

Rendering and Perception of Depth Cues on a Multi-Focal Plane Stereo Display

Nicholas G. Timmons
Downing College



*A dissertation submitted to the University of Cambridge
in partial fulfilment of the requirements for the degree of
Master of Philosophy in Advanced Computer Science*

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
UNITED KINGDOM

Email: ngt26@cl.cam.ac.uk

June 10, 2016

Declaration

I Nicholas G. Timmons of Downing College, being a candidate for the M.Phil in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: —

Signed:

Date:

This dissertation is copyright ©2016 Nicholas G. Timmons.
All trademarks used in this dissertation are hereby acknowledged.

Abstract

The aim of this project is to develop the hardware and software implementation of a multi-focal stereoscopic 3D display and perform research into the effects of visual depth cues on the perception of depth in a virtual scene. The results of which show interesting relationships between 2D and 3D depth cues, particularly binocular disparity, vergence and accommodation, and how they interact when other depth cues have been removed. We found that the accommodation depth cues from rendering to a multi-focal plane do not adversely affect the stereoscopic cues and may be able to enhance the sense of depth with other depth cues. A number of further experiments have been proposed to further investigate the effects of accommodation depth cues using multiple focal planes on 3D scene perception.

Acknowledgement

I would like to thank and express my sincerest appreciation to my supervisor, Dr Rafal Mantiuk, for his patience and guidance on this project without which I would not have been remotely prepared for. He encouraged me to use tools and systems I would otherwise have avoided and provided a very hands-on project, for which I am grateful as it has taught me a lot of useful skills. I didn't think I would need, or get the opportunity to play with a laser cutter as part of my project in a computer science department but he has provided a fun and definitely quite unique project for me to work on that let me do just that and many other fun things.

I would also like to thank so many people on the Advanced Computer Science course and members of the Computer Laboratory, especially the members of graduate tutorial office who were all more than willing run my experiment with me or offer wise advice when I was stuck, even at this hectic time of year.

Contents

1	Introduction	1
2	Background	3
2.1	Realism	4
2.2	Focal Length	5
2.3	Depth Cues	5
2.4	Standard Stereo VR Implementations	7
2.5	Vergence-Accommodation Conflict	8
3	Related Work	11
3.1	Multi-focal viewing	11
3.2	Depth Perception	13
4	Hardware	15
4.1	Display Requirements	15
4.2	Display Design	16
4.2.1	Display Panels	17
4.2.2	Display Configuration	18
4.2.3	Machine Specification	19
4.2.4	Known limitations	20
5	Software	23
5.1	Software Requirements	23
5.1.1	Projection Modes	24
5.1.2	Depth Configurations	27
5.1.3	Reflection Depth	31
5.1.4	Rendering costs	31
5.2	Software Configuration	33
5.2.1	Rotational Consistency	33
5.2.2	OpenGL Multiple Render Targets and Contexts	34
5.2.3	X11 Window Controller	35

5.2.4	Colour Calibration	36
5.2.5	Full pipeline	39
6	Methodology and Testing	41
6.1	Method	42
6.1.1	Depth Comparison Setup	42
7	Results and Discussion	53
7.1	Results	53
7.1.1	Erroneous data	54
7.2	Discussion	59
7.2.1	Lacking Stereopsis	59
7.2.2	Trends with No Depth Cues	60
7.2.3	Stereoscopic Cues	63
7.2.4	Multi-focal Depth	64
7.2.5	Extending our experiment	66
8	Summary and Conclusions	73

List of Figures

2.1	Basic comparison of the stereo and accommodation depths in a mismatched situation, such as a head mounted display.	8
2.2	This diagram shows the basic physics behind the effect accommodation has to bring near and far objects in and out of focus by converging light towards the retina.	9
2.3	This is an illustration showing the different configurations which can cause a vergence-accommodation conflict. Any of the combinations which show a mismatch between the image depth and the convergence point can cause discomfort for the user.	9
4.1	A high level view of the concept layout to achieve the stereo multi-focal images.	16
4.2	The front view of the display with the mirrors, beam-splitters and screens labelled.	18
4.3	This image demonstrated the limited field of view when using this method to display multi-focal stereo.	20
5.1	Example of negative parallax	25
5.2	Example of positive parallax	25
5.3	Symmetric simplified OpenGL matrix (OpenGL depth is in the range -1.0 to 1.0) [33]	25
5.4	Toe-in projection layout	26
5.5	Standard frustum OpenGL matrix [33]	26
5.6	Off-Axis projection layout	26
5.7	Oblique projection layout	27

5.8	This is an illustration of how the near and far planes have different images rendered on them for each pixel depending on depth. In this example the cube is having the nearest corner which is in front of the near plane rendered onto the near screen and then the parts of the cube which are beyond the far plane rendered onto the far screen. The points which exist between these two depths are blended between the two screens. When these two images are combined through the beam-splitter and presented to the user they appear once again as a whole image, however the appropriate parts of the image will now be at the correct focal depth.	28
5.9	This is an illustration to show the different of position relative to the eye position of the same depth when using projective depth instead of world space depth.	34
5.10	Equation for colour correction.	38
5.11	CIE XYZ to RGB colour matrices for each display panel. . . .	39
5.12	A high level overview of the steps to separate work between GPU's and display panels.	40
6.1	This is an example of the render modes we are using for the experiment and how they look on the screen. The multi-focal render modes split the depth of objects across the two places where as the ‘Near plane’ render modes only use the left and right near planes and as a result the rendered objects only appear on a single focal plane.	43
6.2	These photographs are from a run of the experiment showing a large offset between the two objects and displaying the images in the multi-focal render mode. As you can see by changing the focal length of the camera from 0.54m to 0.81m we are able to selectively focus on both objects and there is a large visible difference when that changes. The images were taken with an aperture of f/1.4 which is larger than the aperture of the human eye which gives this image a softer focus than what is experienced by a user of the display.	44
6.3	This image shows the error from incorrect geometric alignment. It shows the far plane out of alignment with the near plane and the ‘shadowing’ artefact becomes apparent. When this happens it is very clear to a user that the image is comprised of two separate planes. We aim to reduce the occurrence of this in our experiment using a calibration screen and frequent updates.	44

6.4	This is an example of the deformed sphere models that are being used in the experiment. They are scaled within the scene based on depth from the camera so they maintain the same size in screen space and are then scaled by 5% up or down in the scene to help reduce any false cues from perspective.	45
6.5	This diagram different offset locations and the relative distances between them. This diagram shows the right object at different locations but in the experiment either blob can be in the offset position but one is always on the far plane.	47
6.6	These graphs show the binomial graph of expected error.	49
6.7	Software Geometric Calibration: The near plane calibration screen shows a central point and two ‘L’ characters. The point helps in centering the image at the correct position and the ‘L’ is a simple way to confirm the orientation of the screens are correct. The far plane has a series of cross marks to align with the near plane as well as a central point to align.	50
7.1	(Part 1 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean. The point of subjective inequality is marked in red at $y = 0.5$	55
7.2	(Part 2 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean.	56
7.3	(Part 3 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean.	57
7.4	(Part 4 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean.	68
7.5	This graphs shows the combined results of all participants for all rendering modes and offsets.	68
7.6	These graphs show the separate results of participants who were or were not able to perceive depth from only stereographic cues.	69
7.7	Personal runs of the experiment	70

List of Tables

2.1	Depth cues in two dimensional images.	6
2.2	Depth cues in stereoscopic and/or multi-focal images.	6
3.1	Interocular distances from the US Army data [1]	13
4.1	iPad Display Panel Specification	17
4.2	Controlling Machine Specification.	19
5.1	Memory consumption of the display buffers used in this display. The large linear colour buffers used for rendering the scene are the most costly. We are using a significant portion of the GPU memory with just the rendering pipeline. If we were to test a more complex scene this could become prohibitive and accuracy of the colour calibration transform might need to be sacrificed.	33
5.2	Black levels - Representing the light leaking through the LCD panels when attempting to display a completely black screen. .	38
5.3	Gamma correction values for each display panel.	39
6.1	This is the distance configurations the display has been setup to use in this experiment. It approximately maintains the optimal 0.6 dioptres between focal planes so that the difference in focal depth should be very perceptible to the user.	45
6.2	This table shows us the four possible combinations of the render modes we will be testing and which cues they allow. We are particularly interested in the combination of ‘Near Plane’ and ‘No Disparity’ as this combinations will show us how well we have isolated other depth cues. If that combination is still able to be accurately judged for its depth then there is other depth cues available or we have failed to block the ones we wanted.	47

- 6.3 This table shows the offset positions in distance from the camera. The depth is central position of the object. On the far plane at zero offset the value is not 81cm as we have adjusted for the size of the object and random scale to ensure that is entirely behind the far plane in its starting position. 48

Chapter 1

Introduction

The current solution for virtual reality and 3D experiences is to render a scene with binocular disparity for two displays which are presented to the eyes separately to form stereo pairs and give the user a sense of depth. This has been very effective for presenting compelling experiences but has been known to induce discomfort and visual fatigue in some people. This is a well understood problem but there is less information available on possible solutions.

One proposed solution is to partially solve the vergence-accommodation problem [21] through the use of multiple focal planes[9]. These displays use multiple screens per eye to represent screen depth instead of just the one to present the user with multiple slices of the scene at different depths. This way the user is able to correctly converge and focus at the depth that the object is presented as existing at.

Our aim in this project is to construct a functional multi-focal stereo display and supporting software to investigate the effectiveness of multiple focal planes for accurately perceiving depth cues. We want to investigate if through the introduction of accommodation depth cues to a stereoscopic display we can improve the accuracy of depth perception by presenting depth cues which more closely represent the depth cues we use in the real world.

The results of our experiment with the display do not conclusively support our hypothesis but do show some interesting behaviour of the complicated relationship between the different depth cues and gave enough information for us to be able investigate more thoroughly in future experiments. We can say conclusively that introduction of an extra cue does not hinder the effectiveness of other cues and for some people did give amplified results.

This report includes a background of current techniques being used for VR and a short overview of similar methods using multi-focal planes (Section 2 and 3). There is then a breakdown of the requirements and techniques implemented in software and hardware for this multi-focal display with details of how they have been configured to achieve our results (Section 4 and 5).

We also detail our experimental methods and results (Section 6.1 and 7.1) before summarising our findings and discussing future work which could take our research further (Section 7.2 and 8).

Chapter 2

Background

Stereo displays such as those used for showing 3D movies and in HMD (head mounted displays), are in use in many commercial products and have seen significant improvement over the past decade but still have some large technical and usability problems related to rendering techniques which could restrict the wide-scale appeal of the products [20] and do not yet successfully mimic all the visual cues that the visual cortex processes when viewing objects in the real world.

When considering the comfort of the user, the results of many psychometric and usability studies have shown that the current implementations of binocular disparity using stereo rendering can lead to problems and discomfort from a problem known as “Simulation Sickness” [2]. These symptoms include distortion of perceived depth [11], short term degradation of the oculomotor response[7], diplopia vision (commonly known as double vision) [6] and can cause a lot of visual fatigue [3].

There are many factors contributing to cause these conditions varying from low quality images, such as the high persistence in the early Oculus Rift displays [30], or incorrect interocular distances to having no comfortable point to allow the eyes to rest. A major cause that is often mentioned is the discrepancy between accommodation and vergence when using displays

with a fixed real focal distance. In this context accommodation refers to the focusing of the lens in the eye on objects at different distances and vergence is the motion of eyes rotating to bring the convergence point of the optical axis to intersect at the distance of the object. These oculomotor actions are coupled, they work together when looking at an object in the real world but do not function correctly when decoupled due to being shown objects with stereo correspondence at one depth and vergence correspondence at another - which is the general case for objects shown on standard stereo head mounted displays (see Figure 2.3 and 2.2).

Past research has suggested [11] that breaking of the link between accommodation and vergence cues can lead to a reduced perception of depth, which will affect how the user understands the space they are looking at and could have an adverse effect on the perceived realism of the space as the scale may appear inconsistent with what the user is accustomed to. This is of particular importance to artificial reality environments where the virtual environment is mixed with the real world. In this situation the depth cues from the display would have to match with those from the real world in order to maintain consistency to not induce the conflict.

2.1 Realism

In the context of this research we are considering realism to consist of simulating all of the separate visual cues. In a perfectly realistic scene the user would not be able to tell the difference between looking a scene in the real world and one that is rendered as the cues would all match.

To achieve this, we would want a high quality rendering of a scene with correct lighting and reflectance within the full colour and intensity range that the human eye can perceive as well as being seen as 3D to the user through correct visual and depth cues.

Within the limited scope of this project we will not be able to develop all of those features but will be focused on improving the realism of a rendered

scene through the use of more sophisticated depth cues to measure whether that improves how real the depth in the scene appears to the user when compared to a scene which is lacking such cues.

2.2 Focal Length

As we will be discussing multi-focal displays it is important to clarify what the focal length is. In this context the focal length is the measure of the distance to the point that an optical system converges. The unit for Focal Length is the Dioptre which is measured as the reciprocal of the distance in meters to the convergence point.

$$X_{\text{Dioptries}} = \frac{1}{\text{Distance}_{\text{Meters}}} \quad (2.1)$$

2.3 Depth Cues

The human visual system has many cues for determining depth. Some that are visible in two dimensional displays (Table 2.1) and some that are only possible with stereo or multi-focal displays (Table 2.2). To attain maximum realism in the images shown the visuals would have to be delivering all of these cues [31].

For most people ‘Binocular Disparity’ is a very important depth cue [8] which is relatively easy to replicate with two screens and as such is the cue that is most well represented by consumer 3D products at the moment. However, conflicts with other cues can cause detrimental effects to the visual cortex’s ability to form a single image and extract the depth like it would when viewing the real world. Patterns such as repeating identical vertical bars are particularly problematic for finding stereoscopic matches without other cues and this can lead to visual strain and fatigue.

Perspective	Objects are scaled based on the distance to the eye.
Known Sizes	We can judge relative depths of objects from known sizes. For example, if an image shows a football and a house as the same size we assume the house is further away.
High Frequency Detail	We assume when we can see more small details when the object is closer to the user.
Occlusion	An object that occludes another is closer than the object it is occluding.
Lighting	The human visual system is accustomed to seeing objects under different lighting conditions and can reason the position of objects in a scene by the lighting and shadowing between them[19]. It is also able to judge distance by the slight dimming of objects in the distance due to atmospheric scattering. These are part of the group known as pictorial depth cues
Relative motion/Parallax	An object closer to the camera appears to move across the view faster than one further away.

Table 2.1: Depth cues in two dimensional images.

Binocular disparity	The difference in the apparent position of objects under projection caused by the interocular distance between the two eyes. (<i>Stereo Only</i>)
Accommodation	Changing the focal length of the lens in the eye to focus on an object at a particular depth. (<i>Multi-Focal Only</i>)
Convergence	Bringing the two images from each eye to overlap more coherently and face a specific point at a particular depth through rotation of the eyes to converge at the distance of the object.

Table 2.2: Depth cues in stereoscopic and/or multi-focal images.

2.4 Standard Stereo VR Implementations

The current standard for VR in a number of commercial hardware implementations consists of a head mounted display with the screen split vertically to display a separate image of the scene for each eye and a lens to distort the image to increase the amount of the screen that is visible to reduce the “Screen Door Effect” [24] caused by low pixel density and visible lines between them. The software is then implemented using interocular distances of the screen in the display with an appropriate field of view and a standard Off-Axis parallel projection for each eye (see Section 5.1.1).

The correctly calibrated projection of the same scene for the two separate views gives a feeling of depth and “realism” through the user picking up on stereo-correspondences from the disparity in the two images and perceiving them as a single object at the correct distance.

Since the screen that is being used is a fixed short distance from the users eyes at all times the user is always focusing at a fixed focal distance which is different than the identified distance for the motion and stereo correspondences that are being shown in the scene.

This triggers the Vergence-Accommodation conflict from the conflict in visual depth cues. There are examples of trying to use simulated Depth of Field blur in early VR and large screen 3D to try and simulate one of the missing visual cues but this was found to induce “Simulation Sickness” and is it now strongly advised against [32].

There is currently a push in 3D technologies to try and increase the frame rate of applications that are using these techniques beyond the normal high bar of 60 frames per second. This has been seen as a way to reduce the discomfort some users feel from uneven or clunky motion caused by low or varying frame-rates [27]. As resolutions and visual quality increases, making it harder to achieve, this is becoming a particularly important area to consider when developing for these technologies.

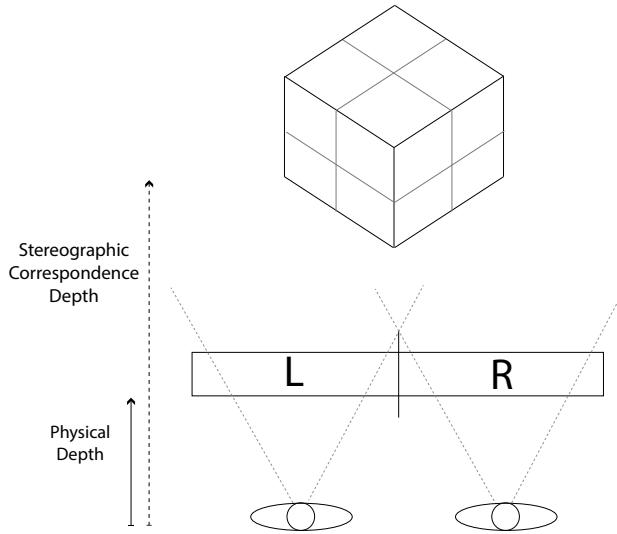


Figure 2.1: Basic comparison of the stereo and accommodation depths in a mismatched situation, such as a head mounted display.

2.5 Vergence-Accommodation Conflict

The Vergence-Accommodation conflict is a source of discomfort and disorientation for a lot of users of stereo displays [13]. It is caused by a mismatch of the focal depth and convergence depth cues. It is a problem which originates from trying to simulate depths of objects which have a different focal depth in software leading to correct stereoscopic disparity between eyes but an incorrect focal depth for those objects. This is particularly a problem for standard stereo displays due to the use of a single focal depth because of the fixed display panel position.

This conflict can cause eye strain, headaches and dizziness in some users and is strongly associated with “simulation sickness” [32].

While this conflict does have some effect on the comfort of the user [20], in the area we are looking at we are more interested in the effect this has on how the users perceives the distances to objects in the world when this conflict is reduced.

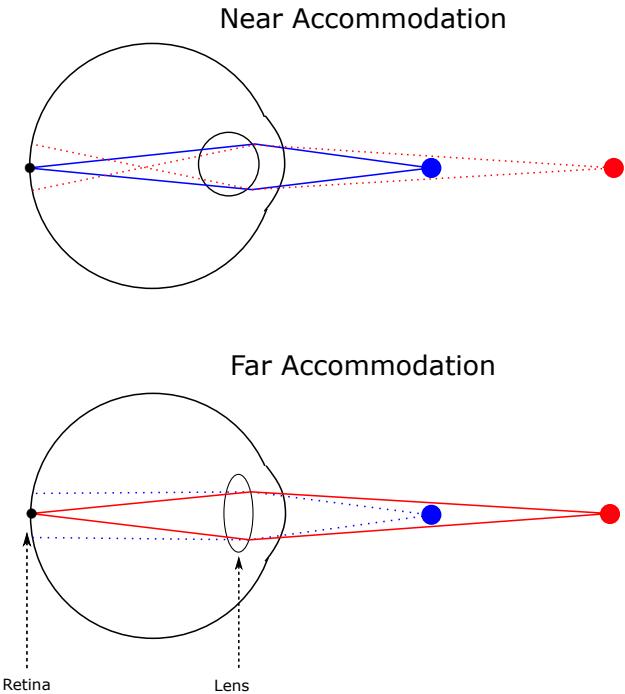


Figure 2.2: This diagram shows the basic physics behind the effect accommodation has to bring near and far objects in and out of focus by converging light towards the retina.

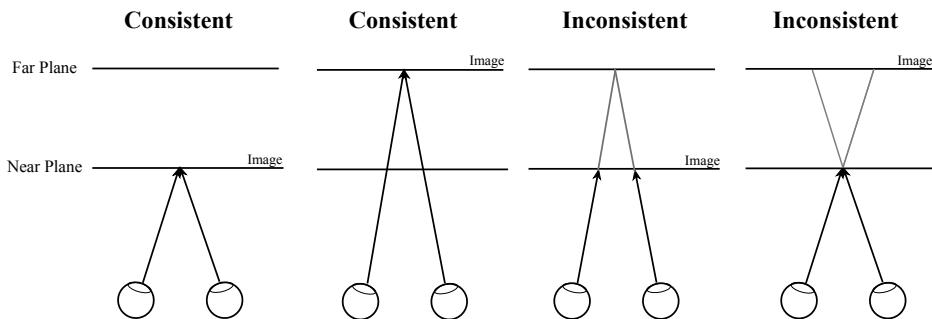


Figure 2.3: This is an illustration showing the different configurations which can cause a vergence-accommodation conflict. Any of the combinations which show a mismatch between the image depth and the convergence point can cause discomfort for the user.

It appears that when these visual cues are mismatched the users ability to judge the distance to objects becomes limited, vision can become less clear and the speed at which stereoscopic correspondences are matched is

decreased. These symptoms can all interfere with the stereoscopic technique as it reduces the ability of the user to create an internal model of the scene correctly.

If the user is unable to judge the depth and position of objects clearly in the scene then it would not be comparable to the real-world and therefore, not realistic.

One of the main aims of using multi-focal cues is that it will reduce the confusion of depths and allow it to be properly processed the same as any real-world scene.

Chapter 3

Related Work

3.1 Multi-focal viewing

Multi-focal displays aim to simulate real world focal lengths to provide correct depth cues to the user. It is an area that has been investigated for over a decade but has recently became more popular with the rise of VR as a viable commercial platform due to the problems the single focal displays cause.

The idea for multi-focal viewing is to use multiple displays, or sections of displays [14] and a series of lenses, mirrors and beam-splitters to combine images at different depths into a single image for the user to see. The different distances to the screens will then give the user different focal points to focus on that will behave like real screens at those distances.

Through clipping the viewed images to have only the sections at the correct depths displayed on each screen you can effectively create scenes with discrete visible focal depths.

These screens were used to investigate the comfort and physical reactions of the human eye when viewing simple scenes [9] and were found to increase the comfort of the user and reduce the negative effects of single focal plane stereo rendering solutions.

A downside to this method is that it is very sensitive to calibration and view alignment as the images need to align correctly to create the illusion of a single image, see Figure 2.1. This means that displays have to be invariant to user motion and match the physical position and separation of the users eyes at all time to avoid unwanted parallax between planes.

Alternative, more complex, methods have been suggested such as a lenticular display setup [22], however, like many autostereoscopic methods, this method suffers from crosstalk which you do not get in simpler designs. Other alternative designs include methods which make use of the active stereo techniques to limit the number of screens required [15] but this has its own implications on the frame rate and how active a scene can be as well as restricting the user to wearing special glasses, which can be a problem for users who already wear corrective lenses.

For less complex display models, it is possible to avoid the eye tracking by using fixed focal depths and designing to accommodate different eye positions - which can be quite varied among a population.

There was a study carried out by the US Army which investigated the interocular distance of its members and gave some very good data on the topic [1]. They showed a mean distance of 6.47cm for men and 6.23cm for women, see Table 3.1. Using this data work has been carried out to support many users with varying interocular distances with adjustable hardware [17]. There was a more in depth analysis of the mean interocular distances which took a more widespread look at the general population, including children [10] which are not represented in the US Army study. The general results within adults fell in line with what we see from the US Army work. As we are testing on adults we will be using the data for adults only to configure our display.

The downside to of multi-focal displays is that the user is required to maintain a calibrated head position through out to keep the effect working. To prevent the movement of the user causing problems some very *interesting* solutions have been used in research such as bite-bars to bite down on to prevent the users head from moving [9]. However, these are a little impractical for a study

covering multiple users which will need to support multiple focal distances and interocular distances.

	Male	Female
Mean	6.47cm	6.23cm
Min	5.20cm	5.20cm
Max	7.80cm	7.60cm

Table 3.1: Interocular distances from the US Army data [1]

3.2 Depth Perception

There have been a number of papers investigating the effect of stereo displays on perceived depth. One such study performed an investigation into the effect of different fields of view when using a stereo setup and how that changes our interpretation of the depth which points to the conclusion that relative scale and rate of change of scale is important in our perception of depth.

In one experiment [26] they were able to show that a user in the correct configuration could correctly navigate to a specific physical scene based on interpreting the depth and positions of objects in a 3D scene using a head mounted display, where as a user who was not calibrated would have difficulty due to “minification”.

This shows the ability to accurately determine depth and position of objects is affected by the accuracy of depth cues, whether they are from stereo correspondences, projection or otherwise. This gives us reason to believe that the accuracy and depth perception can be improved by the introduction of further depth cues.

Chapter 4

Hardware

The goal of this implementation is to create a display which we will be able to use to investigate the relative perceived realism of rendered scenes by measuring the effectiveness of depth perception when configured to give specific focal depth cues compared to standard stereoscopic rendering and allow it to be easily configurable for more specific tests in the future.

4.1 Display Requirements

The display is designed to allow us to investigate perceived depth for this work but also to be configurable to support future work beyond this project.

This project requires that the display supports four high resolution, good quality displays which are able to be viewed at configurable focal distances in a stereo setup that allows the user to view two screens per eye and support a wide range of users. This is to allow the testing of combinations of zero disparity, stereoscopic and multi-focal scenes.

To support male and female users it should be configurable to support both adult male and female average interocular distances, see Figure 6.1.

We have been provided with a pre-assembled frame with slots for the display

panels, mirrors and beam-splitters. We will be assembling any additional components such as viewing goggles, coverings and supporting frames as needed.

4.2 Display Design

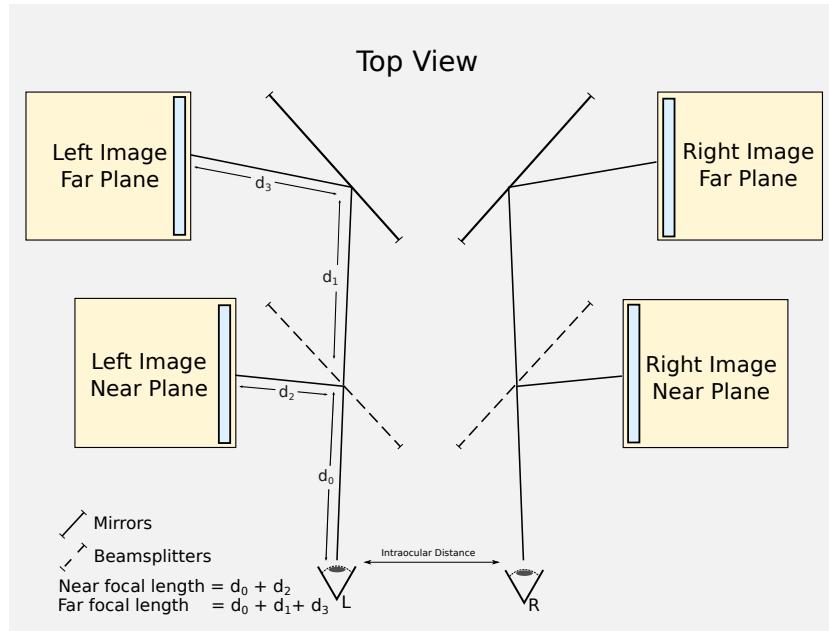


Figure 4.1: A high level view of the concept layout to achieve the stereo multi-focal images.

To achieve the requirements of Section 4.1 we will be using four high resolution displays to get maximum quality in rendering per view and reduce visible problems from low pixel density. Full details of the panels we will install in Table 4.1.

These will be configured with two screens per eye. One at the distance of the required near plane and one at the distance of the required far plane. The rendered images are seen through a combination of two mirrors and a 50/50 beam-splitter which merges the two images into one to be displayed to the user. See Figure 4.2 for a view of the physical layout.

As the image is merged through the beam-splitter it is essentially additively combining the colour values of the two screens. To ensure a clear image of both views we need to isolate the beam-splitter, mirrors and all screens from external light to prevent obscuring or offsetting colour as it is merged towards the eye. To achieve this, we will be shielding the constructed display with matte black boards to prevent any light entering the display from outside and to reduce internal reflectance.

An extra consideration for combining the images and maintaining stereoscopic correspondences is that the images being displayed on each screen must match in colour range and intensity when they reach the eye. This will mean that the displays will have to be calibrated for any differences in the displays as well as error from light absorption from the mirrors or beam-splitter. Additionally, this will also help mitigate the error from any external light or internal reflection we are not able to remove.

4.2.1 Display Panels

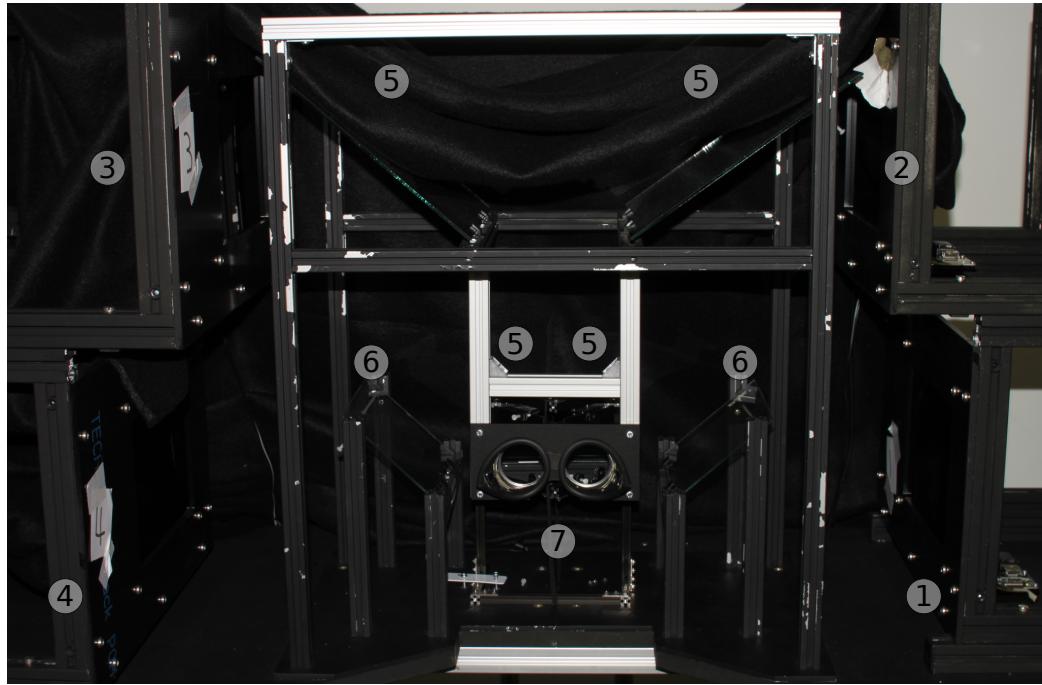
The display panels used in this display are replacement screens for the iPad 3. These have been chosen due to the high pixel density and good colour reproduction for the price point.

These displays are also a good size for our test setup. Any larger screen would have made the desk prohibitively large and impacted other features we would like to look into in the future, such as HDR support.

Resolution	2048 x 1536
Width	169.5 mm
Height	240.0 mm
Type	LED-backlit IPS LCD
Colours	16 million
Model Number	LP097QX1 6091L-1579A

Table 4.1: iPad Display Panel Specification

The panels are mounted into the display and fixed in place on the configurable



① Screen 1 (Right near plane) **③** Screen 3 (Left far plane) **⑤** Mirror
② Screen 2 (Right far plane) **④** Screen 4 (Left near plane) **⑥** Beamsplitter
⑦ View goggles

Figure 4.2: The front view of the display with the mirrors, beam-splitters and screens labelled.

near and far planes. They are powered and controlled through a 3rd party control board and each screen is connected to the PC using *mini display port* with the two left screens being connected to one GPU and the two right screens to the other.

4.2.2 Display Configuration

The display has ten configurable components to allow support for varying distance and angle from the screens to the eye.

The screens themselves are on fixed beams that allow the screens to slide to

be nearer or further from the mirrors so the total distance to each screen can be easily calibrated.

There are two mirrors and a beam-splitter for each side of the display, see Figure 4.2. All three components can be rotated and skewed to reach alignment. Additionally, the mirrors that are placed directly in-front of the user can also be translated to account for offsets in eye position.

4.2.3 Machine Specification

Although this display will work on any machine supporting 4 *mini display ports* it is worth mentioning the setup we are using and our reasoning to allow for easier replication.

CPU	Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz
GPU	2 x NVidia 970
Memory	16GB
OS	Ubuntu 14.04 LTS

Table 4.2: Controlling Machine Specification.

CPU The CPU in our test machine is a little over specification for what is required as we will only be running simple scenes, however, we want to ensure that any analysis or logging applications we are using alongside the main multi-focal program will not interfere with the performance of the display. It also allows for future research which may involve more intensive video processing that may not be suitable for the GPU.

GPU This machine has two GPU's to allow us to attach many monitors. In this case we have four display panels for the multi-focal display and one monitor attached to control the testing. In the future we would like to expand this to allow for up to another four outputs to possibly allow the iPad panels to be used as High Dynamic Range displays using additional projectors [12].

We have quite powerful GPU's due to the need to render to multiple screens at quite a high resolution which requires a lot of memory and to process potentially very complex scenes.

OS We have chosen to use Ubuntu on this machine to maximise the support for research projects in the academic community and maintain compatibility with other projects in the Cambridge University Computer Laboratory.

4.2.4 Known limitations

Limited Field of View. As our screens will only cover a limited field of view when compared to the nearly full coverage of HMDs (see Figure 4.3), so we are limited in our ability to give as full an immersive effect, rather for full scenes it will look like peering through a window into another room than the full VR experience people may be accustomed to. This could be improved in a less experimental setup through the use of lenses.

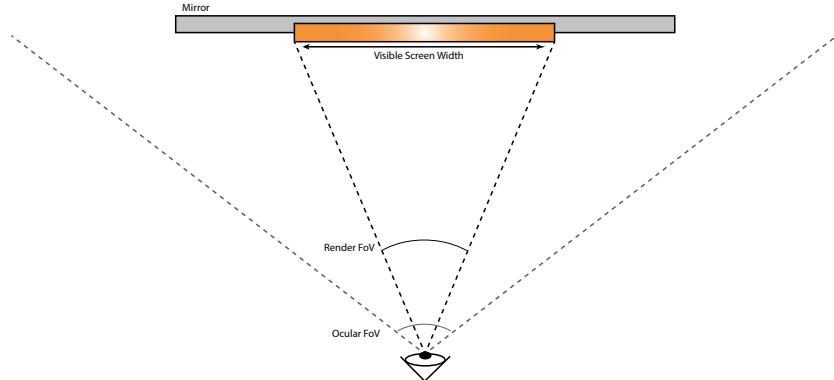


Figure 4.3: This image demonstrated the limited field of view when using this method to display multi-focal stereo.

iPad Displays While working with the displays we have found that the glossy coating on the screen can cause a problem within the display as from some angles within the display there is a clear reflection of other screens. This can be mitigated with more obstructing panels to prevent it interfering

with any testing but it raises the potential of improvement by using matte display panels in the future, although this would limit the quality of the colours displayed on the panel.

Relative Head Position As this display is not physically attached to the user they are able to slightly move their head which will cause parallax from misalignment with the screens. This is especially true when using all four displays as small head motions will result in a large change of position of the near screen from the user's perspective. To alleviate this, extra consideration will have to be made in software design and experimentation to ensure correct calibration throughout the use of the display.

Large display. The users view eye coverage is limited by the size of the screen at the furthest distance. This means that if we want to have a display which covers all of the user's view then the screen would have to be of the correct size to cover the full view at the given focal distance. Because we are using iPad panels in our display we are limited to quite a small view of the scene due to the limited size. With a more complex setup using custom lenses it would be possible to reduce this problem but it would be more expensive and less easily modified and is therefore out of the scope of this project.

Benefits of head mounted displays As this display is desk mounted instead of head mounted we lose the ability to do head tracking and rotation which greatly help the user by giving subtle relative motion depth cues. The fixed view point means we need to use motion of the scene rather than motion of the user to gain these cues. Motion of the camera in VR without actual head motion can cause motion sickness from mismatching signals [27]. If user movement was possible it would have been particularly useful for measuring how small motions may help determine depth.

Varying resolutions Due to the cameras being at physically different distances and no use of lenses to increase the size of the displayed screens

we have a reversal of the ideal resolutions. In the best case we would want the objects which are appearing closest to the user to have the highest pixel density and the objects further away to have the lowest. In our display the image which is closest is scaled down to match the distant screen and as such is only using a small portion of the possible resolution.

Chapter 5

Software

5.1 Software Requirements

Rendering As this implementation will require the software to output four high resolution configurable planes each showing a real-time 3D scene on four separate screens controllers by two GPUs. We will need to implement and test different blending and projection modes. This will all require the use of OpenGL as we are using Ubuntu as our operating system (see Section 4.2.3).

Experimental Support The software will need to be able to support real-time configuration of the screen position within each display panel to allow it to support multiple different users when running experiments. This will mean accounting for error from head position and slight motion during the test. It will also be required to support and switch between different rendering modes without significant delay to allow for comparison of different techniques.

Input & Output The software will be required to be able to send keyboard input to separate windows to support configuration during any exper-

iments. It will also be required to store experimental data and configuration preferences.

5.1.1 Projection Modes

When rendering stereo pairs there are a number of techniques available that each have unique features. The aim of the stereo pair is to simulate the viewing conditions of the user in the real world To achieve this these methods use a projection plane at a fixed distance from the cameras and the cameras have different offsets and sometimes view directions. At the distance to the projection plane there should be zero vertical or horizontal parallax between the left and right eye.

Vertical parallax is generally avoided due to it causing discomfort and effecting stereo correspondence [4]. This error can break the stereoscopic effect, cause diplopia and cause discomfort in the user. However, a certain amount of *horizontal parallax* is required for the stereo effect to work. This comes from positive (Figure 5.2) and negative parallax (Figure 5.1). Positive parallax is when the point is beyond the projection plane and negative is when the point is in front of the projection plane. Generally, points in positive parallax are more comfortable to look at and we aim to avoid too much negative parallax as it can quickly become extreme or cause clipping from the screen when the object moves very close to the eye position.

For early testing of the display we are going to support three main projections of the scene, ‘Toe-In’, ‘Off-axis’ and ‘Oblique’ and then select the appropriate method to use in our experiments.

Toe-in This projection mode points both cameras at a single focal point. It produces reasonably correct stereoscopic vision on the projection plane but points in front and beyond that plane and particularly towards the left and right side of the image suffer from quite severe vertical parallax caused by the different rate of change of position due to the non-parallel projection

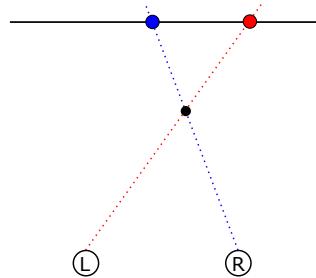


Figure 5.1: Example of negative parallax

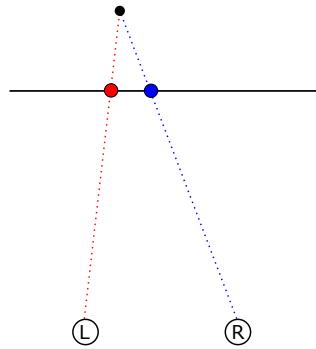


Figure 5.2: Example of positive parallax

planes. This conflict is increased when the field of view of the viewing camera is decreased as the difference in depths between the left and right images become more prominent nearer to the centre of the image.

$$\begin{bmatrix} \frac{near}{right} & 0 & 0 & 0 \\ 0 & \frac{near}{top} & 0 & 0 \\ 0 & 0 & \frac{-(far + near)}{far - near} & \frac{-2far * near}{far - near} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (5.1)$$

Figure 5.3: Symmetric simplified OpenGL matrix (OpenGL depth is in the range -1.0 to 1.0) [33]

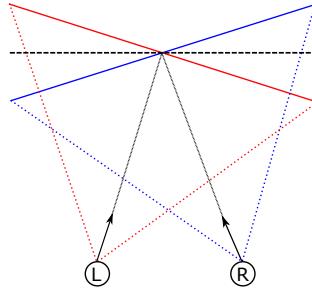


Figure 5.4: Toe-in projection layout

Off-Axis Off-Axis projection corrects the non-parallel projection planes of the ‘Toe-In’ method and forces the camera view direction to be parallel. These changes remove all vertical parallax making for a more comfortable viewing experience. This method requires the creation of non-symmetric camera frustums which can look incorrect when viewed in non-stereo individually. An added benefit of this method, which is used in 3D cinema experiences, is that it can alter the interocular distance to exaggerate depth to an extent.

$$\begin{bmatrix} \frac{2near}{right - left} & 0 & \frac{right + left}{right - left} & 0 \\ 0 & \frac{2near}{top - bottom} & \frac{top + bottom}{top - bottom} & 0 \\ 0 & 0 & \frac{-(far + near)}{far - near} & \frac{-2far * near}{far - near} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (5.2)$$

Figure 5.5: Standard frustum OpenGL matrix [33]

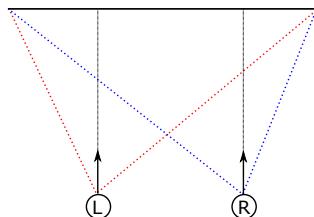


Figure 5.6: Off-Axis projection layout

Oblique Our oblique projection is very similar to the Off-Axis projection in that it also removes vertical parallax on the projection plane by sharing the same plane between the two eyes, but it has the added benefit of better modelling vergence for the central position on that plane. It keeps the benefits of the Off-Axis projection while emphasising depth due to the converging eye direction. This leads to increased disparity between the left and right image as objects move towards and away from the projection plane. In our implementation the oblique projection is an accurate model of vergence at the projection plane and matches our non-parallel optical axis in the display shown in Figure 4.1. For the construction of this matrix see the Appendix.

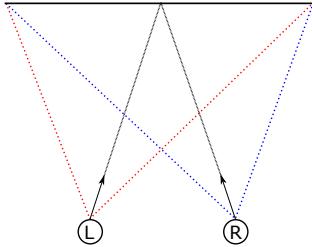


Figure 5.7: Oblique projection layout

5.1.2 Depth Configurations

The scene is being modelled in real world units to easily match our physical configuration and allow for comparison in the future to real world scenes such as those taken from light field cameras.

When rendering the scene all the points closer than the near focal distance will be displayed on the near screens and the points beyond the far focal distance will be displayed on the far screens.

For the points which lie in the distance between the two planes we do not have a physical screen to display them at the correct focal distance so we will try different blending methods to represent the points partially on each screen and determine if it is possible to convince the user that these points exist at the appropriate focal distance between planes.

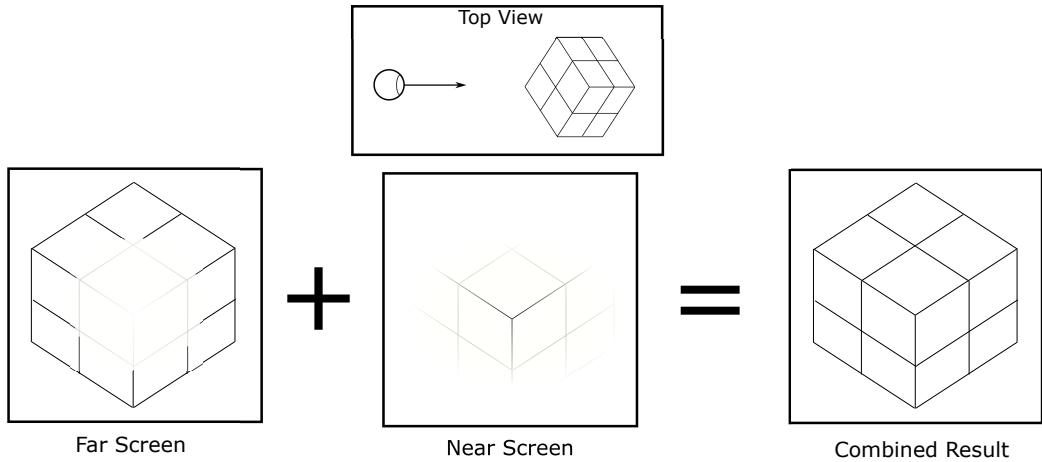


Figure 5.8: This is an illustration of how the near and far planes have different images rendered on them for each pixel depending on depth. In this example the cube is having the nearest corner which is in front of the near plane rendered onto the near screen and then the parts of the cube which are beyond the far plane rendered onto the far screen. The points which exist between these two depths are blended between the two screens. When these two images are combined through the beam-splitter and presented to the user they appear once again as a whole image, however the appropriate parts of the image will now be at the correct focal depth.

As we are using beam-splitters which work additively we display black for points which are being displayed on the other screen and not within the blend depth range.

The primary methods we will try are:

Box: All points less than half way across the distance between focal lengths will be considered on the near plane and all the ones more than half way across will be considered fully on the far plane.

$$n = \frac{n_i - n_1}{n_2 - n_1} \quad (5.3)$$

$$col_{out} = f(n) = \begin{cases} col_{near} & \text{if } n \text{ is } < 0.5 \\ col_{far} & \text{if } n \text{ is } > 0.5 \end{cases}$$

We expect this to produce the effect of the scene feeling like it is made out of two pictures shown to the user. As there is a clear “focal seam” where the edges of each depth are visible and this is exacerbated by any calibration error.

Linear: As the points move across the middle space they will be linearly interpolated between the two views.

$$n = \frac{n_i - n_1}{n_2 - n_1} \quad (5.4)$$

$$col_{out} = (n * col_{near}) + ((1 - n) * col_{far})$$

Non-linear: As the points move across the middle space they will be non-linearly interpolated using a modified sigmoid curve between the near and far plane.

$$n = \frac{n_i - n_1}{n_2 - n_1}$$

$$blend = \frac{1}{1 + exp((-n * 2 + 1) * 6))} \quad (5.5)$$

$$col_{out} = (blend * col_{near}) + ((1 - blend) * col_{far})$$

Projective: In this mode the calculation of the depth interpolation value through the blending area is scaled inversely to match the depth scaling in the perspective projection transform.

$$n = \frac{(1/n_i) - (1/n_1)}{(1/n_2) - (1/n_1)} \quad (5.6)$$

$$col_{out} = (n * col_{near}) + ((1 - n) * col_{far})$$

Fixed: Fix all to either the near or far plane. This method exists as a way to test and configure the views and will only be used in experiments as a comparison to none focally split images.

$$col_{out} = col_{near}$$

or

$$col_{out} = col_{far} \quad (5.7)$$

The aim of the blending is to produce a sum result of combined rays which would approximate the rays from the target distance and to make the shift from one viewing plane to another less noticeable.

Blend Comparison

It showed in our early testing that linear and non-linear blending gave inaccurate results when compared to the projective blend.

The linear and non-linear blends caused a false sense of depth as objects remained in the near focal image too long and the depth appeared inconsistent as well as showing a visible seam.

The non-linear blend was particularly susceptible to calibration errors similar to that of the box blend as the change in the focal plane was too quick in the middle ranges causing very strong border artefacts.

It makes sense that these symptoms were relieved with the projective blend as it is mapping the change in focal depth to the same type of scaling that is used in the projection matrix. Combining this with the linear depth calculation we are appearing to maintain correct focal depth consistency throughout the blend.

5.1.3 Reflection Depth

In our setup we are interested in providing correct depth cues through light rays reaching the eye from the correct focal distance.

When light hits a given diffuse object the light is scattered with varying amounts of uniformity which results in the light hitting the eye with an angle appropriate for distance to the object.

This is different for reflective objects where a portion of the light is directly reflected without diffusion towards the eye. In this case the rays of light are arriving at an angle similar to that of the object at the distance to the reflecting object plus the distance to the source of the reflection.

As we are not modelling the scene to take into account multiple reflections or the depth of those reflections, we are not able to successfully map these reflections onto the depth that is being used to split the scene into different focal ranges.

A naive approach could be attempted but any mismatches could potentially break the illusion for the surface we are mapping.

As a result of this limitation all objects in our experiments will only have diffuse lighting.

5.1.4 Rendering costs

State Change

For rendering simple scenes with OpenGL a high proportion of the costs can be the switching of GPU render state [25] which can cause stalls if enough data is not being submitted and the GPU is sat idle during the process.

When using two cameras the common approach is to render all the left view and then all of the right view. This means that the camera state is being changed many times per object per camera view.

A more optimal approach is for each set of objects being rendered switch the currently bound camera and render target (or multiple render targets and mask out the opposing view).

In this configuration it is only a maximum of two sets of camera and render target state changes per object instead of twice the textures, meshes and data per object.

$$\begin{aligned}
 m &= \text{Object count} \\
 c &= \text{View count} \\
 n &= \text{State changes per object} \\
 \text{Eye change per scene} &= c * m * n \\
 \text{Eye change per object} &= m * n + m * c \\
 c * m * n &> m * n + m * c
 \end{aligned} \tag{5.8}$$

GPU Memory Usage

Memory consumption is an important consideration when using multiple screens, particularly at a high resolution and high texel bit rate.

The displays being used with this software requires four screens outputting at 8 bits per pixel so we start with a minimum of 96MB of GPU memory being dedicated to the screen buffers alone.

We would like to render in a linear XYZ colour space High Dynamic Range setup to allow for future tests with HDR displays. This means we are required to use a linear (non-srgb) format for rendering scene geometry. When we come to present the rendered result on screen we will be transforming that result with an XYZ to RGB colour space matrix and gamma correction, and want to ensure that each texel colour has a high enough bit-rate to ensure that the rendered result will be transformed accurately into its corresponding RGB colour value without banding or artefacts in the resulting image. Because of this we will be using one of the OpenGL floating point formats and aim for a higher bit rate per pixel.

As this is purely an experimental setup we have selected to use the 32 bit per pixel floating point format, GL_RGBA32F, to ensure maximum quality and reduce the chance of errors from unwanted quantisation before the transform to RGB colour space.

This has big impact on the total memory being used for the render buffers as it is four times as large as the standard RGB8 formats but it is worth the cost to ensure correct results in our tests and to ensure that any colour calibration can be correctly supported now, or in the future.

Texture	Resolution	Bits/Pixel	OpenGL Format	Size (MB)
Near Plane Render	2048 x 2048	32	<i>GL_RGBA32F</i>	128
Far Plane Render	2048 x 2048	32	<i>GL_RGBA32F</i>	128
Near Screen Buffer	2048 x 1536	8	<i>GL_RGBA8</i>	24
Far Screen Buffer	2048 x 1536	8	<i>GL_RGBA8</i>	24
TOTAL: Single GPU				304
TOTAL: Both GPUs				608

Table 5.1: Memory consumption of the display buffers used in this display. The large linear colour buffers used for rendering the scene are the most costly. We are using a significant portion of the GPU memory with just the rendering pipeline. If we were to test a more complex scene this could become prohibitive and accuracy of the colour calibration transform might need to be sacrificed.

5.2 Software Configuration

5.2.1 Rotational Consistency

A major part of this software is deciding where we need to partition the images being displayed. That partitioning needs to be correct for the world space distance to objects and should be the direct distance from the camera to the point in space.

The depth that is produced from a standard projection matrix is not the world space scene depth. The non-linear divide gives a non-linear depth

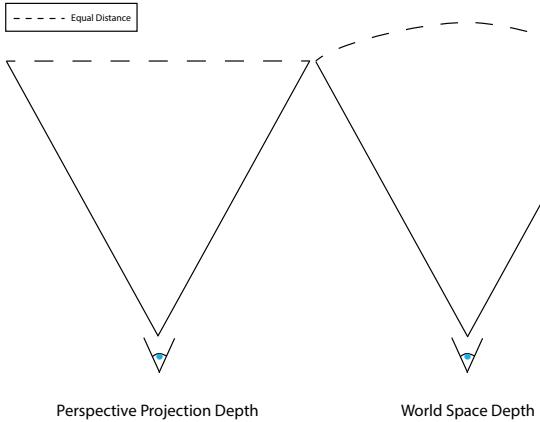


Figure 5.9: This is an illustration to show the different of position relative to the eye position of the same depth when using projective depth instead of world space depth.

from zero (or minus one) to one when what we need for calculating depth is the real distance from the ocular centre to the point in the scene. The perspective depth also gives all points at the far plane an equal distance from the camera, as shown in Figure 5.9 which causes changes in the depth value as the screen is rotated by causing points in space move from being on the far plane to be in front or behind it.

To calculate accurate depth, we will be using the objects positions multiplied by its world space transform matrix to get its world space coordinates and then subtract from them the world space position of the camera and calculate the length of the resulting vector.

This will give us a linear depth to each rendered point in the scene which will be consistent under any scene projection or camera motion.

5.2.2 OpenGL Multiple Render Targets and Contexts

When rendering to multiple screens we want to try and do it as optimally as possible to reduce wasted CPU or GPU time so that this display can support as complex a scene as needed without performance issues.

The hardware setup has two GPUs powering four screens in a non-SLI configuration. As we are not able to share data between the two GPUs we are forced to render the scene at least twice. As this is also a requirement of rendering stereo views without reprojection this is not a limiting factor in performance.

To be able to take advantage of the data sharing we do have available we can make use of multiple render targets in our shaders where it is possible so that we are only processing the vertex data once per eye and then in pixel shader we perform our depth calculation and write the appropriate blend of the lighting value to separate render targets representing the near or far screen. This limits our lighting calculations to once per eye and by running through the same GPU and OpenGL context we can be more sure of maintaining synchronisation on both screens.

This is more of a factor for matching the views between each eyes as we currently have no method to ensure that the left and right views are refreshing on the same schedule. To overcome this, we are using screens with a high refresh rate and low persistence and ensuring the test scenes are running above the required frame rate so that even if they are out of sync the difference between the two scenes should be very low.

If we are not able to share context across multiple screens, then the costlier method of using four individual contexts will have to be used. This would involve storing the data twice on each GPU and will prevent any fragment shader optimisation to write to multiple render targets, essentially doubling the pixel shading cost of the application.

5.2.3 X11 Window Controller

With modern Operating Systems a windowed applications maximum frame rate is tied to the refresh rate of the desktop. This would not be suitable for our application as we need to maximise frame rate to maintain realism and ensure persistence of vision [28].

Only full-screen applications are completely decoupled from the desktop screen refresh rate. Since we are using multiple discrete non-SLI GPU's it is not possible to create an OpenGL graphics context controlled by both, so we need to support multiple full screen context windows which is not possible in the standard "extended desktop".

In our setup we are using X11 as the windowing system which supports a mode to allow the user to run multiple independent X11 screens, rather than just extending one, and the user can select which screens are driven by which GPU. Using this we are able to launch fullscreen windows directly on the separate screen which allows us full control of the framerate of rendered images up to the maximum supported refresh rate of the display panels and GPU.

This setup lets us remove as many barriers to controlling the present rate of the screen to try and ensure that the screens are kept in sync. If the scenes are not correctly synchronised it could cause strange effects when objects in the scene are in motion.

5.2.4 Colour Calibration

As we are sharing the rendering of an image across multiple displays it is important that the parts of the image from each display are indistinguishable by colour and intensity.

In our display we are using four separate displays and a number of mirrors and beam-splitters. Each of these can interfere with the colour and intensity of light that is reaching the user. To maintain the merging of images for multi-focal rendering and correspondences for stereographic rendering this has to be avoided.

The source of error in the displays come from how they are provided from the manufacturer. Some are produced with variance in light intensity and colour ranges. This is unavoidable for our setup.

The beam-splitters suffer error as light is absorbed or diffused in a direction other than the expected reflected direction. To try and reduce this we are using high quality first surface mirrors to prevent diffusion or absorption within the protective medium. They are more susceptible to damage but as in our design they are covered and can be calibrated without touching them directly so they should be safe from damage or wear.

To correct for this error, we need to calibrate the screen to show the true colour being represented to the eye from each display.

Implementation

To begin to calibrate the screen we must use a high quality spectro-radiometer to measure the output from each screen for a range of colours and intensities. We do this with the display setup as if it is going to be used normally so that it is being configured for the real lighting conditions of regular use.

For each screen the spectro-radiometer is setup and all the other screens are set to display black, to allow calibration for the light they still emit when in a blank state which will be mixed in the beam-splitters. The screen then displays a wide range of colours at different intensities which are measured to produce a map of what the spectro-radiometer expected to see and what it actually received. From this map we can construct the calibration parameters needed to ensure that each display for each colour input will output a matching colour out. See Section 5.2.4.

The calibration from each screen produces a black level, gamma value and matrix to map from the CIE1931 XYZ colour space to RGB for each display [29].

We have chosen to use XYZ colour space as our scene rendering colour space as it is a good approximation of the human eye colour reception and has a larger supported colour gamut than RGB. It is also a good fit for our 32-bit per pixel linear render target textures which we are using to gain more accuracy during the transformation into RGB colour space than the standard

$$col_{rgb} = (screen_{XYZtoRGB} * (col_{xyz} - screen_{black}))^{(1/screen_{gamma})} \quad (5.9)$$

Figure 5.10: Equation for colour correction.

8 bit per pixel render targets would allow.

Once the scene is rendered in XYZ colour space it is then transformed with the equation showing in Equation 5.10 into the calibrated RGB colour space which is sent to the screens.

Calibration Results

The results of calibrating came out roughly how we expected apart from the results in the blacks levels (Figure 5.2, which show a small amount of variance mostly in the Red and Green channels. Interestingly screens 1 and 2 have a similar black level profile and screens 3 and 4 have a similar black level profile. This is interesting as screens 1 and 2 correspond to the screens handling the right eye and screens 3 and 4 handle the left. We expected that it would be the two near and two far planes that would get similar results because they would be passing through similar numbers of mirrors.

Screen ID	Black Level Adjustment(CIE31)
Screen 1	[0.5074, 0.4216, 1.1063]
Screen 2	[0.5045, 0.4233, 1.0869]
Screen 3	[0.4420, 0.4321, 0.8115]
Screen 4	[0.4373, 0.4325, 0.8009]

Table 5.2: Black levels - Representing the light leaking through the LCD panels when attempting to display a completely black screen.

Once this data was applied in the software the screen colours matched almost completely when viewed from outside the display and the difference was difficult to notice when looking through goggles.

Screen ID	RGB Gamma value
Screen 1	[2.2368, 2.2064, 2.2247]
Screen 2	[2.2319, 2.1663, 2.1936]
Screen 3	[2.4438, 2.4475, 2.4438]
Screen 4	[2.2406, 2.1956, 2.2554]

Table 5.3: Gamma correction values for each display panel.

$$\begin{array}{ll}
 \left[\begin{array}{ccc} 0.0324 & -0.0095 & 0.0006 \\ -0.0147 & 0.0180 & -0.0017 \\ -0.0051 & 0.0006 & 0.0081 \end{array} \right] & \left[\begin{array}{ccc} 0.0345 & -0.0078 & 0.0003 \\ -0.0159 & 0.0149 & -0.0011 \\ -0.0058 & 0.0007 & 0.0062 \end{array} \right] \\
 \text{(a) Screen 3} & \text{(b) Screen 2} \\
 \left[\begin{array}{ccc} 0.0115 & -0.0038 & 0.0004 \\ -0.0050 & 0.0070 & -0.0009 \\ -0.0018 & 0.0002 & 0.0034 \end{array} \right] & \left[\begin{array}{ccc} 0.0118 & -0.0031 & 0.0001 \\ -0.0055 & 0.0059 & -0.0005 \\ -0.0020 & 0.0002 & 0.0028 \end{array} \right] \\
 \text{(c) Screen 4} & \text{(d) Screen 1} \\
 \end{array} \tag{5.10}$$

Figure 5.11: CIE XYZ to RGB colour matrices for each display panel.

5.2.5 Full pipeline

The pipeline for rendering the images to separate displays, shown in Figure 5.12, is relatively simple.

At start up the geometry and any scene data such as camera position and scene calibration is loaded into each OpenGL Context. This is either two contexts with left screens sharing one context and the right screens sharing one context, or it is the less efficient four contexts if there is no support for shared contexts on the machine.

Once the data is loaded in, the OpenGL Contexts rendering can begin. If contexts are being shared then the shaders bound for rendering will have two output render targets bound. One representing the near plane and one representing the far plane. When geometry is sent to be rendered it is projected as normal and then passed to the pixel shader where we run the selected

blend mode (see Section 5.1.2). This will split the colour of the point being rendered across the two displays according to the blend function by writing appropriately to the correct render target.

When using a single OpenGL Context per display the process is the same but when it comes to the blending stage it will simply discard the data that would have went to the other focal plane and writes to a single render target.

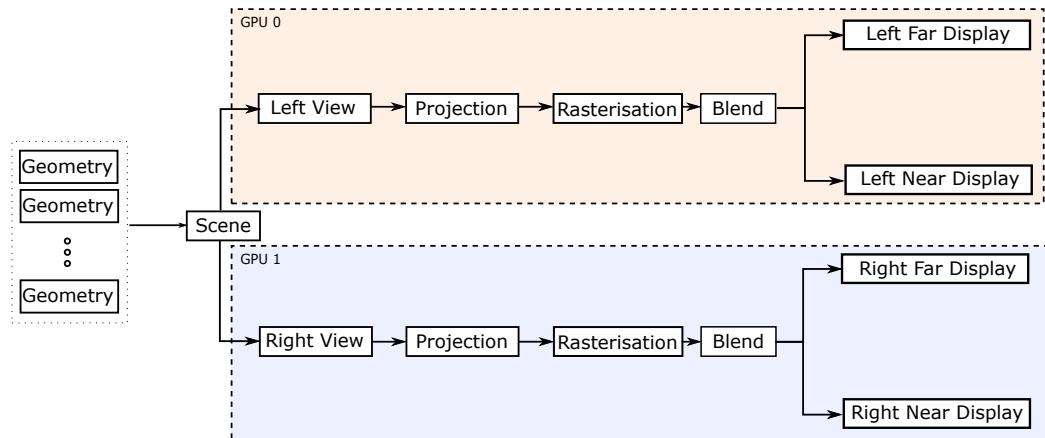


Figure 5.12: A high level overview of the steps to separate work between GPU's and display panels.

Chapter 6

Methodology and Testing

In experimenting with the device we are aiming to find out to what degree, if any, the use of multi-focal viewing planes increases the users perception of depth in a scene when compared to stereo and standard 2D viewing conditions. Our hypothesis is that by alleviating the vergence-accommodation conflict we should be introducing a more comfortable and realistic depth viewing experience which will allow the user to more accurately judge depth than when the focal cue is missing. However, we are open to the idea that the change in focal depths as the user looks around a scene could introduce discomfort from other means, which are less researched as it is not what users are typically accustomed to when using stereoscopic 3D devices.

The display and software design has given us lots of options for testing through varying projection, blending and real focal distances. However, to keep the experiment simple and within scope we have performed some early testing to narrow the total number of experiments. As such we will be investigating using the “oblique” projection method (Section 5.1.1) as it provided correct near and far plane correspondence and with projective blending as it gave us the least error from misalignment in our test scenes. Misalignment is a large concern for us during experimentation. We have seen that misalignment can completely ruin the effect we are trying to achieve and can lead to a lot of discomfort for the user.

In order to address our hypothesis, we will be testing the users perception of depth by showing two objects at different depths to the user and asking them to select which object appears closer. This will be performed with the user viewing objects with zero disparity, zero disparity multi-focal, stereoscopically and stereoscopically with multi-focal planes (see Table 6.2. This will allow us to compare how accurately the user was able to determine the relative depth when given different depth cues and therefore determine if the additional focal depth cue allowed for a higher perception of depth.

6.1 Method

This is an experiment that will be measuring the perception of depth so it is necessary for this to be tested on a range of volunteers rather than just ourselves to ensure that any conclusions are conclusions for the general case of the larger population. The participants selected for this experiment will not be specifically selected for any visual strengths or weaknesses, such as stereoscopic perception.

For this experiment we have setup the display to have the distance between the far and the near plane approximately 0.6 dioptres (see Section 2.2) apart, as this is seen to be the optimal distance for retinal image quality [17] and is a large enough distance for optimal stimulation of the accommodation response [18] so the transition between the two objects at the largest offsets should not be jarring. For this experiment the near plane will be at 0.54m and the far plane set to 0.81m.

6.1.1 Depth Comparison Setup

The depth comparison scene is a small scene in which the user is asked to look at two objects (see Figure6.4) in the display and then report which of the two objects appears to be closer. This is then repeated for different distances between the two objects (see Table 6.3). The aim of this scene

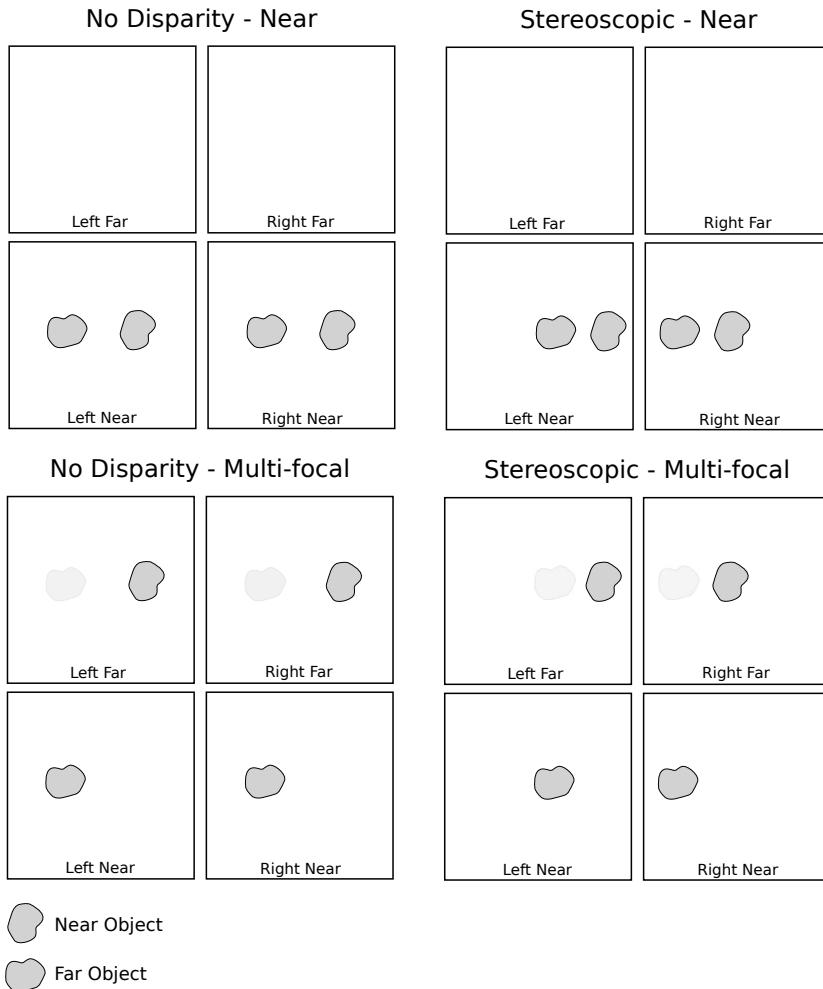
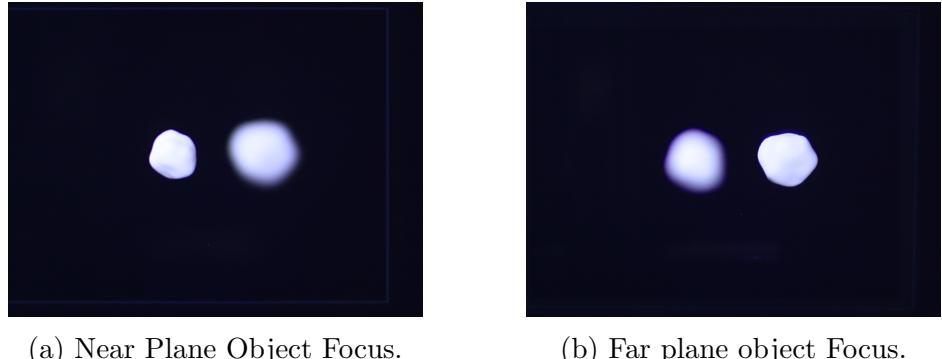


Figure 6.1: This is an example of the render modes we are using for the experiment and how they look on the screen. The multi-focal render modes split the depth of objects across the two places where as the ‘Near plane’ render modes only use the left and right near planes and as a result the rendered objects only appear on a single focal plane.

is to accurately measure the effects of the different depth cues to be able to determine if the addition of focal depth cues from accommodation improves user perception of depth. To do this we need to be able to isolate the user from all other cues except the ones we are investigating, namely the presence binocular disparity, vergence, accommodation and the absence of all cues.

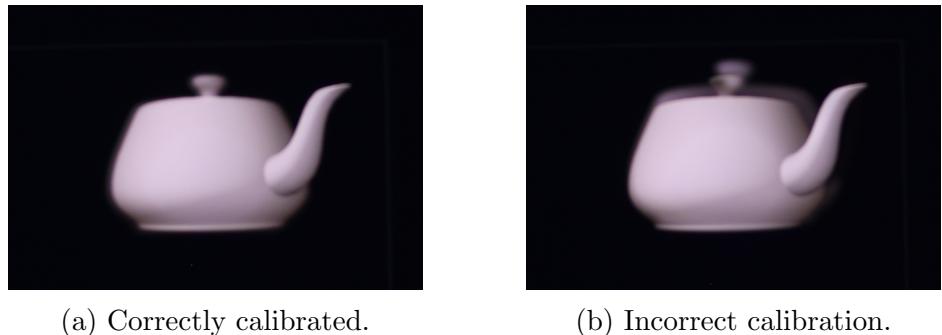
In a regular scene the user is given many hints for the depths of objects, see



(a) Near Plane Object Focus.

(b) Far plane object Focus.

Figure 6.2: These photographs are from a run of the experiment showing a large offset between the two objects and displaying the images in the multi-focal render mode. As you can see by changing the focal length of the camera from 0.54m to 0.81m we are able to selectively focus on both objects and there is a large visible difference when that changes. The images were taken with an aperture of f/1.4 which is larger than the aperture of the human eye which gives this image a softer focus than what is experienced by a user of the display.



(a) Correctly calibrated.

(b) Incorrect calibration.

Figure 6.3: This image shows the error from incorrect geometric alignment. It shows the far plane out of alignment with the near plane and the ‘shadowing’ artefact becomes apparent. When this happens it is very clear to a user that the image is comprised of two separate planes. We aim to reduce the occurrence of this in our experiment using a calibration screen and frequent updates.

Near Focal Distance	54cm — 1.85 Dioptres
Far Focal Distance	81cm — 1.23 Dioptres
Near-Far Range	27cm — 0.62 Dioptres
Interocular Distance	6.5cm

Table 6.1: This is the distance configurations the display has been setup to use in this experiment. It approximately maintains the optimal 0.6 dioptres between focal planes so that the difference in focal depth should be very perceptible to the user.

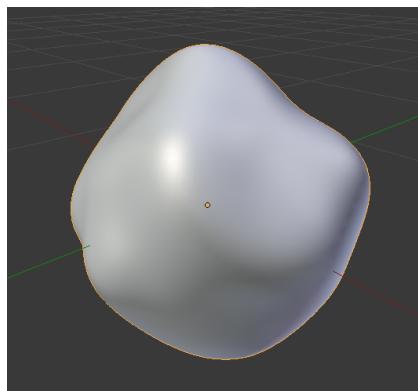


Figure 6.4: This is an example of the deformed sphere models that are being used in the experiment. They are scaled within the scene based on depth from the camera so they maintain the same size in screen space and are then scaled by 5% up or down in the scene to help reduce any false cues from perspective.

Section 2.3. In our scene we have taken measures to isolate the user from them so we can accurately measure the ones we are interested in.

- **Perspective:** To eliminate perspective cues we enlarged more distant objects to revert the effect of perspective projection. The objects being displayed also have no geometric parallel lines for comparison of convergence. Additionally we have introduced a random scale of up to 5% of the objects size
- **Known Size:** We are using randomly extruded spheres (Figure 6.4) so they are not a recognisable object.
- **Detail:** The objects have no texture detail. The only cues from detail

come from a slight pattern of shading from the shape of the object but as the two objects have different shape this is not comparable as a cue.

- **Occlusion:** The objects sit either side of the optical axis and do not occlude each other.
- **Lighting:** We have used fixed directional lighting from points at infinity so the lighting is invariant to the position of objects in the scene. This prevents the distance being judged from the rate of change of lighting on the surface.
- **Relative motion:** There is no motion in the scene to allow the user to judge depth from rate of change of size and position.

We have also had to consider hints at position that can come from the display setup:

- **Colour:** As we are using multiple mirrors and beam-splitters in the display there is a small amount of chromatic aberration which was quite obvious when using separate colours for each object in the scene. This gave hints to the number of mirrors being used to display that object which in turn told the participant whether it was on the near or far plane when testing the multi-focal part of the experiment. As a result of this we had to switch to using a single colour for the objects in the scene.
- **Head Position:** If the user is misaligned when they change scene they will be able to notice outlines of any objects partially drawn on the near plane overlaid with the percentage of the object on the back plane, Figure 6.3. This will show the participant which object is physically closer. To avoid this we have added a calibration step between each display of the scene at different depths and advised the participants to try and avoid head motion while viewing the scene.

With these changes we are left with two randomly deformed spheres which will allow us to isolate and measure the cues we are interested in by varying which render mode (Table 6.2) we use.

RENDER MODES	Near Plane Only	Both Planes
6.5cm Disparity	Stereoscopic	Stereoscopic + Multi-focal
No Disparity	No depth cues	Multi-focal

Table 6.2: This table shows us the four possible combinations of the render modes we will be testing and which cues they allow. We are particularly interested in the combination of ‘Near Plane’ and ‘No Disparity’ as this combination will show us how well we have isolated other depth cues. If that combination is still able to be accurately judged for its depth then there is other depth cues available or we have failed to block the ones we wanted.

From our early pilot tests we found that in stereo and multi-focal viewing modes we were able to very accurately determine the relative depths of objects when the distance between the objects were large. To confirm this and determine at what distance this breaks down we have selected to keep one object on the far plane and then move the other object away from the far plane at fixed offsets up to the near plane with more steps being taken near the far plane (see Figure 6.3) to help determine the largest range that works within the multi-focal setup and determine where it breaks down.

In our chosen offset distances, we have included a zero offset. At the zero offset the two objects are at the same distance from the camera. In this situation the computer chooses the correct answer randomly. This has been included in the test as a check for our data collection. The correct answers on this offset should trend towards $50\% \pm$ error from sample size correctness (the point of subjective equality), see Figure 6.6. If this is significantly different in our data, it will indicate that we may have a problem in the experiment.

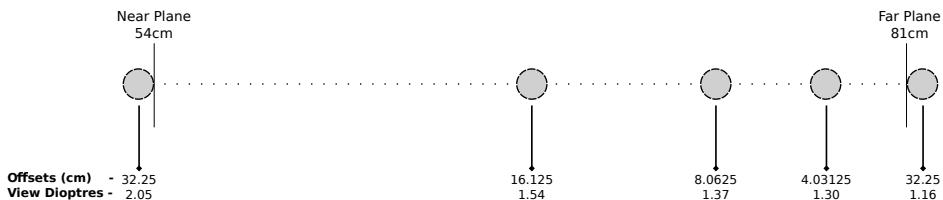


Figure 6.5: This diagram different offset locations and the relative distances between them. This diagram shows the right object at different locations but in the experiment either blob can be in the offset position but one is always on the far plane.

Index	Offset (cm)	Depth (cm)
0	0	86.25
1	4.03125	76.97
2	8.0625	72.94
3	16.125	64.88
4	32.25	48.75

Table 6.3: This table shows the offset positions in distance from the camera. The depth is central position of the object. On the far plane at zero offset the value is not 81cm as we have adjusted for the size of the object and random scale to ensure that is entirely behind the far plane in its starting position.

To be able to get accurate and unbiased results we have made sure the sphere that is closer to the observer is randomly selected and both spheres are randomly rotated.

In experiments such as these the participant will become accustomed to what they are seeing and are expected to improve as it goes on. Because of this we will be randomising the order of offsets which the user is shown so that the offsets or certain view modes are not unevenly improved. This is randomised for each user performing the experiment so that we are accounting for any pattern that may arise from a single randomisation.

For the experiment we are asking the participants to do 20 samples per render mode and offset. Ideally we would have wanted to perform up to 40 comparisons per offset per rendering mode but time is a limiting factor for our participants. With 20 samples we are able to reduce the margin of error from random choice (Figure 6.6) to be low enough to extract the information we need but for further study on any findings we would want to want to be more thorough.

From our pilot studies with this test we found 20 samples for all render modes and offsets took around 20 minutes. As we are accustomed to using the display we are assuming participants will take between 20-30 minutes including calibration steps.

Calibration takes a short amount of time. The display is setup in its default

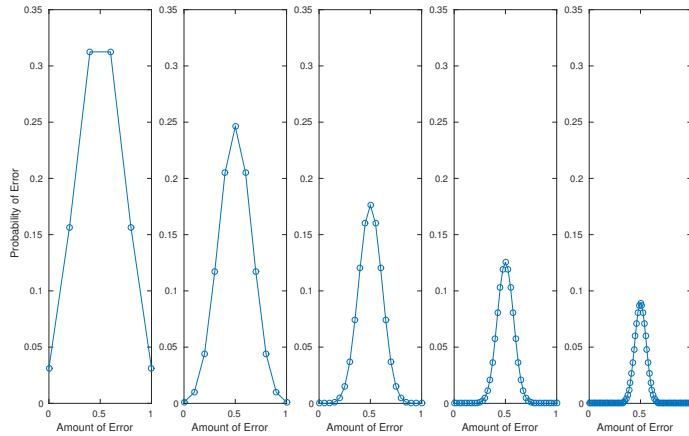


Figure 6.6: These graphs show the binomial graph of expected error.

calibration assuming the mean interocular distance with a convergence point on the near plane. If a user's interocular distance varies from this then we will need to spend a few minutes using the software's calibration screens (Figure 6.7) to adjust the positions of the displayed visual planes. The software supports scaling and translation of the different planes to accommodate for participants whose eyes may differ from the average in vertical alignment as well as horizontal interocular distance. This also allows us to ensure that the display is calibrated for a head position which is comfortable for the user.

Experiment Procedure

The user will first be given a guide explaining what the experiment is and what is required. See Appendix for the briefing form. Then the participant will be shown the display with the calibration screen and asked to get into a comfortable position. The screens are then aligned so that the user can comfortably see the near and far plane and converge the images into a correct stereo pair.

The participant will then be shown an example of the scene and shown the controls for how to select between the two objects.

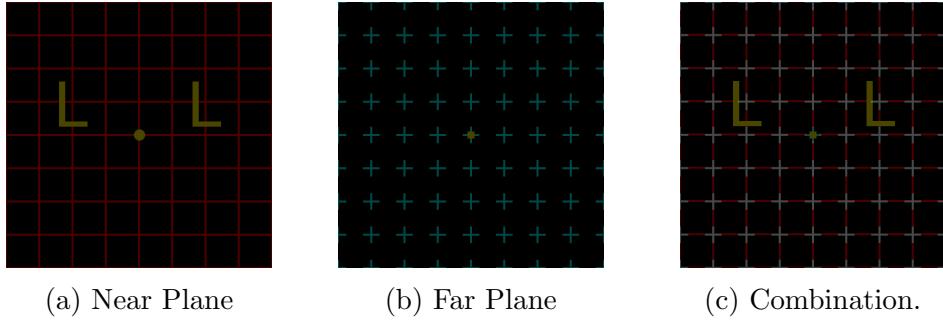


Figure 6.7: Software Geometric Calibration: The near plane calibration screen shows a central point and two ‘L’ characters. The point helps in centering the image at the correct position and the ‘L’ is a simple way to confirm the orientation of the screens are correct. The far plane has a series of cross marks to align with the near plane as well as a central point to align.

Once the participant is comfortable at the display with the controls the experiment can start.

The participant will be shown the scene, make a decision on which they believe is closer and then they will be shown the calibration screen to ensure the calibration did not change during the scene. This is repeated twenty times for each combination of render mode (Table 6.2) and each offset distance (Table 6.3) in a random order.

During the experiment the participant is advised to try and avoid head motion during each scene.

After the participant has completed all the required decisions an image will inform show them that the test is over.

Data Collection

From our experiment we will be storing an ID to represent each participant and a list of whether the participant was correct or incorrect for each sample of each offset and render-mode configuration.

We will then analyse each participants results individually to determine any obvious patterns in multi-focal depth perception for that user and compare

it to other users. We will then look at any apparent groupings of types of results and the combination of all results.

We will look at each participants results individually first as we expect there to be some quite interesting results that may be unique to each participant or groups of participants. In our early testing we found some quite large discrepancies in the perception of different depth cues which we hope to elucidate.

Once we have analysed the data individually we are also going to look for any trends in the whole data, with consideration to any groupings we may find in the individual analysis.

Chapter 7

Results and Discussion

7.1 Results

The experiment was ran with 15 participants but we are only including the results of seven, plus the pilot test data in this section due to an error that caused the data for the first eight people to be unusable, which is discussed in Section 7.1.1.

Most runs of the experiment took between 20 and 25 minutes, as we expected. The results did not produce results which obviously support our hypothesis but did produce some trends which are interesting for what they suggest about depth cues, it is discussed further in Section 7.2.4.

The results have some noise from the error coming from our small sample size for each offset and render mode but this is consistent in most entries with what we predicted in Section 6.1, however there are a few discrepancies which fell outside of this expected error which are explained in Section 7.2.2.

Individual Results

This section contains the results of each individual who took part in the experiment.

For each participant we have graphed the four different render mode combinations (Table 6.2) against the results of the ratio of answers were correct for each offset position. For each result we have included the standard error of the mean as an indication of expected error and the participant names have been anonymised to letters. These are shown in Figures 7.1, 7.2, 7.3, and 7.4.

Combined Results

As well as looking at the individual performance of each participant we have also found it useful to group the data together so that we can identify any overall trends. The total combination of all results is in Figure 7.5.

There was a very noticeable grouping in our individual results of those who were not able to discern the depth of objects using only stereoscopic depth cues. For further analysis we have separated those into two groups and plotted the results separately (Figures 7.6a and 7.6b).

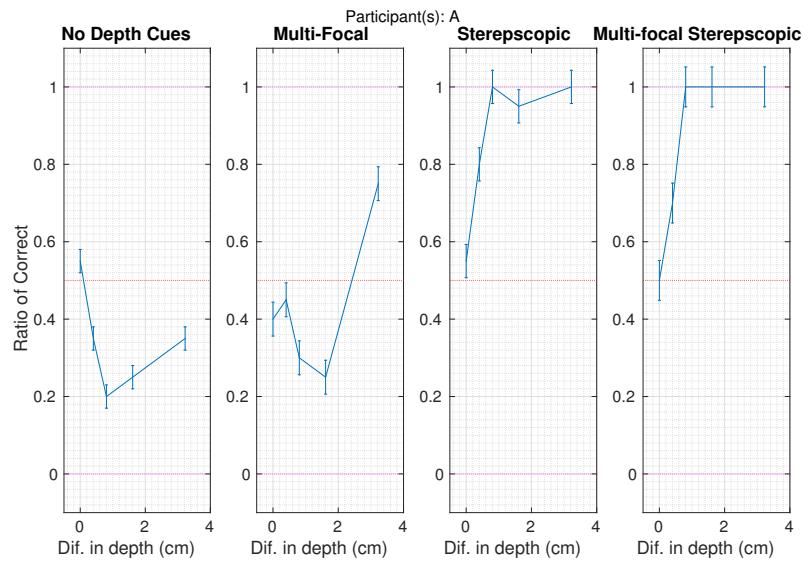
Personal Results

Due to a problem encountered during the initial data collection resulting in not being able to use a lot of participant data we have included the data from the author of the paper in Figure 7.7 to give as broad a picture of the results as possible.

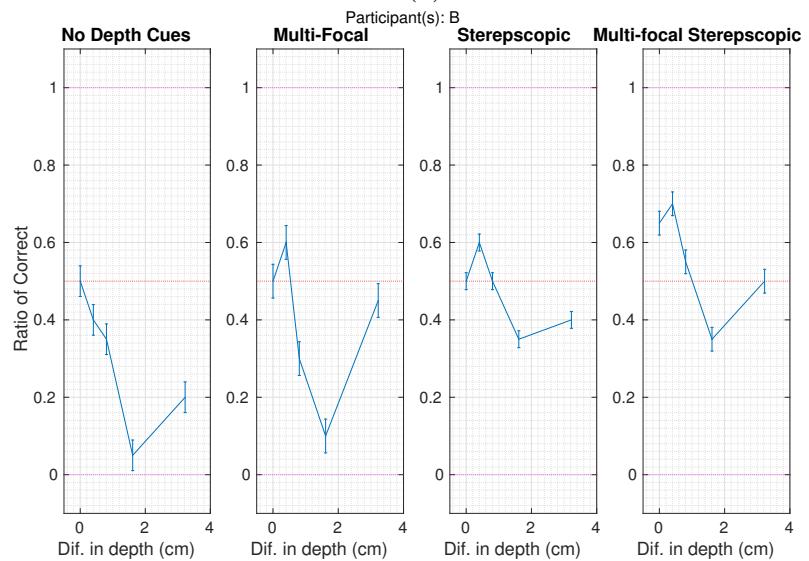
This data is for two sequential runs of the experiment by the author from the pilot runs of the experiment which were performed before inviting participants to take part in the experiment.

7.1.1 Erroneous data

There was a fault in the original run caused by a lack of rotation in the displayed objects which caused the zero offset scenes to be in an arrangement

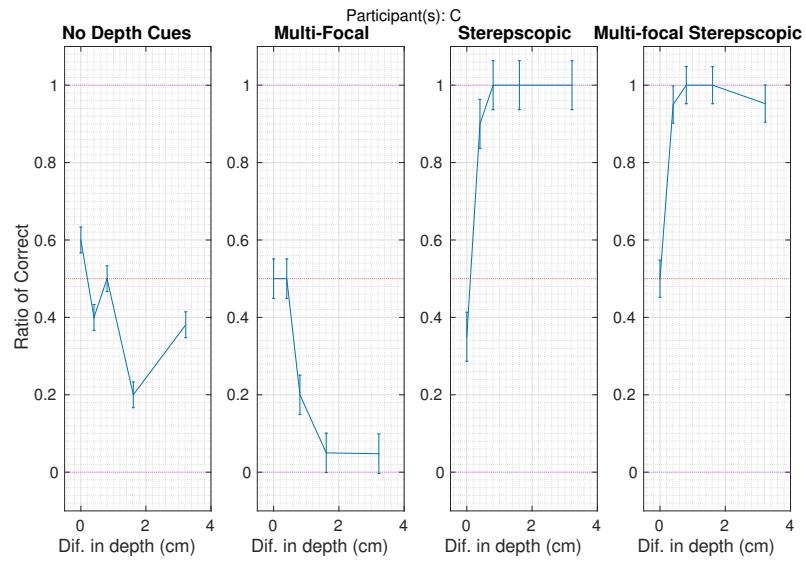


(a)

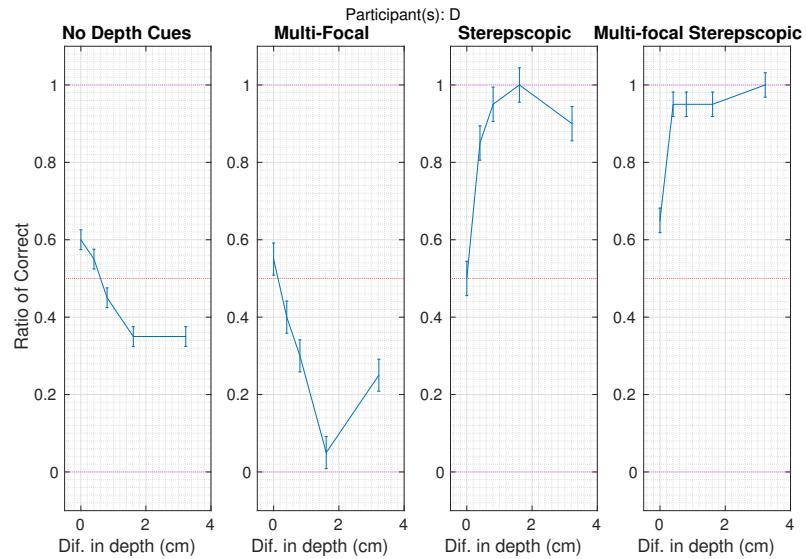


(b)

Figure 7.1: (Part 1 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean. The point of subjective inequality is marked in red at $y = 0.5$

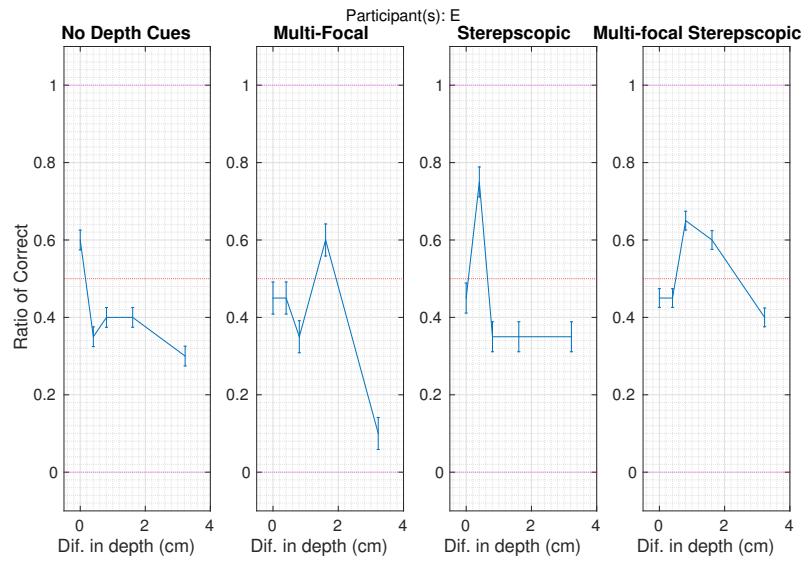


(a)

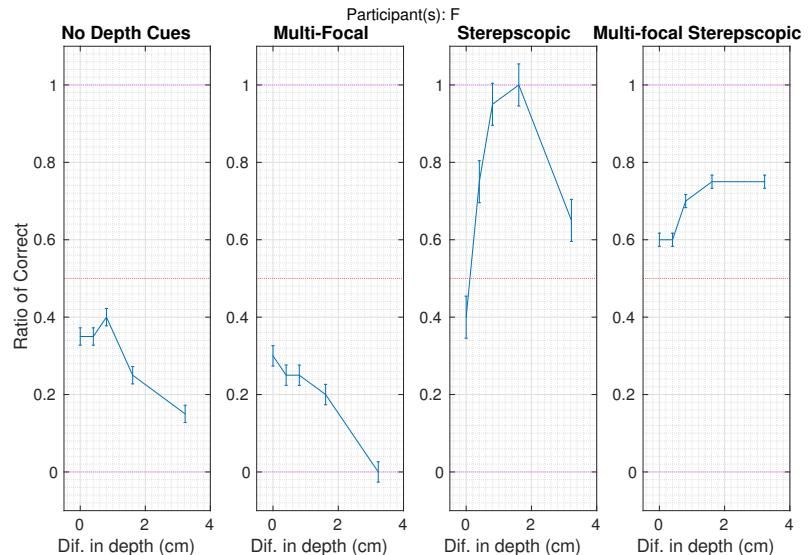


(b)

Figure 7.2: (Part 2 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean.



(a)



(b)

Figure 7.3: (Part 3 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean.

which gave the illusion of one of the objects to appear closer to some users. Coupled with a bug in the seed generation for random numbers for selecting which object is closer when there is no offset this caused a problem in the data resulting in very extreme correct or incorrect selection at this offset.

As we had not used a random rotation this could have caused misleading selection for all objects in the ‘No Depth Cues’ render mode.

As a result, we had a lot of noise in our data and it was difficult to reach any convincing conclusions. This has meant that the experiment has had to be completely ran again with new participants. Due to our limited time we have ended up with a smaller population sample than is ideal.

Minor Adjustments

Aside from the problem which caused us to have to restart the experiment, there were initially a few issues when we ran the test that we had not considered in our initial setup.

The first minor problem was that we had not taken into consideration the height of the user. The desk was too tall for some of our participants and it is not height configurable. This meant that we had to repeat an experiment after it was noticed and the participant mentioned it was difficult to keep the correct head position due to stretching to reach the mounted goggles.

The next minor issue was caused because we had not taken into consideration that people have varying sensitivity to bright images. The combination of the brightness of combined images, particularly the calibration and end screens, which did not have their brightness averaged to be the same as one display, caused some participants discomfort and the brightness had to be reduced.

For some participants the test was a long period of time to maintain the same head position. That led to more breaks and slight head motion during the test than was expected and could have impacted the results of the test particularly the later comparisons which could impact the amount of noise in the data.

7.2 Discussion

From an initial look at the overall results for all users (Figure 7.5) we can see that depth was fairly accurately judged when the participant had any stereoscopic cues available but the performance did not seem very different when multi-focal cues were added.

The data is quite hard to understand for each individual due to appearing to be quite noisy in some of the plots and not being completely random where expected. We have performed a short investigation into this to ensure the quality of the collected data in Section 7.2.2, and we discuss the lack of binocular disparity depth cue perception in some participants in Section 7.2.1.

We see that there may be a small increase in depth perception when multi-focal depth cues are added for participants who were able to correctly judge depth from stereographic cues in Figure 7.6b. This is discussed in detail in Section 7.2.3.

For the multi-focal cues alone, the results across the users appear to show no strong suggestion that the cues from accommodation improved the judgement of depth when used alone. But the data oddly seems to suggest that the lack of binocular disparity cues caused the participants to select the furthest objects rather than the nearest. We discuss this result and what may be causing it in Section 7.2.2 and 7.2.4.

7.2.1 Lacking Stereopsis

As mentioned in Section 7.2 we were surprised by some participants (Figures 7.3a and 7.1b) lack of ability to be able to select the depths of different objects when given only binocular disparity depth cues as participants had reported being able to view the objects stereoscopically during calibration.

This leads us to consider that some people have experience with stereographic systems from going to see 3D films or using VR devices and day to day regular binocular vision, but others may not. If the participant does not

have good stereopsis in a the real world, such as if they have stereoblindness caused by amblyopia, then unless they are specifically trained with stereo pairs they will not be able to accurately process the depth cues as they may not have developed these cues when they were younger and would normally have developed this skill [16].

It is also worth considering that these users may only be suffering from poor calibration. Such as if the calibration steps correctly aligned the positions of the view planes but the software interocular distance was too great or small relative to the participants real interocular distance leading to a larger disparity than the user is able to view comfortably. This theory is backed by some participant reporting sometimes seeing three objects when viewing the binocular disparity render modes when the offset was at it's maximum offset value which places one of the objects quite close to the camera. This is the situation where the participant will see the greatest level of disparity between the left and right views and is mostly likely to have difficulty converging the two images.

7.2.2 Trends with No Depth Cues

The results, which are themselves a re-run of the original pass of the experiment (see Section 7.1.1), still exhibited some strong trends in the ‘No Depth Cues’ render mode, for which the observers should give random answers. It is not nearly as large and unpredictable as the error which caused the experiment to be restarted, but it is significant and could point to a problem in our experiment, so we have performed a short investigation to show the integrity of our software and try and identify the cause of this data.

Firstly we wanted to check that the software controlling the experiment was working correctly. To do this we ran the experiment with varying numbers of samples per offset and render mode and allowed the software to select choices randomly to see if the results would converge on the point of subjective equality (50% of the answers being correct). This gave us the results shown in Figure 7.8, which do in fact converge to 50% correct as the sample count

per render mode and offset increased.

To further validate the software we ran the same process again but this time only allowing it to select the left object, and then again only selecting the right object. This is to check that there was a proper balance of each answer in each set of comparisons that we showed the participants, and to check that the 50% convergence we saw when using random selection was not the result of a problem from our pseudo-random number generator. This test also gave us the same convergence to 50%.

This combination of tests gave us confidence in the process of selecting which object is closer and the distribution of results.

We next checked whether the input selection of the right and left objects were being consistently and correctly processed by the software. In this test we altered the generation of the offsets and random selection of which object is closer to allow us to specify these explicitly. We then ran the experiment and selected the correct answers we had told it to use. With our custom list of offsets and answers we then confirmed that the answers we entered were all correctly registered as correct in our output data file. We then repeated this experiment for incorrect answers. This validated that our input control and data storage were all working correctly. It is an area that wasn't expected to have errors due to the good scores from the binocular disparity scenes, but was worth sanity checking.

This leaves the only other source of difference to be in the rendering. When switching between render modes the only differences is the blend mode that is being used to show the objects on the appropriate screens. The 'No Depth Cues' render mode uses the near blend mode which displays objects only on the two near planes. This can be visually validated by running the experiment and checking which screens the objects are being rendered on. This appears to be correct. We have also looked at the source code which sets the blend mode and writes the data to the appropriate screen to check in case there was any differences between what is being displayed on the left and right display panels. The hypothesis being that if there was a bug setting the left

or right screen pixels correctly it may bias the scene somehow. This was not the case however, as the blend modes were all set and behaved correctly.

The only culprits left that could be giving us such differences is a highly unlikely pattern of purely random noise or there were some depth cues in the scene that we were not able to fully eliminate, which would mean the participants are actually seeing the depth of the scene - even if they are seeing it backwards.

We are fairly confident that this is not simply random noise as it has a persistent pattern between most participants and is statistically very unlikely. The trends that we see in the results for this render mode appear to behave very similar to the results for the Stereoscopic render modes. When the offset is large the answers are consistently more likely to be correct and as the offset becomes smaller they trend towards random.

That points us to how we are rendering the scene as the thing that is giving the participants incorrect clues to the depths of object.

In Section 6.1.1 we outlined the depth cues we were looking to remove from the scene. There is a chance that we have not been totally exhaustive in our removal of depth cues. Our suspicion is that the error we are seeing is from perspective convergence to the vanishing point. We did not consider this screen space translation when setting up the experiment, we only considered the perspective scaling and geometry of the objects being transformed, but it will be causing a visible change on screen between different offset people.

If we place two objects in front of a camera, as the object moves closer it also becomes less central in screen space if it is not exactly on the optical viewing axis. Each of our objects are only slightly offset from this axis so we did not consider this motion to be significant but given that 200 images with no disparity were shown to the user, many of which may have appeared without an image with disparity in between, it is possible the user may have subconsciously noticed this part of the perspective depth cue.

This does not, however, explain why the error gave consistently incorrect

results rather than correct ones but it is a depth cue we can identify in our experiment that would effect these scenes only and could give the curves we have seen. Ideally we would like to have access to more participants to see if this is effecting everyone as this is not a depth cue which we picked up on in our pilot testing (Figure 7.7) which also show patterns of stereo-blindness but does not have a consistent trend of erroneous selections in the render mode with no depth cues.

7.2.3 Stereoscopic Cues

For the majority of participants in this experiment the presence of depth cues from binocular disparity was a strong hint at the depth of objects and those who could pick up on it were able to answer correctly fairly consistently up to a short offset between the objects. This is what we expected from the binocular disparity cues as stereo vision is known to be a strong depth indicator [8].

It is difficult to discern whether the presence of accommodation depth cues with binocular disparity gave an increase to the accuracy of the judgements as in many cases there is only a small difference in the results between the binocular disparity results with accommodation depth cues and the ones without accommodation depth cues. Frequently these are nearly all judged correctly (Figures 7.2a, 7.2b, 7.1a) for the large offsets which suggests that the number of samples was not enough to accurately measure the difference between the two render modes when the depth cues were strong.

However, it is clear than when the user could perceive the cues from disparity that the addition of accommodation cues did not reduce the judgement of depth for those who could see the binocular disparity cues, but it is difficult to say with any certainty whether or not they were improved.

To be able to know with any certainty we would have to run an experiment that would allow a greater number of samples, possibly over a wider range of offsets, to be taken so that we would be able to match it to the psychometric

curve and with greater accuracy find the points of just noticeable difference for each stereoscopic render mode and see if we were able to maintain the accuracy over an increased number of offset positions with or without accommodation cues.

We had a hypothesis that if the multi-focal cues would not give an improvement to the stereo rendering than the higher chance of error from misalignment or bad calibration would cause the stereoscopic rendering to be less effective at expressing depth. In our experiment we haven't seen a convincing level of improvement from multi-focal cues but we also have not seen a detrimental effect from it either. It is possible that we over estimated the amount of error we expected to receive from alignment with multi-focal modes.

7.2.4 Multi-focal Depth

Our original plan to analyse the accuracy of accommodation cues was to compare the ‘No depth cues’ render mode against the the multi-focal only render mode then any differences we saw would be as a result of the extra focal plane as the render mode with no depth cues should be completely random.

However, as described in Section 7.2.2 we found that we were not able to isolate that render mode from depth cues entirely. As we are not able to attribute any trends directly with the accommodation cues we are instead comparing it to the *trends* found in the ‘No Render Cues’ render mode. This way we are looking to see whether the addition of the accommodation depth cues increased or decreased the ratio of correct comparisons in the trend similar to how we compared against the stereoscopic only render mode.

When comparing the render modes this way we can see that there is approximately the same trend between the two non-stereo render modes but the trend is are emphasised in the multi-focal results. This is particularly evident in Figures 7.2b and 7.3b. The ‘Multi-focal’ render mode gets a very strong consistent response from many of the participants and participant C

(Figure 7.2a) in particular. For participant C the multi-focal only result looks very similar in scale and trend to the stereoscopic results but inverted.

The amplification of the result when the extra focal cues are added is a strong indication that the cues are being received by the participant. Additionally the point of just noticeable difference appearing to fall roughly around the same offsets as both the stereoscopic render modes, for some participants (such as participant C Figure 7.2a, suggests that it may have a fairly strong influence and should be further investigated.

Inverted Results

In the same way as the render mode which should have no depth cues caused the choice of depth to be reversed, the multi-focal only mode has also been reversed. In both of these modes the user is consistently choosing the furthest object instead of the nearest object.

This is very unexpected as conflicts with the binocular disparity modes and our pilot data (Section 7.1), but the data is very consistent and these participants are not experienced with these cues on this kind of display where as we have spent a lot of time with it.

Where stereo is present and depth is correctly recognised by the participants, the addition of multi-focal cues keeps the perceived depth order of the object correct. The depth order is perceived wrong in the ‘No Depth Cues’ render mode and the addition of accommodation cues keeps the perceived depth of the objects reversed but increases the rate of consistently reversed answers.

In a study performed on the relative strengths of different depth cues [8] it was found that different depth cues give people varying degrees of depth perception and the combination of different cues can give different results. More recent work [23] pointed at binocular disparity and blur being complementary depth cues.

If we follow that the visual system is able to combine depth cues, it may

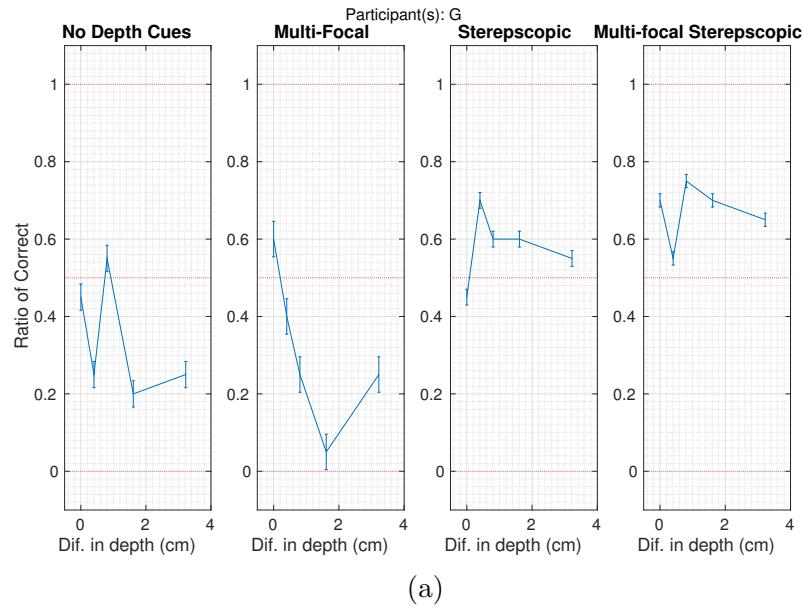
be the case that the depth cues given from the ‘No Depth Cues’ render mode without accommodation cues are stronger or more dominant than the accommodation cues which are added in the ‘Multi-focal’ render mode. There has been research performed into models of how the human visual system fuses different depth cues [5] but we have been unable to find any significant data related to accommodation to back up our claim beyond the simple observation of our limited results.

If we were able to correctly isolate the results of only accommodation cues then we would have been able to say more certainly whether the correct ordering of depth is something that can be achieved on accommodation alone at our experimental focal distances. But, with just our results we can only conclude that when accommodation depth cues are paired with another reversed order depth cue the perceived order remains incorrect. We suggest attempting this experiment with much the larger range of focal depths and compare against a render mode that really does not have any depth cues and that may reveal if directionality is absent from accommodation cues or if this behaviour is just a product of the unknown depth cue in our zero disparity mode.

7.2.5 Extending our experiment

When using the device the sense of ‘blur’ from accommodation was quite small and could have contributed to the weak response it saw at times. It would be interesting to extend the focal distances used in this test and see if that gives more precise results. Similarly it would be interesting to experiment with varying lighting conditions of the objects being shown so that we could effect the pupil dilation of participants. Similar to what we can see in Figure 6.2a where the wider aperture of the camera gives a larger amount blur on the out of focus object we could, through darker and controlled test images, cause the pupil to be larger and see if that could also emphasise depth by increasing the focal blur. Then perform a similar experiment to one performed in this work to measure if that increased the accuracy of the

depth cue.



(a)

Figure 7.4: (Part 4 of 4) Individual results from participants in the depth comparison experiment. The graphs show the ratio of correct answers at each offset. The error range on each point is the standard error of the mean.

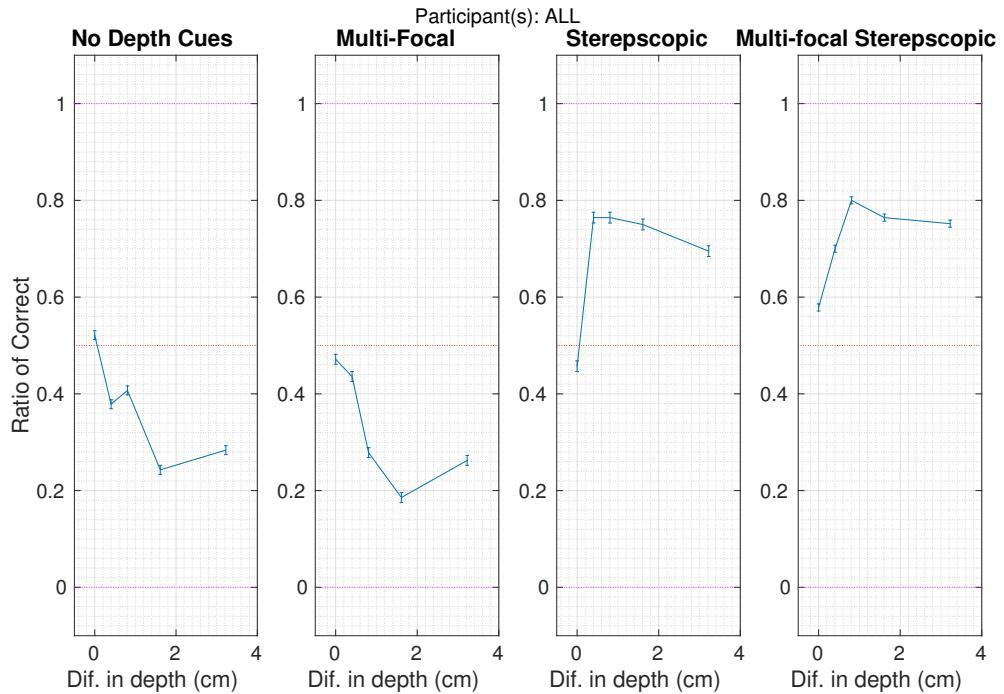
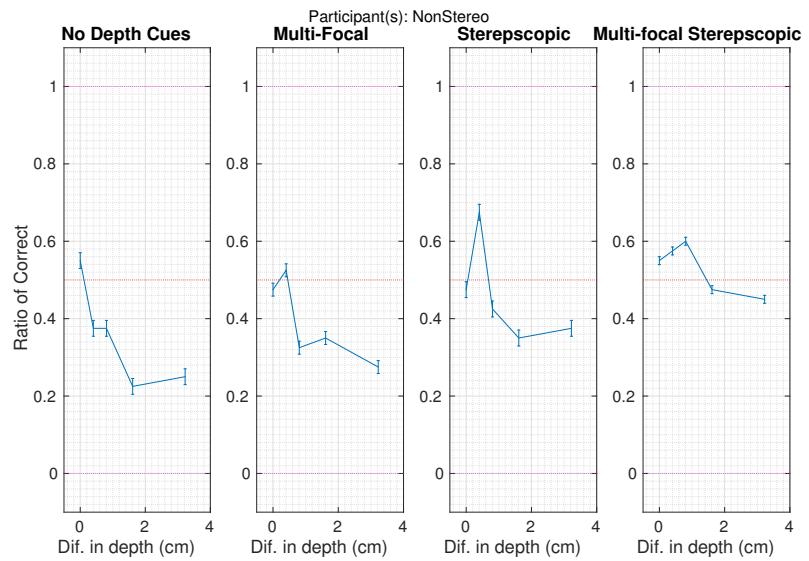
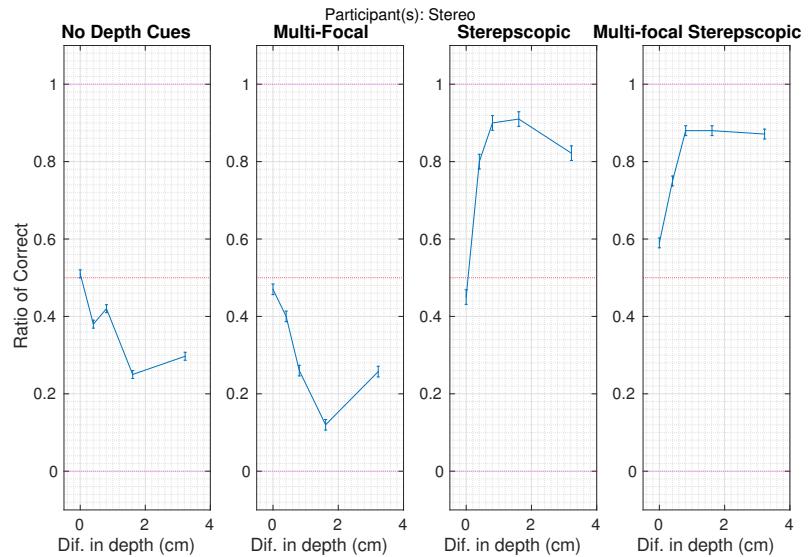


Figure 7.5: This graphs shows the combined results of all participants for all rendering modes and offsets.

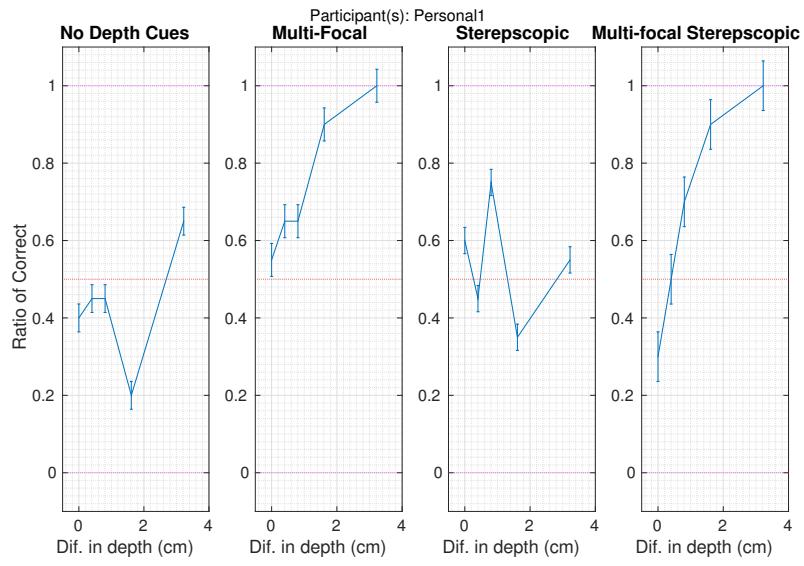


(a) Participants unable to see stereo depth. (Participants B and E)

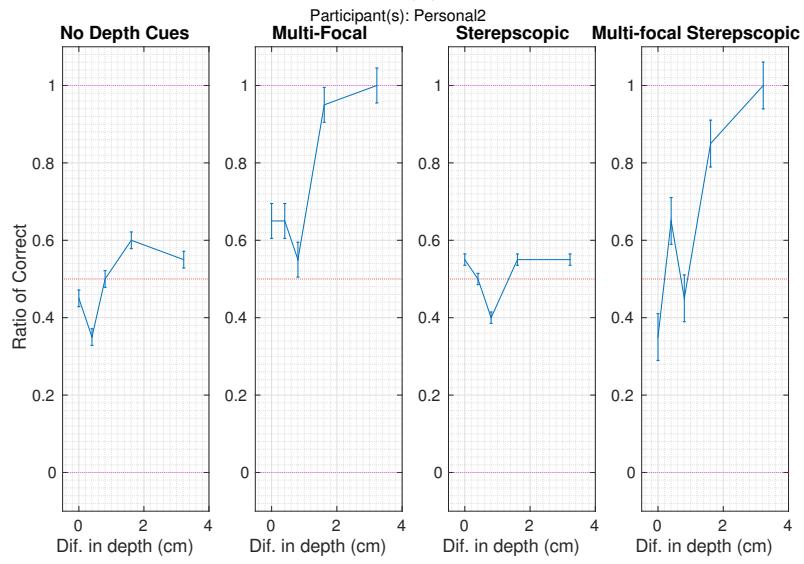


(b) Participants able to see stereo depth. (Participants A,C,D,F,G)

Figure 7.6: These graphs show the separate results of participants who were or were not able to perceive depth from only stereographic cues.



(a)



(b)

Figure 7.7: Personal runs of the experiment

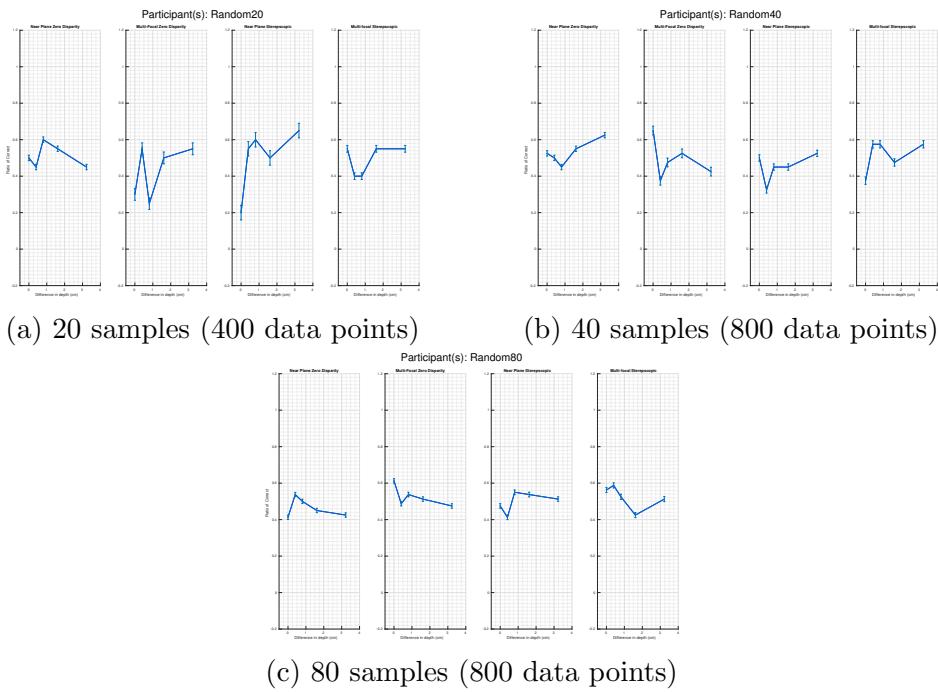


Figure 7.8: Random choice for different samples sizes per offset and render-mode. The random noise is consistent with what expect for this distribution (Figure 6.6).

Chapter 8

Summary and Conclusions

In this project we have constructed a multi-focal plane stereo display and accompanying software that is able to test a wide range of multi-focal stereoscopic rendering problems particularly new techniques for rendering with realistic depth cues.

We then ran an experiment to identify and measure the perception of depth across different rendering modes which attempted to isolate depth cues. Our results show that multi-focal rendering does have an affect on the perception of depth in some instances but the exact measure of that effect is difficult to judge from our data due to problems in the experiment. We can, however, conclude that the use of multi-focal depth cues did not negatively effect the perception of depth within our tested depth ranges when used with a stereo configuration.

The display that has been constructed for this work has performed exactly as intended and met all the requirements we set forth in Section 4.1. The software written to control the display also functioned well and performed all of it's required tasks, with the exception of one small bug in the experiment setup script which forced us to restart the experiment, but that was unrelated to the rendering goals of the software which it performed without fault.

Both the display and software are extensible and configurable so they will be

able to run experiments discussed in Section 7.2 in the future. Due to time restraints we are not able to run these experiments ourselves but with the provided software they should be trivial to implement.

Additional to the experiment mentioned in Section 7.2 in the future it would be interesting to see investigation into the comfort of the user when using this display or other multi-focal displays, and how that varies when different depth cues are enabled. Particularly it would be interesting to measure the comfort as objects pass across the screen as the distance to the near plane is only correct at the centre of the screen, would that conflict of depth cues be noticeable and would it be more or less comfortable than the vergence-accommodation conflict in standard stereoscopic rendering.

Overall the results from our exploratory experimentation are limited but have identified interesting complications we did not expect at the outset of this project such as the varying strength of responses different people have to depth cues, the possible dependencies between depth cues including how they may alter when combined differently and the scale of experiments which are needed with this type of display to account for all the possible variables that can affect it. We hope that it gives additional insight into the areas which this display can be used to further investigate multi-focal planes as a means to improve stereoscopic 3D rendering.

Appendix

BRIEFING FORM

Experiment: Depth Perception with a Multi-focal Stereo Display Date: 31 May 2016

Thank you for taking part in the experiment. The experiment will take approximately 20 minutes and will be comprised of one 20-25 minutes section. Please read the following instruction carefully before starting the experiment.

The purpose of this experiment is to compare the effect on perceived depth when given extra depth cues. The results of this will help in the design of future displays.

You will be shown two objects at different depths. The size, shape and lighting on these objects is no indication of the depth of the object. You will have to select which of the objects is nearest using the ‘.’ and ‘,’ keys to select the left or right object. After each selection you will be shown a calibration screen to ensure the screens remain calibrated correctly for your eyes, press ‘/’ when this is correct to move onto the next comparison. This will be performed for 400 times.

In this experiment you are asked to avoid making any decision which based on misalignment of the screens and are encouraged to take a break between comparisons if you feel any discomfort.

[END]

Oblique Matrix

```
//Left Eye
mat4 mlt(1.0f);
mlt[0][0] = 1.f/aspectx;
mlt[1][1] = 1.f/aspecty;
mlt[3][3] = 1.f;

float eyeDepth = -5.4f;

float near = 8.1f;
float far = 5.4f;

vec3 screenNormal = vec3(0.f, 0.f, 1.f);
vec3 screenPos = vec3(0.f, 0.f, 0.f);

float pr22 = -1.f/(far-near);
float pr23 = near/(far-near);

vec3 eye = vec3(-_eyeOffSet, 0.f, eyeDepth);
vec4 nn = vec4( screenNormal, -dot(screenPos, screenNormal) );
vec4 eyePosW = vec4(eye, 1.0f);

mat4 PR = outerProduct(eyePosW, nn )
    + ( mat4(1.0f) * dot(screenNormal, screenPos - eye) );

PR[2][2] = pr22;
PR[3][2] = pr23;
Proj_Oblique = PR * mlt;
```

Bibliography

- [1] Claire C Gordon et al. *1988 Anthropometric Survey of US Army Personnel-Methods and Summary Statistics. Final Report.* 1989.
- [2] Frank Biocca. “Will simulation sickness slow down the diffusion of virtual environment technology?” In: *Presence: Teleoperators & Virtual Environments* 1.3 (1992), pp. 334–343.
- [3] Mark Mon-Williams, John P Warm, and Simon Rushton. “Binocular vision in a virtual world: visual deficits following the wearing of a head-mounted display”. In: *Ophthalmic and Physiological Optics* 13.4 (1993), pp. 387–391.
- [4] Andrew J Woods, Tom Docherty, and Rolf Koch. “Image distortions in stereoscopic video systems”. In: *IS&T/SPIE’s Symposium on Electronic Imaging: Science and Technology*. International Society for Optics and Photonics. 1993, pp. 36–48.
- [5] Michael S Landy et al. “Measurement and modeling of depth cue combination: in defense of weak fusion”. In: *Vision research* 35.3 (1995), pp. 389–412.
- [6] John P Wann, Simon Rushton, and Mark Mon-Williams. “Natural problems for stereoscopic depth perception in virtual environments”. In: *Vision research* 35.19 (1995), pp. 2731–2736.
- [7] Tetsuri Inoue and Hitoshi Ohzu. “Accommodative responses to stereoscopic three-dimensional display”. In: *Applied optics* 36.19 (1997), pp. 4509–4515.
- [8] Geoffrey S Hubona et al. “The relative contributions of stereo, lighting, and background scenes in promoting 3D depth visualization”. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 6.3 (1999), pp. 214–242.
- [9] Kurt Akeley et al. “A stereo display prototype with multiple focal distances”. In: *ACM transactions on graphics (TOG)*. Vol. 23. 3. ACM. 2004, pp. 804–813.

- [10] Neil A Dodgson. “Variation and extrema of human interpupillary distance”. In: *Electronic imaging 2004*. International Society for Optics and Photonics. 2004, pp. 36–46.
- [11] Simon J Watt et al. “Focus cues affect perceived depth”. In: *Journal of Vision* 5.10 (2005), pp. 7–7.
- [12] Rafal Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. “A perceptual framework for contrast processing of high dynamic range images”. In: *ACM Transactions on Applied Perception (TAP)* 3.3 (2006), pp. 286–308.
- [13] David M Hoffman et al. “Vergence-accommodation conflicts hinder visual performance and cause visual fatigue”. In: *Journal of vision* 8.3 (2008), pp. 33–33.
- [14] Sheng Liu, Dewen Cheng, and Hong Hua. “An optical see-through head mounted display with addressable focal planes”. In: *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*. IEEE. 2008, pp. 33–42.
- [15] Gordon D Love et al. “High-speed switchable lens enables the development of a volumetric stereoscopic display”. In: *Optics express* 17.18 (2009), pp. 15716–15725.
- [16] Daphne Bavelier et al. “Removing brakes on adult brain plasticity: from molecular to behavioral interventions”. In: *The Journal of neuroscience* 30.45 (2010), pp. 14964–14971.
- [17] Sheng Liu and Hong Hua. “A systematic method for designing depth-fused multi-focal plane three-dimensional displays”. In: *Optics express* 18.11 (2010), pp. 11562–11573.
- [18] Kevin J MacKenzie, David M Hoffman, and Simon J Watt. “Accommodation to multiple-focal-plane displays: Implications for improving stereoscopic displays and for accommodation control”. In: *Journal of Vision* 10.8 (2010), pp. 22–22.
- [19] Stephan Reichelt et al. “Depth cues in human visual perception and their realization in 3D displays”. In: *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics. 2010, 76900B–76900B.
- [20] Takashi Shibata et al. “The zone of comfort: Predicting visual discomfort with stereo displays”. In: *Journal of vision* 11.8 (2011), pp. 11–11.
- [21] Takashi Shibata et al. “Visual discomfort with stereo displays: Effects of viewing distance and direction of vergence-accommodation conflict”. In: *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics. 2011, 78630P–78630P.

- [22] Yasuhiro Takaki, Kosuke Tanaka, and Junya Nakamura. “Super multi-view display with a lower resolution flat-panel display”. In: *Optics express* 19.5 (2011), pp. 4129–4139.
- [23] Robert T Held, Emily A Cooper, and Martin S Banks. “Blur and disparity are complementary cues to depth”. In: *Current biology* 22.5 (2012), pp. 426–431.
- [24] Parth Rajesh Desai et al. “A review paper on oculus rift-A virtual reality headset”. In: *arXiv preprint arXiv:1408.1173* (2014).
- [25] Cass Everitt. “Beyond Porting”. In: *SteamDevDays*. NVidia. 2014.
- [26] Bochao Li, Ruimin Zhang, and Scott Kuhl. “Minication affects action-based distance judgments in oculus rift HMDs”. In: *Proceedings of the ACM Symposium on Applied Perception*. ACM. 2014, pp. 91–94.
- [27] Nick Whiting. “Lessons from Integrating the Oculus Rift into Unreal Engine 4”. In: *Oculus Connect*. Epic Games. Oculus, 2014.
- [28] Vernon Harmon. “Embrace Virtual Reality with PlayStationVR”. In: Sony. MIGS15, 2015.
- [29] Rafal K. Mantiuk. “Perceptual display calibration”. In: *Displays: Fundamentals and Applications*. Ed. by Rolf R. Hainich and Oliver Bimber. 2nd. CRC Press, 2016.
- [30] Michael Abrash. *Down the VR rabbit hole: Fixing judder*. <http://blogs.valvesoftware.com/abrash/down-the-vr-rabbit-hole-fixing-judder/>. Accessed: 2016-06-05.
- [31] Paul Bourke. *Calculating Stereo Pairs*. <http://paulbourke.net/stereographics/stereorender/>. Accessed: 2016-06-05.
- [32] Ben Lewis-Evans. *Designing to Minimize Simulation Sickness in VR Games*. <http://www.gdcvault.com/play/1022772/Designing-to-Minimize-Simulation-Sickness>. Accessed: 2016-06-05.
- [33] OpenGL. *OpenGL - GLFrustum*. <https://www.opengl.org/sdk/docs/man2/xhtml/glFrustum.xml>. Accessed: 2016-06-05.