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

Nicholas G. Timmons
Downing College

UNIVERSITY OF CAMBRIDGE

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 2, 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:

Abstract

This is the abstract. Write a summary of the whole thing. Make sure it fits in one page.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research has been performed to fulfil the requirements of the MPhil Advanced Computer Science course. It is an investigation into the effects of the presence of multiple focal planes, and different interpolations techniques through the construction of a custom display setup on countering the vergence-accommodation problem and how it impacts the perceived realism of the images being shown. This is of particular interest in the areas of VR (Virtual Reality) where the user is faced with two conflicting focal depths for images appearing at the same location.

The goal of this research is to contribute to the knowledge base for VR(Virtual Reality) and AR techniques so that in the future similar techniques and ideas can be used in development of hardware to improve the user experience of depth in 3D displays.

Add some of the results of the work here

It's often useful to bring forward some "highlights" into this chapter (e.g. some particularly compelling results, or a particularly interesting finding).

This report includes a background of current techniques being used for VR/AR and a short overview of similar methods using multi-focal planes. There is then a breakdown of the requirements for the software and hardware for a multi-focal display with details of how they have been implemented to achieve

our goals. Finally there is a more in depth analysis of our results, what they reveal and how they can be built upon in the future.

Chapter 2

Background

Current trends in VR/AR Brief overview

Stereo displays such as those used for showing 3D movies and used in HMD (head mounted displays) which are in use in many commercial products have seen significant improvement over the past decade but still have some large technical and usability problems which could restrict the wide-scale appeal of the products [REF: The zone of comfort...] 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 psycho-physical and usability studies have suggested that the current solutions can lead to various problems including distortion of perceived depth [REF], visual fatigue [REF], diplopic vision (double vision) and degradation in the oculomotor response (as measured by slight movements within the eye). There are many factors contributing to cause these conditions varying from low quality images such as the high persistence in the Oculus Rift display, incorrect interocular distances or the inability to allow the eyes to rest. A major cause that is often mentioned is the discrepancy between the accommodation and vergence when using these displays with a fixed real focal distance. In this context accommodation refers to the focusing of the eye on objects at different dis-

tances and vergence is the motion of eyes rotating to bring the convergence point of the visual axis to intersect at the distance of the object. These two oculomotor actions are coupled when looking at an object in the real world but cannot 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 diagram.

past research has suggested [WATT by [Watt el al]] that the breaking of the link between accommodation and vergence cues can lead to a decreased perception of depth, which will effect 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 AR environments where the virtual environment is mixed with the real world and depth cues from the display would have to be correct to maintain consist with the real world.

2.1 What is realism?

In the context of this research we are considering realism to consist of many 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 rasterisation 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 scene appears to the user when compared to a scene which is lacking such cues.

2.2 Depth Cues

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

Perspective	Objects further away from the eye appear smaller.
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 more detailed objects are closer.
Occlusion	An object that occludes another is closer.
Lighting	The human visual system is accustomed to seeing objects under different lighting conditions and can there reason the position of objects in a scene by the lighting and shadowing between them. It is also able to judge distance by the slight dimming of objects in the distance due to atmospheric scattering.
Relative motion	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.

For most people 'Binocular Disparity' is often considered to be the most important depth cue and as such is the cue that is most well represented by consumer 3D products at the moment [REF], however conflicts in the 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 vertical bars are particularly problematic for finding stereoscopic matches without other cues and this can lead to visual strain and fatigue. [ref]

Binocular disparity	The difference in the apparent position of objects under projection caused by the interocular distance between the two eyes.
Accommodation	The changing of the focal lenfth of the lens in the eye to focus the view on an object at a particular depth.
Convergence	Bringing the two images from each eye to overlap more coherently and face a specific point at a particular depth through rotation towards and away from the interocular midpoint.

Table 2.2: Depth cues in three dimensional or multi-focal images.

2.3 Standard Stereo VR Implementations

The current standard for VR in use in a number of commercial hardware implementations consists of a head mounted display with the screen split 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 and to reduce the "Screen Door Effect" [ref] caused by the pixel density. The software is then implemented using the interocular distances of the screen in the display with correct field of view and a standard parallel projection for each eye (see section offaxis).

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 between the two images and perceiving them as a single object at an expected 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 distance which is different than the expected distance for the motion and stereo correspondences that are being shown in the scene.

This triggers the Vergence-Accommodation conflict mentioned earlier, give section num from a conflict in visual cues. There are examples of trying to

use simulated Depth of Field blur in early VR and large screen 3D for very near objects 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. cite Unreal talk

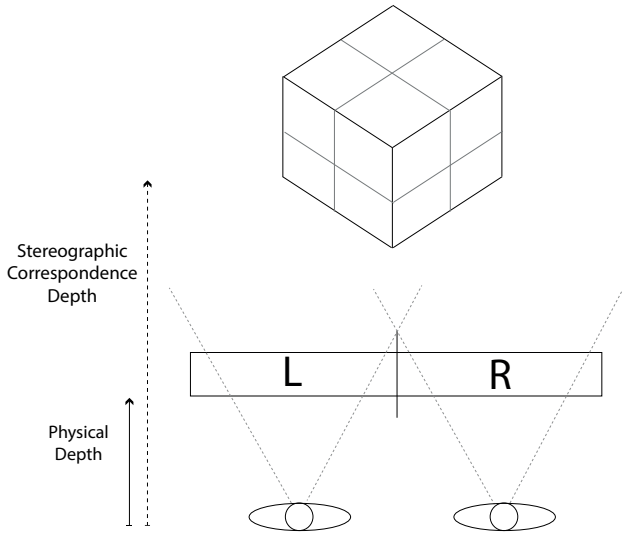

Figure 2.1: Basic comparison of the stereo and accomodation depths.

Papers:

Minification Affects Action-Based Distance Judgments in Oculus Rift HMDs
Speccifications: A Review Paper on Oculus Rift & Project Morpheus

Reference system requirements from current state of the art (75fps etc.)

2.4 Vergence-Accommodation Conflict

Papers:

The zone of comfort: Predicting visual discomfort with stereo displays (2011)
Immersive stereo displays, intuitive reasoning, and cognitive engineering (2009)

Vergenceaccommodation conflicts hinder visual performance and cause visual fatigue(2008)

The Vergence-Accommodation conflict is a source of discomfort and disorientation for a lot of users of stereo displays. It is caused by a mismatch of the focal depth and vergence depth cues. It is a problem which originated from trying to simulate depths of objects which have a different depth in software leading to visual disparity between eyes but an incorrect focal depth for those objects. This is particularly a problem for standard stereo displays because of 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" [ref].

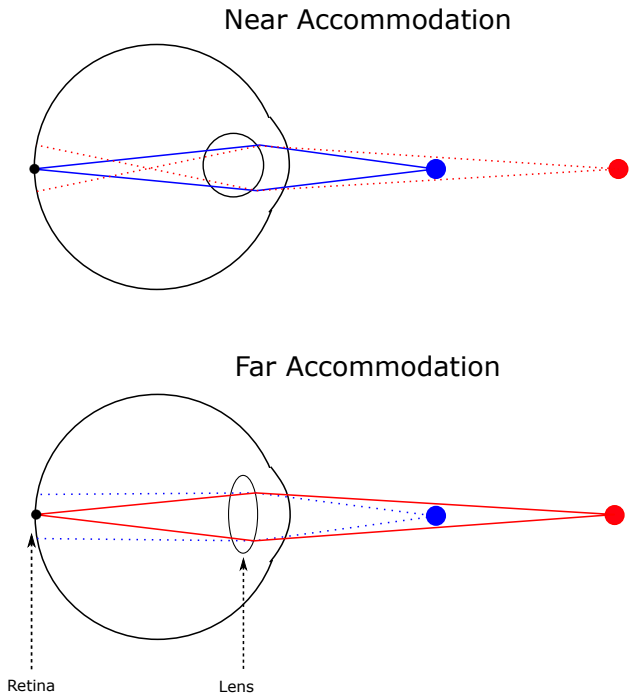

Figure 2.2: This diagram shows the basic physics behind the effect accommodation has to bring near and far objects in and our of focus by converging light towards the retina.

While this conflict does have some effect on the comfort of the user REFD the zone of comfort paper, 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.

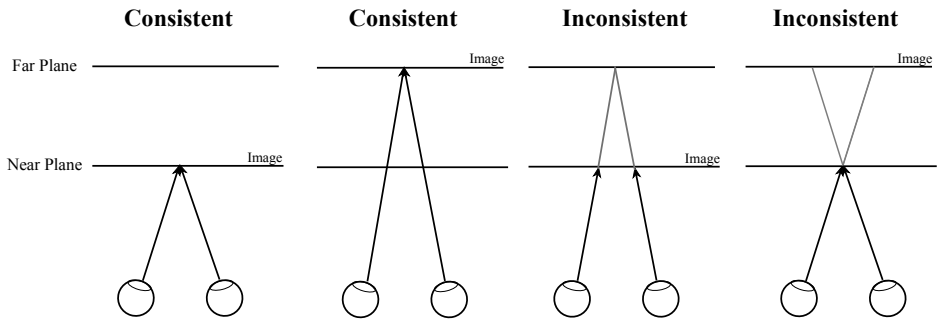

Figure 2.3: This is an illustration showing the different configurations which can cause a vergence-accommodation conflict

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 method as it reduces the ability of the user to view 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 in the viewed scene and allow it to be properly processed the same as any real-world scene.

This will need diagrams - borrow fancy camera and use images to help clarify

Chapter 3

Related Work

Discuss papers:

A Stereo Display Prototype with Multiple Focal Distances (2004) Super multi-view display with a lower resolution flat-panel display An Optical See-Through Head Mounted Display with Addressable Focal Planes A systematic method for designing depth-fused multi-focal plane three-dimensional displays

talk about pixel limitations with regard to requirements of modern setups as well as real distances.

The related work chapter should usually come either near the front or near the back of the dissertation. The advantage of the former is that you get to build the argument for why your work is important before presenting your solution(s) in later chapters; the advantage of the latter is that don't have to forward reference to your solution too much. The correct choice will depend on what you're writing up, and your own personal preference.

3.1 Multi-focal viewing

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

The idea for multi-focal viewing is to use multiple displays, or sections of displays ref 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 ref and 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 the images need to match perfectly to create the illusion of a single image add diagram. 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 the two planes.

Alternative more complex methods have been suggested such a lenticular display setup [REF], however this method suffers from crosstalk and, like many complex 3D viewing displays, requires eye tracking. Other alternative designs include methods which make use of the active stereo techniques to limit the number of screens [High speed switchable lens allows development of a volumetric stereoscopic display] 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 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 gives some very good data on the topic. They showed a mean distance of 6.47cm for men and 6.23cm for women, figure ref. Using this data work has been carried out to support many users hl[ref sytematic method for designing depth fused multi-focal plane three-dimensional display]

The downside to this type of method 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 ref such as gum-shields to bite down on and fixed eye positions. However, these are a little impractical for a study covering multiple users which will need to support multiple distances and interocular distances.

http://www.dtic.mil/dtic/tr/fulltext/u2/a209600.pdf

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

Figure 3.1: Interocular distances

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 [ref minication affects action-based distance judgements in Oculus Rift HMDs] were able to show that a user in the correct configuration could correctly navigate to a specific position in the scene, where as a user who was not calibrated would have difficulty due to "minification".

This shows the ability to accurately determine the depth and position of objects is affected by depth cues, whether they are from stereo correspondences, projection or otherwise. Which gives us reason to believe that the accuracy and depth perception can be improved by the introduction of further depth cues. Particularly, focal length was shown as a way to determine the depth of objects from separate images static scenes under varying levels of focus [http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4767940].

Chapter 4

Hardware

Speak about how we want to measure the down sides of the current solutions and how we can compare them to the improved view.

The goal of this implementation is to create a display which we will be able to use to investigate the relative perceived realism of the scene through the effectiveness of depth perception when given the extra focal depth cue 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 realism for this work but also to be configurable for 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 only 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 would be ideal that the displays were configurable to support both male and female average interocular distances.

4.2 Display Design

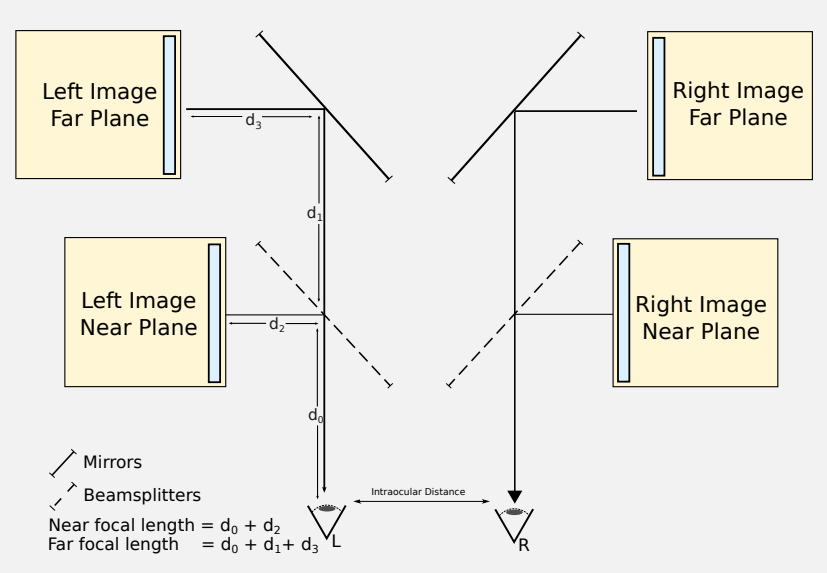

Figure 4.1: A high level view of the layout to achieve the stereo multi-focal images.

To achieve the requirements of the display we will be using four high resolution displays [ref ipad displays] to get maximum quality in rendering per view and reduce visible problems from low pixel density.

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. For each side of the display the images are seen through a combinations of mirrors and a 50/50 beam-splitter which merges the two images into one and then another mirror directs the combined image to the users eye. See fig layout for a more detailed plan 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 attain 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 each of the images being displayed on each screen must be matching 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 for 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 taken from dismantled iPad Air 2's. 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

Figure 4.2: Display Panel Spec

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

ref diagram / new component diagram.

4.2.3 Machine Specification

Although this display will work on any machine supporting 4 *mini display ports* it is worth mentioning the setup fig:machineSpecwe are using and why 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

Figure 4.3: Controlling Machine Spec

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 loggins applications we are using alongside the main multi-focal program will not interfere with the performance of the display. It also allows for more other 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 for a lot of monitors to be attached. 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 [Ref rafal].

We have quite powerful GPU's due to the need to render to 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. EXPAND

4.2.4 Known limitations

Limited eye coverage (low FOV). As our screens will only cover a limited FoV compared to the full coverage of HMDs, so we are limited in our ability to give as full an immersive effect, rather 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 or different focal lengths. ref fov image

iPad Displays While working with the displays we have found that the glossy coating on the screen increases light pollution within the display and from some angles within the display there is a clear reflection of another screen. This can be mitigated with more obstructing panels to prevent it

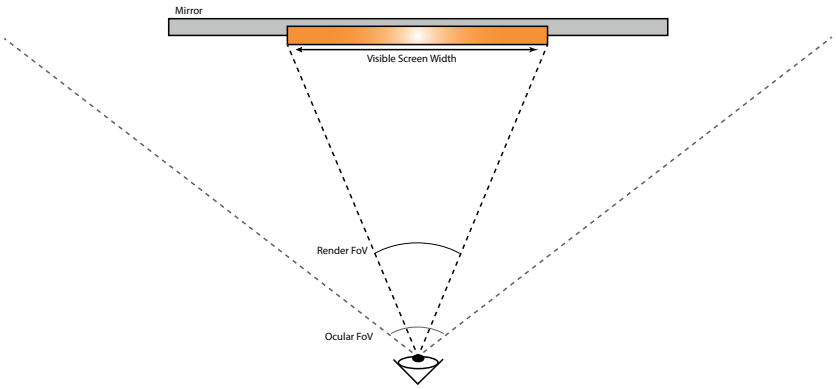

Figure 4.4: Showing the limited field of view when using this method to display multi-focal stereo.

interfering with any testing but it raises the potential improvement from using matte display panels in the future.

Relative Head Position As this display is not physically attached to the user the user is able to slightly move the 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 in the view direction. To alleviate this, extra consideration will have to be made in software design and user testing 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.

Lacks 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 which is less intuitive. 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.

Head motion As this is a desk mounted display the head position in the scene should be fixed. Motion of the camera in VR without actual head motion can cause motion sickness from lacking motion cues from ears [ref]. So in order to give the user cues of depth from relative motion we are relying on objects in the scene moving relative to the camera and avoiding any scenes which could feel like the user is moving. This may mean relying on static object to be points to ground the user perception. Correct depth in the scene is only maintained when looking at the centre of the view so to avoid the user from moving focus around the displayed image we will want objects to try and only be in motion in a single position unless we are measuring how distance tracking is changed when they are in motion.

Chapter 5

Software

This section will cover the considerations in the design of the software and the solutions to problems specific to multi-focal rendering.

5.1 Software Requirements

Configuration: Four render outputs - One per screen Full standard rendering support - To be able to support objects to test depth + textures and what have you. Configurable positions for each screen - For alignment. Adjustable scene depth planes - to allow testing of mismatch.

In addition to the blending we also need to ensure that the field of view being rendered in the scene is the same as the real field of view of the human eye. Investigations have shown minification action-based distance judgements in oculus rift how decreasing the field of view causes the user to misjudge distances. This could effect the perceived distance to the near plane and as such in our software the FoV will have to be correct to maintain that depth cue accurately.

Rendering As this is implementation will require the use of multiple outputs and be displaying real-time 3D it will also need to support 3D mesh,

texture and shader loading as well as lighting control. For this we will be using OpenGL as the graphics API and building support onto that.

Testing The software will need to be able to support real-time configuration of the screen position within each display panel to support multiple users during testing. This will mean accounting for error from head position, interocular distance 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.

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 and rely on having the plane of projection at a fixed convergence distance and at that distance there should be zero vertical or horizontal parallax between the left and right eye.

Vertical parallax is generally avoided due to it causing disparity between the eyes in the vertical plane which there should only be a very small amount of as the eyes sit approximately on the same horizontal plane.This error can break the illusion, cause diploptic vision 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 and negative parallax. 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 with positive parallax are more comfortable and we aim to avoid too much negative parallax as it can quickly become extreme or cause clipping from the screen when it 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'. Talk about maintaining correspondence...

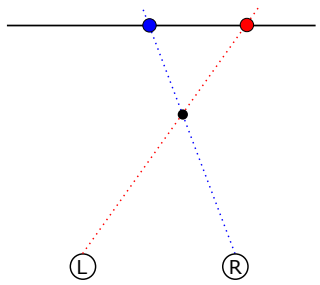

Figure 5.1: Example of negative parallax

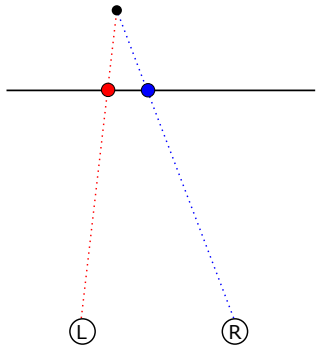

Figure 5.2: Example of positive parallax

Toe-in This projection mode points both cameras at a single focal point. This method produce 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 depth due to the non-parallel projection 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.

Off-Axis Off-Axis projections corrects the non-parallel projection planes of the 'Toe-In' method and forces the camera direction to be parallel. These

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

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

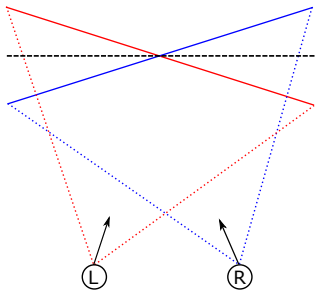

Figure 5.4: Toe-in projection layout

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 individually. An added benefit of this method which is sometimes used in 3D cinema experiences is to alter the interocular distance to exaggerate depth.

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

Figure 5.5: Standard frustum OpenGL matrix [ref]

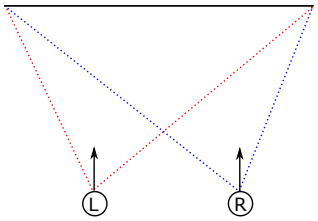

Figure 5.6: Off-Axis projection layout

References: Hodges, L.F. and McAllister, D.F. Stereo and alternating pair techniques for display of computer generated images IEEE Computer Graphics and Applications 5,9, September 1985, 38-45

Oblique Our oblique projection is very similar 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 also 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 this is a good thing as it is an accurate model of vergence at the projection plane.

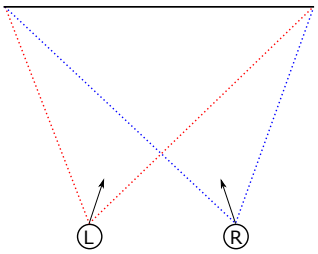

Figure 5.7: Oblique projection layout

See appendix for projection matrices used.

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 only be displayed on the near screens and only the points beyond the far focal distance will be displayed on the far screens.

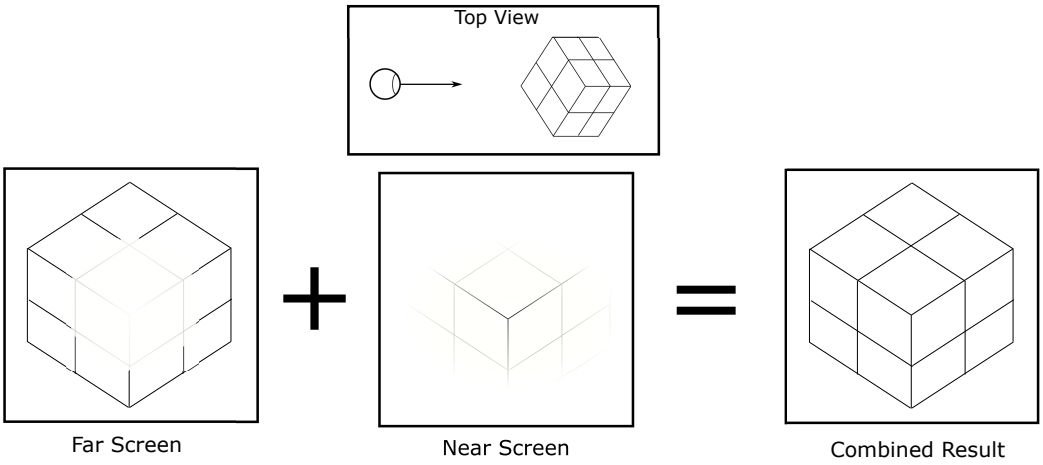

Figure 5.8: Simple example of how the near and far images are rendered and combined.

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 of the two distances to try and determine if it is possible to convince the user that these points exist at the appropriate distance between planes.

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 middle space will be considered on the near plane and all the ones more than half way across will be considered fully on the far plane.

$$
\begin{aligned}
n &= \frac{n_i - n_1}{n_2 - n_1} \\
col_{out} = f(n) &= \begin{cases} col_{near} & \text{if } n \text{ is } < 0.5 \\ col_{far} & \text{if } n \text{ is } > 0.5 \end{cases}
\end{aligned}
\tag{5.3}
$$

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.

$$
\begin{aligned}
n &= \frac{n_i - n_1}{n_2 - n_1} \\
col_{out} &= (n * col_{near}) + ((1 - n) * col_{far})
\end{aligned}
\tag{5.4}
$$

Non-linear: As the points move across the middle space they will be non-linearly interpolated using a modified sigmoid curve between the two distances as shown in fig.blend.

$$
\begin{aligned}
n &= \frac{n_i - n_1}{n_2 - n_1} \\
blend &= \frac{1}{1 + exp((-n * 2 + 1) * 6))} \\
col_{out} &= (blend * col_{near}) + ((1 - blend) * col_{far})
\end{aligned}
\tag{5.5}
$$

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

$$n = \frac{(1/n_i) - (1/n_1)}{(1/n_2) - (1/n_1)} \tag{5.6}$$
$$col_{out} = (n * col_{near}) + ((1 - n) * col_{far})$$

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

$$col_{out} = col_{near} \quad \text{or} \quad col_{out} = col_{far} \tag{5.7}$$

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 (ref) and non-linear(ref) 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 vert 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 '1/Depth' projection that is used in the projection matrix. Combining this with the linear depth calculation we are maintaining correct focal depth consistency throughout the blend.

Add examples of this working - with camera on nice scene.

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.

For a given diffuse object when light hits it 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 from the reflected object 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. REWORD We will be able to fully test whether this method of distance splitting is effective using a purely matte test scene.

Show diagram of diffuse distance vs specular distance

5.1.4 Rendering costs

State Change

for (all object) (all eyes) is better than (all eyes) (all objects) For rendering simple scenes with OpenGL a high proportion of the costs can be the switching of GPU render state 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 state changes per object instead of twice the textures, meshes and data per object.

$$\begin{aligned} &m = \text{Object count} \qquad c = \text{View count} n = \text{State changes per object} \\ &\text{Per view} = c * m * n \\ &\text{Per object} = m * n + m * c \end{aligned} \tag{5.8}$$

GPU Memory Usage

The display being used with this software requires four screen outputting at eight bits per pixel so we start with a minimum of 96MB of GPU memory being dedicated to the screen buffer.

We would like to render in a HDR(High Dynamic Range) setup to allow for future tests with HDR displays and we want to render in linear XYZ colour space. This means we are required to use a linear (non-srgb) format

and since we will be transforming that result with an XYZ to RGB colour space matrix and gamma correction, ideally we would to store the values in a higher bit rate than eight per pixel to ensure accurate representation and transformation to reduce banding or artifacts in the resulting image. As such it is necessary to use one of the 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 the maximum quality and reduce the chance of errors from unwanted quantisation before the transform to RGB colour space.

Texture	Resolution	Bits Per Pixel	OpenGL Format	Size (MB)
Near Plane Render	2048 x 2048	32	*GL_RGBA32F*	128
Far Plane Render	2048 x 2048	32	*GL_RGBA32F*	128
Near Screen Buffer	2048 x 1536	8	*GL_RGBA8*	24
Far Screen Buffer	2048 x 1536	8	*GL_RGBA8*	24
Single GPU				304
Both GPUs				608

Figure 5.9: Per GPU

5.2 Software Configuration

Which blend methods were used? How will it accommodate different people. What problems will it overcome (colour correction, etc).

5.2.1 Rotational Consistency

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

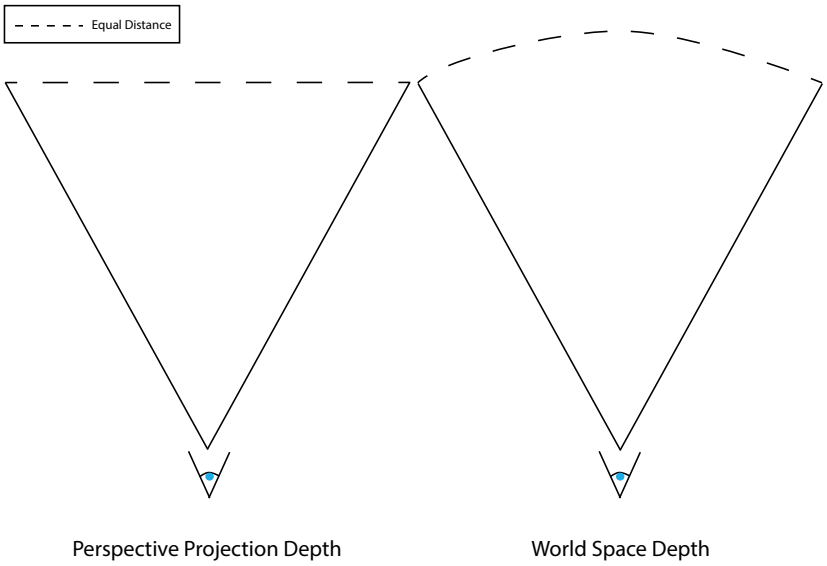

Figure 5.10: Showing the different distance to reach 1.0 unit of depth when comparing projected depth and world space depth.

dist which causes changes in depth as the screen is rotated as points move from being on the edge of the far plane to the centre.

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

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

Show image of different depth

Depth has to be distance from the eye position, not distance in view space. Show comparison of different depths.

5.2.2 X11 Window Controller and shared contexts

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 maximise the effects of persistence of vision ref].

Only full-screen applications are completely decoupled from the desktop screen present rate. Since we are using multiple discrete GPU's it is not possible to create a graphics context being 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 called "Zaphod" to allow the user to run multiple screens rather than just extending one and the user can select which screens are controller by which GPU. Using this we are able to launch windows directly on the separate screen and run them full screen which allows us full control of the presenting of rendered images up to the maximum supported refresh rate of the displays.

As we can select which GPU controls which displays it is possible to align the left side displays to share a context on one GPU and the right side to share a context on the other.

We want to remove as many barriers to presenting to the screen as possible to try and ensure that the screens are kept in sync and we do not fall out of alignment which could cause strange effects when objects are in motion,

5.2.3 OpenGL MRT's

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 ref] this is not a limiting factor

in performance.

In order to take advantage of the data sharing we do have available we are making use of multiple render targets in our shaders so that we 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 matching VSync on both screen so we are less likely to suffer from screen mismatches from the screens not being synchronised.

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 recommended 75fps [ref] so that even if they are out of sync the difference between the two scenes should be low enough to be imperceivable by the human eye.

5.2.4 Colour Calibration

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 and 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 have slightly different light intensity and colour ranges.

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 touch them they should be safe from damage or problems.

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.

For each screen the spectro radiometer is setup and all the other screens are set to display black, to account 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 how? needed to ensure that each display for each colour input will output a matching colour out.

The calibration from each screen produces 'black level', 'gamma' value and matrix to map from the XYZ linear colour space to RGB for each display.

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 RGBREF CIE. 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 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 figure ref into the calibrated RGB colour space which is sent to the screens.

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

Add pictures with and without calibration

Calibration Results

Show curves for our displays and highlight the top displays black levels being increased due to mirrors.

Explain different screen configurations - Get graphs from Rafal

5.2.5 Full pipeline

tidyup.... Once the screen is fully calibrated and the environment has been setup, the process of rendering scenes for the multifocal view is relatively straight forward if you are used to rendering regular 3D scenes.

First you load all of the geometry and configure the position of the cameras and lights in the scenes. The cameras should be setup to match the projection that you would like to use and a blend mode should be chosen and implemented in the shaders.

The geometry is then loaded into each OpenGL context, and the cameras and projection matrices for the correct views are applied to each from the appropriate camera and projection are computed.

When it comes time to render the scene for each view we set the fragment shaders to render to two linear 32-bit render targets (Frame buffer objects) each. One represents the near plane and the other the far plane. We then apply the pixel shader which we have implemented some lighting and the blend functions as described in section somesection.

When the geometry is submitted is then correctly positioned in the scene

and lit using the shader. At the end of the shader the blending is performed on its real world depth and then based on the depth and the blend mode selected the output is then applied to both render targets. For example, if a triangle is submitted which sits in front of the near plane and we have selected “projective‘ blending then 100% of the value of each pixel in front of near plane will be applied to the near plane render target and black will be written to the far plane. This effectively splits the scene between the two planes as shown in figsplit.

After all of the rendering is complete the resulting render targets are then written to a plane on each screen in there configurable position.

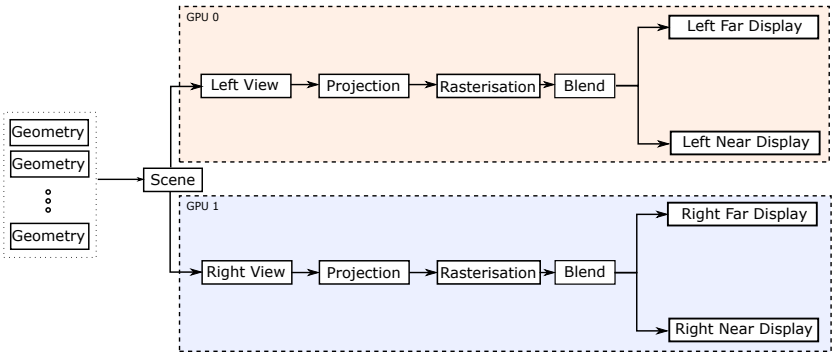

Figure 5.11: A high level overview of the steps to separate work between GPU’s and display panels.

Chapter 6

Methodology and Testing

In testing the device we are aiming to find out to what degree, if any, the use of multifocal viewing planes increases the users perception of depth in a scene when compared to stereo and standard viewing conditions. We are also interested in how this change in viewing condition effects the comfort of the user. It is hypothesised that by alleviating the vergence-accommodation conflict we should be introducing a more comfortable and realistic depth viewing experience however it is possible that the change in focal depths as the user looks around a scene could introduce discomfort from other means.

What does our early work with the display suggest? 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 tests simple and within scope we have performed some early testing to narrow the scope of the tests. As such we will be investigating using the "Oblique" projection method as it provided correct near and far plane correspondence and with the projective blending when gave us the least error from misalignment in the scenes. Avoiding error from misalignment was a large consideration in the decision of how to test as we will have little control over ensuring the user is probably calibrated beyond some simple calibration screens so we wished to reduce the amount of error in our results from this.

In order to address our hypothesis' we will be testing comfort through scenes showing objects moving in and out of the screen, objects moving across the screen and a static scene with varies focus points at different depths. This will allow us to determine if the users sense of depth is greater or less than other methods and the relative comfort for each type of motion. Additionally we will also be testing the users ability to judge depth from 3D depth cues alone to measure the effect of the addition of the focal cue.

Display Distances

Add maths for calculating distance based on dioptres. Assumptions about parallel rays beyond far distance Targetting 0.6 dioptres

Why have we selected these distances specifically?

Near Focal Distance	54cm — 1.0/0.54m Dioptres
Far Focal Distance	81cm — 1.0/0.81m Dioptres
Interocular Distance	6.5cm ref
Convergence Distance	67.5cm

6.1 Method

All of the testing will be performed on the display and software as described in this work. It will be used to run tests with voluntary participants who will be guided through a number of tests scenes and a short game.

There are four scenes in total which will be used. Three of which are scenes for testing depth perception and comfort. The first of these involves three horizontally aligned blobs objects moving towards and away from the user, the second is a series of objects spinning objects in place and the third is a static grid of objects at different depths. The final scene is three blobs at different depths.

Translation Scene: The translation scene is designed to give the user multiple objects to look at which all traverse the same depth from past the far plane to very close to the camera and back. As the object transitions near to the camera the user will be effected by disparity between the images as horizontal parallax will effect the users ability to converge on the image from the stereo pair. When using the both focal distances it will also force the user to accommodate from very near to far. This will allow us to test aspects of stereo, mono and multi-focal rendering for its ability to infer depth to the user and measure how comfort is under each view mode.

Rotation Scene: Similar to the translation scene this scene is asking the user to focus on a moving point. However, unlike the translation scene this point is moving from the left of the screen to the right and in and our of the screen as it travels through its rotation. This allows us to measure the effect varying depths as the eye is in motion and whether the user is able to remain focused on the point comfortably through that process.

Static Scene: This scene is for allowing the user to select different depths and positions for focus in a scene where everything is static. This will allow us to measure the comfort and sense of depth when the scene is not requiring constant adjustment from the user.

Depth Comparison Scene: This is a more interactive scene. The user is shown three 3D blobs. The blobs are a 3D mesh generated with no obvious parallel lines or features which the user could use to determine size, shape or position. The blobs are placed at different distances from the camera, scaled based on the depth to appear the same size in screen space to the user, this reduces the ability of the user to determine the position from projective cues, and then they have an additional random scale to ensure that size on screen is not in anyway corresponding to depth and remove the chance of small error from projection giving clues. The user is then asked to select which blob is closest to the screen. By removing as many visual cues as possible

we are hoping that the only cues available should be binocular disparity, accommodation and convergence, then by comparing the scene with stereo and/or multi-focal rendering disabled we should be able to measure the effect each has on the perceived depth by the accuracy of the answers. Furthermore we will be able to determine at what depth each of these methods breaks down and the accuracy of each when objects are placed at small differences in depth together.

These first three scenes will be shown to the user as part of the *Comfort and Perception Test* and then a separate test will be ran for the *Depth Comparison*. In the comfort and perception tests we want the user to focus on how it feels using the device and whether they feel like they can see more depth where as in the depth comparison tests we will be looking to use the user data tro measure how accurately they can actually see the depth.

6.1.1 Comfort and Perception Test

The comfort and perception test uses Oblique projection for the projection mode and for the blending we will be comparing the projective blend to show multi-focal depth and then the near and far plane blending to test standard stereo rendering at different focal distances.

Each scene will show the user a comparison between two of the three blend modes and then ask the user to choose which is more comfortable. This is repeated five times per combination per scene. They will be shown in a random order to provide varied images so it isn't too boring for the user and each scene is shown for five seconds. The scene is only shown for a limited time to prevent the user from the using head motions to determine the rendering methods through breaking the calibration.

For each scene being shown the user is also shown a calibration screen beforehand ref and asked to press a key when the display is correctly positioned. It is shown each time to prevent the user from losing calibration as the tests progress.

As the comparison between the near focal stereoscopic, far focal stereoscopic and multi-focal rendering methods are recorded as single A/B tests we are able to then analyse the results using piecewise comparison to measure the differences between each methods for each scene and should be able to determine if there is any correlation in the results to prove or disprove parts of the hypothesis.

6.1.2 Depth Comparison Test

The depth comparison test uses a very similar projection and blend setup as the Comfort and Perception Test. Except there is the addition of switching to using a monoscopic eye setup for some of the tests. We will be testing the accuracy of depth perception using a multi-focal stereo setup, a near focal distance stereo setup and a far focal distance stereo setup then we will repeat the same tests using a mono setup.

By separating into mono we will be removing any depth cues from binocular disparity and vergence. It allows us to test the effectiveness of accommodation alone on depth perception with two multi-focal planes against a standard monocular display.

To isolate and measure where the most effective depth are we split the scene into five fixed depths and six depth ranges (figure). For each depth the user will be shown the three blobs at that depth with a maximum offset and minimum offset. This way we can isolate at which difference in depth around a given depth in the scene the user is able to distinguish depth and how that is effected under the different configurations.

As with the Comfort and Perception Test a calibration screen is shown between each test. However, there is no time limit for selecting the nearest blob. In this situation we want to give as much time as needed to the user to try and focus on the different objects.

table of depths and offsets

6.2 Results

Results

Show and discuss the results. Any problems, or limitation ran into during the gathering process etc.

Chapter 7

Evaluation

For any practical projects, you should almost certainly have some kind of evaluation, and it's often useful to separate this out into its own chapter.

What are the results? How did it perform? GRAPHS.

Does this match the hypothesis? Why?

What new information does this show?

By what margin is it better or worse?

Chapter 8

Summary and Conclusions

Mention if it reduced the vergence problem. Mention if it appeared more real. Which blend methods were convincing? Was allignment consistent and believable?

potential future work.