting, contact points, spatial relationship, etc., and gives information about the shape of both objects and background. These aspects are well illustrated in Figure 3.7.

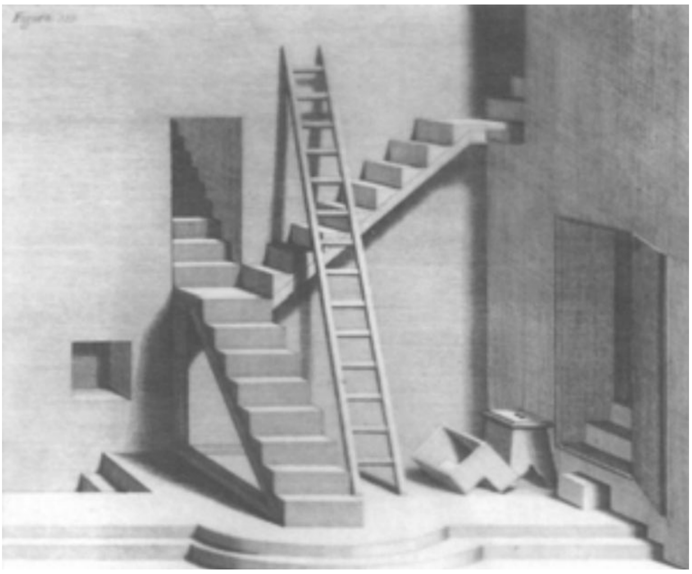


Figure 3.7 Classic picture that demonstrated the importance of shadows in an image. It is very clear how it support the perceptual understanding of depth, contact points, size of objects and spatial relationship

Shadows shades are often defined to be separated into different areas; shading is formed on surfaces that directly faces a light source, and depict the variation of reflected light, whereas shadows are regions that are occluded from the light sources. Shadows are split into two areas, attached shadows and cast shadows. Attached shadows are formed on the very surface which is occluding the light the light, where cast shadows are formed when one surface occludes another surface from the light source.

This necessarily restricts the information from attached shadows and shading to reveal clues about the surface or object on which they are formed. Where cast shadows reveal more potentially information about shapes of the object and about the spatial layout of the scene.

In Figure 3.8 the effect of shading is very impressive as it is the only cue shown in the 2d image, yet it provides the clear impression of raised surfaces appropriated for crumpled paper.

Figure 3.8 A picture of crumpled paper. Illustrates how shading provide clues about the raised surfaces. (Matlin & Foley, 1997)

Both static and dynamic cast shadows shows to be great cues for shape and spatial layout. This is of great importance for AR applications, as they in most cases will be creating dynamic shadows. A classic example to show the effect of shadow displacement on the perception of relative depth is shown in Figure 3.9. The closer an object is to its cast shadow, the closer it appears in depth to the background surface. Another finding is that cast shadows in many cases are more likely to make the observer perceive motion and overriding conflicting cues that suggest that the square is stationary; the lack of any change in size of the square and the lack of any 2d translation of the square.

Figure 3.9 The closer an object is to its cast shadow, the closer it appears in depth to the background surface. (Kersten, et al., 1994)

In a dynamic scene the motion of an object that casts a shadow on a background surface, will cause the cast shadow to move in the image. The motion of the shadow therefore, provides information about the relative motions of the object and the background surface. This is illustrated in Figure 3.10.

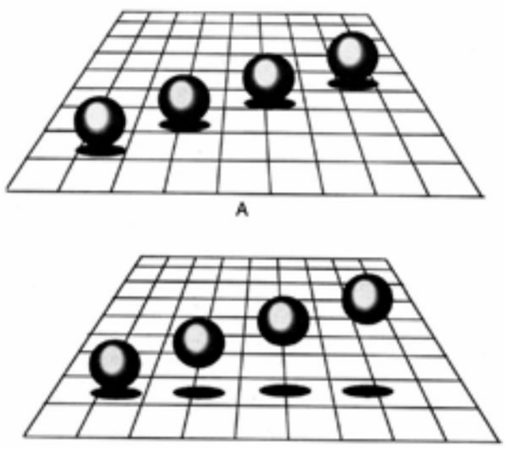


Figure 3.10 Relative motions of an object, the ball, and the background surface, in accordance with the movement of the cast shadow.

3.1.3 Conclusion on Shadows in Augmented Reality

This section has shown different aspects for getting virtual objects, which are augmented onto a real environment, to look as if they are actual part of the scene. Shadows have shown to be of great importance in achieving this realism in augmented reality. At the moment no easy method to find and calculate a dynamic lighting condition in an environment used for augmented reality. If such a solution could be achieved it would greatly enhance the realism, possibilities and usage of applications in the field of augmented reality.

4 State of the Art Technology

This section will describe the state-of-the-art technologies that are available at the moment.

The first part of the section will be a guide through different technologies that are common known at this time, which makes it possible to generate consistent illumination in mixed reality. The section will try to describe the overall concept of the different technologies and then describe which open problems and limitations exist for each of them.

The second part of the section will describe various papers and works that have been published in the field of shadows generation, shadow detection, and light source detection. A description of which technologies that have been used, what limitations they have, and what could be used in this work.

The third part of the section will be a discussion and summarization of the findings, and will also describe some of the approaches to the problem that could be used.

4.1 Part1 - Different technologies to calculate a real scene illumination

4.1.1 Image based Lighting (IBL)

Based on a tutorial about image-based lighting (Debevec, 2005a).

IBL is the process of illuminating scenes and objects, real or synthetic, with images of light from the real world. When used effectively, IBL can produce realistic rendered appearances of objects and can be an effective tool for integrating computer graphics objects into real scenes.

To use IBL you obtain a measurement of the real-world illumination, also called a light probe image. Light probe images are photographically acquired images of the real world, with two important properties. First, they're omnidirectional. For every direction in the world, there's a pixel in the image that corresponds to that direction. Second, their pixel values are linearly proportional to the amount of light in the real world. That is why high dynamic range (HDR) images are used.

There are different approaches to acquire a light probe of the real scene. You could use a 180 degree wide angle camera to capture the environment. Alternatively, a highly reflective sphere could be used. As the sphere reflect all incoming light from the scene, it is captured in an HDR image from all angles by a camera.

There are a few but significant limitations when using image based lighting. It is time consuming to construct a light probe to use as an environment map. Secondly, the approach makes use of special hardware, so it could not be done by an average consumer. Lastly, it does not work with changing lighting conditions, and a new measurement of the scene illumination must be made each time the lighting changes. At the same time the light probe is a measurement for one, and only one point in the scene, which means that moving virtual objects to a different point in the scene could create bad lighting. Of course this problem gets bigger when light sources are close to the object, like an in-door scene, and isn't affected as much in out-door scenes where the sun and sky are the light sources.

4.1.2 Detecting shadows using a known pre-defined object

It is possible to estimate light sources in an environment by tracking shadows cast by a known object, placed within the frame of the camera. These objects come in many forms and sizes, but common for them is a known geometry and size. Light sources can be estimated by looking at the shadow characteristics cast from the object.

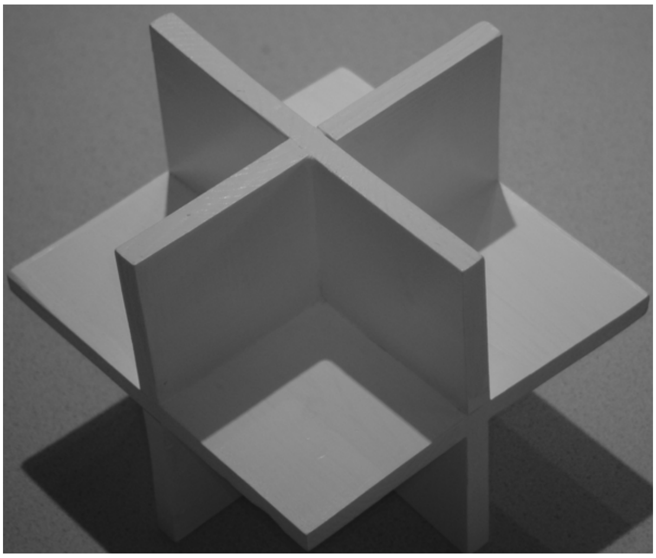


Figure 4.1. The Shadow Catcher object to detect shadows (Hartmann, Zauner, Haller, Luckeneder, & Woess, 2003)

For this method to work, the predefined object must be built. Furthermore the methods typically only support one light source with fairly hard shadows (Hartmann, et al., 2003). Figure 4.1 shows the Shadow Catcher used in (Hartmann, et al., 2003) to track shadows in the scene.

4.2 Part2 – Description of different papers and their technologies

4.2.1 Using Real Shadows to Create Virtual Ones (Claus B. Madsen)

The paper tries to come up with a solution to detect shadow regions in images of natural scenes, without the use of special-purpose objects, like cameras or reflective spheres. The shadow detection can distinguish between directly lit areas, penumbra areas, and umbra areas. Furthermore the paper describes how this leads to the creation of virtual shadows which appear very similar to real shadows.

To detect shadows the system uses color segmentation to extract objects from an image. This is done by an interactive training system, where the user selects a small rectangular area, for example a table. The object is then analyzed to detect shadow regions, where the system statistically and automatically is calculating two thresholds from the intensity histogram of each color segment, thus dividing the pixels into three intensity categories: umbra, penumbra, and directly lit. In this way the system knows where shadows are cast, and what type of shadow it is.

The information about the real shadows in the image is used to create virtual shadows by using alpha blending ("Alpha blending," Wikipidia). The paper does not address how to compute where to place virtual shadows, but only looks at how to change the spectral properties of an image area so that it looks like a real shadow.

Though the paper does not mention in which environment the system works, it is assumed that the objects that are to be extracted like a table, must have similar colors.

Future research involves developing methods for estimating the light source positions. Another open discussion is whether it would be possible to use a statistical approach to find pixels that are similar in color, like a table, without the need of user interaction.

4.2.2 "Shadow Catcher": A Vision Based Illumination Condition Sensor Using ARToolkit (Hartmann, et al., 2003)

This paper deals with the problem of creating consistent illumination in real time mixed reality. This is done using a so-called "sensor geometry". The sensor geometry consists of three squares that are mounted orthogonally to each other. This geometry allows it to use each square either as a base plate, where shadow patterns can be observed or as two adjacent shadow-generating plates. The Shadow Catcher approach is based on the idea of using texture changes that are caused by illumination changes to detect the position of the light sources. This is done by using the knowledge about the geometry's shape, together with information about the shadow silhouettes.

The approach works perfectly for well controlled indoor setups but fails for outdoor applications. The position of the geometry is known by the system, and ARToolkit is currently used to track the position of the sensor geometry, though the solution is independent of the underlying tracking system.

The canny edge detection algorithm (Canny, 1986) is used to enhance edges within the rectangle containing visible sensitive areas. A Hough transform (Duda and Hart, 1972) is used to detect lines that form the shadow silhouettes. The 3d position of a shadow corner and the 3d position of the vertex that causes this corner can be used to determine a ray that contains the light-source. The intersection of two of those rays is the position of the light source. The proposed method strongly depends on clearly interpretable shadow patterns that the sensor casts on itself. Thus, the method relies on a clear view between the sensor and the light source.

4.2.3 Estimation of Light Source Environment for Illumination Consistency of Augmented Reality (Feng, 2008)

The paper introduces two spheres with Lambert surfaces. Through image analysis and analytical geometry of three dimensions, the illumination parameters of the real scene are estimated automatically. These parameters describe a point light source and ambient light.

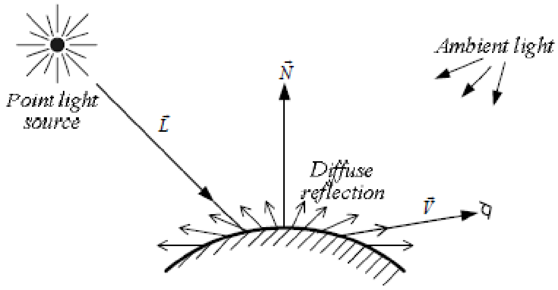


Figure 4.2. Illumination of a lambert surface

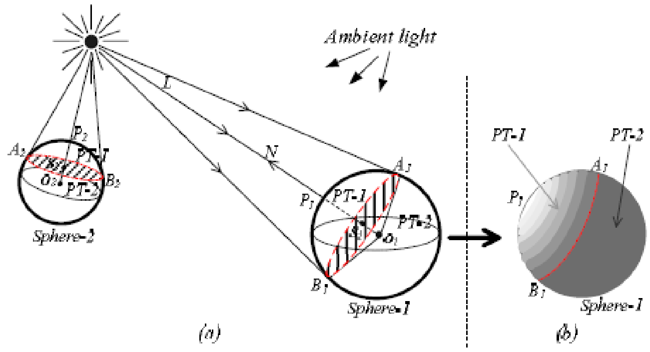


Figure 4.3. Illuminated multiple spheres

The marker spheres have a Lambert surface which means each surface point appears equally bright from all viewing directions *V,* as illustrated in Figure 4.2 and Figure 4.3. A point on marker sphere is called a *Special Point* if the surface normal at it, is parallel to *L*, the illumination direction of point light source. The luminance at the Special Point is higher than at any other point, that is, the Special Point appears the brightest on the sphere. Some light from the point light source is tangent to the spheres at some points. All the points of contact compose a circle and the two spheres are respectively divided into two parts by ʘS1 and ʘS2 . Let PT-1 be the one part the Special Point is in, and PT-2 the other one. The points across the circle S_i (i=1, 2), namely the points in *PT-2*, have common brightness because they are illuminated only by ambient light. Owing to the characteristic of point light source, *PT-1* is less than half a sphere, that is, *PT-1* is only a smaller part of the whole sphere, and its area is less than the one of *PT-2*.

The real scene including the marker spheres is captured by two cameras mounted on video see-through HMD (head-mounted display). After segmenting the two spheres from video sequences, the Special Point on spheres should first be detected. As for the images of marker spheres captured by cameras, there are three different cases as follows: For the first case, the Special Point is included in the image. For the second one, the Special Point is not included in the image, but the image includes some points of *PT-1* and of *PT-2* simultaneously. The third case is when the camera only captures the points of *PT-2*.

If the images belong to the above first case, the Special Point will be included in the images. As for *sphere-1*, after detecting the Special Point P_1 in the left and the right image respectively, we can calculate the coordinates of P_1 using binocular vision technique. The line O_1P_1 can be represented by the line equation. As for *sphere-2*, the operation is the same. By combining the two line equations, O_1P_1 and O_2P_2, the solution is the coordinates of the point light source. For the other cases the paper also describes mathematical approaches on how this is done.

Future work should focus on study of the properties of an arbitrary light source, such as multiple light sources.

4.2.4 Automatic generation of consistent shadows for Augmented Reality (Jacobs, et al., 2005)

This paper presents a method for creating consistent shadows for augmented reality. The scope of the project is to detect and protect the original shadows and generate virtual shadows for the augmented 3D models that do not overlap with the existing shadows.

The approach is three steps consisting of shadow detection, shadow protection and shadow generation.

In order to find the existing shadows, an approximate 3D model of the scene is used, along with a single major light position. By using this information, a virtual shadow is generated, and from that a mask, which is used with canny edge detection to find the real shadows. A scaling factor is also calculated at this point, to be used when comparing the color of the real and virtual shadows. The more precise light position and 3D reconstruction, the better the result.

Next a protective mask is placed on the shadow to avoid further rendering to the area. The virtual shadow is then rendered, and the color factors of the overlapping and non-overlapping shadow areas are compared to the scaling factor to give a result closely related to the real shadows. For generating the shadows, shadow map or volumes are used.

In all testing the method ran in real-time, with 30 fps.

For further development, the authors proposed scaling the method to more complex environments, and the algorithm made more robust to any types of shadows, and several light sources.

4.2.5 Real-time Estimation of Light Sources Environment for Photorealistic Augmented Reality (Kanbara & Yokoya, 2004)

The paper describes a method for estimating light sources in an environment relative to a 3d marker using a reflective sphere and how to use these findings to simulate shadows on a 3d object in real time. Two problems are addressed in the paper: estimating the position of the light sources and the user's viewpoint.

Estimating the position and viewpoint of the user is done using a vision-based method: Images captured by a camera attached to the user's viewport. Estimation of the viewpoint is done by tracking a 2d square marker placed within the frame of the camera. By finding the four corners of the square and knowing the distance between these points, the plane in which the 2d marker is positioned as well as the relative distance to the marker can be estimated. The relative tilt, position and distance can be retrieved and a 3d object can be rendered accordingly.

The paper also describes a method for estimating the light sources using a reflective ball placed on top of the 2d marker. Using the information found by tracking the marker, the reflective ball can easily be tracked in the center of the 2d marker. Knowing the surface normal's of a ball and the camera position, the direction, intensity and color of light sources can be estimated by looking at the corresponding pixel values for the reflective ball. To avoid dynamic range problems when tracking the light sources, the reflective ball is painted black.

The method described in the paper is implemented and tested on a 2GHz, 256MB memory computer using a Sony DFW-LV500 camera and achieved a frame rate of 20 FPS and provides a photorealistic augmented reality in the same lighting environment as the real scene. But since the region size of the mirror ball in the image is limited, the directions of light sources is estimated separately which result in the shadows being rendered with discontinuities.

4.2.6 Estimating Position and Radiance of a Small Number of Light Sources for Real-Time Image-Based Lighting (Claus B. Madsen, Sørensen, & Vittrup, 2003)

Realistic 3d objects rendered in real world images can be obtained by using image-based lighting, achieved by using an omnidirectional image to represent the irradiant light at the virtual object's location. This method typically creates thousands of tiny light sources to simulate the lighting in the real world.

The paper describes a method for estimating these lighting conditions using only a small number of light sources which makes it feasible to create real-time rendering of the objects. To do this, a sphere mapped with an omnidirectional image of the scene is made. Inside this sphere a smaller white sphere is created, and the radiance of the large sphere is now rendered on the smaller sphere like in classic IBL.

The small sphere now holds information of all the lighting on the current position in the image. The lighting on the small virtual sphere is now estimated on another sphere using only a user defined number of light sources. The result is a small number of light sources that create approximately the same lighting as the thousands of light sources used in classic IBL.

The result is a realistic rendering of a 3d object in a real world environment in real-time at around 100FPS using a GeForce 4 graphic card. Comparison with classic IBL techniques shows almost no visible differences.

The system needs a user defined number of light sources, but the authors believe that it can be further improved by letting the computer calculate the optimal number of light sources, based on a threshold on allowed shading errors.

4.2.7 Segmentation of Soft Shadows based on a Daylight- and Penumbra Model (Nielsen & Madsen, 2007)

This paper describes a method for segmenting soft shadows in an image. A model is described, which accounts for sunlit, umbra and penumbra regions. The mode estimates shadows in the images and removes those shadows using an alpha overlay. The alpha overlay is estimated by analyzing the umbra and penumbra regions and applying a graph cut energy minimization.

The paper assumes only one major light source in the environment, in an outdoor daytime scene. Moreover, the model makes a series of assumptions in order to work:

- Narrow-band, delta function, sensitivity
- Linearity or simple gamma correction which does not change the log illumination direction
- The log illumination direction must be known through calibration
- Angle of surface and ambient occlusion are not taken into account

The results in the paper are found with experiments using both a professional grade camera, Canon EOS-1 Mark II and a consumer grade camera, Minolta Dimage 414.

Umbra were removed completely, however, the model sometimes over- or underestimated the penumbra regions. More of the penumbra is segmented away in the non-linear consumer camera images, but there were number of false-positive segmented as well.

According to the paper, the model can be used to model sunlit areas from shadow regions, as well as shadows from sunlit areas.

For future work, the results also demonstrated that it may be possible to extract information of the sun position, given simple geometry knowledge.

4.2.8 Estimation of Dynamic Light Changes in Outdoor Scenes Without the use of Calibration Objects (Andersen, Jensen, & Madsen, 2006)

This paper describes a model for estimation of illumination in an outdoor daytime environment. The model makes use of existing inverse rendering techniques is used to acquire diffuse surface reflectance in an offline procedure. This reflectance model is then used in an online procedure to estimate illumination in an outdoor scene.

The model reduces the light estimation problem to a reverse Phong model with only two unknown parameters, which can be determined using linear equations. However, in order for this to be reduced to a reverse Phong model with two unknown, the following constraints and assumptions are made:

- Outdoor only, during daytime. Only one major light source
- Predefined diffuse surfaces
- HDRI environment map
- Known sun position
- Simple 3D model with most significant surfaces

The illumination estimate is then able to analyze images by taking 500 randomly chosen pixels of the image. In order to determine the shadow area, a shadow map must be computed for every frame, to get a more correct sample reading.

This paper does not describe and further development or problems to be handled.

4.2.9 A Survey of Real-time Soft Shadows Algorithms (Hasenfratz, et al., 2003)

Shadows are important visual cues about the position and size of an object and its shadow object. This visual effect can be realized in real time simulations. The approach is however crucial to both achieving credible shadows while taking frame rate into consideration.

The paper is a good introduction to a general overview real-time soft shadow algorithms, from 2003. The analysis have picked 9 algorithms from various authors and implemented them and analyzed their quality and their achieved frame rates. There is a good description of each implementation but there is no exhaustive algorithmic analysis that could conclude what the limiting behaviors of the implemented algorithms are. The primary implementation technique for all implementations are shadow-buffer based, the buffer is manipulated by the various algorithms that are either image-based or geometry-based.

By 2003 standards in graphics hardware, it is possible to render realistic shadows onto planar surfaces by various means. The visually most impressive is the soft shadow volume, which is implemented - but this approach requires, as all other geometry-based a priori knowledge of the objects in the scene or that the algorithm itself can segment the contents of any given scene. The team expects that soft shadows will soon become standard in real-time rendering.

4.2.10 Illumination from Shadows (Sato, Sato, & Ikeuchi, 2003)

This paper introduces a method for estimating shadows by investigating the relationship between incoming light and light in shadows.

The method recovers the illumination distribution from image brightness inside shadows. These shadows must be cast by an object of known size and shape. By looking at light inside shadow areas, an estimate of the illumination distribution can be made.

Assumptions for the method are:

- Known geometry
- Distant illumination, parallel light rays
- No interreflection

The method uses a method for relating illumination radiance with image irradiance, and approximates the illumination distribution. The radiance distribution is estimated based on reflectance properties of shadow surfaces.

The method is tested using known and unknown surfaces. In both cases the estimate of shadows are correct for a high number of samples.

The authors mention no future work.

4.2.11 Image Based Shadowing in Real-Time Augmented Reality (Supan, Stuppacher, & Haller, 2006)

This paper presents an approach for rendering shadows with image based lighting. The approach is real-time for Augmented Reality.

The approach uses a system with dual camera setup, with one camera focusing on a marker for tracking, using ARToolkit, and one focusing on a reflective sphere. Based on the input from the reflective sphere an environment map and an irradiance map are generated.

Shadowing is then performed using a reasonable amount of light sources around the scene to cast light sources.

The prototype creates shadow volumes from the light sources, but since rendering is expensive only one shadow is updates per frame.

The test results show the system is running at 35fps with 0 shadows and 15fps with 64 shadows.

The authors propose to investigate more advanced rendering techniques for use with other surfaces.

4.3 Part3 – Discussion and approaches to the problem

The research and development in the fields of augmented reality, shadow detection and generation is very advanced and diverse in approaches and techniques. However none of the above mentioned techniques are implemented in such way that they are usable for the casual user of augmented reality. Of the techniques described, all but a few require substantial preparation and are very reliant on the right requirements in the environment in order to be effective.

A simple and effective technique is sought to implement shadows that gives a convincing and reliable result. All the augmented reality trackers would be greatly enhanced if the implementation could accommodate shadows without straining the host computer or mobile device significantly. A minimum criteria for tracking of shadows given any of the techniques implemented is important to be established, hence it can be used to determine if the shadows can be generated.

An interesting approach would be to establish a threshold for how accurate the shadows must be in order to create the illusion of realistic shadows. This survey is using State-of-the-art analysis, HDR techniques and renders different shadows with varying positions according to parameters from absolute accuracy to inaccurate. The survey needs to be designed so the parameter for the threshold for acceptable shadows is established in a sound scientific manner. This threshold will be established with a test.

5 Establishing criteria for realistic virtual shadows

The test will seek to establish the number of light sources needed to provide sufficient realism in augmented virtual shadows. Using a heuristic derived from the test, the sufficiency of realism for shadows ensures that there is a sensible benchmark for the lower bound for realistic shadow. The test will also seek to find if it is possible to enhance virtual shadows by applying visual effects. It is assumed that an effective shadow generation algorithm can be produced, by finding an optimal compromise between realism and performance.

5.1 Test hypotheses

- It is possible to create computer generated virtual shadows, that the user wouldn't rank as less credible than genuine shadows.
- By using a different number of light sources to create virtual shadows, it is possible to find a lower bound of light sources necessary to make the shadows believable to the user.
- By using visual effects, like blurring the virtual shadows, it is possible to lower the lower bound of necessary light sources even more.
- By using visual effects to reduce the number of necessary light sources, the calculation time needed to create virtual shadows can be reduced.

From the hypotheses, two main problems for the test can be composed:

1. Determine the light sources necessary to generate sufficient shadows.
2. Determine whether the use of visual effects can enhance credibility of virtual shadows.

To find an optimal compromise between realism and performance, the actual system must be developed. Therefore this compromise should be sought in the technical test of the system. This compromise will also be very system-dependent.

5.2 Target audience

To identify which test setup would be most suitable for the problems addressed, the target audience must be identified.

The direct user of the final product will be the software developer, who probably has a high level of technical knowledge. It is likely he will have more or less understanding of shadows, how they work, and how they should look. The developer therefore probably will find it easier to detect imitated shadows than the average person.

However, the direct user is not the audience the test should be designed for. This is due to the fact that the developer is merely developing applications for other users. These applications could be commercials, simulation, entertainment, games and so on. All these applications have their own target audience which could range from a shadow expert to a teenage pop girl.

These persons are the indirect users of the system, and they are the audience the test should be designed for. Most likely the majority of this audience does not necessarily have any knowledge of shadow generation. Moreover, this kind of application would probably not be targeted for old people, as they have not appropriated modern technology in the same degree as younger individuals. Therefore the average person is the target audience for this test.

5.3 Test objects

The main focus of the test is virtual shadows. In the final system the virtual shadows will be part of a moving picture, which will run in real time. As no product or system is available at this time, a method to simulate the final product must be made. To simulate this, a number of small footages with augmented shadows created by a different number of light sources could be produced. However this demands a lot of time in crafting the different footages. To reduce production time, still images were used instead.

To create the images, a picture of a real scene was taken with and without an object that is easy to model in a 3d program. By inserting different number of light sources into the 3d program, virtual shadows can be calculated and generated. Where to place the light sources was calculated using an HDR environment map of the scene and using the Median Cut algorithm (Debevec, 2005b). The images were then composed in Adobe Photoshop ("Adobe," 2009) combining the picture without the object, the object segmented from the scene without shadows, and the shadows generated from the 3d program.

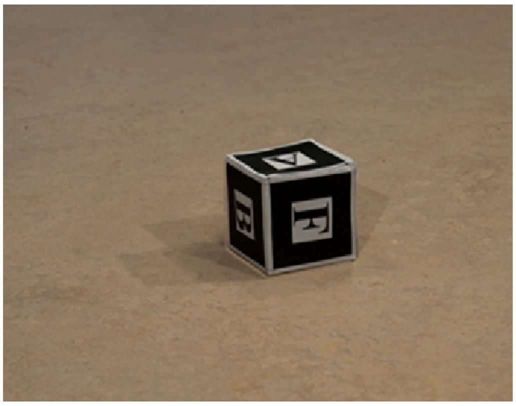


Figure 5.1 An example of a real image with superimposed virtual shadows created with 16 light sources, using Median Cut and Adobe Photoshop

To determine the range of light sources needed to do this test, a heuristic evaluation was made by the group. By looking at different images composed with different numbers of shadows, it was found that the use of more than 256 light sources would be unnecessary. Due to the nature of the Median Cut algorithm, the number of light sources that can be created has to be a power of two.

5.3.1 Test setup up and different settings

Choosing images from only one environment could easily lead to test errors, or results that only work for a given environment, and therefore have usage limitations. To compensate for this error, images from three different settings were composed.

1. First setting features an environment with one object placed on the floor and light generated from two large fluorescent lamps.
2. Second setting features an environment with three objects on a table and light generated from one major light spot, and some indirect light from the surroundings. Only the middle object, a book, has augmented shadows, while the two other objects have untouched shadows.

3. Third setting features an environment also with three objects, but the light is generated from more than 15 light spots and some large windows, all placed on the ceiling in a large hall. Only the middle object, a box, has augmented shadows, while the two other objects have untouched shadows.

Figure 5.2 Example images from 3 different settings for use in the test. All with superimposed virtual shadows created with 256 light sources. In a) and b) only the middle objects have virtual shadows

These three settings together should give a fair amount of different setups, featuring different light settings and different sized objects. Having settings featuring objects that cast real shadows, it is presumed that it will be easier to detect the imitated shadows of the object in the middle. On the other hand, environments including real objects are most likely to be used with the final product, and the settings are therefore considered to be more realistic for test setups.

From this it can be concluded that the test set is composed of 10 images with augmented shadows, [0,1,2,4,8,…, 256] light sources. Also 1 image with real shadows should be used as a control. This gives a total of 11 images in each of the three different settings.

5.3.2 Test images with visual effects

As stated in the introduction for this test, it was also the intention to test whether visual effects could enhance the credibility of the virtual shadows. A major problem by observing imitated shadows is the fact that very sharp edges are visible. To eliminate sharp edges a blur should be very effective. This gives the test an additional set of images which is composed from the same set as before, but with a box blur effect applied to the shadows. A box blur was chosen, because it is easy to implement and is much less computationally expensive than for example a Gaussian blur ("Box blur," Wikipidia).

Figure 5.3 a) Image with superimposed virtual shadows created with 16 light sources b) Same image as a but with a box blur applied

As the images with a low amount of light sources have very sharp and edgy shadows, and the images with a high amount of light sources have much more soft and transparent shadows, the size of the box blur should differ with the amount of light sources. The size of the box blur filter was again chosen using a heuristic evaluation by the group of what looks the best.

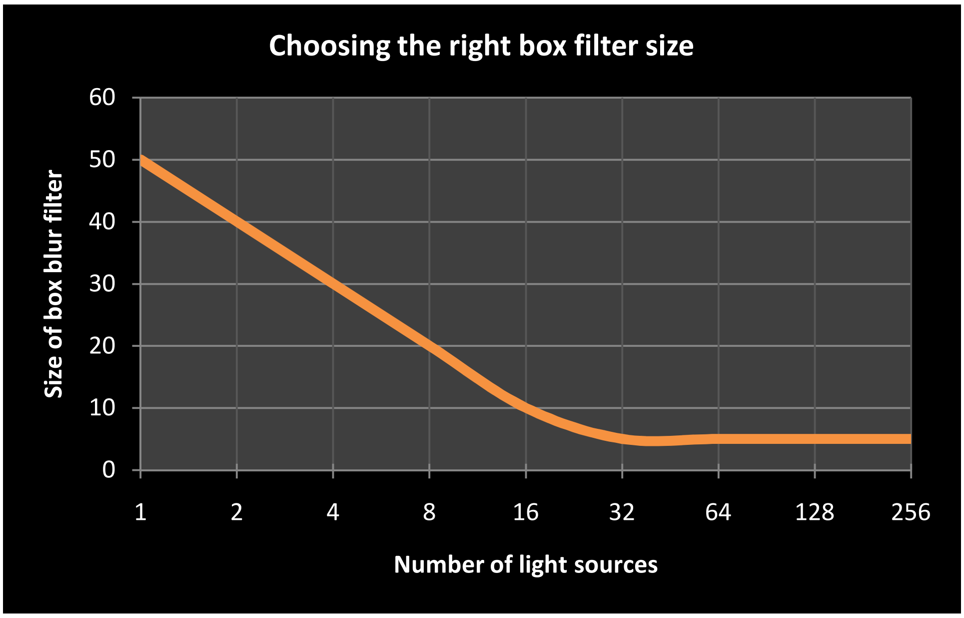


Figure 5.4 How to decide filter size of box blur depending on the number of light sources used

To sum up the test now includes the following images for each of the three different test settings:

- 10 images with augmented shadows using [0, 1, 2, 4, ... , 256] number of light sources
- 10 images with augmented shadows using [0, 1, 2, 4, ... , 256] number of light sources, with box blur applied
- 1 real image as reference

This gives 21 images in 3 different settings, giving a total of 63 different images to test on.

5.3.3 Qualitative vs. quantitative

The data produced by tests can be of qualitative and quantitative nature. Qualitative investigations focus on trying to inspect the users' understanding of a specific thing, what they think of it and what it means to them. Qualitative analysis can also cause the discovery of useful unexpected information about a certain topic. Quantitative analysis focuses on the measurable impact a specific thing has on the reality, that is, if there are any tendencies or patterns caused by the thing that are being tested. Quantitative data is most useful for statistical research and analysis. The main difference between quantitative and qualitative data is that quantitative data is measured and gives a numerical value while qualitative data does not give any numerical values and usually involves the participant giving qualitative judgments. (Sharp, Rogers, & Preece, 2007, p. 536).

As this test seeks to find patterns in the average user's perception of virtual shadows, quantitative test is valued over qualitative test. The next phase is to find a proper quantitative test method.

5.3.4 Test methods

There exist various different quantitative test methods, but not all of them could be used to show the correct information that is wanted from this test. As described the main goal is to find some kind of threshold for how many light sources is needed to make credible shadows for a user with average information about shadows.

Four different test methods were analyzed:

1. Two-alternative forced choice (TAFC)
2. Paired comparison
3. Rank order
4. Dose-response relationship

5.3.4.1 Two-Alternative Forced Choice

Based on the online book by Geoff Martin (Martin, 2004) and a website from Center of Image Science ("RIT CIS - Center for Imaging Science," 2009)

The TAFC test is usually used to test for the smallest difference in stimuli a target group can perceive. The test uses a method, were the subject is exposed to two stimuli of different size or strength. The user must choose between the two stimuli corresponding to a criterion, and does not have the option of not making a choice; i.e. he cannot choose "I don't know". The method could also be carried out the other way around where the subject is exposed to one stimulus and then has to make a choice of two different criteria. Since the participant always has to decide between two different choices, the acceptance rate ranges from 50% to 100%. This is a result of when the stimuli becomes too dim to detect the subject is simply guessing on the test subjects behalf. The percentage of correct responses is plotted against stimulus intensity.

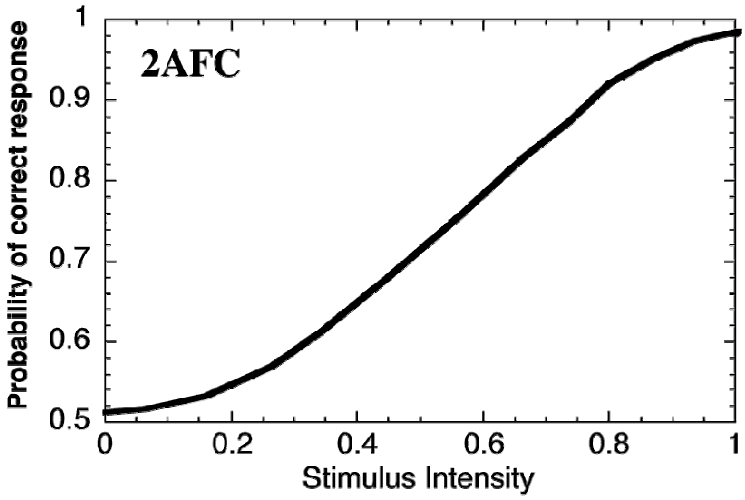


Figure 5.5 Show a typical graph from a two-alternative forced choice test. As the intensity in stimuli is decreased the accuracy of reaction tends towards 50%. When the difference in stimuli is 0 the reaction is naturally just a guess.

5.3.4.2 Paired Comparison (Martin, 2004)

Paired comparison is a good way of weighing up the relative importance of options. The TAFC and paired comparison tests are much alike in their approach. While the TAFC changes the strength of stimuli between each trial, based on previous answers, the paired comparisons analysis tries to compare all options against each of the other options. This gives a kind of ranking of the different options and the option with the highest score is the preferred option.

5.3.4.3 Rank order (Martin, 2004)

Another way to rank a number of stimuli is just to use the rank order test. The subject receives all stimuli at the same time and is asked to put them in order of least to most of some attribute. It is much easier for the subject to understand this type of test, but it also means that the subject can compare each stimulus against each other at any time. Another disadvantage is that the subject can be confused if there are too many different options. Also there should be a predefined set limit to the number of options should be available to the subject.

5.3.4.4 Dose-Response Relationship (Kane)

The dose-response curve can be used to plot many experiments and is used to show the relationship between an amount of exposure and the resulting response. It could be how rats are affected by poisons. The x-axis would be a logarithmic scale of the amount of dose the rats are exposed to, and the y-axis would show the response, for example if a rat died or not. Depending on the type of exposure and which response is measured, dose-response curves can have almost any shape. However, in very many systems dose-response curves follow a standard shape, shown in Figure 5.6.

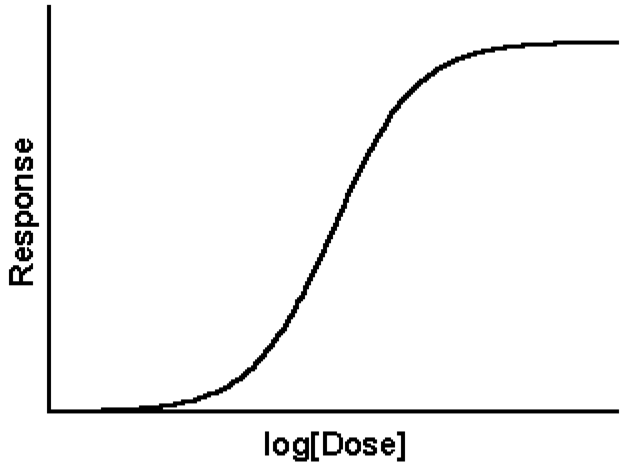


Figure 5.6 Standard dose-response curve showing the relationship between a given dose and its response

The relationship can be used determine how large a dose should be used to achieve a certain percentage-wise response.

5.3.5 Choosing the best test method

The TAFC test would be most useful if it was sought to find the smallest difference in light sources that is perceptible for a subject. Another scenario where the TAFC procedure would be useful is if it only was sought to find the number of light sources where a subject couldn't differentiate between a real and an imitated shadow. The test aims to find where the boundary for acceptable virtual shadows is, that is when a subject would not notice that anything is wrong, and therefore the TAFC test is not really the best choice for this test, because it does not give any insights into when there is no distinguishable difference between shadows.

Paired comparison and the rank test, in theory end up with the same result, as in both tests the different images could be ordered from least to most credible. However, a considerable difference is present. In the rank test all images can be reviewed against each other at any time, and it is likely that it is easier to spot the increasing number of shadows when such a comparison can be made. In the paired comparison test, only two images can be compared against each other at a time, thus eliminating some of this error. On the other hand the paired comparison test requires a large amount of comparisons to get all images from each set compared against each other.

A ranked order of each set of images would not give an exact result to when the numbers of light sources are sufficient for creating credible virtual shadows. This view is also a most subjective opinion as it is really just a "is the shadow good enough" question, and it would not be a clearly outcome from an ordered test set. On the other hand, the real image could end up floating somewhere in between some of the other images in the results. If this would be the case it can be derived that images that are ranked higher than the real image must be at least as credible as the real image. But the real image won't necessarily be placed in the middle of an ordering. It is more likely that the real image will have one of the highest ranking and therefore it wouldn't be possible to say anything about the other images. Even if it is possible to say something about how many light sources that are needed to simulate a real image, it doesn't say anything about how many light sources are needed for sufficient realism.

By taking a look at the dose-response curve, it seems to be a much more logical approach to finding what we are looking for in this test. If the subject is presented with one stimulus at a time, in this case an image from the test set, the response should be whether the subject finds the virtual shadows convincing or not. This would give a percentage of the population accepting a given number of light sources used to create virtual shadows. The plot of the dose-response could then be used to define how many light sources are necessary to satisfy a given percentage of the population and thus make it possible to make a reasonable compromise between realism and performance.

5.3.6 The making of an initial test

The test set contains 63 different images. To make a dose-response test, where all images are considered a reasonable number of times, either a lot of test subjects are needed, or each subject has to be tested on many different images. If the subjects are exposed for too many images, there is a possibility they will begin to know what to look for and thus be more aware of imitated shadows late in the test. To prevent this error in the test design the number of images shown to each subject will be limited.

A quick initial test was chosen to be made for different reasons:

- It acts as a pilot test for the final test, to secure that the test and test sets are reasonable.
- To make a quick test to see whether some setting or test set could lead to errors.
- With an initial test it would be clear if it at all is possible to truly imitated shadows
- A reduction of the test image sets could be made on the basis of the test.

A rank test was chosen as an initial test. The rank test is fast to perform and should make it possible to detect errors in the setup, the possibility to get qualitative feedback from the test subjects in person is also a valid parameter in the initial phase of the project.

5.4 Final test design

Test stages:

- The first initial test is a ranking of credibility of composited images. The test is used to find candidates to a subset for the subsequent test.
- The second final test is a dose-response scenario determining the percentage of the test population that judges an image as real or imitated. This test uses candidates from the first initial test.

Several data gathering techniques will be used:

- A rank order experiment where users sequence the test images in order of credibility (test stage 1)
- Online questionnaire where users determine whether the image is credible or not (test stage 2)

Safety and privacy:

The test subjects will be asked to consent to an agreement that states that:

- Subjects are anonymous.
- The data will be used for scientific purposes.
- That data about their behavior on the webpage they enter.

Limitations:

The test offers only a very crude and limited insight into the perception of shadows for human beings, but will give the developing team a marker for the success criteria of the application. The test itself and the specific approach can be prone to errors by design. The various approaches can themselves yield specific results and therefore the design of the test is critical to establishing a credible test result and conclusions. The problems in comparing shadow credibility lies primarily in the sequence of events that the test consist of. If the images are shown to the test subjects from low complexity to highest, or vice versa, and asked to find whether the picture is credible or not, can be very dependent on the prior images. This reference will most likely affect the test and subsequently the marker that defines the success of the implementation. Therefore the test is designed using a random combination that can satisfy that the sequence of events does not influence the test results.

5.5 First initial rank test

As stated the initial first test will be a sort of short pilot test to the second final test. This section will describe the hypothesis of the test, the exact approach and the results.

5.5.1 Hypothesis for the first initial rank test:

1. The ranking of the images will show when the test subjects cannot distinguish between composited images and a real image.
2. This will lead to a higher bound of light sources necessary to imitate shadows and show that it is possible to do so.
3. By eliminating perceptually similar compositions, a reduction of the test set can be achieved.
4. Reducing the set serves to make the test into a coarser but carefully chosen set that can be subsequently tested for their credibility.

5.5.2 Approach for the first initial rank test:

To speed up the test this initial test only using images without visual effects, i.e. images without blur effect applied to the shadows. The series consist of 11 images in three different settings. Each set contains 10 composited and 1 real control image to ensure that the subjects can distinguish real shadows from virtual. The images are presented to the test subjects to be sorted in order from least to most credible. To avoid confusion and too much comparison between images, the test is not carried out the usual way of a rank test. Instead the images are given one at the time to the subject during the test. The test subject is first exposed to two images and subsequently 1 image until the test set is empty. This image must be put into the sequence from least to most credible. This should prevent some of the comparison between images, though the subject will still have the opportunity to do so.

The results will be an ordering of rank from least to most credible, the recordings will be the indexing of each image set. Also the subject will be asked if they find some of the images convincing, that is, a subjective view whether they think they would have noticed any remarkable mistakes in the image if they had seen the image outside of a test.

5.5.3 Expected results for the first initial rank test:

An ordered set of images are expected from each subject, based on the test subject's personal evaluation. It is expected that the subjects will find the images in the sets more similar, as the number of light sources increases. In the ordered set, it is expected that a lower part of the images will be completely unreal and the upper portion will be more or less identical, and harder to separate from each other.

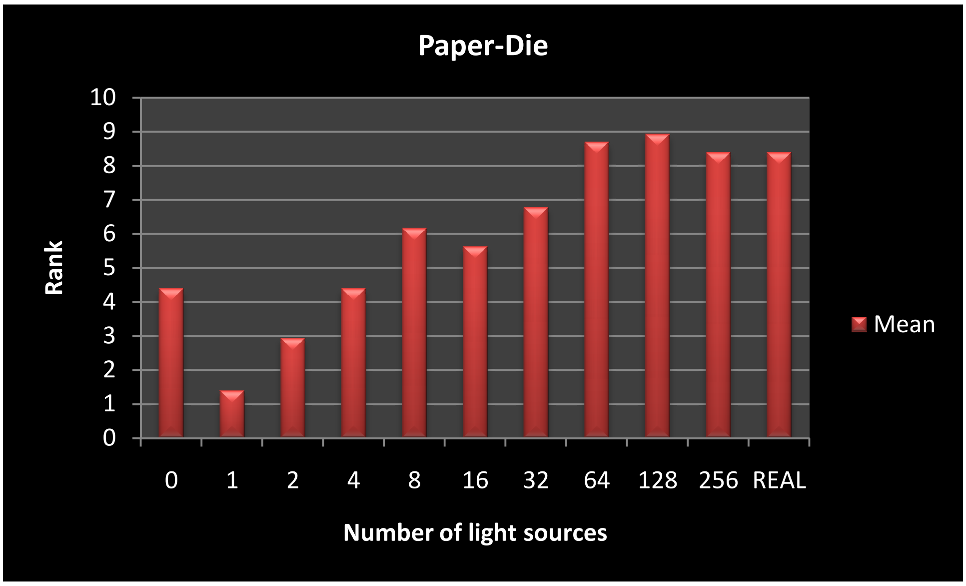
If any clustering in the rank test occurs, for example if people agree on a specific rank of two different images, it could be caused by an overlap, and maybe only one of the two images should be used in the subsequent test.

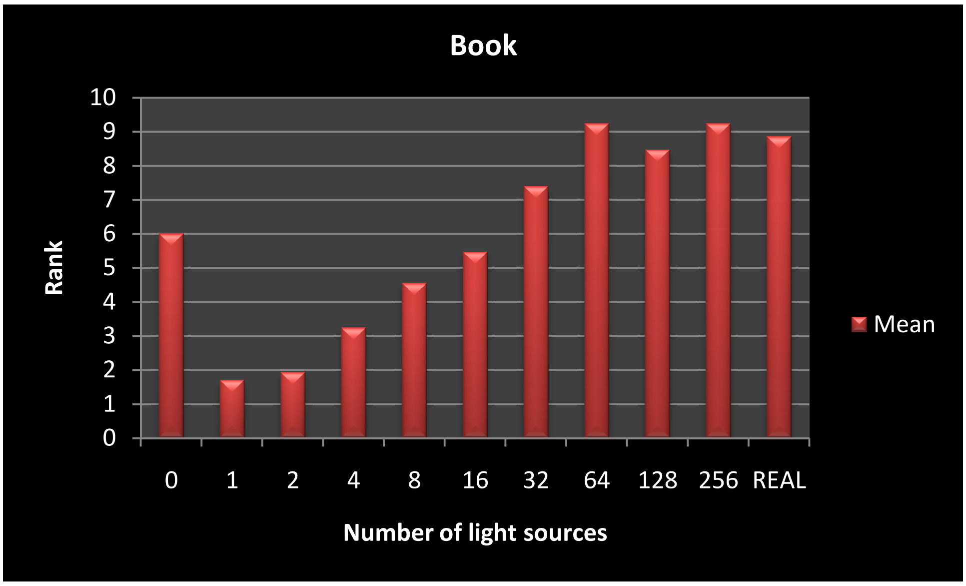
There will be three sets of data acquired, giving a probabilistic insight into the credibility of the different compositions. The questions and tasks will not vary in the different acquisition scenarios, this way it ensures that the answers can be consistently compared with each other.

5.5.4 Results for the first initial rank test

The initial rank test was performed on 13 randomly chosen participants. The participants were found in the cantina at Aalborg University and at Studenterhuset Aalborg. The answers from each participant and their respective ranking of the images were recorded in the rank order table in the appendix. The data was followed up by a statistically analysis, to find if any superfluous images could be held out from the final test.

In the three diagrams below data from each setting has been plotted using the mean value of the recorded data.





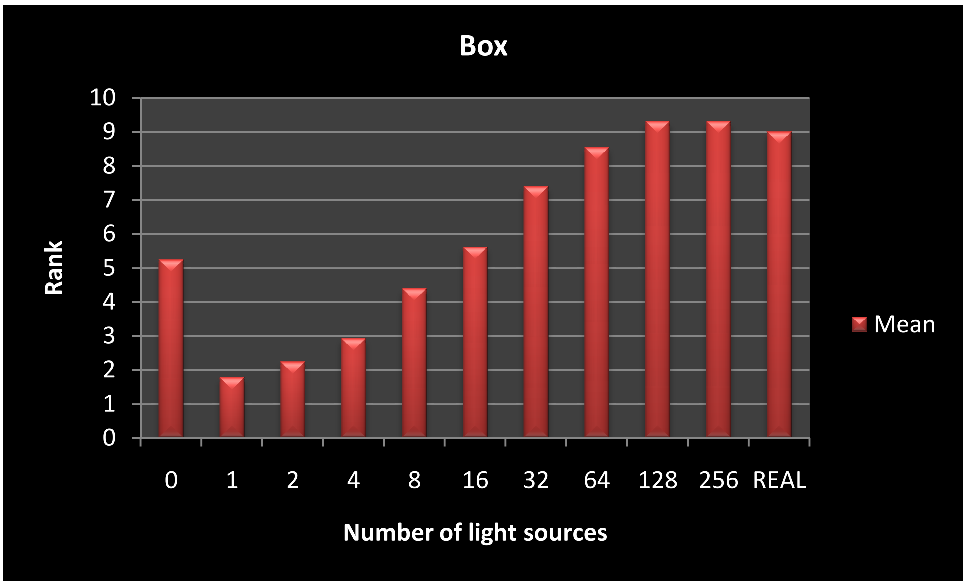


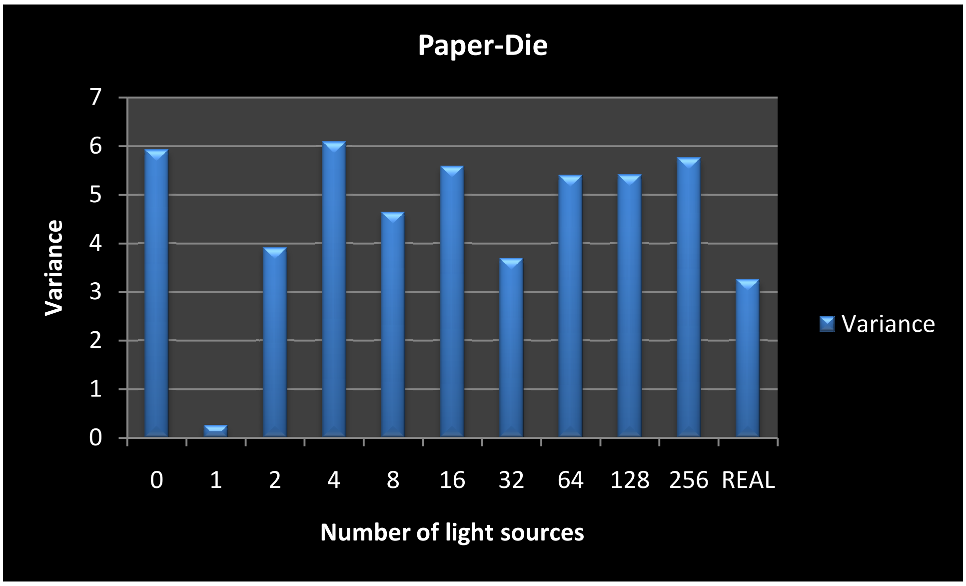
Figure 5.7 Average ranking of images in each test set, in each different test setting

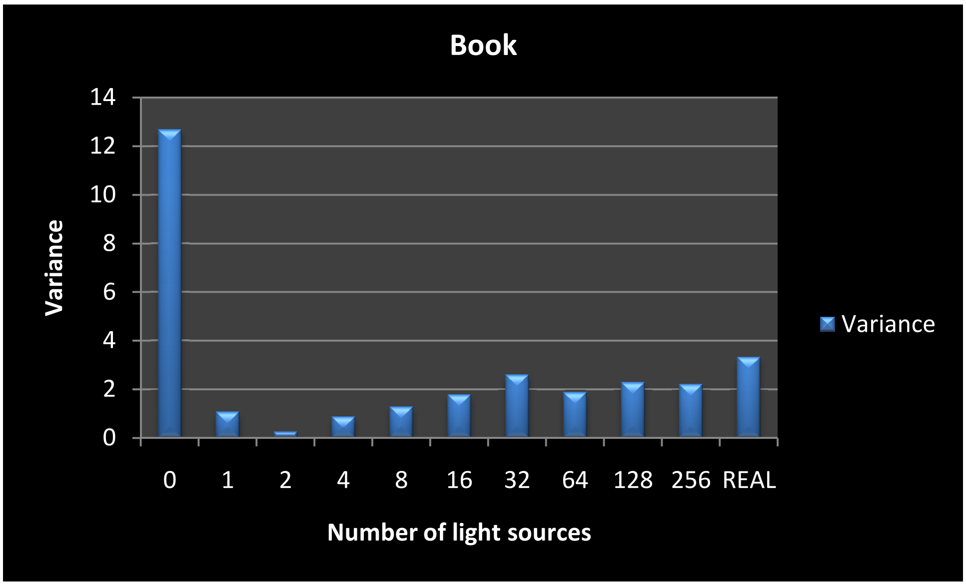
In all three settings, the diagrams indicate that it is possible to imitate the shadows of objects, as the real image, in average is placed below the ranking of some of the imitated images. Also, as all images from 64 light sources up to the real image, scores about equal, this could indicate that 64 light sources would be enough to satisfy a user.

If we look to find any clustering in the results, it is clearly that clustering occurs in the upper part of test images. That goes for all three settings. A closer look on the first setting with the paper-die, actually gives the kind of clustering that the test was supposed to show. The diagram for the first setting gives about three clustering; an upper part which contains 64 light sources and up, including the real image, a middle cluster which contain 8 light sources up to 32 light sources, and then the last cluster which contain the remaining lower end of images. The image with 0 light sources seems to fall into some part in the middle.

But the clustering only holds true for the first setting. The two last setting have a much more smoother curve of ranking from 1 light source up to 32 light sources, but yet again we see some clustering in the upper part of the images.

This behavior could be explained by looking at the variance of the answers, and the diagrams below shows some interesting results.





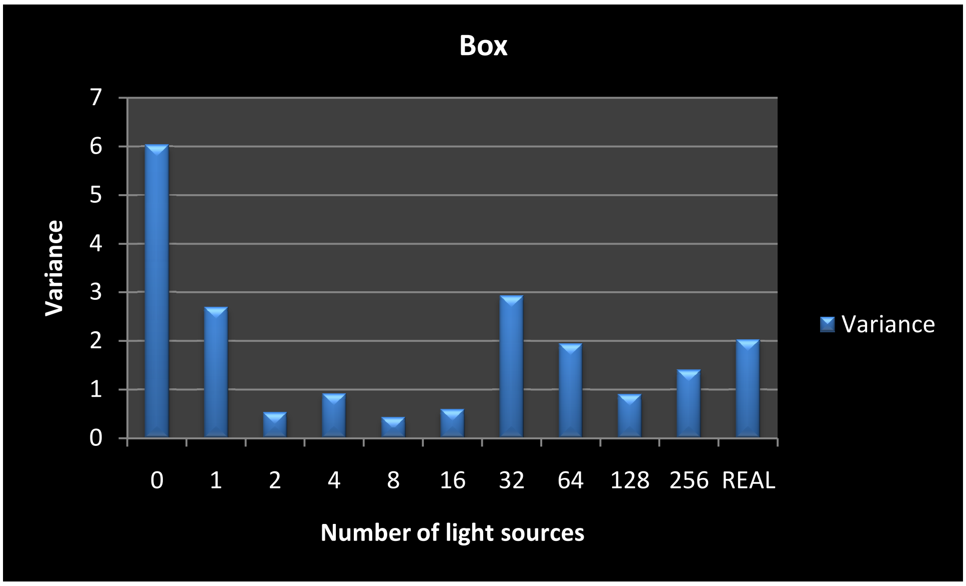


Figure 5.8 Variance of answers of images in each test set, in each different test setting

The variance indicates several things. First of all it is indicates that people have a lot of difficulties placing the image with 0, no light sources; the variance is quite high for this particular image in all three settings.

Second, the variance of the first setting with the Paper-Die is high for almost every image in the test set. This could probably be caused by two reasons. It could have something to do with that fact that it is the first set the test subject are introduced for. Hence they had easier to rank the images the second and third time, because they now know the assignment better and know what to look for. But the assignment should be quite easy to understand and accomplish. Therefore the most obvious reason would be that the test subject was confused because the first setting didn't contain any objects of reference. When there is no object of reference to compare the shadows against each other, it would sound reasonable that it would be harder to rank the images.

Because of this rather scattered ranking of the first setting, this section will continue to focus only on the last two setting. The variances of these setting are quite low, which indicates that people have been quite agreed with the ranking. That of course, is except the image with 0 light sources which people still have problems of defining.

Results from the two last settings are combined to get a more average look of people's opinion. This gives the following mean values when plotted into a diagram.

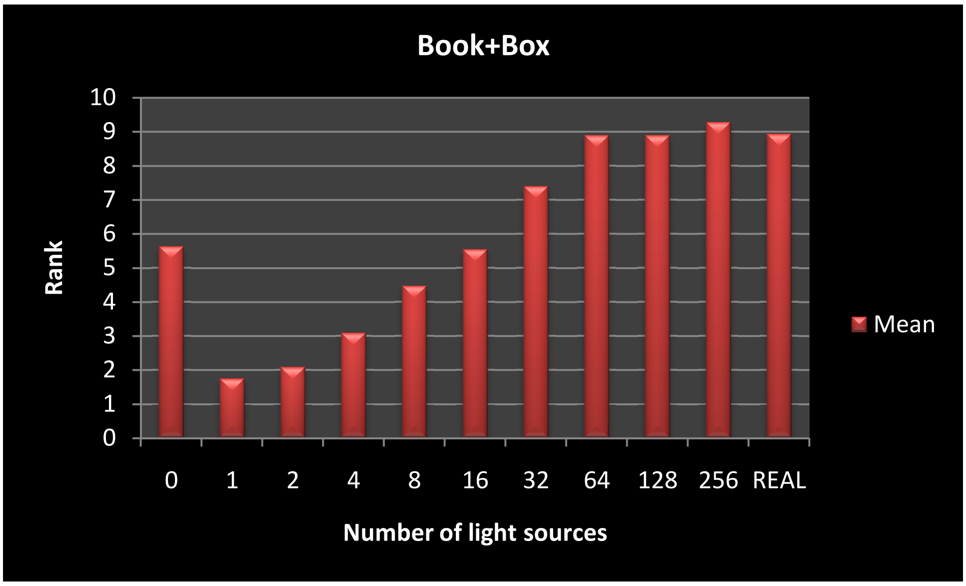


Figure 5.9 Test setting 2 and 3 combined average ranking of each image

Now a more clear behavior of people's perception emerges. The initial test has showed us that the first setting, when no objects of reference are present, confuses the subjects and make them unable to make a clear descision. As such this setting are discarded for the final dose-response test.

Also as mentioned the image with 0 light sources are discarded because of the high variance value in the initial test, which indicates that people find it hard to place in the ranking.

In the upper end of the images there is a clear clustering, which includes the images with 64, 128, 256 light sources and the real image. As all these images scores about the same score, the conclusion is that 64 light sources can be as good as the real image. This was also the impression when participant were asked qualitative of which images looked as real images. Many of the participant replied that those with shadows generated with about 64 light sources looked like they could have been real. This is also approximately in accordance with a paper written about perceptually correct virtual shadows (Nakano, Kitahara, & Ohta, 2008), which states that the use of 128 light sources is sufficient. It should be mentioned that they didn't test on 64 light sources, but jumped right from 32 light sources to 128 light sources.

5.6 Second final dose-response test

The second test will use candidates from the initial test to make a dose-response relationship between the amount of light sources used to create virtual shadows and the acceptance rate. The purpose is also to determine the subjects' sensitivity to shadows with and without visual effects.

5.6.1 Hypothesis for the 2nd dose-response test:

1. The dose-response curve will follow the standard shape for both the blur set and the normal set. This means that people agree on the lower part to be unreal and the upper part to be realistic.
2. The dose-response curve plotted for visual effects, will be placed higher than the normal images, hence the blur effect will have a positive effect on the acceptance rate.
3. Knowing the system performance and having a criteria for percent-wise acceptance of a population, the dose-response curve can determine the minimum requirements for light sources,

and the acceptance of a given composition with a given number of light sources between 1,2,4,..256.

5.6.2 Approach for the 2nd dose-response test:

Using an online questionnaire a series of images are shown to the test subject for a maximum period of 10 seconds and afterwards the image disappears. The time limit serves to prevent the paticipant from studying the image for too long, as this would not be considered count as an intuitive answer. The subject must pick an answer to whether he/she thinks the image is real or not. Images are only proposed for the subject a single time, and the subject will only be exposed to 4 different images from each of the two settings. This should prevent the subject becoming accustomed to the task, influencing his/her choices due to multiple viewings of the same scene.

The main purpose of the test however, was not to find whether people could spot an imitated image just by looking at it. The purpose of the final system is only to be able to make virtual shadows that seem credible in a given environment. Therefore, a merely yes or no question to whether an image is imitated or not, would just make people very suspicious and start hard to look after erroneous effects in the image. What this test should show is when people answer intuitive would say a virtual shadow looks real, without looking after it. Therefore the question after each image was change to:

- The shadows were obviously imitated
- I wouldn’t immediately think of the shadows as being unreal

Because the image disappears after 10 seconds another option has to be present. If the subject is distracted, receiving an SMS or someone walking in the door while he is taking the test, this could remove attention from the test. This could lead to the subject not seeing the image or forget what he saw. So the test must include an option to say that he doesn’t know what to answer.

In the intro it was clear that the test should inform the subject for what he should look for, though this would break the intuitive answer. But if the subject does not know what to look for he might be aware that it is the shadows that are changing, and at the end of the test he will have easier to spot imposters.

5.6.3 Expected results for the 2nd dose-response test:

As the hypothesis stated the expected results should yield two standard shapes of the dose-response curve, one for the blurred images and a second one for the normal images. And it is expected that the blurred images will have higher acceptance rate than the normal images. The expected results can be seen in the diagram below:

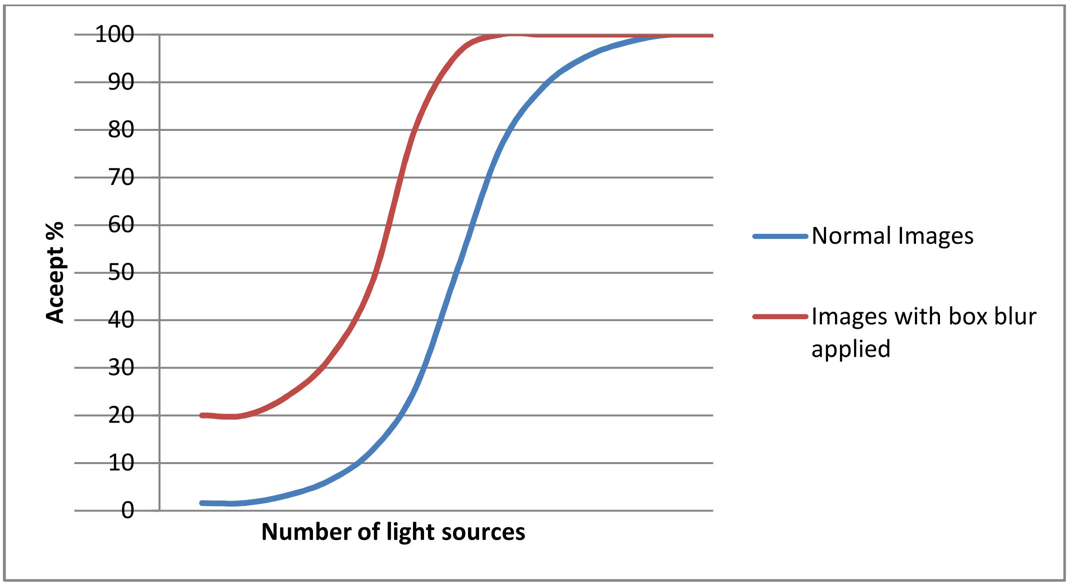


Figure 5.10 Expected result for the final dose-response test. Images with blur applied general have a higher acceptance reate

5.6.4 Results from the 2nd dose-response test

The questionnaire was posted on the internet on two different sites; Facebook ("Facebook," 2004) and on the forum of Nordic-t ("Nordic-T,"). Facebook gave about 100 answers while the Nordic-t forum gave about 400 answers to the questionnaire. This of course, could color the results a bit, because both sites targets specific users.

Although most of the subjects coming from Facebook would fall into the target group as they are just ordinary people with a variety of different backgrounds. Still the majority would be friends of some in the group and hence could have some experience in virtual shadows and augmented reality. On the contrary, it is also good to have some subjects representing this specific group.

The greatest part of the subjects, are from the torrent site Nordic-t.org. Even though it is difficult to say which type of people is visiting this site it could somehow affect the results. One thing that can be said about the users, is that they know how to use a computer, and are somewhat technical minded. But that does not necessary mean they do not qualify as subjects for this test. Looking around the forum tells us that the majority is young males, and probably a lot of them are teenagers. This means a lack of female subjects. Other than that it should be good enough. Augmented reality products would often target young people as they are more open to new technology.

First a summarization of what data can be withdrawn from the questionnaire:

- The time the user takes to submit his answer for each image
- The rating, in terms of credibility, of the image coming from each answer

5.6.4.1 Data verification

At first it is a good idea to see what type of information that is gathered in the test phase. This could eliminate possible errors in the following analysis. First the total amount of answers is plotted and distributed by the different images.

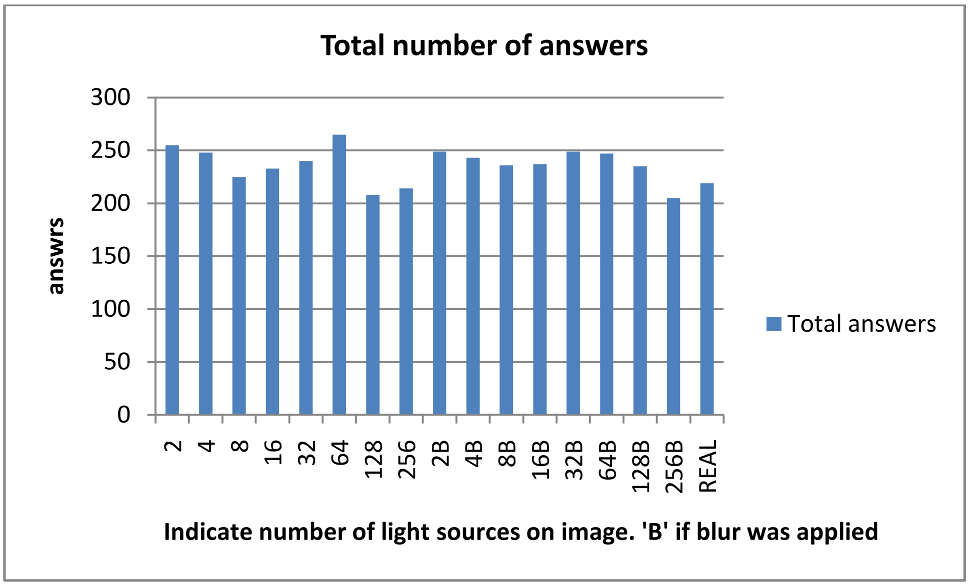


Figure 5.11 Total number of answers distributed by test images

As can be seen on the graph the number of answers is evenly distributed between the different images, given between 200 and 250 answers to each. This means that all images are evenly represented, and can be used in an analysis. Let's see how the different answers are distributed.

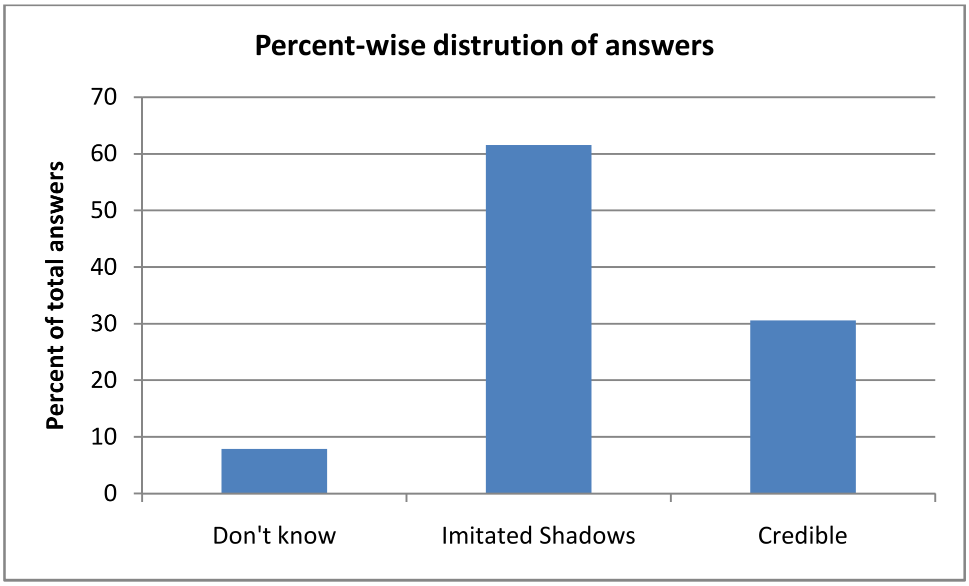


Figure 5.12 Number of answers recorded in each category

The diagram shows that most of the answers are either of the type imitated or credible with most imitated answers.

Now it is time to see how the response time of each answer is distributed.

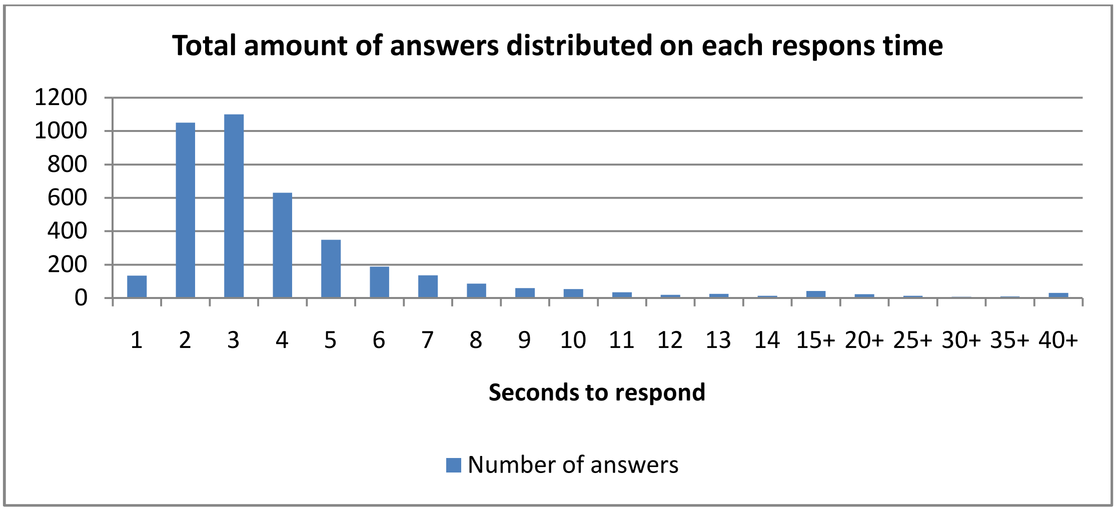


Figure 5.13 Diagram of how many answers were submitted for each response time

The diagram above show that the bulk of answers were submitted within the first 10 seconds, and the largest part of answers is done between 2 and 4 seconds. This tells us that people don't take too long to respond and thus must have made intuitive answers, which was essential for the test. One could ask himself how it is possible to make an answer so fast as 1 or 2 seconds. But it is to be remembered, that the subject is forced to see each image for 10 seconds. This makes it easy for the subject to make a decision while he is looking at the image and then make a quick answer afterwards.

To analyze the response time further, the specific response time of each different image is plotted in a histogram. This should show if the number of light sources affects the time the subject is using to answer. From the initial test it is shown that images with 64 light sources and above are difficult to distinguish from a real image. It is likely that these images would take a longer time to respond to than the images that are obvious imitated.

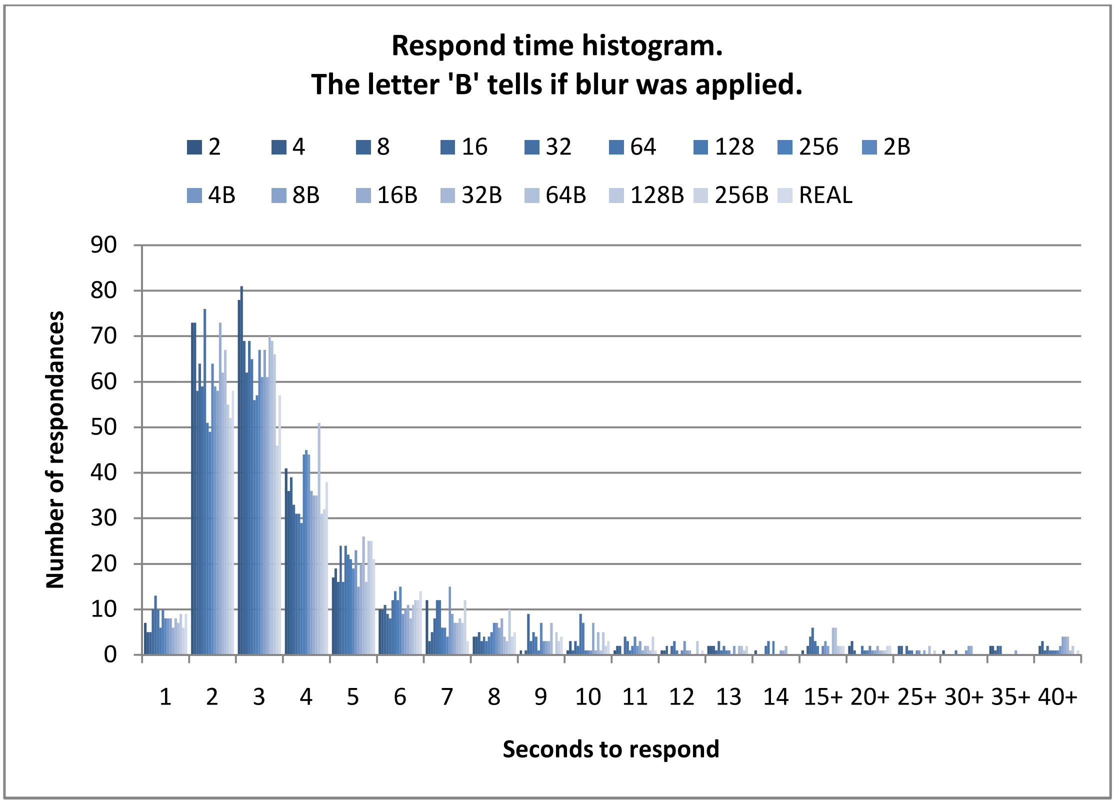


Figure 5.14 The diagram show the response time of each image in the test set.

The diagram shows that the response times are evenly distributed between the different images. The subjects did not take any longer to respond to an image with a high amount of light sources compared with an image with a low number of light sources. It would be interesting to see the specific answers or rating of images, for each response times. The "don't know" button in the questionnaire is the default option, and if this is the major answer of the quick answers it could indicate that, these answers was not considered carefully enough by the subjects and might lead to errors.

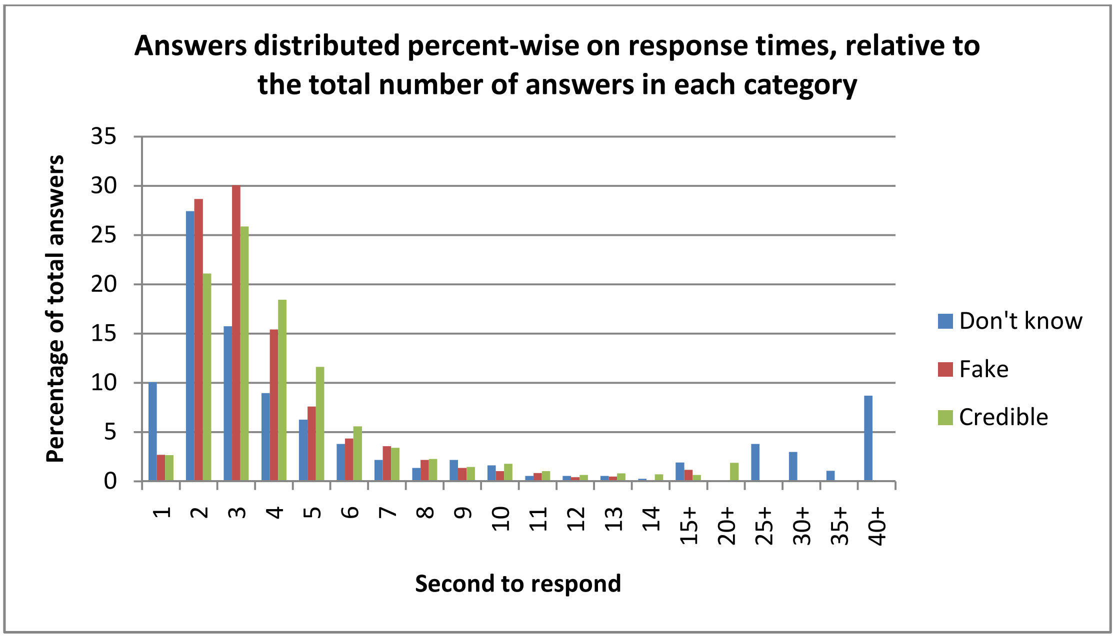


Figure 5.15 The diagram show how the answers from each category are distributed between response times. But it doesn't show the relationship of the total number of answers between the three categories

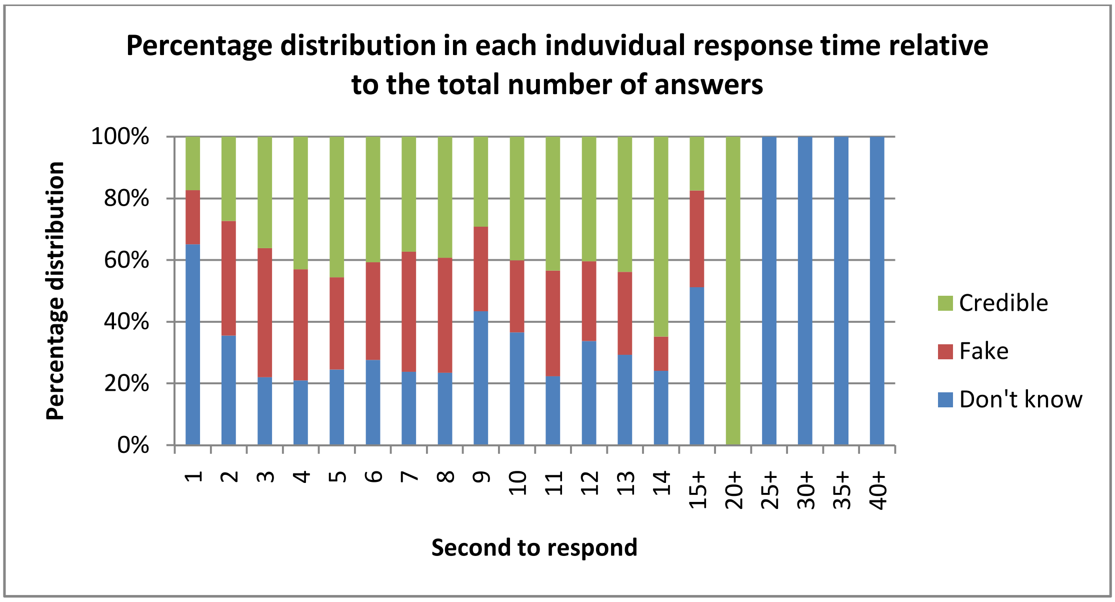


Figure 5.16 The diagram show how the answers from each category are represented in each response time

The two diagrams show that the distribution of the total answers of "don't know" definitely are more represented in the first second of response time than on other response times. As described this could indicate that the answer is merely a quick submitting from a lousy subject. Another remarkable finding from the graphs is that 100% of the answer that took more than 25 seconds to be submitted was a "don't know" answer. This could very likely indicate that the subjects attention was drawn away from the test for a while, and when the subject returned to the test, he couldn't remember the image and therefore the answer becomes a "don't know". Other than that, it seems that the distribution of answers is evenly distributed between 2 seconds of response time and 24 seconds of response time.

It is important to notice that the graphs doesn't show how each category of answers relate to each other. The total percent-wise distribution of answers in each category can be seen in Figure 5.12.

As a consequence of the above findings, data that took more than 25 seconds to be submitted to the database are discarded. Another valid reason for doing so is the fact that these answers would not candidate as intuitive answers.

5.6.4.2 Data analysis

It is time to look at the actual data, to see if any useful information about the credibility of shadows could be withdrawn.

There are three different scenarios in which the data could be plotted. Of course the answers about imitated credible images should be plotted. But the data where the subject has answered "don't know" can be handled in three different ways:

1. "don't knows" data could be discarded and not count in the overall score
2. "don't knows" could count as a answer equal to a "imitated" answer
3. "don't knows" could count as a answer equal to a "credible" answer

It should be pointed out that as stated in the start of the section, the total number of "don't know" answers is quite low, and wouldn't make that of a difference in the big picture.

The obvious solution would be that the "don't knows" should count as a "imitated" answer, because otherwise we would favorite the test result to our side. But the test is about to see what an intuitive answer of an imitated shadows would be. Because of this, it would make sense, that subjects that answer, that they don't know whether the shadow is imitated or real, definitely do not see an obvious imitated shadow. In this case the shadows from our point of view would be considered to be good enough, which is also what is sought in the test.

Therefore the "don't knows" is tried plotted both as discarded data and as if it was equal to a "credible" answer. In compliance with the initial 1. test, the test is carried out with two different setting. The data from the two settings can be seen plotted below:

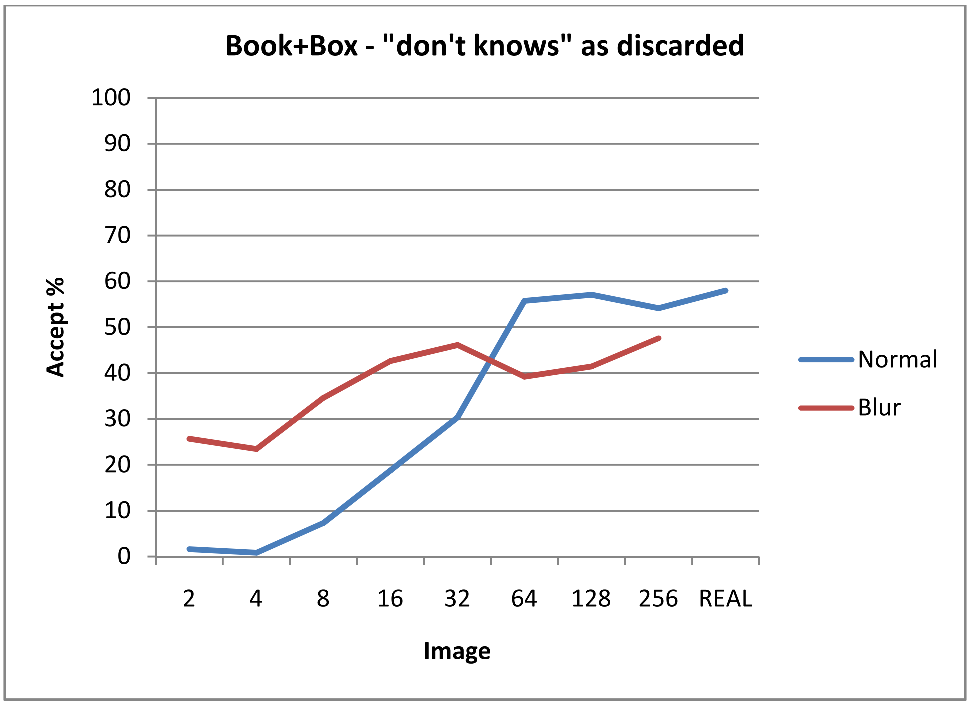


Figure 5.17 Dose-response relationship between acceptance rate and number of light sources. The category of "don't know" answers has been discarded as data

Figure 5.18 Dose-response relationship between acceptance rate and number of light sources. The category of "don't know" answers has been plotted equal to a "credible" answer

The first thing that comes to our mind is that the acceptance rate of all images is quite low. Even the real image only has an acceptance of about 60%. . This is in contrast to the initial first test, where many participants rated images with about 64 shadows as images that could have been real. The skepticism could be caused by some different factors:

1. The text in the questionnaire was somehow not clear and has confused the subjects.
2. Subjects are more skeptical when dealing autonomous things. In the initial first test, the subjects were confronted personally, which could make them more believable, whereas in the second test, they don't have the personal contact, which has made them more skeptical.
3. Subjects see the task as a competition and a challenge to test his skill and therefore are more likely to select images as imitated. As the questionnaire was posted on Facebook and Nordic-t forum, subjects had the possibility of commenting on the test afterwards. What was extracted from the comments was that the subjects really wanted to spot the imitated images instead of just making an intuitive answer, which was the purpose of the test.
4. The images used for testing is images of unnatural environments. It is not normal to see images of a clean table with three different objects placed on the table. It is unnatural and is likely to have caused people to think of it as imitated.
5. The lights in the environment used to create the test images cast shadows that are not normal to what people normally perceive. Although the human should have experienced a lot of different shadows, the shadows cast in the test images are pretty edgy. That could have been a factor as why the shadows look more unnatural. Also the images in the second test are shown on a computer screen rather than printed images as in the first test. This could enhance the effect of unnatural shadows because images on a printed paper look smoother.
6. People have difficulties with solving a task they normally take for granted. The human mind is very well accustomed to perceive shadows, and it is just a subliminal task. As humans we normal don't make notice of shadows, they are just there. It is likely that when asked to point out imitated ones we can't remember how they actual should look like, and therefore are more skepticism.

As for now it is not possible to say exactly why people do not perceive the real image as real. From the first initial test though, we get that these images should look real. Knowing that the real image is only accepted as real 60% of the time, we can however try to normalize the graphs to get a more applicable look.

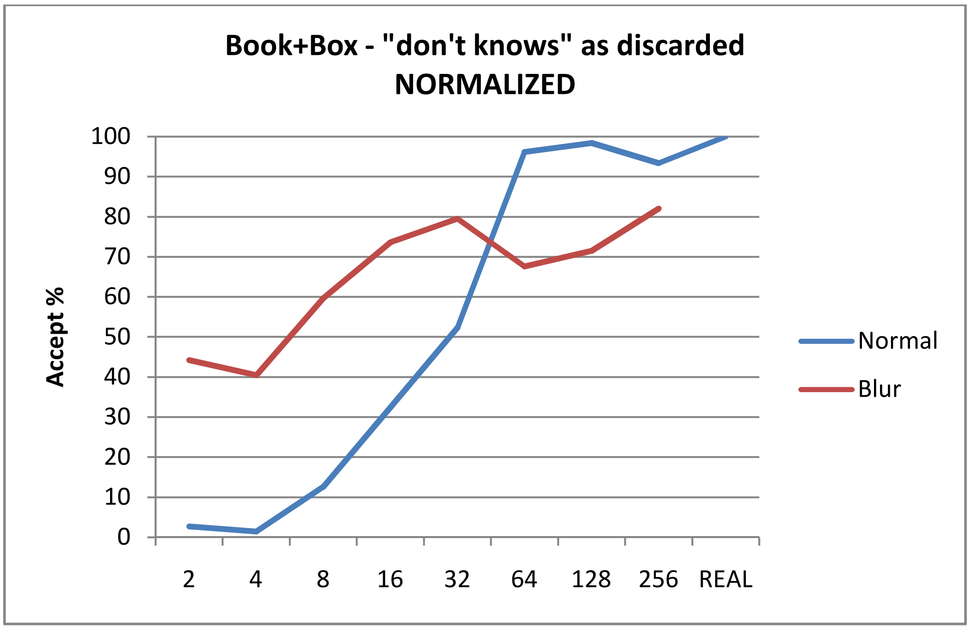


Figure 5.19 The normalized version of the diagram of Figure 5.17. Show the dose-response relationship between acceptance rate and number of light sources. The category of “don’t know” answers has been plotted equal to a “credible” answer

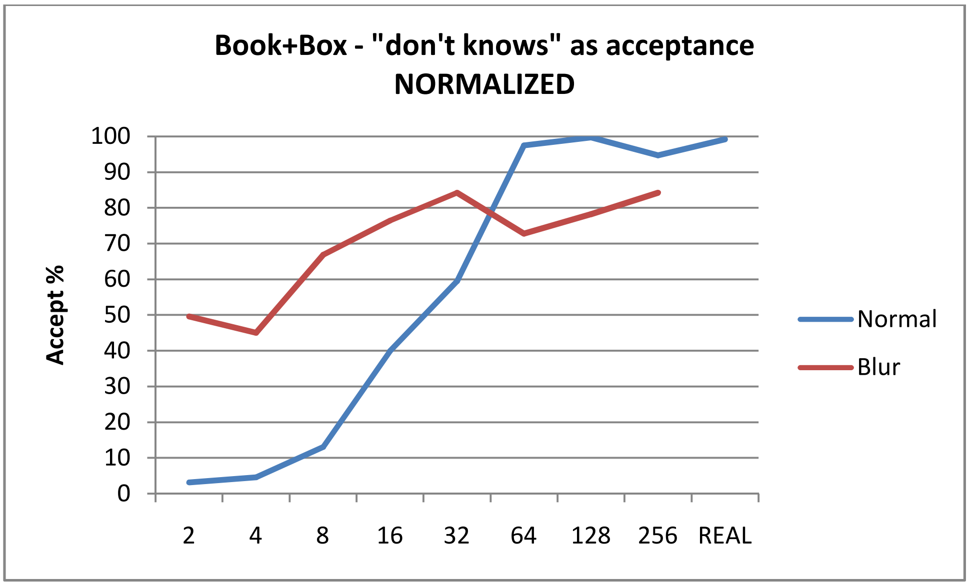


Figure 5.20 The normalized version of the diagram of Figure 5.18. Show the dose-response relationship between acceptance rate and number of light sources. The category of “don’t know” answers has been plotted equal to a “credible” answer

What can be said about the results is that, as the initial first test also showed, 64 light sources or higher, should be enough to replicate a real scene. Also the visual effects seem to have a very positive effect on images with a low number of light sources. But as we pass 64 light sources or higher, the blur effect seems to have the opposite effect and just make the image less credible. On the other hand, when 64 light sources are enough to simulate a real image, it wouldn’t be necessary to use the blur effect here anyway.

As a conclusion it can be stated that if it is acceptable that an acceptance of about 60% for a real image is enough, then the graphs gives a good indication what further should be implemented. As for performance it is up to a technical test on the system to test how many light sources it can handle. If the system can handle 64 light sources or more, then the shadows should look pretty good. If the system is to use less than 64 light sources then visual effect might be a good way to enhance the look of the virtual shadows. The graph could also be used by the user of the system to give a clue of which setting to use.

6 Creation of images used for testing

HDRshop ("HDR Shop," 2001) allows for creation of high dynamic images (HDR) by combining multiple almost identical images to a single image. The images must the identical, though with different expositor times (less/more light gets to the sensor). The same can be done in Adobe Photoshop ("Adobe," 2009).

The median cut algorithm (Debevec, 2005b) splits the image into n regions, specified by n number of light sources wanted, and places a light source in the centroid or center of that part. The median cut plugin for HDRshop allows for light positions and brightness to be exported to scripts for Autodesk 3Ds Max and Autodesk Maya ("Autodesk," 2009).

For the test, images were taken of static objects using a Nikon D90 camera with a 50mm lens. The objects were placed in their settings and images were taken. An image was also taken with no object, in case we needed to remove the objet completely. Then a mirror-ball was placed in the setting and multiple images using different exposure times were taken to get the lighting information from the room. Using the mirror-ball made sure that we got the complete lighting information, as it gets the information as an environment map. The mirror-ball was shot using a 300mm lens at a distance of approximately 2 meters making the ball fill the entire frame.

The images were composed to a series of HDR images, which was then cropped to only the mirror-ball. This image was then converted to a longitude/latitude image for it to be used in the median cut. HDRshop provided with ability to do this. Once this was done, the median cut was applied and script for lighting created.

Once the images were taken and the HDR images were created, the original image of the setting and object was loaded into both Photoshop and Maya. In Photoshop the shadows were removed and layered. In Maya, an artificial scene was created to match the geometry of the image, and the camera was placed as the original camera. The camera's setting was matched to the Nikon D90 as well, to render the exact view of the camera.

The scripts generated from the median cut plugin were altered to fit the current version of Maya (2009) by replacing the number of parameters in the light source generation. Then the scripts were loaded into Maya one by one and the shadows were rendered into a separate layer and saved in the alpha channel of a .tiff image.

The numbers of lights rendered were $[0, 2^0, 2^1, 2^2, \dots , 2^8]$.

The shadow layers were then imported into Photoshop and overlaid as alpha channels to achieve the correct shadows for the scene.

7 Design Specification

The designs requirements are based on the tests conducted in order to determine the perception of artificial augmented shadows compared to real ones. The tests will later serve as the requirements for the final implementation and a reference when making default and recommended settings for the final implementation.

The first test showed that the realism of the augmented shadows were dependent on reference objects in the scene, as people generally could not tell the difference between shadows when no reference object was present. The tests also showed that more than 64 shadows are indistinguishable from real shadows for the test subjects. In addition it showed that the number of light sources necessary for creating convincing shadows can be reduced further by adding a blur effect on the shadow. The smaller the number of shadows the larger the blur kernel needs to be.

The project seeks to implement an application that can estimate light source positioning and generate shadows. For the implementation to be successful, there are some basic requirements that the implementation must be able to handle:

1. An implementation that can track and process a known marker object and generate at least 64 light source positions, using the median cut algorithm. Possible to select 2^N shadows, based on the computing power of the individual computer running the project.
2. A real-time graphics renderer to display the virtual objects and render the shadows.
3. An implementation that can be downloaded and used by laymen and designers using available materials. Staying in the spirit of ARToolkit; making it a minimalist ease-of-use package.

These three criteria can each be elaborated further to specify the implementation even further.

7.1.1 An implementation that can track and process a known marker object and generate 64 light source positions using the median cut algorithm.

Tracking light sources in a scene has its limitations in the 2D representation of the frame it captures. The loss of information from the 3D representation to the 2D image is the loss of depth information. Using known geometry it is possible to reestablish the relationship between the camera and the marker. This exploit is the basis for ARToolkit. The markers are similar with a black rectangle around them for easy recognition by the computer vision algorithms in ARToolkit. The transformation from the marker coordinate system to camera is used as a path from viewport-coordinates to object-coordinates.

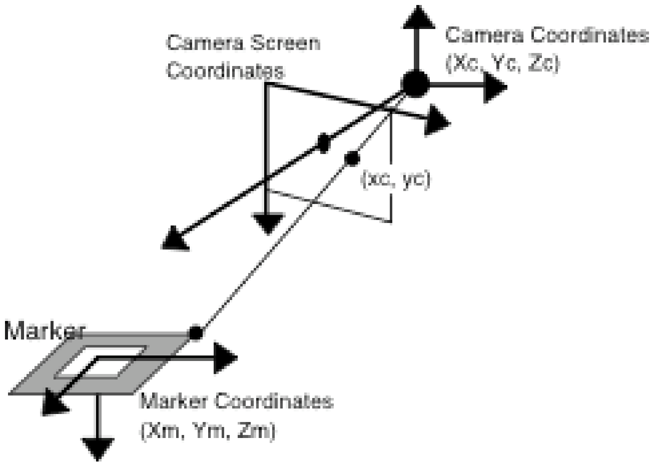


Figure 7.1 The conversion –matrix of coordinate-system describes the transformation in 3D space for a virtual object (Hirokazu Kato, Billinghurst, Morinaga, & Tachibana, 2001)

7.1.2 Augmented reality implementations

A precompiled and working Augmented Reality tracker is necessary for the development process, with additional libraries that can be added if necessary. The tracker chosen for the project is ARToolkit, because it has proved useful in numerous multimedia and design projects in recent years. Its popularity is primarily based around the usability and reproduction of results that ARToolkit offers, it is downloadable in a package free of charge and can be used for non-commercial development, and the markers can be printed out with little or no effort. ARToolkit can even be set to recognize drawings of markers. The program segments an image using image processing to find the orientation and distance of the marker relative to the camera, and determines the ID by template matching. If known geometrical markers are detected in the image, ARToolkit generates a matrix of the position and orientation of the marker according to camera pose and position. The resulting matrix is used as the identity matrix for the virtual object associated with the marker.

7.1.3 Light approximation

Applying shadows in a dynamic environment where camera position and light sources can change in an instant constrains the algorithm used to be of a complexity that allows for real-time tracking of light sources, while leaving overhead for other processes. The light detection must happen in an interval that generates the light sources necessary for the implementation to generate credible shadows, and the algorithm could only be executed when necessary and not be a continuous process unless the environment light conditions changes continuously as well.

7.1.4 Shadows from known 3D geometry

To track light sources from the environment the camera must have a way of tracking the surroundings that exceed the frame of the image. The relationship between lights and shadows can establish an estimation of the location of a light source. However the implementation is costly in execution time because it needs extensive image processing to determine the light sources using the shadow tracking algorithm. The algorithm analyses the full image checking the maximum luminosity and then evaluates these regions whether they are light sources or not. At first glance, depending on the conditions, the algorithm can at best be of O(N^2), which will take up the most of the processing overhead for light source tracking. This is not desirable and therefore another technique that has been described in several papers will be considered: Use of reflective spheres as light probes.

7.1.5 Reflective spheres

The reflective sphere can obtain information of the environment that exceeds the frame. The lighting information in the environment is captured in the reflections of the sphere, giving almost complete information of the lighting in the environment. By knowing the location and center of the sphere, the directions of the light sources can be estimated mathematically.

This can then be mapped to an environment map in a longitude-latitude format, which can then be used to find areas of similar light intensity using a median cut algorithm (Debevec, 2005). The centroids in these areas can then be used as light source positions to create shadows from.

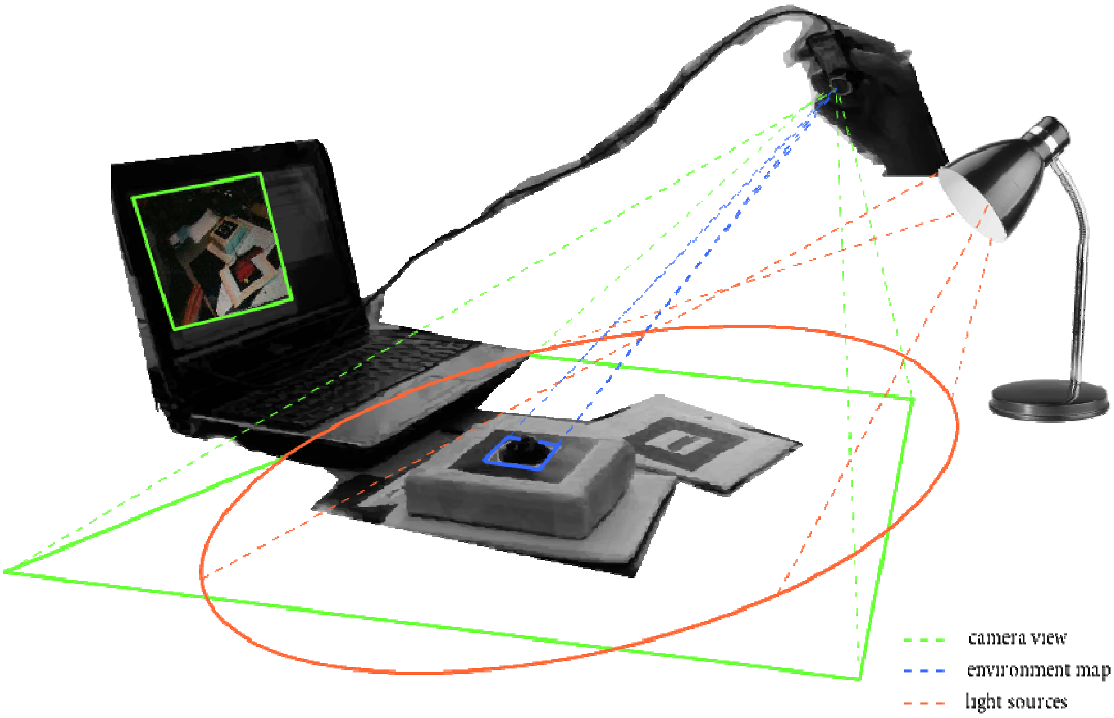


Figure 7.2 A mockup of the desired setup, notice the custom marker that exposes information from the environment beyond the image frame

To sum up, the light approximation technique requirements are:

- An implementation that can track and process a 3D object with a reflective sphere
- Generate at least 64 light source positions using the median cut algorithm.
- Can create 2^N light source positions, depending on the users computing power
- Leaving a reasonable overhead for 3rd party implementations.

7.2 A real-time graphics renderer to display the virtual objects and render the shadows

The project is not centered on coding a graphics pipeline, so for this project OpenGL is chosen, as ARToolkit already makes use of OpenGL and GLUT. This way the user does not need to install additional programs/libraries for the program to run.

The renderer must be able to render a 2D image of a virtual 3D environment with a virtual object overlay. The information from the tracking software should pass pose and position from each marker to the renderer that displays the object in 3D virtual space. This renderer must be able to handle 3D object formats that can be exported from 3D modeling programs and downloaded from the internet

Requirements for the renderer:

- Real-time graphics pipeline.
- Accessible and programmable.
- A 3D model reader.
- Visual FX processing of the created shadows.

7.3 Staying in the spirit of ARToolkit; making it an ease-of-use package.

The limited accessibility by the vast field of research in shadow generation techniques makes it very much a niche for experts. This is great in research environments but it can prove difficult to implement for laymen that have not worked in the field or extensive experience in implementation of technical descriptions. The easier the final implementation is to use, the greater the likelihood of it being used by a 3rd party.

User and distribution requirements:

- The implementation and instructions is in a format that can be downloaded from the internet.
- The implementation uses available materials and as little of them as possible.
- The implementation is fairly easy to set up and use implement and comes with a demo program.

7.4 Limitations

The shadows represent a very complex domain. The extensive visual cues the shadows give to a given scene is very complex. Because the implementation is real-time, a lot of details must be left out, so that the final implementation can qualify itself as being a true real-time application. It must be able to carry out shadowing of virtual objects, leaving enough overhead computing power to track markers, generate graphics and the likes of the developers working with the implementation. The limitations will be a guideline to what the final implementation will not take into account.

All shadows are projected shadows; the visual cues of the underlying geometry will be discarded. The reason being that most Augmented Reality trackers available are only tracking known geometry. Hence the algorithm is oblivious to everything but the marker geometry. This handicap makes it much faster at tracking the objects designed for it. The test shows that there is substantial disagreement of real and imitated shadows, especially detailed shadowing makes most test subjects confused. Therefore the assumption seems fair, that shadows are cast on a plane even though it physically is nonexistent i.e. the end of a tabletop, the curvature of a car bonnet etc.

Self-shadowing will not be of the highest priority in the initial implementation of the implementation. However the technique can be implemented using guidelines from (Wright, Lipchak, & Haemel, 2007, chapter 14). This is a minor aspect of the implementation and will only be implemented if time constraints allow it. However it contributes to the realism of the composited scene, and that must be taken into account eventually.

The implementation will not take the following parameters into account in the first development stages.

- Underlying geometry of the scene.
- Self shadowing

Depending on the time available for further development of the implementation these parameters can be implemented if deemed necessary by user tests.

Advanced expensive solutions to the problem, such as light sensing using sensors or high definition cameras, are discarded due to constraints from the 3rd design criteria. The techniques have been reviewed and evaluated, and been useful in the initial tests. But for real-time and easily accessible solutions it is not a useful approach.

7.5 Conclusion on Design

The implementation must be as lightweight as possible, using the derived test results as a minimum benchmark for the specification to be fulfilled. This approach guarantees the initial premise of the project – creating credible shadows in an augmented reality space. The limitations are based on computational constraints and the prerequisite of a real-time based application. Using a sphere in the image solves the problem of the limitations of an image frame. If the sphere is in the image, the reflections of the sphere, gives additional information about the lighting conditions in a given environment and is therefore a useful and practical approach. Spheres are generally accessible and so is the materials to process them such as paint, gloss etc, and prior research have already established that it is feasible approach for real-time estimation of light source positions.

8 Implementation

Implementing a real-time light source tracker in a constrained time period involves using available tools and applications, these will be described below as well as the specific techniques implemented in the project.

8.1 ARtoolkit

ARToolkit is a software library for tracking markers and calculating the translation and rotation in 3D to find the position relative to the camera (Lamb, 1999).

The ARToolkit library uses a fixed size marker region to track, which can be seen in Figure 8.1. The outer bound of the figure is the area tracked and the inner part is the actual marker that has an id.

Figure 8.1 The Hiro marker, which comes with ARToolkit. The inner part labeled "Hiro" and the white space are defining the marker ID, where the black outer bound is for tracking (Lamb, 1999).

For the implementation, the markers were printed as 80mm x 80mm rectangles. The size of the markers are not that important, as long as all markers have the same size, as ARToolkit uses the size information to calculate the distance to the marker in 3D space.

The positioning of the marker compared to the camera is calculated from the image data of the marker available after an initial threshold of the input camera image.

The ARMarkerInfo type in ARToolkit holds information of the found marker, such as area, which is the area of all pixels, the id compared to known markers, the position in 2D and all four corners. From the gathered data, it is possible to calculate the position and rotation in a 3D space. This is then saved in a 4x3 matrix, which can be used later for translating the OpenGL coordinate system of that position in space for drawing.

Figure 8.2 shows a general outline for how ARToolkit works, by first capturing the image, using image processing, calculating marker position and rotation and finally augment an image on top of the marker (H. Kato, Billinghurst, Poupyrev, Imamoto, & Tachibana, 2000).

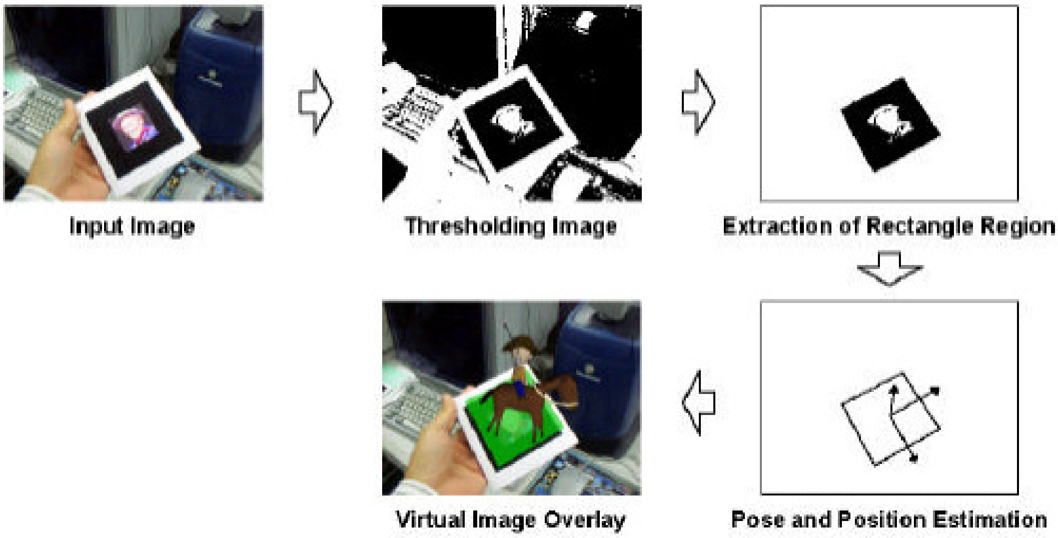


Figure 8.2 ARToolkit tracking of a marker. First, an image is captured by the camera, afterwards a marker is extracted and the position in 3D space is calculated and a virtual object is being augmented into the image (H. Kato, et al., 2000).

ARToolkit can be setup to find multiple markers in a single frame, in which case a struct or class can be used to keep track of the individual markers, their ID, their current positions, rotations, etc. and having a loop traverse through the found markers in each frame and update the data, as needed.

8.2 Placing the reflective sphere

To dynamically map the light settings in an environment using a reflective sphere it is necessary to locate and track the sphere in the individual video frames retrieved from the camera. One approach to locate this sphere is to use digital image processing to look for a circle within the image. This approach would make the location of the sphere independent of the location of the marker, but would also require an extra amount of processing power for the image processing

ARToolkit provides us with position, rotation and distance to any marker found in an image and it would seem obvious to place the reflective sphere in a known position relative to this marker and use the ARToolkit provided data to locate the sphere. This approach result in a few limitations since ARToolkit is only able to track a given marker when the whole marker is visible with the camera and therefore the reflective ball must not cover any part of the black edge of the marker.

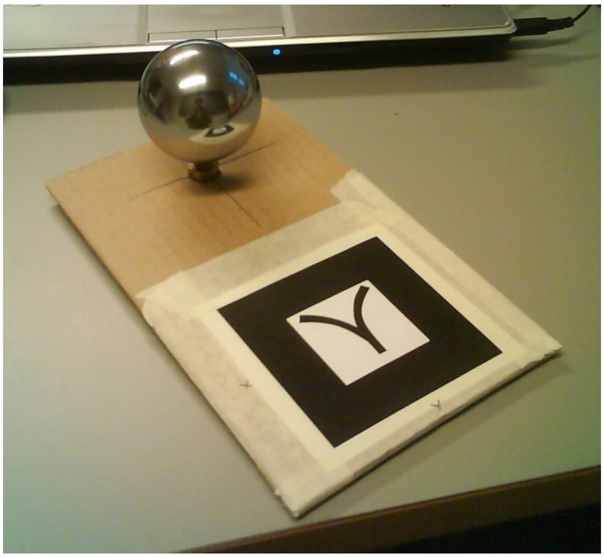


Figure 8.3 Approach where reflective ball is placed away from the marker

Figure 8.3 illustrates an approach also used by (Supan, et al., 2006) where the reflective sphere is placed at a known position outside the marker. This approach would make the marker visible from any height and roughly 270 degrees around the marker, as long as the reflective ball is positioned away from the camera. Unfortunately testing of this setup showed that the lens distortion of the camera made the tracking of points outside the marker inaccurate even thought the camera had been calibrated to ARToolkit.

Alternative to the first approach, the reflective sphere can be placed in the middle of the marker, as illustrated in Figure 8.4. This resolves the inaccuracy caused by the lens distortion as well as the problem where the entire marker is only visible in 270 of 360 degrees. Contrary the edge of the marker will be blocked unless the marker is viewed from an angle close to zenith.

By assuming that all relevant light sources is positioned above the plane represented by the marker, the bottom half of the marker can be discarded and the sphere can be lowered so the center of the sphere is placed in the center of the marker.

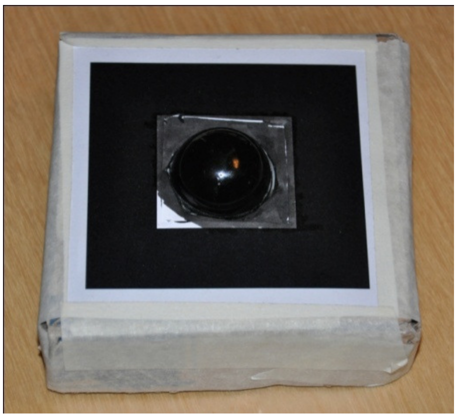


Figure 8.4 Approach where the center of sphere of circle is placed in the center of the marker

8.3 Color of the reflective sphere

When using reflective spheres to map the light in an environment, a mirror or chrome sphere is used to catch the colors of the surrounding environment. This approach however mirrors light intensity as-is and the intensity range from the darkest reflected spots to the lightest. To solve this problems a high dynamic range image is created by taking multiple images of the reflective sphere using different exposure time on each image.

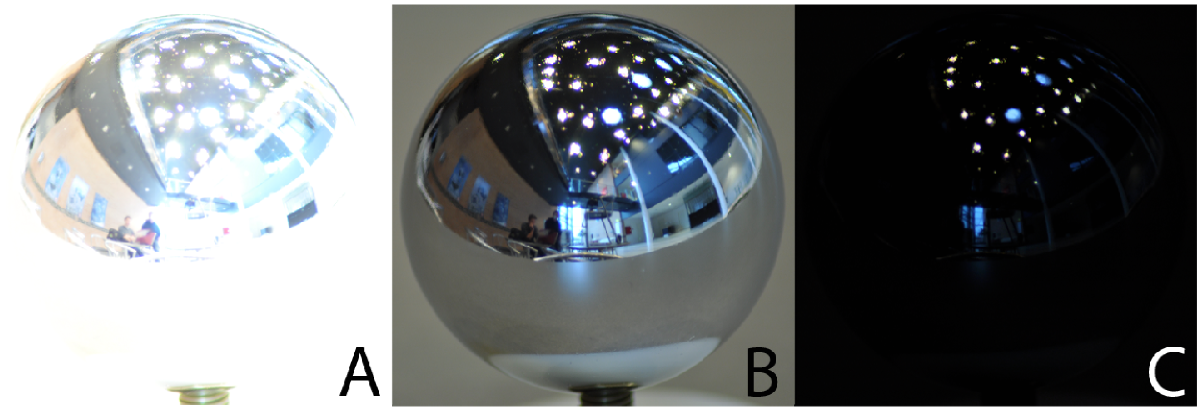


Figure 8.5 Images of a reflective sphere taken with different exposure times. The images can be combined into a HDR image. A) Overexposed image taken to retrieve information from very dark places in the image B) Image taken with normal exposure time to retrieve information from the normal lid parts of the image C) Underexposed image taken to retrieve information of the brightest parts of the image.

A normal image usually only have 256 values for each color channel. This renders a problem since the color range of most images is far too great to great to be represented in a scale of 0-255. Figure 8.5 shows 3 images of a reflective sphere taken with different exposure times. Figure 8.5a is taken with long exposure time, rendering the image very bright. This allows for details in the otherwise dark areas of the image to become visible. The same problem consists in Figure 8.5b and Figure 8.5c where some color channels of the image is either all 0 or 255. The solutions in high dynamic imaging is to interpolate between several images with different exposure times to create a single HDR image where the color channels are represented as a floating point instead of a char like in normal images, solving the limited range problem.

The high dynamic image approach is however impossible when working with real-time images and webcams since it is not possible to create multiple images with different exposure in every frame. Instead a way to discard some of the data in the image, so only the most useful data is left. The most intense lights are also the lights that create the most shadows. By coloring the reflective sphere, it is possible to shift the average colors to the lower end of the color scale, leaving only the brightest sources in the image visible in the higher end of the scale.

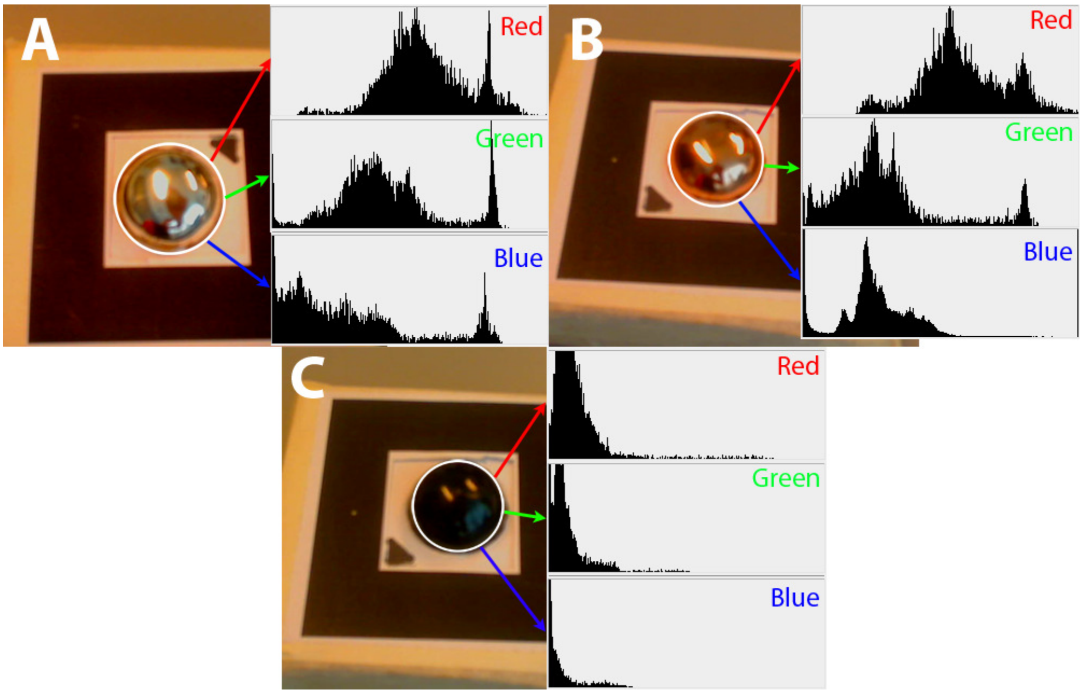


Figure 8.6 Histogram of different colored spheres positioned in the same environment A) Crome sphere B): golden sphere C) black sphere

Figure 8.6a shows an image of a chrome reflective sphere that could have been used with normal HDRI. The picture is a normal image with a color range from 0-255 is taken with the webcam used for the implementation. By looking at the histogram it is clear that the color values are spread all over the color space. The distance between the light sources and the rest of the image is not immense and therefore hard to distinguish. The same goes for the golden sphere in Figure 8.6b. Figure 8.6c of the black sphere shows that most of the information in the image has been shifted to the lower end of the color scale, and the strongest lights in the scene take up the rest of the color scale. This makes it a lot easier to distinguish the lights in the scene from the rest, although some information is lost in the process.

8.4 Segmenting the reflective sphere

All ARToolkit information on the marker is provided in 3 dimensions relative to the camera position of openGL. To make computations on the reflective sphere, it is first needed to segment the area of the image where the sphere is visible.

A sphere projected onto an image plane will appear as a circle. To segment this circle in pixel coordinates, two in formations are needed: center and radius. The center of the circle can be found by projecting the center of the sphere onto the image plane. The radius can be found by projecting any of the points on the periphery of the sphere that are orthogonal to the camera-center vector, onto the image plane and calculate the distance from this point to the projected center.

Figure 8.7 Segmentation of the sphere in the image by finding the center and radius and center in pixel coordinates. The two white dots is the center and a point on the periphery of the sphere orthogonal to the camera-center vector

The openGl function *glProject()* uses the supplied modelView, projectionView and viewport matrix to calculate the projection of point (x, y, z) in three-dimensional space $\mathbf{R}^3$ to the point (x, y, 0) in the image plane, $\mathbf{R}^2$.

By loading the identity matrix using glLoadIdentity(), the camera is reset to P(0,0,0). The center can now be calculated using the center coordinate $C(c_x, c_y, c_z)$ of the marker supplied from ARToolkit. The point on the periphery is found by first calculating any of the normalized orthogonal vectors to the position vector of C and scaling it with radius of the sphere.

One of the solutions is $R(r_x, r_y, r_z) = \{-c_z\ , c_y , c_x\}/|\{-c_z\ , c_y , c_x\}| * radius_{sphere}$

Projecting C and R to image coordinates using glProject() results in the pixel coordinates of the two points. An example of the two points found can be seen as white dots in Figure 8.7. Using Pythagoras to calculate the distance between the two points gives the radius of the circle in pixels. The radius and the center is all that is needed to segment a circle in an image.

8.5 Reflection mapping

The circle segmented in the image represents the visible half of the reflective sphere located in the center of the marker. Light from the surroundings of the sphere is reflected on the surface and into camera. This means that each pixel in the segmented circle in the image represent the light reflected from a given direction around the reflective sphere. This is illustrated in Figure 8.8.

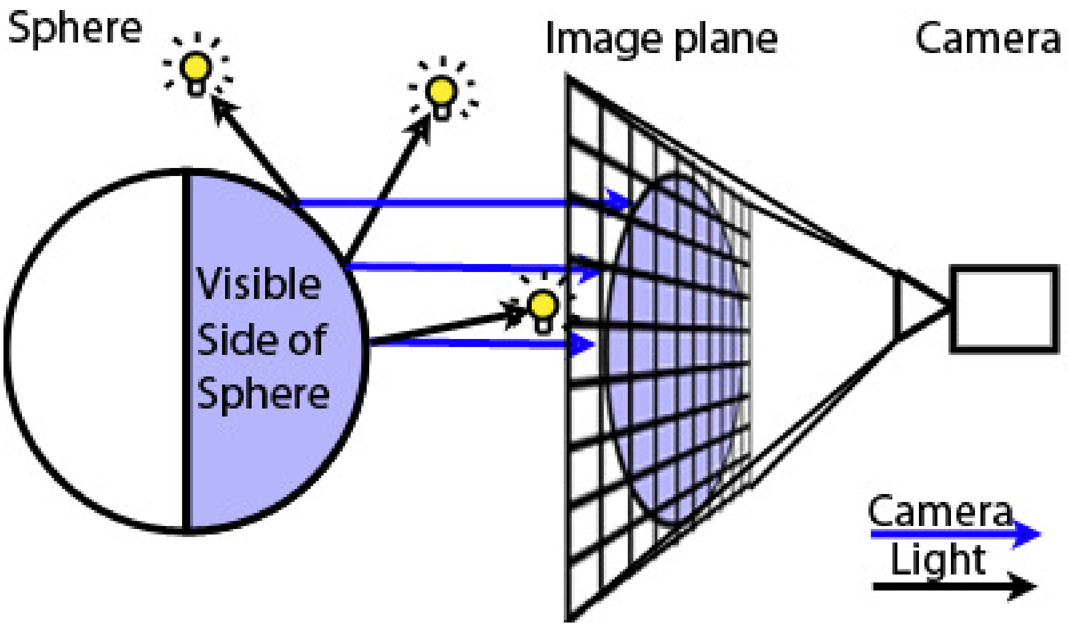


Figure 8.8 Light from the surroundings of the reflective sphere is reflected on the surface and into the camera, making each pixel in the segmented image correspond to a reflected light direction relative to the sphere. It is assumed that the camera is placed at an infinite distance from the sphere, and the camera vectors are therefore parallel

When calculating the reflection seen in the reflective sphere, it is assumed that the camera looking at the sphere is placed at an infinite distance from the sphere along the positive z-axis pointing towards origin and having $[0,1,0]$ as up vector. This makes the calculations easier, since all the camera direction vectors are parallel and therefore the same. Furthermore it is assumed that the sphere is a point in space and the radius is not taken into account when calculating the reflection vector.

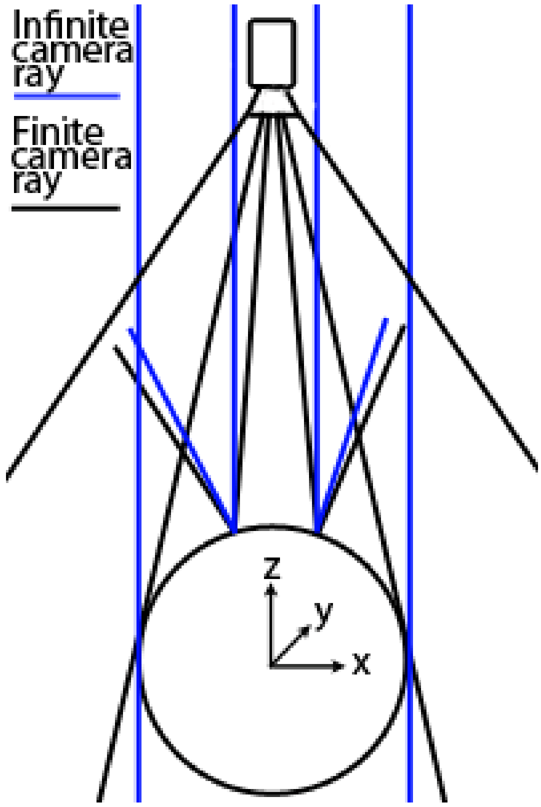


Figure 8.9 Comparison between the camera rays from a camera positioned at finite distance from a sphere, black rays, and the camera rays of a camera at an infinite distance from the sphere, blue rays.

These assumptions make the calculations easier, and thereby faster, but also result in a source of error. Figure 8.9 shows the different errors when comparing a camera placed at an infinite distance with a camera placed at a finite distance. The blue lines represent camera vectors from the camera placed at an infinite distance, while the black lines represent vectors to the camera placed at a fixed distance. It is obvious that the error rate decreases the further away the camera is from the sphere and since the segmented circle only take up a small amount of the camera feed, and the camera is placed at a long distance relative to the sphere size, this error rate becomes insignificant.

When mapping the reflective sphere both spherical and Cartesian coordinates are used. Spherical coordinates are defined as$[altitude, \theta, \varphi]$. All calculations will be done on the unit sphere which means that the altitude equals 1. Theta $\theta \in [0; \pi]$ is calculated in radians and is defined as the angle between the z-axis and the position vector to P. Phi $\varphi \in [-\pi; \pi]$ is calculated in radians and defined as the angle between the x-axis and the projection of the position vector P onto the xy-plane. This mapping is shown in Figure 8.10.

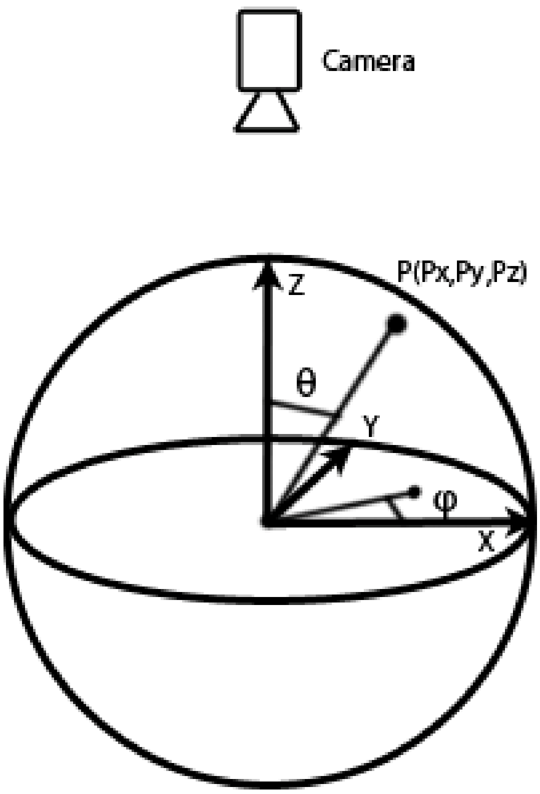


Figure 8.10 Definition of spherical coordinates. Theta $\theta \in [0; \pi]$ is the angle between the z-axis and the position vector to P. Phi $\varphi \in [-\pi; \pi]$ is the angle between the x-axis and the projection of the position vector P onto the xz-plane

Any light visible in the sphere is assumed to be a reflection of a light ray reflected in the tangent plane to the sphere in the corresponding point. Since it is assumed that the camera is positioned at an infinite distance along the z-axis, the coordinate of a pixel in the segmented image would correspond to that point projected to the xy-plane. As illustrated on Figure 8.11a where φ is calculated as the angle between the x-axis and the position vector to the pixel coordinates to the point P. As illustrated in Figure 8.11b the reflection is calculated by doubling the angle between the input angle from the camera and the normal vector to the tangent plane in the point P.

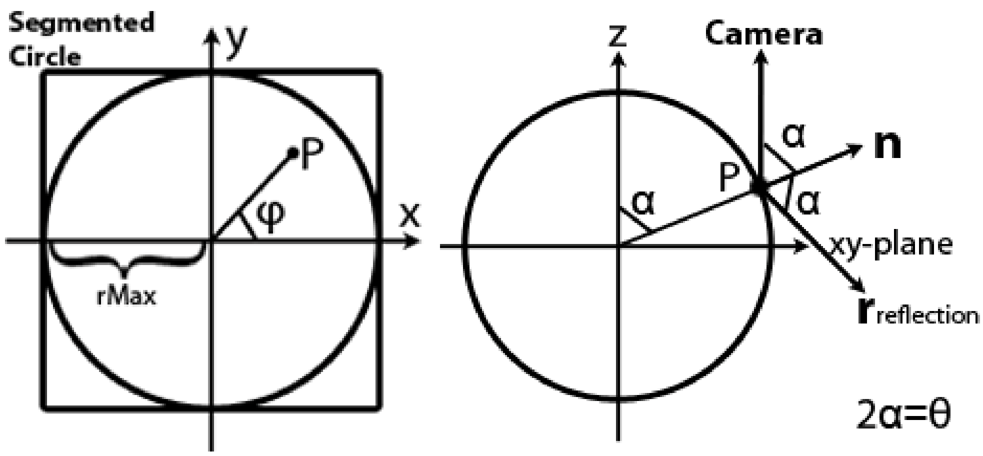


Figure 8.11 Approach for finding spherical coordinates for the reflected ray in a point P on the segmented circle.

With the spherical coordinates in place, an environment map of the light direction can be defined by mapping the spherical coordinates of the light reflections to a 2:1 image with theta on the x-axis and phi on the y-axis.

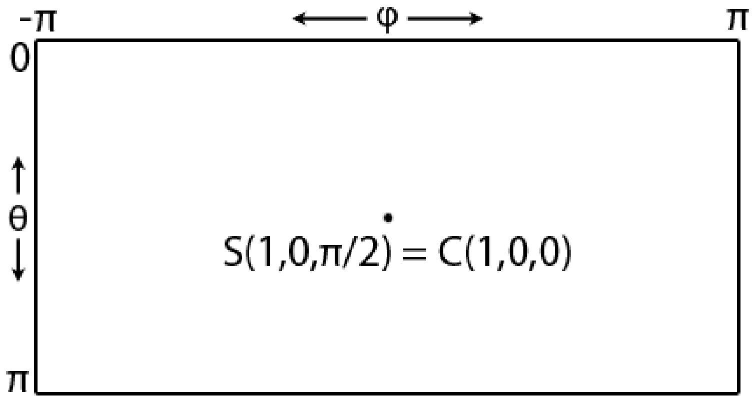


Figure 8.12 Environment map used to map spherical coordinates of the reflections to a 2d image.

8.5.1 Forward mapping

Forward mapping is done by iterating through every pixel in the segmented circle of the reflective sphere and mapping each pixel value to the corresponding pixel in the environment map. Since the environment map usually consists of more pixels than the segmented sphere, this approach would leave untouched pixels in the environment map. This approach is the fastest since it avoids pixel duplicates in the environment map. The spherical coordinates for the reflection in a given point P in image coordinates can be calculated as:

$$altitude = 1$$

$$\theta = 2 * \sin^{-1} \frac{\sqrt{Px^2 + Py^2}}{rMax} \in [0; \pi]$$

$$\varphi = \mathrm{atan2}(\mathrm{Py}, \mathrm{Px}) \in [-\pi; \pi]$$

Where Px, Py are the pixel coordinates for the segmented image and rMax is the radius of the segmented circle.

8.5.2 Backwards mapping

Backwards mapping is iterating through every pixel in the environment map and grabbing the color value of the pixel in the segmented circle that corresponds to this reflection. This approach ensures that all pixels in the environment map receive a value although some pixels may be duplicated since the environment map usually holds more pixels than the segmented sphere.

$$Px = \begin{cases} -\sin\left(\frac{\theta}{2}\right) * |\cos(\varphi)| * rMax, & \varphi > \frac{\pi}{2} \mid \varphi < -\frac{\pi}{2} \\ \sin\left(\frac{\theta}{2}\right) * |\cos(\varphi)| * rMax, & \varphi < \frac{\pi}{2} \mid \varphi > -\frac{\pi}{2} \end{cases}$$

$$\mathrm{P}y = Px * \tan(\varphi)$$

Where Px, Py are pixel coordinates within the segmented circle and rMax is the radius of the segmented circle.

8.6 Averaging the environment map

By averaging the environment map over a period of time, any sudden changes in individual pixel values can be smoothed depending on the sample size. A mean filter can be applied by using the formula

$$M = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Where n is the number of samples and a is the current pixel value. This method require n stored versions of the environment map and lots of computation time since n environment maps must be iterated at every frame to calculate a mean. To avoid the large amount of stored versions of the environment map and computation time, a modified moving average, MMA can be implemented. The MMA uses information from the current frame as well as the previous frame to imitate a mean filter described above.

$$MMA = \frac{(n-1) * MMA_{previous} + a}{n}$$

Where n is the number of samples, $MMA_{previous}$ is the MMA of the pixel value calculated in the previous frame and a is the value of the current pixel.

The MMA is not as accurate as the mean value, but requires significant less computational time since environment map only needs to be iterated once every frame.

8.7 Tracking light from the environment map

A median cut algorithm can be used to distribute the 8 available openGL lights as well as the directions from which the object should cast shadows. The algorithm ensures that all the lights in the scene is distributed according to the total amount of light in the image. It works by dividing the whole image along the longest side to the two new smaller images have the same amount of light within them. Repeating this step on the new images until the desired amount of squares is found. In the center of each square, a light is placed. Figure 8.13 shows the pseudo code of the implemented median cut in the final product.

```
medianCut(x, y, width, height, iterations, sumOfLights = -1)
{
    if(iterations >= 0){return}

    if(sumOfLights == -1)
    {
        sumOfLights = calculateSumOfLight()
    }

    sum = 0;
    if(width > height)
    {
        for(i=x; x<width +x; x++)
        {
            if(sum * 2 > sumOfLights)
            {
                medianCut(x, y, i-x, height, iterations-1, sum)
                medianCut(i, y, width+x-y, height, iterations-1, sumOfLights-sum)
                return
            }
            for(j=y; y<height +y; y++)
            {
                sum += image[i][j][red]
                sum += image[i][j][green]
                sum += image[i][j][blue]
            }
        }
    }else
    {
        for(j=y; y<height +y; y++)
        {
            If(sum * 2 > sumOfLights)
            {
                medianCut(x, y, width, j-y, itterations-1, sum)
                medianCut(x, j, width, height+y-j, itterations-1, sumOfLights-sum)
                return
            }
            for(i=x; x<width +x; x++)
            {
                sum += image[i][j][red]
                sum += image[i][j][green]
                sum += image[i][j][blue]
            }
        }
    }
}
```

Figure 8.13 pseudo code of the median cut algorithm

The rectangles found in each iteration would have roughly the same amount of light in them. After the first three iterations the center of the rectangles is stored and later used to place the 8 openGL lights around the object. Similarly, when the algorithm is finished with all iterations and the latter is used to place shadows. Figure 8.14 shows the implemented median cut on an environment map of a black reflective sphere. Each of the red rectangles has the same sum of pixels, the white crosses represent the openGL lights and the green pixels represent a shadow each.

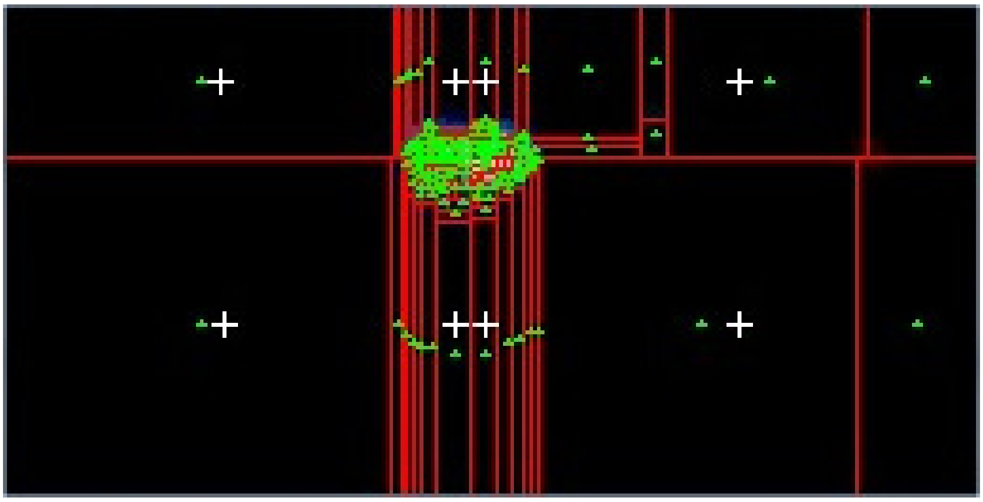


Figure 8.14 Median cut implementation used on an environment map of a black reflective sphere. Green dots represent shadows and white crosses represent the 8 light sources used for openGL

When the lights has been found, they can be translated back to Cartesian coordinates using the following formula:

$$c_x = radius * \sin(\theta) * \cos(\varphi)$$

$$c_y = radius * \sin(\theta) * \sin(\varphi)$$

$$c_z = radius * \cos(\theta)$$

Since all the lights are calculated where the camera was positioned at an infinite distance to the z-axis, the output reflection matches the coordinate system within openGL when the identity matrix is loaded and the directions is directly usable within the global space of openGL.

8.8 Lights and OpenGL

OpenGL default supports four types of lighting – point lights, spotlights, ambient lighting and distant, directional lights (Angel, 2008) and the following subsection is based on this source.

For elements to receive lighting in OpenGL, lighting first needs to be enabled:

```
glEnable(GL_LIGHTING);        //lighting enabled
```

With lighting enabled, each of the eight light sources must be setup individually. Each light source needs to be enabled, have a position, as well as diffuse, ambient and specular values:

```
glEnable(GL_LIGHT0);          //first light source enabled

glLightfv(GL_LIGHT0, GL_POSITION, position);     //position sat for the first light source

glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);     //ambient color sat for the first light source

glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);     //diffuse color sat for the first light source

glLightfv(GL_LIGHT0, GL_SPECULAR, specular);     //specular color sat for the first light source
```

The last entry for position, ambient, diffuse and specular is a reference for a vector containing the information. The vector needs to contain four floating point numbers. The fourth floating point number in the position is the coordinate for specifying whether it is a point light (1.0) or a directional light source (0.0).

The two remaining light types are a global ambient light that is independent of the other light sources and a spotlight. That the global ambient light is independent means that even with all lights disabled, there will be a small amount of the specified color everywhere in the scene. The global ambient light is used in a similar fashion to how ambience is set by a light:

glLightModelfv(GL_LIGHT_MODEL_AMBIENT, globalAmbient);

This sets the light model and *globalAmbient* is the vector containing the color.

The spotlight is basically a point light with parameters set for GL_SPOT_DIRECTION, GL_SPOT_CUTOFF and GL_SPOT_EXPONENT. This turns the point light into a spotlight.

Many other options are available for lighting in OpenGL, but this section has been limited to the information necessary in order for **the** implementation to work.

8.9 Creating shadows in OpenGL

There are three types of real-time generated shadows which will be discussed here: The projection shadow, shadow mapping and shadow volumes.

Projective shadows are shadows that are projected onto a surface, knowing the surface placement, the geometry placement and the light source placement. The projected geometry is rendered using another darker color and an alpha value for transparency (Chuang, 2009).

The following sequence illustrates the steps needed to create projective shadows:

1. Render the scene as normal
2. Translate back to local 0,0,0 where object is created
3. Multiply the modelview matrix with the projective shadow matrix calculated based on the ground and light
4. Translate to the place where the light-object vector crosses the plane
5. Draw the shadow projection using a dark color

The implementation for the shadow matrix is:

```
void setShadowMatrix(float ground[4], float light[4])
{
            float dot; // dot product of ground plane and light 
position
            GLfloat shadowMat[4][4]; // shadow matrix

            dot = ground[0] * light[0] + ground[1] * light[1] + 
ground[2] * light[2] + ground[3] * light[3];
            float scale = 0.0;

            shadowMat[0][0] = dot   - light[0] * ground[0];
```

```
        shadowMat[1][0] = scale - light[0] * ground[1];
        shadowMat[2][0] = scale - light[0] * ground[2];
        shadowMat[3][0] = scale - light[0] * ground[3];

        shadowMat[0][1] = scale - light[1] * ground[0];
        shadowMat[1][1] = dot   - light[1] * ground[1];
        shadowMat[2][1] = scale - light[1] * ground[2];
        shadowMat[3][1] = scale - light[1] * ground[3];

        shadowMat[0][2] = scale - light[2] * ground[0];
        shadowMat[1][2] = scale - light[2] * ground[1];
        shadowMat[2][2] = dot   - light[2] * ground[2];
        shadowMat[3][2] = scale - light[2] * ground[3];

        shadowMat[0][3] = scale - light[3] * ground[0];
        shadowMat[1][3] = scale - light[3] * ground[1];
        shadowMat[2][3] = scale - light[3] * ground[2];
        shadowMat[3][3] = dot   - light[3] * ground[3];

        glMultMatrixf((const GLfloat *) shadowMat);
}
```

The transformation flattens the object into a shadow of itself.

Some limitations to this approach are that it is hard to cast shadow to other objects than a ground plane, and no self shadowing. Since it is merely a projection of the object, controlling the color is difficult as well, with multiple overlays when using alpha, which is needed when augmenting on top of a camera image.

A bonus is, that it is fast to implement and does not take much computation time, so creating multiple shadows should not pose a problem.

Shadow mapping (Williams, 1978) is a simple approach using the depth buffer to store information of objects seen from the lights viewpoint. Each pixel is then compared to the image in the depth buffer to determine if the image should be in shadow or not (Chuang, 2009).

The following sequence illustrates the steps needed to create shadows mapping ("Shadow mapping," Wikipidia):

1. Create the shadow map by rendering the depth information from the light source point of view, and store the information in a depth map
2. Draw the scene as normal, applying the shadow map:
3. Find the coordinates as seen from the light
4. Compare the point to the shadow
5. Draw the object in light or shadow

For the test to work, the scene coordinates must be transformed into the position seen by the light.

Figure 8.15 shows an example of shadow mapping.

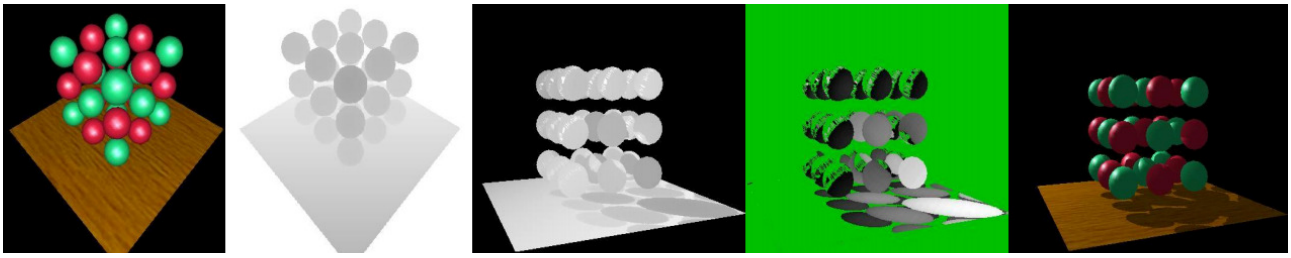


Figure 8.15 Shadow mapping. The images at the left side are the light source point of view and the depth information. The three remaining are the depth information projected to normal view, the test to find objects in shadows and the resulting final image (Chuang, 2009).

A downside to this approach is that the resolutions of the shadows are dependent on the resolution of the depth map, which can lead to aliasing if the resolution on the depth map is low.

Shadow volumes (Crow, 1977) are used to find areas in the virtual space that are in shadow ("Shadow volume," Wikipidia). Shadow volumes are generally slower than other shadow generation methods, as the area must be computed per frame. The major advantage is that the resulting shadow is correct to the pixel ("Shadow volume," Wikipidia).

Shadow volumes are created by computing the contour of the object, seen from the position of the light source. The shadow polygon is subsequently found for each light source. An example of this can be seen in Figure 8.16 and Figure 8.17.

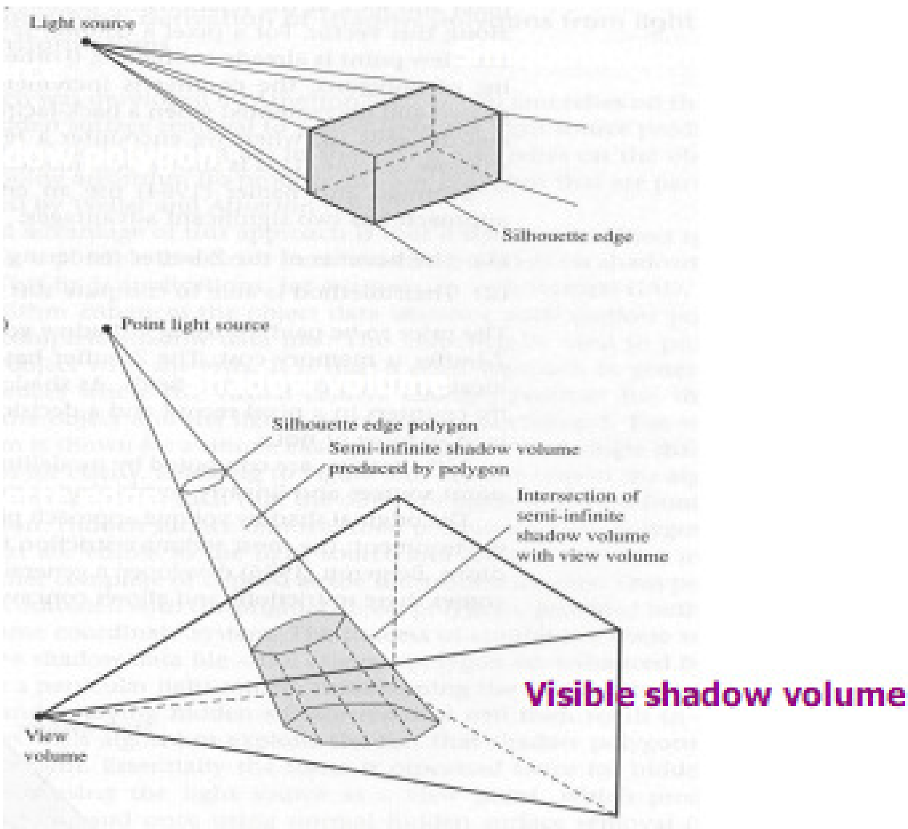


Figure 8.16 Example of shadow volume based on light source and view volume (Chuang, 2009)

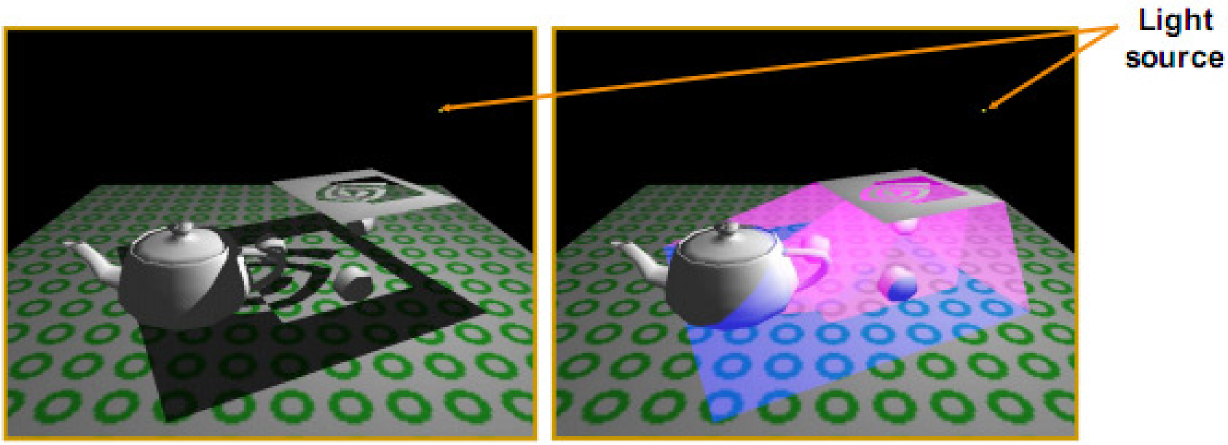


Figure 8.17 Example showing the final shadow on the left, and the shadow volume on the right as pink and blue (Chuang, 2009)

Another benefit from shadow volumes is that everything can be in shadow, including self shadowing. The stencil buffer is required to create shadow volumes, which is possible in both OpenGL and DirectX (Chuang, 2009).

The disadvantages with shadow volumes are that the light sources must be directional or point light sources, so an area light cannot be used to generate soft shadows using shadow volumes. Also, the shadow polygon must be closed and computations of the object silhouettes are required, which can be computationally hard for dynamic scenes (Chuang, 2009).

Tim Heidmann showed how to implement shadow volumes using the stencil buffer (Chuang, 2009). There are multiple ways to implement shadow volumes using the stencil buffer, and according to Wikipedia, they all have the same general approach described here ("Shadow volume," Wikipidia):

Assume the entire scene is in shadow and render it

For each light source:

1. Use the depth information from the scene to generate a stencil mask with holes where there are visible surface not covered in shadow

Render the scene again, this time lit, using the stencil information to mask the shadow areas, and use additive blending.

9 Test of implementation

The final review of the implementation includes a review of the initial conditions for the project. The project premise was to generate a real-time application that could generate shadows. The definition of real-time for graphics and human perception, is described by Möller (Möller & Haines, 2002). This definition cannot be generalized into other fields hence real-time is dependent on the implementation requirements, but will work as a general guideline.

> *"The rate at which images are displayed is measured in frames per second (fps) or Hertz(Hz). At one frame per second, there is little sense of interactivitity; the user is painfully aware of the arrival of each new image. At around 6 fps, a sense of interactivity starts to grow. An application displaying at 15 fps is certainly real-time; the user focuse on and reaction. From about 72 fps and up, differences in the display rate are effectively indetectable."* (Möller & Haines, 2002, p. 1) .

The second premise was to generate convincing shadows for human perception. Therefore testing the implementation objectively against real shadows is not of interest though other projects might find this premise interesting. The issue is that only the human perception of the shadows must be adequate to be defined as credible. There are a multitude of options for testing against credibility, the obvious being asking experts in the field or peers with an extensive knowledge of virtual spaces.

The requirements specified in the design document were 3 main functionalities:

1. An implementation that can track and process a known 3D object and generate up to 64 light sources using the median cut algorithm. Possible to select 2^N shadows, based on the computing power of the individual computer running the project.
2. A real-time graphics renderer to display the virtual objects and render the shadows.
3. An implementation that can be downloaded and used by laymen and designers using available materials. staying in the spirit of ARtoolkit making it a minimalist ease-of-use package.

Only the real-time graphics renderer openGL, specified, documented, and solid render architecture was a good functional choice. The remaining requirements should be tested to verify whether the implementation fulfills the requirements.

9.1 Test of framerate.

The framerate should from the definition in the introduction of this chapter be up and above 15 fps. The resolution of the input image is limited by the bandwidth of the usb 2.0 specification, this limit will be the highest framerate possible for the running implementation. The subsequent limits are set by the platform of execution, meaning the computer executing the application. The results will be compared with the framerates described by (Supan, et al., 2006, p. 5).

Platform specification:
The test was conducted using the windows operating system running on a:

Core 2 Duo T7500 -2.2 Ghz with 2.0 GB RAM, a 32 bit NVIDIA GeForce GO 8600M GT graphics card with 256 MB RAM, at a screen and camera resolution of : 800x600 pixels. Using a virtual test object *glutSolidSpere(40,20,20)*

9.2 Performance Results

Nr. of shadows	Framerate (fps)	Reference Framerate (fps) (Supan, et al., 2006, p. 5)
0	15.0	35.0
8	15.0	32.0
48		21.0
16	15.0	
32	15.0	
64	15.0	15.0
128	15.0	
256	15.0	
512	12.1	
1028	8.8	
2048	5.4	

9.2.1 Evaluation of results

The test shows that there is a limit in the hardware setup somewhere, the most obvious candidate causing the uniform framerates until 256 shadows, must be the usb-camera. In comparison with [2] it does not show a signifant difference in performance, our approach might be able to render substantially more shadows than the proposed method by Supan et alt.

9.3 Testing the credibility of the generated shadows.

There are several issues involved in both the physical model and the implementation where there is a lot of assumptions made to acquire results that are believable to humans. The issue that arises is the deliberate discarding of information that the approach uses. There is no detection of the underlying geometry and therefore the error will increase significantly especially if the geometry is different to a plane. This error can be avoided if the test image is casting shadows on a plane only. The images will be screenshots of the application in runtime.

9.3.1 Detecting a statistical difference from the HDR-shop implementation.

Testing whether there is significant difference between the two approaches. HDR-shops median cut algorithm estimates the light positions in an environment map. The implemented median cut algorithm is a simplification of the HDR-shop implementation and can therefore not be used as a direct performance reference method estimating the light sources. The implementation does not average the centroid by the luminescence distribution; therefore testing against this method is discarded.

9.4 Casual peer reviews.

The unstructured interview involves an indepth conversation with peers in the specific field of research the project is involved with (Sharp, et al., 2007, p. 298). The tests were two interviews & video presentations of the live application. This is an opportunity for the team to discover more about the application and the general project problem.

The final implementation has been evaluated by peer students at the department, interested parties have volunteered to see and review the results.

The interview form is based on a casual conversation to make the subjects feel comfortable and answer in an honest fashion to the best of their abilities. It ensures that no obvious issues have been overlooked and gives further insights into details of the implementation and its functionality.

The qualitative interview was structured around a presentation of a few videos

from the hardware test. What we sought to find was whether direction, length, softness and general impression of the generated shadow. The interview was in Danish, bear with us, we present the quotes and translate them to the best of our ability.

9.4.1 Results

First interview was with a fellow technical group, group 734, that has prior knowledge in the fields of image processing, augmented reality and 3D compositions.

> *"Den flakkede"*
> *(It flickered, about the shadow)*
> *"Hvorfor er skyggen ikke hård"*
> *(How come the shadow has no hard edges)*
> *"Vurderer I intensiteten på de enkelte lyskilder"*
> *(Do you evaluate the single light sources?)*
> *"Det ser meget cool ud"*
> *(It looks pretty cool)*
> *"Uden reference object set det fint ud, men der er noget med farven"*
> *(Without a reference object it looks just fine, however there is something about the colouring)*
> *"Skyggen virker kold"*
> *(The shadows seems cold)*
> *"Mætningen af skyggerne mangler noget"*
> *(The saturation of shadows are missing something)*
> *"Hvad med et median filter, modsat et running average for at fjerne støj"*
> *(How about a median filter, in contrast to a running average, to remove noise)*
> *"Hvad med at analysere et sort/hvid billede istedet for et farvebillede I environment mappet"*
> *(How about analyzing a black and white image of the environment map instead of a colour image)*

The second interview was with fellow student Tom Jensen, who is a skilled 3D modeler and has extensive experience working with lighting in virtual scenes.

"Vinklerne på lyskilder passer perfekt"
(The angles of light sources fits perfectly)
"Lagene af skyggen, kan den sættes efter lyskildens intensitet"
(The layers of the shadows, can they be set to the individual light source intensity)
"De hårde lys I demoen, hvad med at gøre antal skygger dynamisk med en slider mellem bløde og hårde skygger til brugeren"
(About the hard lights in the demo, how about making the number of shadows dynamic using a slider for the user to choose the hardness/shadows of the shadows)
"Den bløde skygge ser mere realistisk ud end den hårde. Den hårde ser digital ud"
(The soft shadow appears more realistic than the hard one. The hard shadow looks digital)
"Reference objektet viser den relative forskel mellem de to"
(The reference object shows the relational difference between the two)
"Højere alpha ved færre lyskilder, ville give mening"
(A higher alpha value when the number of light sources a are low, would make sense)
"En geometrisk marker, til at finde skyggernes intensitet ud fra"
(A geometrical marker, to track the shadow intensity could be useful)
"Hvorfor er skyggen blålig med en rund krans?"
(Why are the shadows blueish and with a round circumference)
"Hvad med stencil shadows og blande typen af skygger?"
(How about stencil shadows and mix the types of shadows)
"Generelt meget cool, der mangler små improvements"
(Generally very cool, only missing a few improvements)

9.4.2 Evalutation of results.

The results are indeed positive from our peers, the direction and length of the shadows seems to be coherent with expectations, there are however some problems with the coloring of the shadows and the number of shadows generated especially from a limited number of real world light sources. This problem can be avoided if as suggested in the second interview that the number of shadows could be set dynamically by the user – who knows what to expect, especially if there is reference objects. The presence of reference objects seem to make it harder to imitate the shadows, hence shadows are a very relative topic to all humans even experts. There is a distinct color bleed but as suggested in the first interview could be corrected by using a black and white image to sample intensity values. This approach could reduce the number of calculations needed in the environment map and could prove a useful optimization. Furthermore the composition of different shadow cast techniques could improve the implementation even further as suggested in the second interview.

9.5 Implementation equipment

Only accessible materials are used and users with a bit of technical flair should be able to achieve similar results. The light tracking marker consists of a table tennis ball painted with a glossy black paint that can be bought in most convenience stores. The implementation is based on ARtoolkit which is freely available, and any webcam will suffice.

9.6 Conclusion

The framerate of the implementation satisfies the real-time requirement, though there is no direct idea of the implementations runtime over 15 fps, this problem is due to hardware limitations from the usb-webcam protocol and the fairly high image resolution used in the test. The qualitative interview with peers showed that the implementation had the desired general impression and even tricked a few especially if there were no reference objects in the scene. A user test of the implementation could prove useful but would be a waste in the current phase of the project, hence the initial premise derived from the analysis of generating at least 64 shadows in real-time is achieved.

10 References

Adobe. (2009). *image editor | Adobe Photoshop CS4* Retrieved 12. Dec, 2009, from http://www.adobe.com/products/photoshop/photoshop/

Alpha blending. (Wikipidia). Retrieved 20. Dec, 2009, from http://en.wikipedia.org/wiki/Alpha_compositing#Alpha_blending

Andersen, M. S., Jensen, T., & Madsen, C. B. (2006). Estimation of dynamic light changes in outdoor scenes without the use of calibration objects. *18th International Conference on Pattern Recognition, Vol 4, Proceedings*, 91-94

955.

Angel, E. (2008). *Interactive Computer Graphics: A Top-Down Approach Using OpenGL* (5 ed.): Toronto: Addison Wesley.

Autodesk. (2009). Retrieved 21. Dec, 2009, from http://usa.autodesk.com/

Box blur. (Wikipidia). Retrieved 12. Dec, 2009, from http://en.wikipedia.org/wiki/Box_blur

Chuang, J.-H. (2009). Shadowing. Retrieved 20. Dec, 2009, from http://cggmwww.csie.nctu.edu.tw/courses/cg_eecs/2009/slide/6-shadowing.pdf

Cloverfield. (2007). Retrieved 14. Dec, 2009, from http://www.cloverfieldmovie.com/

Crow, F. C. (1977). Shadow algorithms for computer graphics. *SIGGRAPH Comput. Graph., 11*(2), 242-248.

Debevec, P. (2005a). *Image-based lighting*. Paper presented at the ACM SIGGRAPH 2005 Courses.

Debevec, P. (2005b). *A Median Cut Algorithm for Light Probe Sampling*. Paper presented at the International Conference on Computer Graphics and Interactive Techniques. Retrieved from http://gl.ict.usc.edu/Research/MedianCut/

Digital rebellion. (2009). Retrieved 14. Dec, 2009, from http://www.digitalrebellion.com/glossary.htm#C

Dobbert, T. (2005). *Matchmoving : the invisible art of camera tracking*. San Francisco, Calif. ; London: SYBEX.

Ekelund, K. L., Larsen, J. V., Laursen, J. S., Nielsen, D., & Pedersen, F. E. S. (2008). Fakta/Fiktion - undersøgt med augmented reality.

Facebook. (2004). Retrieved 12. Dec, 2009, from www.facebook.com

Feng, Y. (2008). Estimation of Light Source Environment for Illumination Consistency of Augmented Reality. *Cisp 2008: First International Congress on Image and Signal Processing, Vol 3, Proceedings*, 771-775.

Hartmann, W., Zauner, J., Haller, M., Luckeneder, T., & Woess, W. (2003). *"Shadow Catcher": a vision based illumination condition sensor using ARToolKit.* Paper presented at the Augmented Reality Toolkit Workshop, 2003. IEEE International.

Hasenfratz, J.-M., Lapierre, M., Holzschuch, N., & Sillion, F. (2003). A Survey of Real-time Soft Shadows Algorithms. *Computer Graphics Forum, 22*(4), 753-774.

HDR Shop. (2001). Retrieved 21. Dec, 2009, from http://www.hdrshop.com/

Jacobs, K., Nahmias, J.-D., Angus, C., Reche, A., Loscos, C., & Steed, A. (2005). *Automatic generation of consistent shadows for augmented reality*. Paper presented at the Proceedings of Graphics Interface 2005.

Kanbara, M., & Yokoya, N. (2004). *Real-time Estimation of Light Source Environment for Photorealistic Augmented Reality*. Paper presented at the Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02.

Kane, A. S. DOSE RESPONSE CONCEPTS. Retrieved 12. Dec, 2009, from http://aquaticpath.umd.edu/appliedtox/dose-response.pdf

Kato, H., Billinghurst, M., Morinaga, K., & Tachibana, K. (2001). The Effect of Spatial Cues in Augmented Reality Video Conferencing. Faculty of Information Sciences, Hiroshima City University; Human Interface Technology Laboratory, University of Washington.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., & Tachibana, K. (2000). *Virtual object manipulation on a table-top AR environment.* Paper presented at the Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on.
Kersten, D., Mamassian, P., & Knill, D. C. (1994). Moving Cast Shadows and the Perception of Relative Depth. Max-Planck-Institut für biologische Kybernetik.
Klein, G., & Murray, D. (2007). *Parallel Tracking and Mapping for Small AR Workspaces.* Paper presented at the Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on.
Lamb, P. (1999). ARToolKit Retrieved 20. December, 2009, from http://www.hitl.washington.edu/artoolkit/
Madsen, C. B. Using Real Shadows to Create Virtual Ones.
Madsen, C. B. (2003). The Importance of Shadows in Augmented Reality. Aalborg University.
Madsen, C. B., Sørensen, M. K. D., & Vittrup, M. (2003). Estimating Positions and Radiances of a Small Number of Light Sources for Real-Time Image-Based Lighting. *Eurographics*, 37-44.
Mamassian, P., Knill, D. C., & Kersten, D. (1998). The perception of cast shadows. [doi: DOI: 10.1016/S1364-6613(98)01204-2]. *Trends in Cognitive Sciences, 2*(8), 288-295.
Martin, G. (2004). Introduction to Sound Recording Available from http://www.tonmeister.ca/main/textbook/index.html
Matlin, M. W., & Foley, H. J. (1997). *Sensation and perception* (4th ed.). Boston, Mass. ; London: Allyn and Bacon.
Milgram, P., & Kishino, F. (1994). A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS. *IEICE Transactions on Information Systems, E77-D*.
Möller, T., & Haines, E. (2002). *Real-time rendering* (2nd ed.). Natick, Mass.: AK Peters.
Nakano, G., Kitahara, I., & Ohta, Y. (2008). Generating perceptually-correct shadows for mixed reality. *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 173-174.
Nielsen, M., & Madsen, C. B. (2007). Segmentation of Soft Shadows Based on a Daylight- and Penumbra Model.
Nordic-T. Retrieved 12. Dec, 2009, from http://nordic-t.org/
Pixar Animation Studios. (2009). *Feature Films* Retrieved 14. Dec, 2009, from http://www.pixar.com/featurefilms/index.html
RIT CIS - Center for Imaging Science. (2009). *Forced Choice* Retrieved 8. Dec, 2009, from http://www.cis.rit.edu/people/faculty/montag/vandplite/pages/chap_4/ch4p5.html
Robinson, L. (2009). A summary of Diffusion of Innovations. *Enabling Change* Retrieved 20. Dec, 2009, from http://www.enablingchange.com.au/Summary_Diffusion_Theory.pdf
Sato, I., Sato, Y., & Ikeuchi, K. (2003). Illumination from shadows. *Ieee Transactions on Pattern Analysis and Machine Intelligence, 25*(3), 290-300.
Shadow mapping. (Wikipidia). Retrieved 20. Dec, 2009, from http://en.wikipedia.org/wiki/Shadow_mapping
Shadow volume. (Wikipidia). Retrieved 20. Dec, 2009, from http://en.wikipedia.org/wiki/Shadow_volume
Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction design : beyond human-computer interaction* (2nd ed.). Chichester ; Hoboken, NJ: Wiley.
Supan, P., Stuppacher, I., & Haller, M. (2006). Image Based Shadowing in Real-Time Augmented Reality. *The International Journal of Virtual Reality, 3*, 1-7.
Texture mapping. (Wikipidia). Retrieved 21. Dec, 2009, from http://upload.wikimedia.org/wikipedia/commons/3/30/Texturedm1a2.png
WIKITUDE. (2009). Retrieved 20. Dec, 2009, from http://www.wikitude.org/developers
Williams, L. (1978). Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph., 12*(3), 270-274.
Wright, R. S., Lipchak, B., & Haemel, N. (2007). *OpenGL superbible : comprehensive tutorial and reference* (4th ed.). Harlow: Addison-Wesley.