School of Computing

FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

Final Report

Alleviating Anaglyph Stereo Ghosting with Real Time Focus Plane and Individual Adjustments

Zachariah Woon Yin Lee

Submitted in accordance with the requirements for the degree of
MEng, BSc Computer Science with High Performance Graphics and Games Engineering

2023/2024

COMP3931 Individual Project

The candidate confirms that the following have been submitted*:*

Items	**Format**	**Recipient(s) and Date**
Final Report	*PDF file*	*Uploaded to Minerva (07/05/24)*
Scanned participant consent forms	*Zip file*	*Uploaded to Minerva (07/05/24)*
Link to online code repository	*URL*	*Sent to supervisor and assessor (07/05/24)*

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) Zach ____________________

Summary

Anaglyph 3D is a method to show a stereoscopic 3D effect on images or videos where colour encoded images combined with coloured lenses allows our eyes to see two different images in one which is perceived in our minds as an 3D image with depth.

Currently anaglyph 3D suffers from ghosting, which is when parts of the image meant for one eye is improperly separated and filtered resulting in it being perceived by the other eye, which leads to visual distractions and a reduced ability to focus and perceive the depth information. This effect is exacerbated by parallax, where in 3D objects far away and close depth wise to where the eyes are focusing on make ghosting more visible due to the images meant for the left and right eye being shown more separately. This can be alleviated by adjusting the colours of the images such that less light leaks through each lens, but this results in poor colour quality and accuracy.

This project aims to improve the anaglyph 3D experience by adjusting the focus plane such that what the user is focusing on has a minimal amount of parallax, theoretically resulting in a minimal adjustment of colour necessary to reduce ghosting, improving colour quality and preservation.

This was achieved by building on top of a renderer with the OpenGL rendering API, results show a significant decrease in the amount of perceived ghosting in the rendered scene, and in some situations the ghosting was minimal enough to warrant no correction needed.

Acknowledgements

I would like to firstly thank my supervisor Dr. Rafael Kuffner Dos Anjos for invaluable feedback and support throughout my project, and Dr. Markus Billeter for providing much needed support with the OpenGL API used in this project.

I would also like to thank my coursemate Oliver Gill for being in my group for the COMP3811 Computer Graphics module where we produced the renderer which formed the base of what I would build upon for this project.

I would also like to thank the many things that have supported my mental health throughout the development time of the project such as but not limited to: Lebanese Bites, Minecraft, Roblox, Mauga from the video game Overwatch 2, Diet coke, singer PinkPantheress, rapper CupcakKe, Bayonetta from video game franchise Bayonetta, Rania Concepcion Maggay (ed19rcm), James Parkes (sc222jp), Edward Das (sc20ed), Alisha May Machin (sc21a2mm), Alessia Stanistreet-Welsh (ed19asw), Jingnan Xu (psjx) and many others.

Table of Contents

Chapter 1
Introduction and Background Research

1.1 Introduction

Anaglyph 3D (most commonly known as Red-Cyan 3D) is one of the first widespread methods for displaying a stereoscopic 3D effect for images and videos. It uses colour encoding to show a stereo pair of images on one singular image, each image being filtered out by glasses which have coloured lenses. These lenses are usually separate in colour and provides separate images to each eye, which in the brain is processed as depth information, giving the 3D effect.

Currently, anaglyph 3D suffers from a multitude of issues such as poor colour quality and an effect known as crosstalk (ghosting), which is when light intended for one eye is visible to the other eye when it is improperly separated and filtered. Additionally, it suffers from retinal rivalry, where after filtering by the glasses an object looks bright in one eye but looks dark in another, causing a distracting effect when perceived.

A consequence of it being widespread is that it is often not tailored for any one person in particular. This means that an anaglyph image that looks good for someone can look bad for another due to the individual variations of our perceptual system.

Although other stereoscopic 3D methods like polarized lenses are now more common, anaglyph 3D is still the only way to present stereoscopic images without any special hardware and on printed media. Thus, it is still used in research within the graphics industry or the field of biomedical sciences such as in Neuroscience (Rojas et al., 2014) and Cardiology (Gupta and Gupta, 2021).

1.2 Terminology and Definitions

In literature in stereoscopy there is jargon which sometimes are used interchangeably which can cause confusion (Woods, 2011). Thus this subsection aims to define the key issues with anaglyph 3D and give a reference point to terminology used in this paper.

1.2.1 Anaglyph Stereo/3D

Anaglyph 3D is the technique of showing a stereo pair of images, meaning for the left and right eye, in one image through encoding information on which eye the image is for through colour channels. In combination with glasses with corresponding filters that block out the colour channels encoded for the other eye (Figure 1), this theoretically gives two distinct images for each eye, which is then combined inside the mind and perceived as depth (Ochotorena et al., 2013).

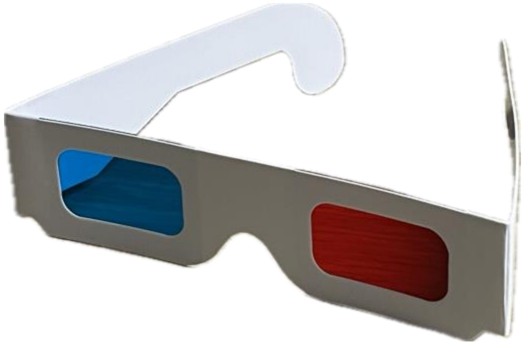

Figure 1 A pair of red-cyan anaglyph 3D glasses

Many different types of anaglyph exist which use different colour channels, such as red/cyan which is the most popular type, green/magenta and blue/yellow types (Woods and Harris, 2010). This project will be focusing on the most common type of red and cyan and all references in this paper to anaglyph 3D will be for red/cyan by default unless stated otherwise.

The simplest and original way of generating an anaglyph stereo pair of images from the original stereo pair is to use the Photoshop Algorithm. This is when you set the channels corresponding to cyan (green and blue) of the left image to 0 and do the same for the red channel for the right image, then combining the two images (Sanders and McAllister, 2003). In this method, the images for each eye are independent of each other and do not influence each other’s end result as an anaglyph pair. This method also does not take into account the display medium properties nor the properties of the anaglyph glasses

1.2.2 Ghosting and Crosstalk

Ghosting and crosstalk (example in Figure 2) are often used interchangeably in literature to refer to the idea of light displayed for one eye being perceived by the other eye, either by improper split of channels or by improper filtering by the glasses.

Andrew J. Woods (2011) has done research in the field of ghosting and has created definitions for key ideas in ghosting, thus for consistency I followed his definitions for separating the terms of ghosting and crosstalk.

“Crosstalk” is thus defined as “the incomplete isolation of the left and right image channels so that one image leaks into the other” (Woods, 2011), meaning in this project’s context as relating to the display and glasses components of Anaglyph 3D.

Furthermore “Ghosting” will be defined as “the perception of crosstalk” (Woods, 2011), in this project's context relating to the human perception component of Anaglyph 3D.

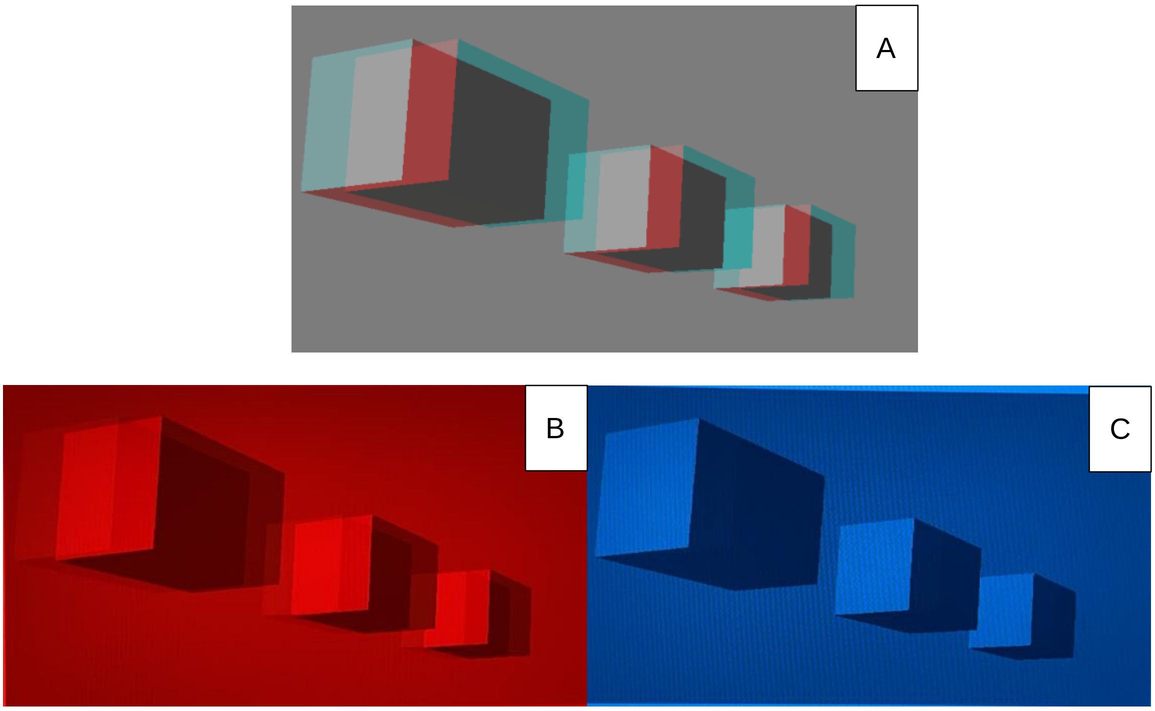

Figure 2 An anaglyph image (A), and a simulation of ghosting by taking a picture of the image through filters of a pair of anaglyph glasses (red (B), cyan (C)).

1.2.3 Retinal Rivalry

Retinal/Binocular Rivalry is the phenomenon where our left and right eye are perceiving two different things at the same time, resulting in a distracting effect where our brain tries to process the situation (Blake and Logothetis, 2002). In anaglyph 3D, this is a concern as due to the nature of the colour encoding, colours can look bright in one eye, while looking dark in another as some colours can be completely blocked by one filter but completely let through by the other (Ochotorena et al., 2013).

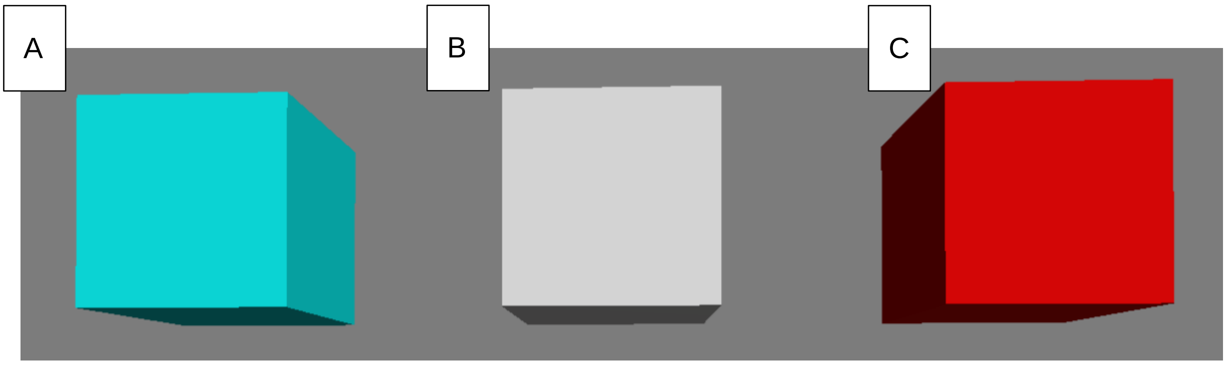

Figure 3 An image with cubes showing retinal rivalry (A, C), and low retinal rivalry (B)

1.2.4 Parallax and Binocular Disparity

Binocular Disparity is the difference in what our left and right eye perceives which is used as a clue by our visual system to form a perception of depth (Qian, 1997). The apparent distance between the same object viewed from our left and right eye is known as parallax. Due to perspective objects that are close to us experience more parallax than objects far away (Tozawa and Oyama, 2006).

In the context of Anaglyph, parallax is important as it is the cue that our brains use to perceive the depth of the image relative to the focusing point of our eyes (Reeve and Flock, 2010).

Positive (sinking) parallax, where the red channel's (left) image is on the left side relative to the cyan channel's (right) image, is perceived as behind the display (Reeve and Flock, 2010).

On the other hand, Negative (rising) parallax is when the where the red channel's (left) image is on the right side relative to the cyan channel's (right) image, is perceived as in front of the display (Reeve and Flock, 2010).

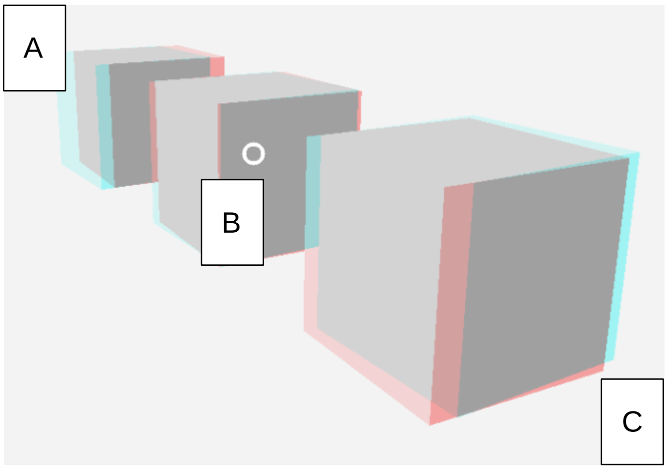

Figure 4 An image with cubes showing positive (falling) parallax (A), and negative (rising) parallax (C), with the focus plane being at circle labelled (B)

Parallax is related to ghosting as the visibility of ghosting is proportional to the absolute value of parallax, meaning objects far away in depth from the focusing point are more susceptable to ghosting as the high separation of the object's left and right counterparts in the anaglyph image is more noticeable to the viewer (Krupev and Popova, 2008).

1.2.5 Eye Dominance

Eye dominance is the preference for humans to use one eyes information over the other. There are two types of eye dominance known as motor and sensory dominance (Ooi and He, 2020).

Motor dominance is when we as humans physically angle our sight lines to align with one eye over the other. This is measured by a "hole in card" test, where subjects align a hole in a card to see a distant object through it; whichever eye was used to see the object through the hole is the motor dominant eye of that person (Lopes-Ferreira et al., 2013).

Whereas sensory dominance is the perceptual preference between the eyes. This is related to retinal rivalry as when it occurs, we can prefer the signal from one eye more than the other, which would be the sensory dominant eye (Ooi and He, 2020).

1.3 Literature Review

As anaglyph 3D has been a well-established method for more than a century, there is an abundance of literature on the topic area and literature tackling existing issues with anaglyph. Thus, I will have subsections on each of the problem areas critically analysing key ideas on issues I have chosen to take on with this project.

1.3.1 Anaglyph 3D Methods Tackling Ghosting

For digital media, Dubois (2001) presented a method of generating a stereo pair through measurements of the 3D glasses' colour filters and the display's colour properties. Called "Dubois Anaglyph", the method improves the visual experience by changing problematic colours that cause ghosting and retinal rivalry, such as bright reds and greens being converted to yellow (Figure 5). This method takes each image and maps each colour to a corresponding shade of red or cyan. These shades have been calculated by the method to not have too much ghosting or retinal rivalry through measuring the colours of the display monitor and the filters of the glasses (Dubois, 2001).

This method is effective in reducing ghosting and retinal rivalry. In addition, it can be used with other anaglyph types other than red-cyan such as magenta-cyan. However, colour accuracy is compromised as accurate colours that are problematic are adjusted to become less problematic but inaccurate to the original image (Dubois, 2001). Although in some images, such as real life photography, the colour inaccuracy can be looked past with the full context of the image (Hurlbert, 1996). Despite no special hardware required for this method, it does require measurements of the display and anaglyph glasses which may be difficult to obtain easily.

In industry, Dubois Anaglyphs have been used by several platforms, including YouTube (Chitu, 2009).

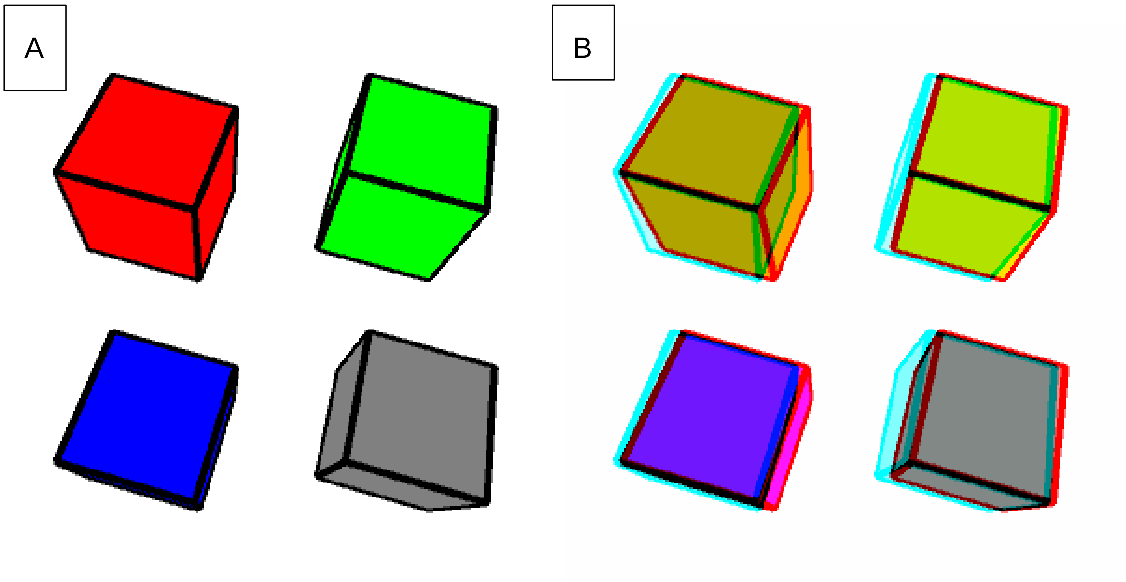

Figure 5 Original left eye image (A), and the Dubois Anaglyph result (B)

Conversely to Dubois' method, Sanftmann and Weiskopf (2011) presents an anaglyph stereo rendering method that tackles ghosting that does not take into account the properties of the display or glasses used, with human perception being the only factor measured. In addition, their method tackles ghosting by targeting differences in luminosity, also known as brightness, between the stereo pair, as opposed to Dubois' method which mainly targeted ghosting caused by the hue of the colour.

Sanftmann and Weiskopf's method (2011) corrects ghosting artifacts by attempting to minimise the difference of luminosity between the red and cyan components of the image. This is done by having the user adjust the luminance of 6 colours (red, green, blue, cyan, magenta and yellow) until it matched the luminance of a corresponding gray value. This was repeated for 20 gray values to achieve 120 measurements. Additionally, measurements can be simplified to just 3 for each eye by estimating the other values.

This method is advantageous as it does not need to account for displays and can be extended to other types of anaglyphs, meaning that the only information prerequisite needed is the type of anaglyph used, and user input can be decreased to just 6 measurements, resulting in a significant increase for ease of use. Conversely its reliance on user input results in inflexibility across users as measurements which all calculation is based on naturally differ amongst each user, as opposed to Dubois anaglyph, where results generally decrease ghosting artifacts across observers as it takes into account the filter and display properties which is user independent.

As opposed to Dubois' and Sanftmann and Weiskopf's methods which aim to reduce ghosting in the pre-processing and processing stage by changing how the left and right full colour images are processed into the anaglyph stereo pair, Krupev and Popova (2008) presents a method for reducing ghosting post-processing after the stereo pair is generated by reducing parallax through shifting the left and right eyes images to match the amount of positive and negative parallax, minimising the absolute value of parallax. And afterwards using gamma correction of the red, green and blue channels based on user input to a point where ghosting did not become a significant distractional issue.

Results indicate a decrease in ghosting artifacts at the expense of detail quality and a darker, more blue image. This presents an issue with gamma correction as although gamma correction does objectively decrease ghosting, it also concentrates ghosting in the most problematic areas, thus discouraging it being used by itself (Krupev and Popova, 2008).

Krupev and Popova's method is advantageous to Dubois' and Sanftmann and Weiskopf's method as it can be applied to already created anaglyph images, as opposed to the previously mentioned methods which cannot be applied due to their reliance on the full colour images for each eye to create the stereo pair, and additionally can be used in conjunction with them due to it not relying on the original full colour images for each eye.

Additionally, although Krupev and Popova's method preserved relative depth information, their adjustment of the left and right images does change the images Inter Pupilary Distance (IPD), which does have an effect on size and by extension depth judgement (Woldegiorgis et al., 2019).

Conversely to the previous methods which include directly changing the colour being displayed or the intensity of colour displayed, Ideses and Yaroslavsky (2005) proposes a method to tackle ghosting by defocusing (blurring) colour compoments. This is based on the notion that ghosting artifacts cause discomfort by interfering with depth cues. Thus, by blurring colour components with ghosting artifacts, our perceptual system does not pick up on the conflicting depth cues as strongly and thus should alleviate the issue of ghosting, (Ideses and Yaroslavsky, 2005).

One advantage of this method is that it is very versatile and flexible, as any colour channel can be the subject of defocusing, and it is not limited to only red-cyan anaglyphs.

Additionally, this method does not interfere with 3D perception such as Krupev and Popova's method (2005). And as defocusing does not majorly interfere with the location of colours, retinal rivalry is not inherently an issue of this method. Also, this method does not require any special measurements which improves ease of use, unlike Dubois' method (2001).

1.3.3 Anaglyph 3D Methods Tackling Colour Quality

As stated previously retinal rivalry is one of the many issues facing anaglyph 3D, and it is a contributor to visual fatigue.

McAllister et al. (2010) proposes a method to compute the anaglyph pair that preserves the colour more accurately to the original images by use of the CIE colour system, which is a colour system that is based on human perception ability rather than pixel colour channel intensity (Tkalcic and Tasic, 2003). Their method includes conversion of the original images from rgb values into the CIE colour space, then alters the stereo pair by finding rgb values which when converted to the CIE colour space, minimise the Euclidian distance between the left eye and right eye's pixel. This theoretically should decrease large differences in brightness between the red and cyan image, which decreases retinal rivalry.

McAllister et al. (2010) presents that results indicate a better representation of red colours, as compared to methods such as Dubois' (2001) method. And although the Photoshop method can show a brighter red colour, the retinal rivalry decreases the stereoscopic effect, resulting in a flat shape.

Qi et al. (2021) builds upon McAllister et al.'s method by also taking into account colour distortion in addition to retinal rivalry issue when calculating the pixel value from the CIE colour space. In addition, the method allows dynamic adjustment between hue preservation and retinal rivalry, allowing balance between hue and brightness adjustment, increasing effectiveness as it can be adjusted for images with high issues in either area.

Results indicate an improvement in colour preservation as opposed to Dubois (2001) method, though only a small improvement on McAllister et al.'s method which it was built upon. Though conversely, there was no significant improvement in visual quality of the image which measures the visual fatigue of the subject, against Dubois' and McAllister et al.'s method. On the other hand, this does suggest that the improvement in colour does not sacrifice visual quality any more than the other methods.

1.4 Project Objectives and Proposed Solution

Based on the review of current existing literature, I have adapted their concepts and ideas into my own solution for tackling ghosting in anaglyph 3D.

Given that perception of ghosting is proportional to the amount of parallax on the screen (Woods, 2010), I propose an anaglyph 3D renderer of which adjusts the focus plane in real time to whatever the user is looking at. Hypothetically this should yield an image where the object in focus has little parallax, which should yield less noticeable ghosting artifacts (Krupev and Popova, 2008), and therefore need less colour correction to alleviate.

I also propose including individual adjustments for Inter Pupillary Distance (IPD) and eye dominance, which hypothetically should make the viewing experience more comfortable for individual viewers by accommodating individual differences in their visual system.

Chapter 2
Methods

2.1 Requirements

To elaborate on my proposed solution, my final deliverable should have the following essential requirements:

- An interactive red-cyan anaglyph 3D rendering application that can render a variety of objects at once at an acceptable framerate to provide a sufficient base for testing.
- It must include methods to adjust the anaglyph image to account for individual Inter Pupillary Distance measurements.
- It must include methods to account for different eye dominances
- It must include a way to adjust individual colour channel intensities
- It must include a way to adjust the focus plane of the anaglyph image, additionally it must include an automatic focus plane adjustment option.

All of these requirements must be achieved to fufill my project aims and objectives and allow a full range of tests to be carried out.

Additionally, there are some non-essential requirements that are of a lower priority, which should be included if and only if all the essential requirements are achieved successfully:

- It can save to and load from individual adjustment configurations of users
- It can export the current rendered frame into an image file

These features are not integral to completing the objectives that I have set out in the previous section thus have been labelled as non-essential.

2.2 Development tools

2.2.1 3D Rendering Base

I decided to use an implementation of a basic 3D rendering program I created as part of the COMP3811 Computer Graphics module's coursework (Billeter, 2023), of which was part of a group collaboration between me and fellow coursemate Oliver Gill (sc21osg) as a base of which I built upon my proposed solution from. Code of which can be found in this github repository (https://github.com/0liverGill/Computer-Graphics-CWK2).

It is important to note that my solution does not rely on this renderer in any specific way, the renderer is just a way to present my solution features and my methods presented throughout this chapter can be applied to extend any other 3D renderer.

The reason for choosing this renderer is due to familiarity and to avoid professional and legal issues with utilising code that I did not create, with more details on this in the appendix.

2.2.2 Programming and Third Party Tools

As a result of the rendering program base used, coding for the main program was done in C++ with the OpenGL API which I have used version 4.6. In addition shaders were coded in OpenGL Shader Language (GLSL), this was chosen to maintain consistency and readability of the code as a whole. As a consequence I decided to develop the code on a Windows machine, with testing done on both Windows and Linux, as MacOS cannot run OpenGL at version 4.6 (Apple, 2018).

I decided to use Visual Studio as the Integrated Development Environment (IDE) and code editor for this program. I chose this due to its support for C++ and GLSL, integration with the libraries I chose to use and its built in version control integration with Git, which I used.

The program utilises the Premake Library to generate the makefiles nessecary for the program. This is because the main program uses several C++ programs and libraries. Premake automates the process for generating makefiles to create the executable program, thus providing consistent executables simplifying the development of this project.

Additionally, the program uses GLFW library to handle platform specific actions such as window creation and input handling, this is such that the program can be platform independent, and simplify the development process as I will not have to include platform specific code for each platform I intended for the program to be used on.

Furthermore, the program uses GLAD, which is a library to generate and OpenGL loader, which loads pointers to the OpenGL functions (Khronos, 2021). This allows the program to access the OpenGL functions up to version 4.6, which is what I am developing on. The program does not contain code to generate the loader, just the generated loader. The program also uses GLAD to access extensions to aid in debugging.

Also the program uses rapidobj for loading in the 3D models that I used in the rendered scene. The library allows loading of Wavefront objects (.obj files) which contain the information of the geometry of the object itself, and their corresponding Material Template Library files (.mtl files), which stores the colour, shading information (properties such as shininess etc.) and texture information. I decided to use .obj files as they are open source and rapidobj is designed to be very fast with large .obj files.

2.3 Solution Design

2.3.1 Anaglyph 3D

I have decided to use the Photoshop Algorithm to generate the anaglyph stereo pair for this project, as stated in the Introduction section, the photoshop algorithm for red-cyan anaglyph forms each eyes image by setting the corresponding colour channels to 0, green and blue for the left eye and red for the right eye. I have chosen this basic method as to make my program display and glasses independent, as the photoshop method does not need measurements of the display nor glasses.

Additionally, I have decided to use the photoshop algorithm as it inherently does not make any adjustments to alleviate ghosting. This is to ensure in testing that all the sources of ghosting alleviation can have a clear source.

I achieved this by doing two passes of rendering, one for each eye. As with the photoshop method, each eye's image is independent of each other. Thus it does not matter which eye is rendered first, as such I have chosen arbitrarily to render the left eye first, then the right eye.

In the main program, this was done by use of the glColorMask() function, which enables and disables which colour channels are written to by the shader programs. For the left eye, the green and blue channels are disabled and rendered, and after the left eyes image is drawn to the screen the right eye's red channel is disabled and drawn (Figure 6).

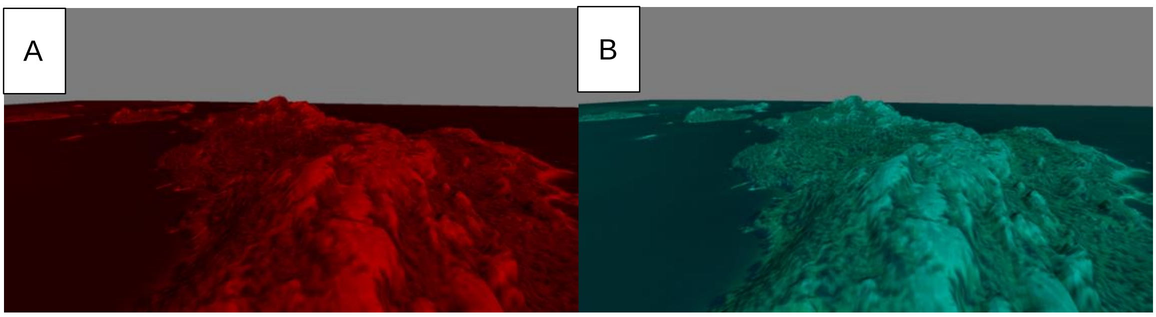

Figure 6 Images showing individually each image for the left eye (A) and the right eye (B)

2.3.2 IPD and Eyeball rotation

For adjustments to Inter-Pupillary Distance (IPD), the distance is factored into the rendering pipeline by having two different World-To-Camera matrices, one for each eye.

Theoretically it should translate each camera away from each other, before all other transformation matrices are applied, with the distance translated being half of the IPD measured for each eye in opposite directions (Figure 7).

The definition of the adjustment matrix T is given below, with d being half the IPD measurement in the positive or negative x direction (1).

$$T = \begin{pmatrix} 0 & 0 & 0 & d \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathrm{d} = \pm\frac{1}{2}\mathrm{IPD} \qquad (1)$$

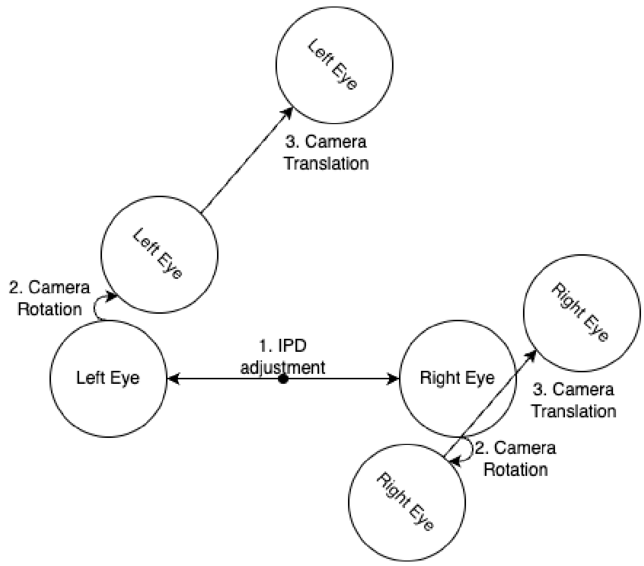

Figure 7 Diagram of the steps of translating each eye

To adjust the focus plane of the renderer, in addition to the World-To-Camera translations a rotation matrix is applied to each eye after translating to the cameras position, this allows for the rising and falling parallax effect to occur. Because of the way that the axes are aligned in OpenGL, the cameras are rotated along the Y-axis to achieve the correct effect (Figure 8).

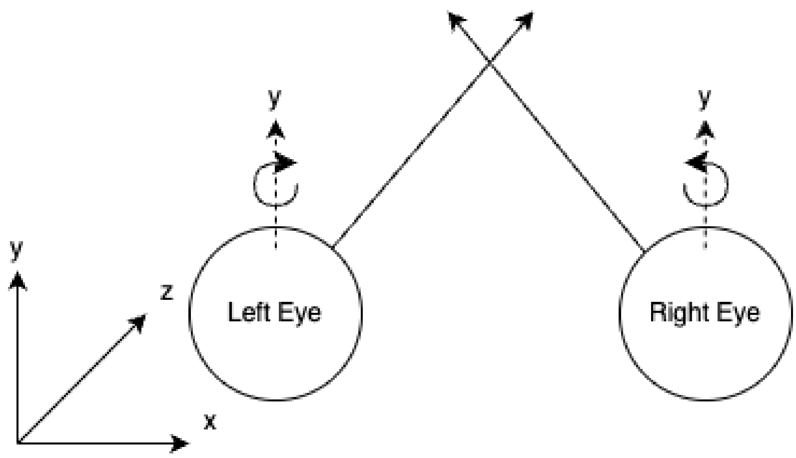

Figure 8 Diagram showing the axis of rotation of each camera

In the main program, I made two different World-To-Camera matrices separate for each eye, as opposed to making one World-To-Camera matrix for one eye, drawing the scene for that eye, then tranformimg it to the other eyes position and doing the same. This is due to later sections where I will make individual adjustments to each eye and need them to be separate.

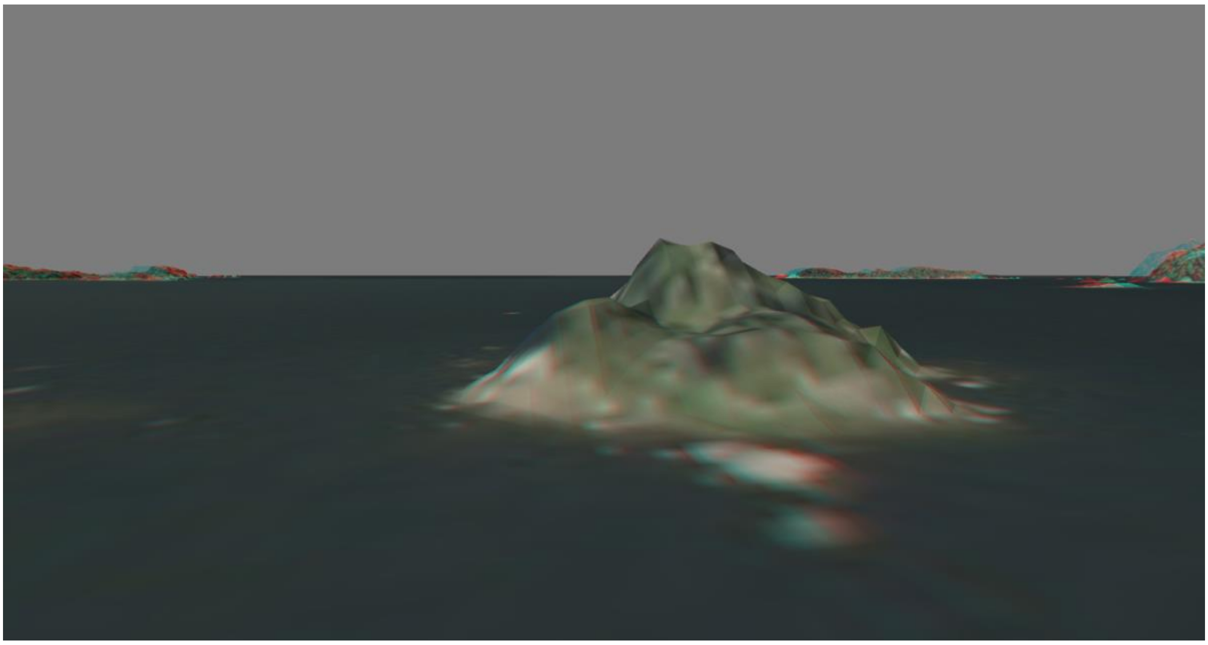

Figure 9 A screenshot of the renderer with IPD adjustments and focus plane adjustment

2.3.3 Automatic focus plane adjustment

Building on top of the previous section, I also implemented an autofocusing mode to the renderer. This is where the focus plane, and by extension the rotation of each eye's camera is automatically adjusted to what the user is looking at.

For finding the object of which the user is looking at, I chose to take the object of which is in the centre of the screen. This is because the 3D renderer is interactive and the user is able to move around freely inside the world. Inside the main program, I enforced this by adding a small circle shaped reticle in the middle of the screen, indicating that this is where the focus of the user should be.

To calculate the correct camera rotation to focus on the object, the distance from the camera to the object, and the IPD information is required. With this we can use equation (2) described graphically in Figure 10 to calculate the angle (in radians) passed to the rotation matrix R, which is described in equation (3).

$$Angle\ to\ Rotate\ Camera = \pi - \tan^{-1}\left(\frac{Distance\ to\ Object}{\frac{1}{2} \times IPD}\right) \qquad (2)$$

$$R = \begin{pmatrix} cos(\theta) & 0 & sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ sin(\theta) & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},\ \theta = \ Angle\ to\ Rotate\ Camera \qquad (3)$$

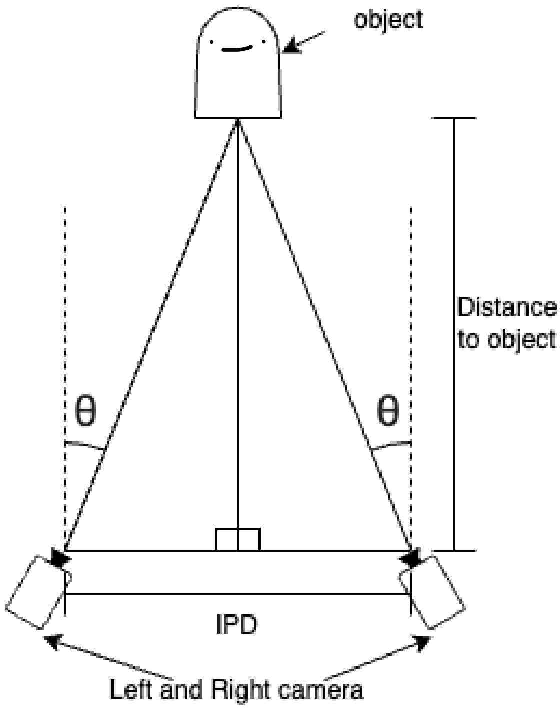

Figure 10 graphical representation of Equation (2)

IPD information is inputted and stored as described in the previous section. For distance from the camera, this information is actually already calculated by OpenGL as part of rendering in the variable gl_FragCoord, which stores the window relative coordinates of the pixel to be drawn to the screen. With the X and Y coordinates referring to the placement of the screen, and the Z coordinate being used to denote depth, or distance away from the camera.

This value is between 0 and 1, based on its position between the near and far plane, which is the distances from the camera of which the renderer will render geometry. And due to the property of the projection matrix, which allows objects nearer to the camera to look bigger than things further away (Santell, 2019), this value is not linear and instead is proportional to 1/z, where z is the actual depth of the object.

As such I needed to convert the non-linear depth value back to the original depth distance. This was achieved by applying equation (4) to first convert the value from the range [0, 1] to [-1, 1] to be align with the normalised device coordinate space, then reverses the calculations done by the projection matrix (de Vries, n.d.).

$$linearDepth = \frac{2 \times near \times far}{far + near - ndc \times (far - near))}, ndc = z \times 2 - 1 \quad (4)$$

Where z is the non-linear depth value, and near and far are the near and far plane distances

As the depth information is stored in the shader program, but the camera transformation matrices are calculated and applied in the main program, I have had to set up communication between the main and shader programs, this was achieved by using a Shade Storage Buffer Object (SSBO). This object is similar to the Uniform Buffer Objects (UBO) which is used to pass data along from the main program to the shader program. Differences include larger, variable storage, and the ability to read and write to the memory (Kronos, 2020), of which is relevant to my usage here as I need the data to be written to in the shader program and then read from back in the main program.

As calculations for the shader and by extension the memory of the SSBO is performed in the GPU, after calculations are finished and the scene is all calculated I needed to use the glMapBuffer() function to map the SSBO's data back to CPU accessible memory space. This is then used back in the main program to calculate the eyeball rotation as described above. As a consequence of this, the eyeball rotation will always be one frame behind of the actual distance, this is not a large issue as the depth change calculations are performed every frame, and the framerate of the program means any depth discrepancy only lasts for a fraction of a second, which is negligible and not noticeable.

Additionally, as OpenGL does not call shader programs if they are not needed I have had to add several checks to each of my shader programs (material, texture) to ensure the correct depth information is returned, of which is described in the flow chart in Figure 12. If the material or texture shader is not used, which means the camera is currently focused on anything, the depth is not updated thus not changed from an arbitrarily high number to simulate looking into the distance.

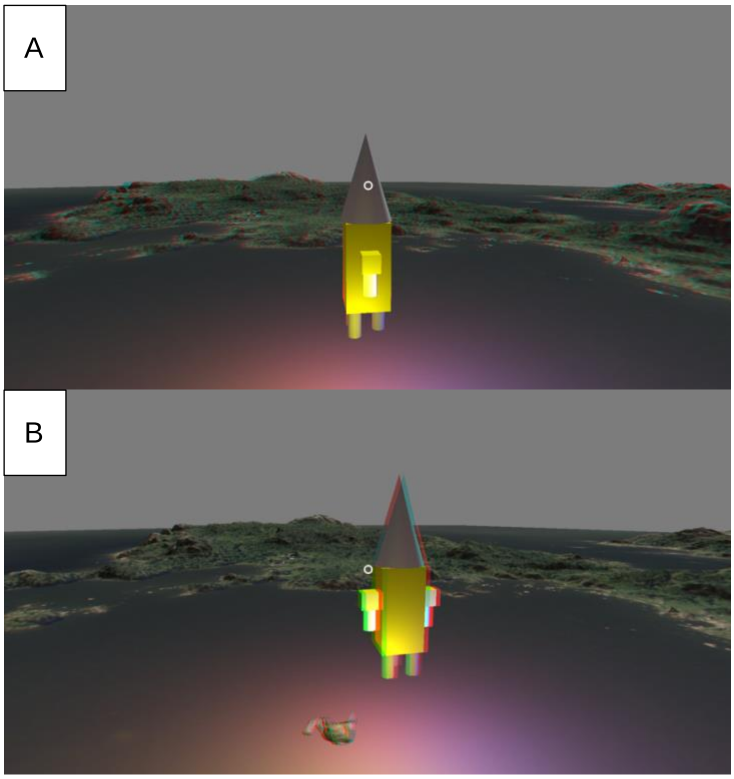

Figure 11 Two images showing focusing on the spaceship (A), and on the background (B)

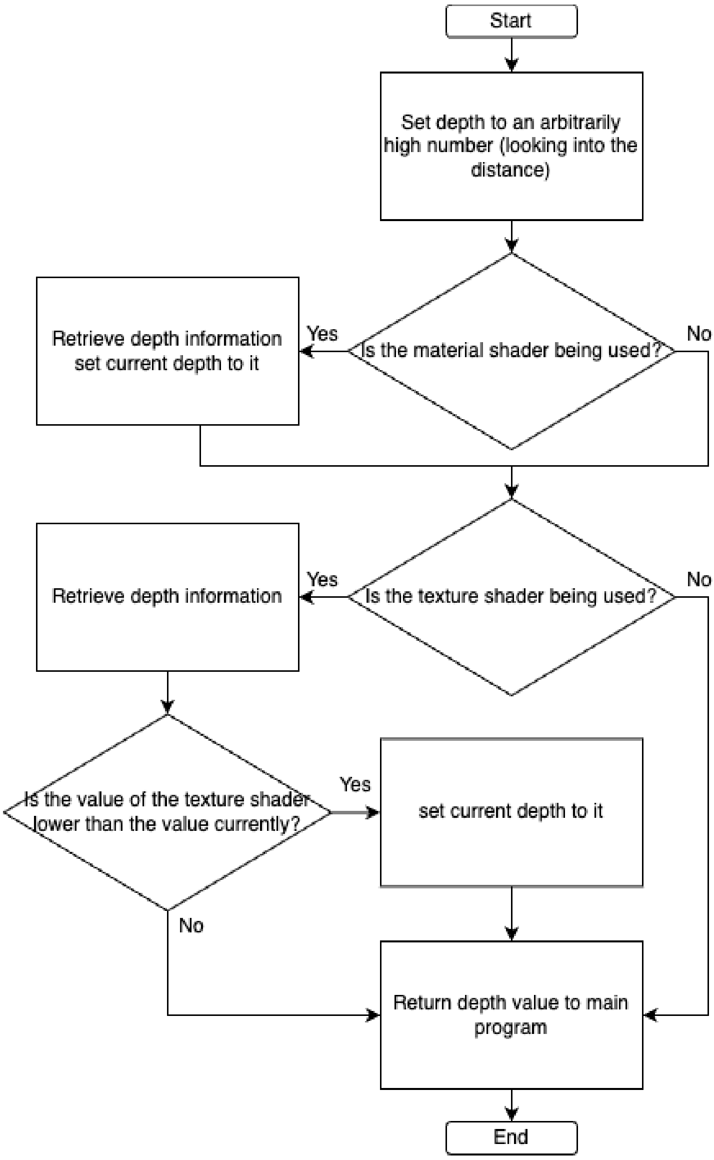

Figure 12 Flow chart showing the procedure of finding the correct depth from the shade programs

2.3.4 Eye dominance

To account for motor eye dominance, I change the way that the eyeball rotations are calculated in the main program. As opposed to moving both eyes towards the focus plane, one eye looks straight at an object whilst the other eye does the rotation (Figure 13). The equation used to calculate the angle of rotation also changes, as the rotation should account for the full IPD of both cameras (5).

$$Angle\ to\ Rotate\ Camera = \pi - \tan^{-1}\left(\frac{Distance\ to\ Object}{IPD}\right) \qquad (5)$$

In OpenGL, due to the orientation of the axes, it is important to note that the left eye is rotated by a negative angle and the right eye is rotated by a positive angle, due to rotation matrixes rotating anti-clockwise.

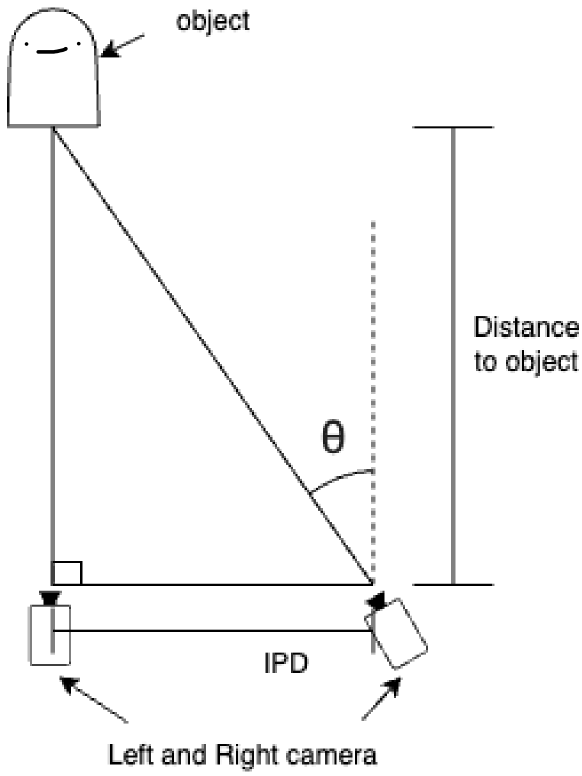

Figure 13 Graphical representation of equation (4) taking eye dominance into account (left eye dominant)

2.3.6 Colour adjustment

To adjust the individual colour channels, I decided to use a similar method as described in Krupev and Popova's method (2008). Alhough I chose not to use thresholds and just use the gamma correction. This like choosing not to apply an already existing pre-processing ghosting correction method such as Dubois' Method (2001) to allow objective and independent metrics for evaluating my method.

In the main program, the variables for storing the amount of gamma adjustment for each colour channel (red, green and blue) are stored in the state structure which keeps track of the current state of the scene; it is stored as a Vec3 (vector of 3 floats) which is passed through to the shader program as a Uniform Buffer Object (UBO). The vector is then used in the gamma correction formula as described in equation (6), where O is the output colour as a column vector containing red, green and blue in that order, I is in input colour formatted in the same way as the output colour, and G is the Gamma correction vector.

$$(O_r, O_g, O_b) = (I_r{}^{G_r}, I_g{}^{G_g}, I_b{}^{G_b}) \qquad (6).$$

2.5 User Interaction and Interface

2.5.1 Camera Controls

For controlling the camera, I have decided to stay with the first-person perspective 3D camera of the base rendering program, although I changed some numbers for speed. This means that the translation matrix for camera position is controlled with keyboard input and the rotational matrices for the camera orientation is controlled by mouse input.

This means that for each object, the equations (7) and (8) used in calculating its depth and location in the final render is as such:

$$WorldToCamera = IPD \times CameraRotation \times FocusPlaneRotation \times CameraTranslation \qquad (7)$$

$$ProjectedCameraWorld = ProjectionMatrix \times WorldToCamera \times ModelToWorld \qquad (8)$$

Where IPD is the interpupillary Distance measurement, FocusPlaneRotation is the focus plane rotation amount from either the manual or autofocusing mode, CameraRotation and CameraTranslation is the camera transformation from the input devices, ModelToWorld is the transformation matrix of the object within the world, and ProjectionMatrix is the projection matrix defined as follows (9):

$$\text{ProjectionMatrix} = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{pmatrix} \qquad (9)$$

$$S = \frac{1}{\tan(\frac{FOV}{2})}, S_x = \frac{S}{Aspect}, \; S_y = S, a = \frac{-(Far + Near)}{Far - Near}, b = \frac{-2 \times (Far \times Near)}{Far - Near}$$

Where Aspect is the aspect ratio of the Display, and Far and Near are the distances of the Far and Near Plane (Billeter, 2023).

The reasons for choosing this method of camera control is to allow the user to have full control of the camera and angle of viewing of the scene. This is to ensure that my collected data will be of the whole scene in testing and for completeness of data. This method also ensures that data is not skewed, for example by limiting the user to only look at a particularly good view and angle of the scene.

2.5.3 Anaglyph Adjustments and Displaying Data

For adusting the anaglyph image, such as for IPD as mentioned before, I chose to have these be inputted through keyboard inputs, and this includes keys for increasing and decreasing the gamma of the rendered image in all colour channels, keys for increasing and decreasing the IPD and the focus plane. Additionally keys to change the dominant eye and whether autofocusing is on or not. Furthermore I include keys to reset the IPD and focus plane adjustments and colour adjustments to an default value.

A total list of the controls are listed in this table:

Key Binding	Description
W, A, S, D, Q, E	Moves the camera forward, backwards, left, right, up and down respectively
Shift, Control	Speeds up and slows down the camera respectively
U, I, O	Increases the gamma value of the Red, Green and Blue channels respectively (Makes it darker)
J, K, L	Decreases the gamma value of the Red, Green and Blue channels respectively (Makes it brighter)
N, M	Increases and Decreases IPD respectively
T, Y	Increases and Decreases the amount of eyeball rotation (adjusting focus plane) respectively
H, G	Resets the colour gamma adjustment to 1 (no change), and resets the IPD and focus plane adjustment to a default value (IPD = 0.063, rotation = 0.04)
C, V, B	Sets the renderer to No eye dominance (C), Left eye dominance (V) and right eye dominance (B) focusing mode.
Z, X	Turns autofocusing off (Z) and on (X)

Table 1 Table of controls and decriptions

For displaying such data about the state of the renderer, I have decided that besides features that are specific to the visual experience itself, such as the reticle showing where the user is looking at, I have decided to not include GUI elements for data such as buttons or sliders on the window itself. Instead I have the data displayed on a command line window, displaying text on the current state of the renderer.

In the main program, this was achieved by having print statements after each of the function which prints directly to the command line.

The reason for not having major GUI elements is that I do not want any unnecessary UI elements to distract from the visual experience. This is to allow better data collection in testing as I do not want the potential retinal rivalry caused by having bright buttons to disturb the testing process.

2.6 Project Management and Coordination

This project contains a lot of features that are dependent on other features, for example the eye dominance adjustments requiring the focus plane adjustment first, and the autofocusing feature requiring the eye dominance feature to be implemented first. Due to this I have decided to take a waterfall approach to developing this project, also a factor is that this project is not geared towards having multiple iterations and deliverables, and is more geared towards having one final complete project.

As a result, testing for the project will be at the end and thus on a negative impact this had, user feedback will also not be available throughout the project. Additionally a risk is that if retroactively an issue is found that happened in an earlier stage, I will have to start again from that stage. To mitigate these issues I have made an initial plan in the form of a Gantt chart that I have stuck to for this project (Figure 14)

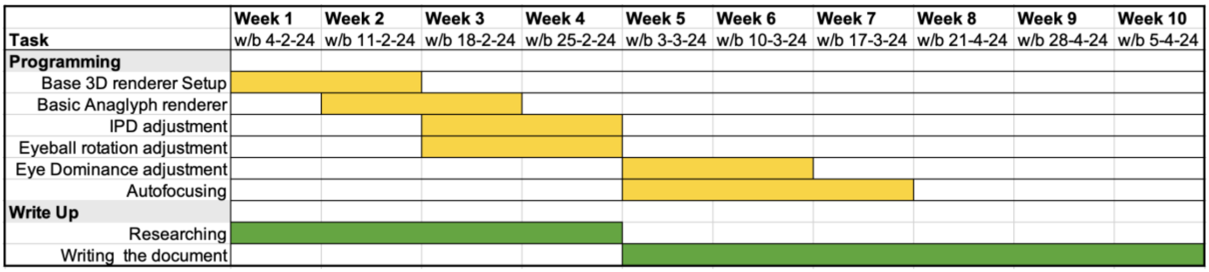

Figure 14 Gantt chart of my plan for this project

I have also used GitHub for version control of my project, due to the waterfall methodology described before, I have not used branches and have worked on the main branch for my project.

Using GitHub for version control was to ensure that any machine I used to run my program, like on a Linux machine or a Windows Machine, would be up to date on my code. Additionally, I created a fork of the base renderer I built the code upon, this is to ensure I had a snapshot of the code I built upon in case I required it for any reason.

Chapter 3
Results

3.1 Testing Environment Design

For the environment that I will be testing the renderer on, I have divided this section into subsections to describe the reasons for including each of the elements in testing.

3.1.1 Bright cubes

To get a baseline for the amount of gamma correction needed without any help from my solution I have decided to add 3 white cubes in a line into the scene. The cubes are not textured and simply use per material colours, of which they all are purely white, as shown in Figure 15.

The reason for choosing this arrangement and colour is so that it is the brightest object in the scene, thus theoretically should need the most gamma correction to fix, additionally the reason for having 3 cubes in a line is to allow the user to experience and adjust for positive and negative parallax.

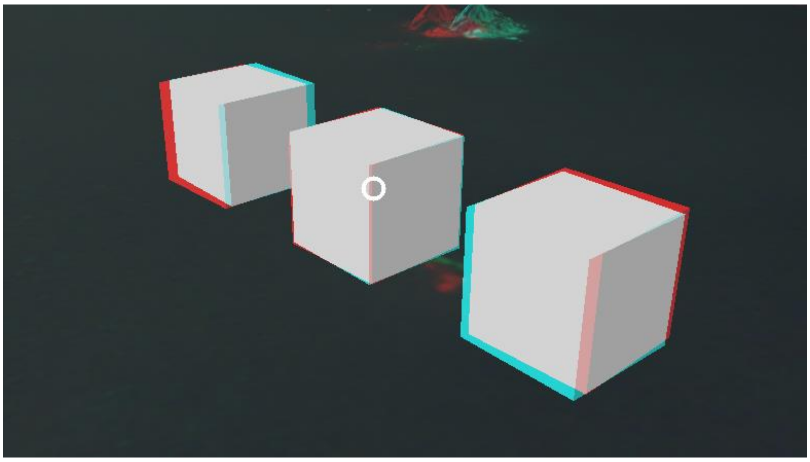

Figure 15 3 cubes within the scene with the partlahti ground in the background

3.1.2 Spaceship

I have decided to keep the spaceship (Figure 16) from the base renderer as it does have a few interesting properties. It contains red, yellow and blue point lights which are reflected on the ships surface and give a gradient of brightness to the object as shown in Figure 16. Additionally, the spaceship does have some animations and motion and thus I also included the animations in to see the effect it has on ghosting.

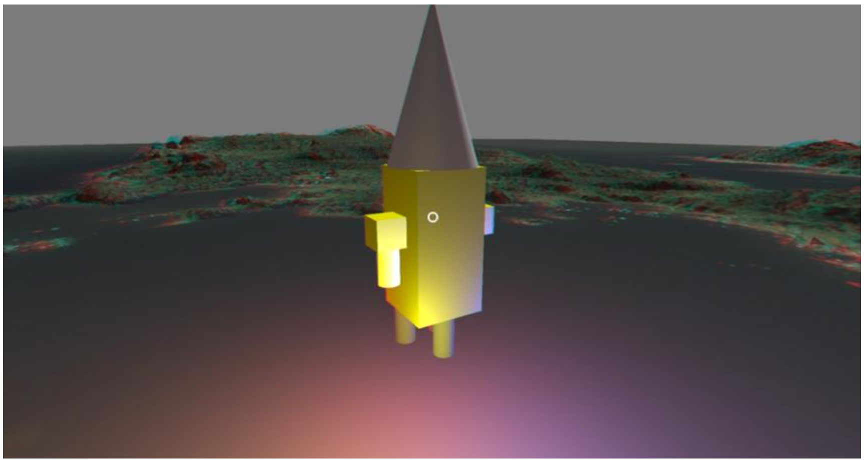

Figure 16 Rendered image of the spaceship in the scene

3.1.3 Spiderman

I have also decided to include a model of Spiderman (Sketchfab, 2024) (Figure 17) in the testing scene. One reason for choosing this is that I wanted to include a humanoid object in the testing. In addition, as the colour palette of the character is mainly red and blue, this means that there is going to be high retinal rivalry on the character, and thus could show an effect on perceived ghosting.

Figure 17 Rendered image of the Spiderman in the scene

3.1.4 3D Scenery

Furthermore, I decided to include the background scenery of the Parlahti Islands (National Land Survery of Finland, n.d) already present in the base renderer (Figure 18), this is due to it spanning over a large area thus can be both in the foreground and background of a rendered scene. Additionally it gives ample room for users to travel across and experience the changes in parallax.

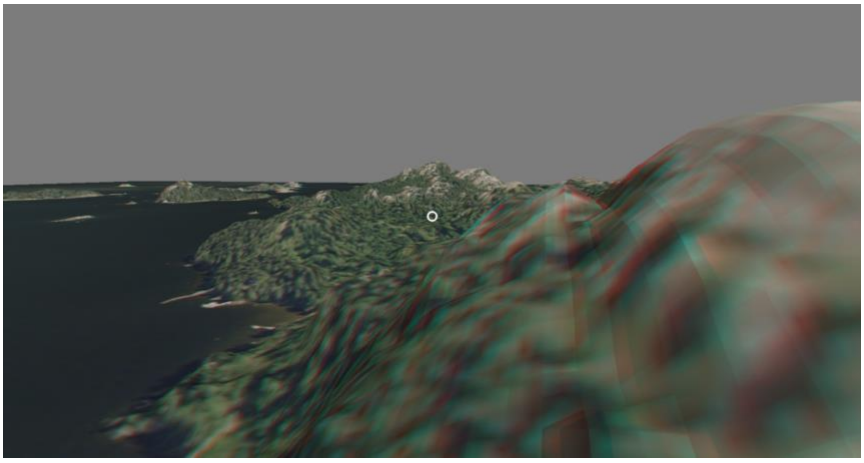

Figure 18 Rendered image of the Parlahti islands in the scene

3.2 Testing Procedure

These testing procedures are approved by the School of Computing block ethical approval. Prior to testing, participants were informed of the procedures and provided written consent with a form which can be found in the Appendix C. I also informed participants that they could withdraw their participation at any time with no negative consequences. As my testing will include changing brightness frequently which can trigger photosensitive epilepsy (Epilepsy Action, n.d.), and that anaglyph 3D in general can cause discomfort in people (Shibata et al., 2011), I have decided to exclude people with photosensitive epilepsy and people who are sensitive to 3D environments.

Once the participant consented to the testing, I first collected the participants dominant eye by use of the ‘hole in card’ test as described by Li et al. (2010). I also collected the participants dominant eye through a method described by Vive (n.d.).

For the conditions of the test, I had the participant in a well-lit room sat on a desk in front of a monitor with the renderer running. This is to simulate the average viewing conditions of a user.

Firstly, I set the program to have a fixed focus plane, no dominant eye adjustments, and an average IPD of 63mm (Murray et al., 2017). I explained the controls of the program and what ghosting was to the user. Then on the bright cubes object as described before I allowed the participant to have free control of the camera and adjust the gamma values of each colour channel until ghosting was negligible to them. I recorded the gamma values and reset the scene then repeated this with the custom IPD measurement, dominant eye adjustment, and the dynamic auto-focusing turned on, on the bright cubes, spaceship, spiderman and 3D scenery objects as described above.

3.3 Results

Note in this section numbers will be rounded to 2 decimal places, additionally a gamma value of 1 indicates no change as it is the default value, and values >1 indicate decreased brightness and vice versa for values <1.

For gamma adjustment before any auto-focusing and personal adjustments, results indicate a large amount of gamma correction is required to alleviate ghosting. As shown in Figure 20, for all channels an average of over a gamma value of 2 is needed for all channels is needed. An interesting thing to note is that the results suggest that the red channel require less gamma correction with an average gamma value of 2.39, compared to the green and blue channels with an average gamma value of 3.39 and 3.36.

After auto-focusing and personal adjustments are enabled, results indicate a significant decrease in gamma correction required to alleviate ghosting. With gamma values for green and blue decreasing to an average value of 1.82 and 1.97 and the red channels average gamma value decreasing to 1.14 with a smaller Inter Quartile Range. Additionally 5 out of 9 of the participants required no red channel gamma correction at all to correct gamma.

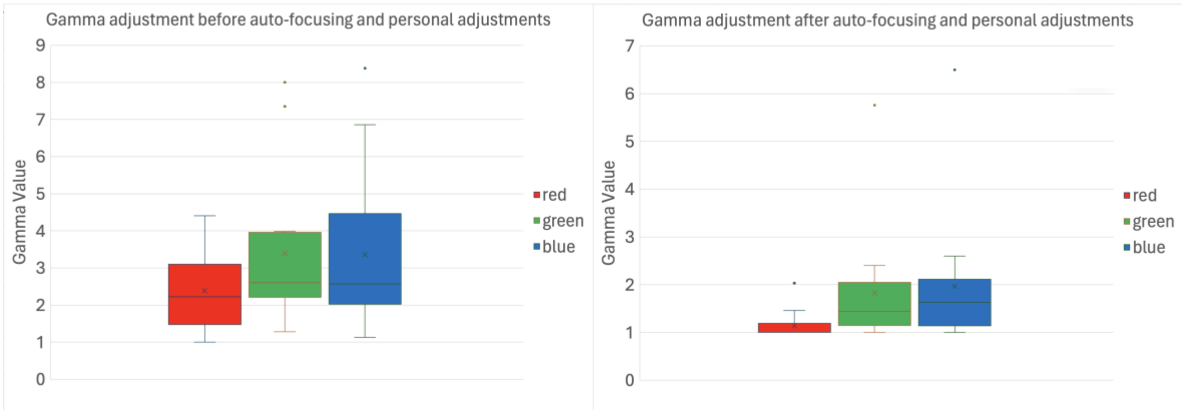

Figure 20 Box and Whisker plots of the gamma values needed to alleviate ghosting before and after autofocusing and individual adjustments

For the other testing elements, results indicate a decrease in gamma correction needed to alleviate ghosting, as seen in Figure 21 for both the spaceship and spiderman object almost no gamma correction was needed. Additionally 8 out of 9 participants did not require any gamma correction for any colour channel at all.

Additionally, through the use of the 3D background, all participants confirmed that they perceived depth from viewing the rendered scene after the autofocusing and individual adjustments have been applied.

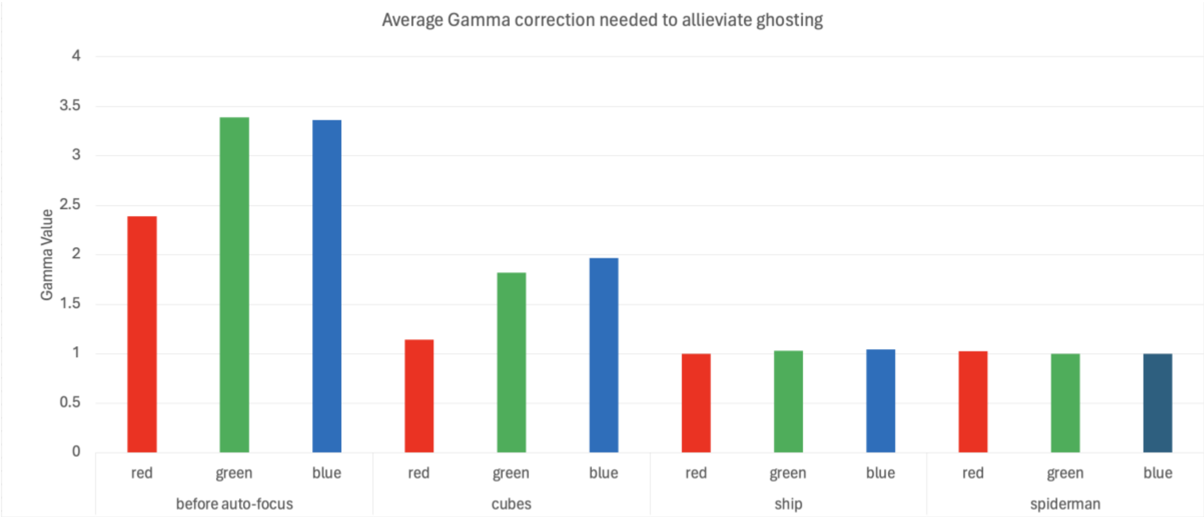

Figure 21 Bar chart showing the average gamma correction needed to alleviate ghosting for each testing element

An interesting thing to note is the decreased ghosting showed in the right eye, meaning red light leaking through the cyan filter, as compared to ghosting in the left eye, meaning green and blue light leaking though the red filter. This could be due to reasons out of the scope of this project such as the properties of the display, or properties of the filter glasses.

For the Spiderman test, it does suggest that retinal rivalry does not contribute negatively to ghosting. In addition, the ship object having low ghosting also suggests that motion and gradients in brightness also do not contribute negatively to ghosting.

Additionally all participants noted that even with the adjustments that there was still a 3D effect in the image, and that the depth information in the image was not sacrificed nor decreased.

Chapter 4
Discussion

4.1 Limitations

There are some limitations of this project due to the choices made during development and planning. This project only tackles Red-Cyan Anaglyph 3D, and thus has not taken into account other types of Anaglyph which may have other methods of tackling ghosting.

Additionally the usage of the Shader Storage Buffer Objects does increase the computational load of the program as it is reading and writing memory to and from the GPU, while in debug mode, OpenGL does give video performance warnings for using the Shader Storage Buffer Object in this way.

Furthermore there were only 9 participants in the user testing phase of this project. Although it was enough data to show some trends, even more participant data would give stronger results.

4.2 Conclusions

All of the aims and necessary requirements for this project have been achieved. I have conducted extensive research of the literature regarding Anaglyph 3D and the issues surrounding the method of showing a stereo pair, and the current methods that tackle these issues. This resulted in a proposed solution which I described the design solution decisions taken and the justifications for each decision taken.

User testing of the final solution was taken which showed an improvement in the perceived ghosting, with some cases resulting in no colour correction needed to alleviate ghosting.

4.3 Ideas for future work

One idea to extend this solution could be to include support for different Anaglyph 3D methods, such as Amber-Blue anaglyph. An additional extension for this project could be to use a eye-tracking module to dynamically change the focus point instead of having it fixed in the middle of the screen and changing focus by moving the camera. Additionally we could go even further by combining this solution with existing pre-processing methods such as Dubois' Method (2001) for potentially better combined effect.

List of References

Reeve, S. and Flock, J. (2010). Stereoscopic 3D. [online] Available at: https://www.ncl.ac.uk/media/wwwnclacuk/pressoffice/files/pressreleaseslegacy/Basic_Principles_of_Stereoscopic_3D_v1.pdf.

Woods, A.J. 2010. Understanding Crosstalk in Stereoscopic Displays. In: 3DSA (Three-Dimensional Systems and Applications conference, Tokyo, Japan, 19-21 May 2010.

Billeter, M. 2023. Coursework 2. [code zip file]. COMP3811 Computer Graphics. School of Computing, University of Leeds.

Abileah, A. 2011. 3-D displays — Technologies and testing methods. *Journal of the Society for Information Display*. **19**(11), pp.749–763.

Anon 2024. *Spider-Man TASM 1 (Andrew Garfield) - Download Free 3D model by tsaphnatmbuyi123* [Online]. [Accessed 1 May 2024]. Available from: https://sketchfab.com/models/77ad207cff4946869c37e0c942bb28f6/embed?autostart=1.

Apple 2018. About OpenGL for OS X. [Accessed 22 April 2024]. Available from: https://developer.apple.com/library/archive/documentation/GraphicsImaging/Conceptual/OpenGL-MacProgGuide/opengl_intro/opengl_intro.html.

Blake, R. and Logothetis, N.K. 2002. Visual competition. *Nature Reviews Neuroscience*. **3**(1), pp.13–21.

Chitu, A. 2009. YouTube 3D. *YouTube 3D*. [Online]. [Accessed 17 April 2024]. Available from: https://googlesystem.blogspot.com/2009/07/youtube-3d.html.

Dubois, E. 2001. A projection method to generate anaglyph stereo images *In*: *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)* [Online]., pp.1661–1664 vol.3. [Accessed 19 April 2024]. Available from: https://ieeexplore.ieee.org/document/941256.

Epilepsy Action n.d. Photosensitive epilepsy. *Epilepsy Action*. [Online]. [Accessed 1 May 2024]. Available from: https://www.epilepsy.org.uk/info/seizure-triggers/photosensitive-epilepsy.

Gupta, S.K. and Gupta, P. 2021. Anaglyph stereo virtual dissection: a novel inexpensive method for stereoscopic visualisation of intracardiac anatomy on CT angiogram. *Cardiology in the Young*. **31**(12), pp.1958–1961.

Hurlbert, A. 1996. Colour vision: Putting it in context. *Current Biology*. **6**(11), pp.1381–1384.

Kellnhofer, P., Didyk, P., Ritschel, T., Masia, B., Myszkowski, K. and Seidel, H.-P. 2016. Motion parallax in stereo 3D: model and applications. *ACM Transactions on Graphics*. **35**(6), 176:1-176:12.

Khronos 2021. OpenGL Loading Library - OpenGL Wiki. [Accessed 22 April 2024]. Available from: https://www.khronos.org/opengl/wiki/OpenGL_Loading_Library.

Kronos 2020. Shader Storage Buffer Object - OpenGL Wiki. [Accessed 25 April 2024]. Available from: https://www.khronos.org/opengl/wiki/Shader_Storage_Buffer_Object.

Krupev, A.A. and Popova, A.A. 2008. Ghosting Reduction and Estimation in Anaglyph Stereoscopic Images *In*: *2008 IEEE International Symposium on Signal Processing and Information Technology* [Online]., pp.375–379. [Accessed 17 April 2024]. Available from: https://ieeexplore.ieee.org/document/4775719.

Li, J., Lam, C.S.Y., Yu, M., Hess, R.F., Chan, L.Y.L., Maehara, G., Woo, G.C. and Thompson, B. 2010. Quantifying Sensory Eye Dominance in the Normal Visual System: A New Technique and Insights into Variation across Traditional Tests. *Investigative Opthalmology & Visual Science*. **51**(12), p.6875.

Li, S., Ma, L. and Ngi Ngan, K. 2013. Anaglyph image generation by matching color appearance attributes. *Signal Processing: Image Communication*. **28**(6), pp.597–607.

Lopes-Ferreira, D., Neves, H., Queiros, A., Faria-Ribeiro, M., Peixoto-de-Matos, S.C. and González-Méijome, J.M. 2013. Ocular Dominance and Visual Function Testing. *BioMed Research International*. **2013**, p.238943.

McAllister, D.F., Zhou, Y. and Sullivan, S. 2010. Methods for computing color anaglyphs *In*: A. J. Woods, N. S. Holliman and N. A. Dodgson, eds. San Jose, California, p.75240S. [Accessed 19 April 2024]. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.837163.

Murray, N.P., Hunfalvay, M. and Bolte, T. 2017. The Reliability, Validity, and Normative Data of Interpupillary Distance and Pupil Diameter Using Eye-Tracking Technology. *Translational Vision Science & Technology*. **6**(4), p.2.

Ochotorena, C.A., Ochotorena, C.N. and Sybingco, E. 2013. Designing anaglyphs with minimal ghosting and retinal rivalry *In*: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* [Online]., pp.2035–2039. [Accessed 17 April 2024]. Available from: https://ieeexplore.ieee.org/abstract/document/6638011.

Ooi, T.L. and He, Z.J. 2020. Sensory Eye Dominance: Relationship Between Eye and Brain. *Eye and Brain*. **12**, pp.25–31.

Qi, M., Cui, S., Du, Q., Xu, Y. and McAllister, D.F. 2021. Visual Fatigue Alleviating in Stereo Imaging of Anaglyphs by Reducing Retinal Rivalry and Color Distortion Based on Mobile Virtual Reality Technology R. Mascella, ed. *Wireless Communications and Mobile Computing*. **2021**, pp.1–10.

Qian, N. 1997. Binocular Disparity and the Perception of Depth. *Neuron*. **18**(3), pp.359–368.

Rojas, G.M., GÃ¡lvez, M., Vega Potler, N., Craddock, R.C., Margulies, D.S., Castellanos, F.X. and Milham, M.P. 2014. Stereoscopic three-dimensional visualization applied to multimodal brain images: clinical applications and a functional connectivity atlas. *Frontiers in Neuroscience*. **8**.

Sanders, W.R. and McAllister, D.F. 2003. Producing anaglyphs from synthetic images *In*: A. J. Woods, M. T. Bolas, J. O. Merritt and S. A. Benton, eds. Santa Clara, CA, p.348. [Accessed 19 April 2024]. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.474130.

Sanftmann, H. and Weiskopf, D. 2011. Anaglyph Stereo Without Ghosting. *Computer Graphics Forum*. **30**(4), pp.1251–1259.

Santell, J. 2019. 3D Projection. *jsantell.com*. [Online]. [Accessed 25 April 2024]. Available from: https://jsantell.com/3d-projection/.

Schmitz, J. 2015. Dubois Anaglyphs. *ixora.io*. [Online]. [Accessed 19 April 2024]. Available from: https://ixora.io/projects/camera-3D/dubois-anaglyphs/.

Shibata, T., Kim, J., Hoffman, D.M. and Banks, M.S. 2011. The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of Vision*. **11**(8), pp.11–11.

Tkalcic, M. and Tasic, J.F. 2003. Colour spaces: perceptual, historical and applicational background *In*: *The IEEE Region 8 EUROCON 2003. Computer as a Tool.* [Online]., pp.304–308 vol.1. [Accessed 20 April 2024]. Available from: https://ieeexplore.ieee.org/abstract/document/1248032.

Tozawa, J. and Oyama, T. 2006. Effects of Motion Parallax and Perspective Cues on Perceived Size and Distance. *Perception*. **35**(8), pp.1007–1023.

Ukai, K. and Howarth, P.A. 2008. Visual fatigue caused by viewing stereoscopic motion images: Background, theories, and observations. *Displays*. **29**(2), pp.106–116.

Vive n.d. How can I find my IPD? [Accessed 1 May 2024]. Available from: https://www.vive.com/uk/support/vive-xr/category_howto/how-can-i-find-my-ipd.html.

de Vries, J. n.d. LearnOpenGL - Depth testing. [Accessed 25 April 2024]. Available from: https://learnopengl.com/Advanced-OpenGL/Depth-testing.

Weissman, M.A. and Woods, A.J. 2011. A simple method for measuring crosstalk in stereoscopic displays *In*: A. J. Woods, N. S. Holliman and N. A. Dodgson, eds. San Francisco Airport, California, USA, p.786310. [Accessed 19 April 2024]. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.877021.

Woldegiorgis, B.H., Lin, C.J. and Liang, W.-Z. 2019. Impact of parallax and interpupillary distance on size judgment performances of virtual objects in stereoscopic displays. *Ergonomics*. **62**(1), pp.76–87.

Woods, A.J. 2012. Crosstalk in stereoscopic displays: a review. *Journal of Electronic Imaging*. **21**(4), p.040902.

Woods, A.J. 2011. How are crosstalk and ghosting defined in the stereoscopic literature? *In*: A. J. Woods, N. S. Holliman and N. A. Dodgson, eds. San Francisco Airport, California, USA, p.78630Z. [Accessed 19 April 2024]. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.877045.

Woods, A.J. and Harris, C.R. 2010. Comparing levels of crosstalk with red/cyan, blue/yellow, and green/magenta anaglyph 3D glasses *In*: A. J. Woods, N. S. Holliman and N. A. Dodgson, eds. San Jose, California, p.75240Q. [Accessed 17 April 2024]. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.840835.

Woods, A.J. and Harris, C.R. 2012. Using cross-talk simulation to predict the performance of anaglyph 3-D glasses. *Journal of the Society for Information Display*. **20**(6), pp.304–315.

Woods, A.J. and Rourke, T. 2004. Ghosting in anaglyphic stereoscopic images *In*: A. J. Woods, J. O. Merritt, S. A. Benton and M. T. Bolas, eds. San Jose, CA, pp.354–365. [Accessed 19 April 2024]. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=836731.

Woods, A.J., Yuen, K.L. and Karvinen, K.S. 2007. Characterizing crosstalk in anaglyphic stereoscopic images on LCD monitors and plasma displays. *Journal of the Society for Information Display*. **15**(11), pp.889–898.

Appendix A
Self-appraisal

A.1 Critical self-evaluation

I would consider this project an overall success as I have completed all my initial goals for the solution and have met all my necessary requirements. Additionally, user testing indicates that my hypothesis that I built my proposed solution upon was not completely false. I attribute most of this success to my planning and research for this project.

One of the issues with this project is also due to my planning, unfortunately I have not completed any of my optional requirements, although they have no bearing on the final program and are there for quality of life improvements I would have wanted to see them achieved rather than not. This can be attributed to my planning as I only gave myself 2 weeks at the end for any optional features. This was coupled with the fact that during this time I would also be in the deep of writing the report, and as a result I simply did not have the time to complete these optional features.

Additionally I think that instead of allocating a time for research, then a time for writing the report, doing both at the same time throughout the project would be a much more efficient way of writing the project report, as I did see myself mentally writing up the report when reading and analysing literature, then struggling to remember what my original thought was when it came to writing the report.

A.2 Personal reflection and lessons learned

This project has been an immensely challenging and rewarding experience, testing my knowledge of concepts learn recently and even as far back as my first year in university. Additionally, through this project I have learnt a great deal of knowledge of C++ programming, the OpenGL API, GPU Hardware and dealing with cross platform challenges.

Furthermore, I have learnt and developed my skills in report writing and researching, and how to compare and critically evaluate sources. This project also enabled me to build on my inter-personal skills with the user testing and interacting with my supervisor and assessor.

A.3 Legal, social, ethical and professional issues

A.3.1 Legal issues

I have chosen to use 3D models of which are not created by me, as such I have chosen models to use which are under creative commons license. I have ensured that my usages of the 3D models in this project do not breach the license and is legal to use.

A.3.2 Social issues

An issue socially with this project is that as stated before, it is not advisable to use Anaglyph 3D on someone who has photosensitive epilepsy or sensitivity to 3D environments. This presents an issue with accessibility as with this certain groups of people are excluded from a technology for simply who they are. To tackle this, full information on what groups are excluded and why are described in the information sheet, and participants who are part of these groups will not be allowed to participate in testing.

A.3.3 Ethical issues

As I have conducted user testing, this was an ethical issue due to the fact I was collecting data from the participants. Of which collection of the eye-dominance and IPD data could be more of an issue as these metrics are specifically from an individual participant. Additionally, I have had to exclude certain groups of people from participating, such as people suffering from photosensitive epilepsy. As such I have endeavoured to anonymise all collected data and have given extensive information about the excluded categories and testing information in the information and consent forms, of which can be found in Appendix C.

A.3.4 Professional issues

I have completed this project in accordance with the BCS Code of Conduct at all stages of development to ensure the integrity of the project. All third party code has been properly cited and referenced in the repository, and all the code specifically for this project was written by me as a sole developer. Furthermore I have aimed to write this report in full honesty and sincerity.

I will reiterate what I have already said in **Section 2.2.1** in that the base code used for this project is simply a way to present the solution of this project, and the solution can be applied to any compatible rendering program. I have chosen to use a base program co-written by me for familiarity and to avoid using code not written by me for professional issues.

Appendix B
External Materials

External Code/ Libraries used in project

- The base renderer used is used from a previous group project under the COMP3811 Computer Graphics module, usage of which has been described in **Section 2.2.1 .** and can be found at https://github.com/0liverGill/Computer-Graphics-CWK2 .
- External Libraries used in the project include
    - Premake
    - GLFW
    - GLAD
    - STB Libraries
    - Rapidobj
    - Catch2
    - Fonstash

    Full referencing and licenses are within the code repository under [Prototype/third_party.md]

External 3D models used in project

Parlahti model (parlahti.{obj,mtl}; L4343A-4k.jpeg):

National Land Survery of Finland, n.d. Karttapaikka - Maanmittauslaitos. [Accessed 22 April 2024]. Available from: https://asiointi.maanmittauslaitos.fi/karttapaikka/.

The datasets are subject to CC-BY 4.0.

Landingpad (landingpad.{obj,mtl}):

Chalmers Computer Graphics Research Group, n.d. Research Group Graphics. [Accessed 22 April 2024]. Available from: https://www.cse.chalmers.se/~uffe/ComputerGraphicsGroup.htm.

This model is licensed under the CC0 Creative Commons Public Domain

The model is modified: it is rescaled by 0.01, removes the "tvCard" texture and sets the ambient colour to the diffuse one.

Spiderman (spiderman.{obj,mtl}):

Sketchfab, 2024. Spider-Man TASM 1 (Andrew Garfield) - Download Free 3D model by tsaphnatmbuyi123 [Online]. [Accessed 1 May 2024]. Available from: https://sketchfab.com/models/77ad207cff4946869c37e0c942bb28f6/embed?autostart=1.

Appendix C

User Testing Information and Consent Form

Information Sheet: COMP3931 Individual Project

Project Title: Alleviating Anaglyph Stereo Ghosting with Real Time Focus Plane and Individual Adjustments

Student: Zachariah Woon Yin Lee
Contact Email: ed20zwyl@leeds.ac.uk

I am a student at the University of Leeds, interested in how to improve how red-cyan 3D looks. If you volunteer for the experiment, you will perform a series of trials. Each trial will involve interacting in an environment in 3D and adjusting various parameters to improve visual quality. Afterwards there is a short survey to fill out regarding the experience.

IMPORTANT: Anaglyph 3D can cause nausea and visual discomfort, thus if you have **Photosensitive Epilepsy** and/or **Sensitivity to 3D Nausea/Disorientation** you **CANNOT PARTICIPATE**, by participating you confirm that you **DO NOT** have these conditions.

The experiment will take approximately 10 minutes. Your responses will be recorded, but you will remain anonymous. The research will be reported in my writeup, but your data will be processed and anonymised and at no point will your identity be disclosed.

Data being recorded specifically from you will be Inter-Pupilary Distance (IPD) and Eye-Dominance, in addition information on existing visual conditions such as colour blindness will be recorded, again this data will be anonymised and will be kept confidential.

By taking part in the experiment, you are indicating informed consent. You are free to withdraw from the experiment with no negative consequences at any time through contacting the email above.

Finally, please let me know if you have any questions or would like to discuss anything with me.

Please sign below to indicate you've read and understood this sheet, and consent to participation.

Name of Participant :____

Student ID (if applicable) :____

Contact Email :____

Signature :____