THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Mitigating the Vergence Accommodation Conflict in VR by Rendering Chromatic Aberration

Michael Hwang

School of Information Technology and Electrical Engineering

A thesis submitted for the degree of Bachelor of Engineering (Honours) at

The University of Queensland

4th November 2019

Michael Hwang
michael.hwang@uq.net.au

November 4, 2019

Prof Amin Abbosh
Acting Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia, QLD 4072

Dear Professor Abbosh,

In accordance with the requirements of the degree of Bachelor of Engineering in the division of Electrical and Biomedical Engineering I present the following thesis entitled "Mitigating the Vergence Accommodation Conflict in VR by Rendering Chromatic Aberration". The thesis was performed under the supervision of Dr Surya Singh and Dr Arindam Dey. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely,

Michael Hwang.

Abstract

Virtual Reality is a fast growing new technology that can provide users with a highly realistic experience by immersing them in a stereoscopic display. A problem that affects the long term usability of a virtual reality headset is a phenomenon called Vergence Accommodation Conflict, which arises from the inability of the human visual system to function normally within virtual reality. Within the stereoscopic headset, a special set of lenses projects the display to a viewable distance, called the virtual projection distance, taking up the user's entire field of view. The user can view any object near or far in the same way they could in reality, but although their eyes can converge and diverge naturally with depth, their accommodation (eye focus) is always at the virtual projection distance, unlike real life where both of these mechanisms are matched. This mismatch in vergence and accommodation distance is referred to as Vergence Accommodation Conflict, and causes symptoms such as eye strain, headaches, nausea and altered depth perception. To solve this issue we take a graphics based software only approach, and develop an image processing algorithm based on an existing method called Chromablur, that produces images with realistic depth cues that can stimulate accommodation to respond like it would in real life. This algorithm works by rendering defocus blur and longitudinal chromatic aberration, the combination of which provides the eye the signal it needs to accommodate correctly. We adapt the original algorithm for use in VR and assess its effectiveness in mitigating the Vergence Accommodation Conflict in two experiments, and find promising but inconclusive results.

Contributions by others to the thesis

The test environments for the user studies in this thesis were obtained and used with permission from Jane Phoon for experiment 1, and An-Yi Yao for experiment 2, suggested and authorised by Dr Arindam Dey.

Research involving human or animal subjects

The user studies conducted for this thesis were approved by The University of Queensland's Human Research Ethics Committee.

Acknowledgments

I would like to thank Dr Surya Singh for his continual support, encouragement and supervision over the course of this thesis, and pointing me in the right directions while giving me freedom to explore.

I would also like to thank Dr Arindam Dey for his advice, support, and for allowing me access to resources already stretched thin.

Thank you to everyone else who gave me a helping hand with anything related to thesis, everyone involved with the EXRPC lab and my personal support network.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

In recent years digital media has evolved rapidly in quality and type of content delivered. With the vast majority of society becoming a consumer of some variant of digital media, whether recreational, professional or business related, the demand for advances in technology for the delivery of this type of content has been eagerly upkept by the joint effort of business and science. Some of the latest platforms to emerge and garner the attention of the general public are virtual reality (VR) and augmented reality (AR), collectively referred to as extended reality (XR).

VR technology allows the consumer to be utterly immersed in the content they view and interact with it in a manner most closely resembling a real experience. Although VR is readily available to consumers, it is still far from a perfected technology. One major problem that affects the user's quality of experience in XR is a phenomenon known as the Vergence Accommodation Conflict (VAC) [7], which we will focus on in the context of VR.

VAC is a problem that occurs when the human visual system tries to interact with virtual imagery in the same way it would real life. The human visual system incorporates various different mechanisms both monocular and binocular to process visual information and give us our sense of sight. The key mechanisms we are interested in are vergence and accommodation. Vergence refers to our eyes' ability to converge or diverge dependant on the depth of the stimulus [7]. It's main purpose is to maintain stereopsis by keeping binocular disparity at the correct level [8]. Accommodation refers to our eye's ability to change it's focal length, also dependant on the depth of stimulus [7]. It is the mechanism by which we keep the image on our retina sharp, or well focussed [9].

These mechanisms are neurally linked and called the accommodation-vergence reflex [1], due to how closely they work together to respond to real life stimuli. However unlike the real world, in a VR headset depth is only a convincing illusion. Despite being able to verge on near and far points in the scene, the incoming light is always coming from the digital display, which is projected to a virtual distance by the lenses within the headset [7]. This means the eyes are always accommodating to this virtual projection distance, and causes a mismatch in how the two linked mechanisms respond to depth cues - vergence can respond normally, but accommodation does not change at all [1,7]. This is what is referred to as VAC, with symptoms including: nausea, visual fatigue and depth perception issues that persist for a short time even after taking the headset off [1,7,9].

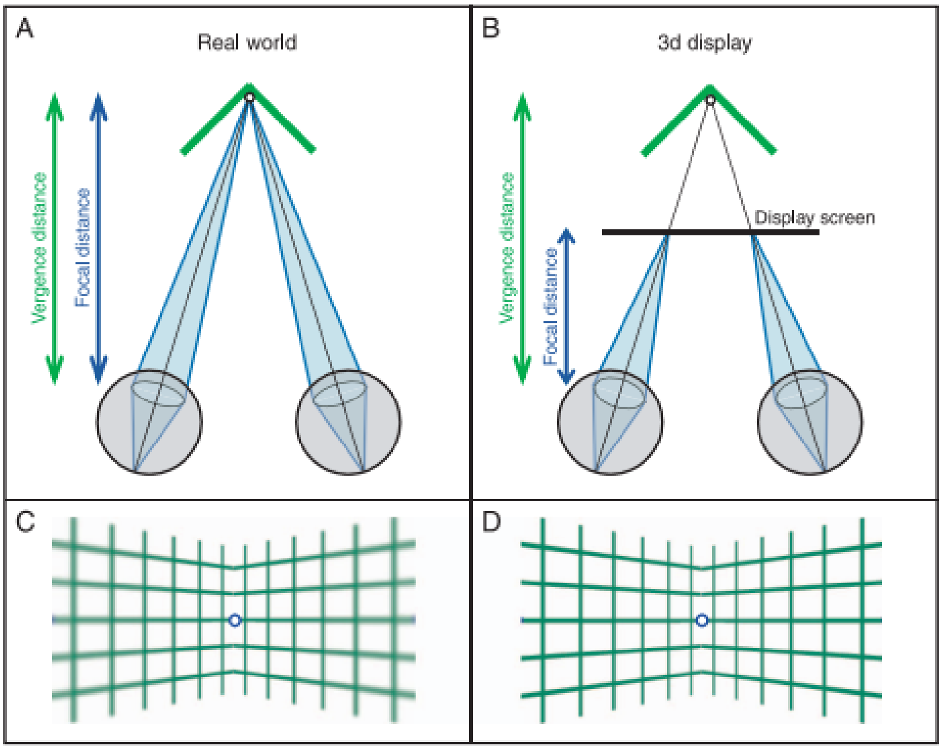

Figure 1.1: Figure taken from Hoffman et al [1] illustrating the mechanisms of vergence and accommodation in the real world (A) and in a 3D stereoscopic display (B)

VAC is not the only cause of feelings of general discomfort when using VR, and it affects different people to different degrees [10]. However it is a major contributor to visual fatigue [8], along with things like resolution, and impacts the user's ability to use the system for an extended period of time. Mitigating VAC is therefore one of the key problems to solve to create VR systems that all people can use for as long as they wish.

1.2 Topic and Scope

There are many solutions to this problem that are quite effective, and the newest models of XR headsets already include some form of VAC mitigation. However one approach that is under explored is a graphics based approach [10], which is distinguished from other more common methods by attempting to drive accommodation through manipulation of the rendered image, rather than manipulation of the light path with special lenses. The advantage of a software only approach is that it circumvents the need to redesign the optical hardware of a headset, instead increasing the load to the graphics processor, allowing mitigation of VAC in both existing and future platforms.

The specific approach chosen for exploration in this thesis was a method of rendering graphics more reminiscent of realistic human vision than the conventional method of attempting to maximise photorealism, based on a concept originally presented by Cholewiak et al in their 2017 paper "Chromablur: Rendering Chromatic Eye Aberration Improves Accommodation and Realism" [2]. The problem with photorealistic rendering is that it does not take into account the fact that even normally functioning human eyes have many optical aberrations present, and that these aberrations often serve the visual system as depth cues [2]. Viewing a photorealistic image of a scene does not generate the same signal as viewing the real scene, as it does not present our eyes with the natural wavefront of incoming light [2, 10]. This is apparent if one tries to focus on a point at a particular depth in an image; because everything is rendered sharply we do not perceive the proper depth of field effect in the unfocussed parts, as we would in real life [7]. Like depth of field, many visual effects we experience in real life are not conveyed properly in photorealistic imagery. In this thesis the focus will be on chromatic aberration and its ability to provide a depth cue to drive accommodation.

Cholewiak et al develop an image processing effect called Chromablur that is very computationally expensive, but produces a highly accurate result which renders in the order of minutes. It is a reference quality result that shows the effectiveness of the concept when viewed on a computer screen or projector, and they suggest that adjustments can be made to make the Chromablur effect applicable to XR systems, where providing accurate depth information is highly desirable [2]. This thesis will attempt to replicate their results by applying their effect within VR. This will require modification to the original algorithm, and design choices that will likely trade off quality for a result that can be rendered in real time, suitable for a VR display. The VR version of Chromablur will need to pare down the original algorithm such that it can be performed on a VR headset on a VR capable PC, while still maintaining the effect of driving accommodation and thus mitigating VAC.

To take the algorithm from a static, vision science context to a dynamic VR context, new assumptions must be made and constraints drawn. This will require an in depth understanding of the assumptions and design choices of the original, such that the relevant elements are retained and unnecessary ones can be discarded or replaced with something more suitable. A thorough understanding of VR and the VAC problem itself is also key in applying this solution. The design must be well grounded in the scientific theory, but make practical and technical compromises where necessary. The available development platform was the HTC Vive, and so design considerations were targeted to the specifics of this hardware. The software used to develop the environment and effect was Unity 3D, using the SteamVR API to provide VR functionality. The design was influenced by the capabilities and limitations of this software, and are further discussed in subsequent sections.

This document will explore all the design choices and compromises made, making justifications and detailing the development and evaluation process.

Chapter 2

Related Work

In this section various methods that have been developed to mitigate VAC are described in general, with an exemplar identified and explored.

2.1 Varifocal Displays

The concept of a varifocal display is simple and can be implemented in a variety of ways. The fundamental idea that addresses VAC is a system with a modifiable focal length to allow the eye to accommodate accurately depending on what is being perceived, rather than at the fixed depth of the display screen. This can be achieved through a focus tunable lens, or a mechanical actuation of the physical display [10].

A challenge this concept faces is the driving factor of the tunable focus. Focus should be tied to the perceived depth of regions in the scene, such that the light coming from those pixels appear to be coming from the correct depth. However with only one tunable element this cannot be achieved accurately across a whole scene with a dynamic depth variance [10]. The solution most successful implementations of a varifocal display utilise is the incorporation of gaze tracking to determine which region of a scene the tunable element will focus on.

A recent study by Padmanaban et al. implements such a system and tests their device on a range of subjects [11]. In their 2017 paper, Padmanaban et al propose two varifocal HMD systems that will not only mitigate VAC but also correct natural optical errors in the human eye. Their first system is based on an electronically tunable liquid crystal lens, which they first use to correct refractive errors based on user questionnaires similar to vision tests. They then attempt to drive accommodation to match vergence in the HMD by tying the focal power of the lens to a rendered target moving in and out of depth. Their results show that this method does indeed drive accommodation for a more natural viewing experience, however the system is limited by the aforementioned inability of such a system to drive accommodation in the entire scene. The second system designed by Padmanaban et al. accounts for this limitation by incorporating a gaze tracker into their system. For this system the team implements a varifocal display using mechanical actuation of the display screen using a small motor attached to the HMD, so as to enable the installation of the gaze trackers within the device. This time the perceived focal distance is tied to the user's point of vergence, naturally matching accommodation and vergence. Their user study of this system achieved similar accommodative gains as the first system, while the limitation was eliminated.

Padmanaban et al's work shows that a VR HMD can be designed using a combination of gaze tracking and a tunable focal element to satisfactorily mitigate VAC for users. They also contribute valuable data for users with refractive errors such as myopia, hyperopia, and presbyopia. While the first two conditions were easily corrected using the varifocal lens, they found that users suffering from presbyopia - the majority of adults aged over 40 - did not experience the same accommodative gains as the subjects with normal or corrected vision. They find that the decline in accommodative capacity associated with presbyopia may even mean a system attempting to dynamically drive accommodation is detrimental to the overall experience.

2.2 Multifocal Displays

Multifocal displays provide accurate depth information by breaking down a 3D scene into multiple 2D layers by their depth and displaying them in a spatially or temporally multiplexed manner such that the eye reconstructs the scene with the addition of more realistically varied depth information [10, 12]. The process by which the 3D scene is decomposed is a major factor that affects the performance of a multifocal system, as well as the manner in which the layer images are presented. Multifocal displays have a wide range of applications in both VR and AR, and are often used to implement a light field display [10].

Initial multifocal display designs had weaknesses in that for a spatially multiplexed system, it was impractical to find a display material with enough transmittance to let light pass through all of the layers and still exhibit required qualities such as resolution and FOV [10]. For a temporally multiplexed system, early hardware was not capable of producing a high enough frame rate and the systems were subject to visible flickering and artifacting [10]. Recent work however finds much success in multifocal methods, with Zhan et al. implementing a VR system that makes use of both multiplexing methods [13], and Lee et al. developing a new optimisation method for scene decomposition that is used in conjunction with a holographic lens for an AR system [14].

Zhan et al. design their system's focal plane stack with a set of Pancharatnam-Berry phase lenses

(PBLs). Similar to the way in which a lens in a standard fixed focus VR HMD creates a virtual display, each of these PBLs creates a virtual image plane at different depths, which are selectively shown by applying a voltage to a specific PBL to put it in “active driving” mode, while the non-selected PBLs remain in “passive mode”. The 2D layer images that each PBL displays are found through a light field factorization method, resulting in a viewing experience of a 3D additive light field. They find in assessing the performance of their system that image quality is content specific, with complex scenes producing low quality display outputs. It is found that increasing the number of focal planes and thus virtual depth planes mitigates this effect, allowing more complex scenes to be displayed at high resolution. The temporal multiplexing method works in conjunction with the spatially multiplexed hardware setup as the PBLs exhibit a very fast response time of 0.54ms, which they calculate to be fast enough for a display screen frame rate of 1kHz. Their experimental results show images with physically accurate depth information while maintaining the original display’s resolution.

Lee et al. take a different approach in designing a multifocal display system, putting much of their focus into the optimization method of scene decomposition and applying it to an AR system. They introduce a new method called “foveated retinal optimization”, which extends existing retinal optimization methods of finding 2D layer images by taking inspiration from the foveated rendering technique [15]. They posit that this method will provide 2D layer images that when combined will give a more accurate depth cue to the viewer by using a holographic lens and light field techniques. They present a novel result in that they achieve this foveated optimization technique without the use of eye tracking. Instead they analyse the effect of the “pupil swim effect”, a problem that causes a loss of resolution due to layer dislocation as gaze angle increases. They design an optimization technique that is robust to this effect, providing a weight matrix that creates a tunable tradeoff between central contrast, peripheral fidelity and layer image separation. Their results show that while this method succeeds into creating continuous and accurate focus cues in images regardless of gaze direction, there are many limitations that must be addressed. Firstly their decomposition method for higher resolution images was too slow to be used in real time fashion despite only finding 2 layer images, although they mention lower resolution and a change in parameters could speed this up. Also, their system only achieves a FOV of 38 x 19 degrees, with a low level of luminance caused by the complex optical hardware they employ to achieve their holographic display. They detail that this work could have applications in VR also, and plan for future improvements that will address the weaknesses of this prototype.

2.3 Focal Surface Displays

In 2017 the Oculus Research team published a paper outlining the development of a new approach to mitigating VAC in HMDs, called focal surface displays [12]. This method is most similar to multifocal displays, however rather than constructing a scene through multiple static focus planes, focal surface displays form the scene by analysing the displayed image's depth map and using a phase only spatial light modulator (SLM) to make the light appear to be coming from different depths in the image. This recreates near-correct natural retinal blur, which is the mechanism by which this method mitigates VAC.

The system works by taking a depth map of an image as input, containing the geometric depth information of the rendered scene along each viewing angle. SLMs can only produce a smooth focal surface, so the depth information must be decomposed into multiple smooth focal surfaces, otherwise a depth map would be all that is required to make light from every pixel appear to come from the correct scene depth. Oculus Research successfully decomposes the depth map into discrete smooth focal surfaces via an optimisation problem; given an arbitrary number of smooth focal surfaces, each viewing angle in the focal surface should match the corresponding point in the depth map as closely as possible. This is optimised using non-linear least squares and a set of smooth focal surfaces is obtained.

These focal surfaces must then be implemented in the SLMs using a set of phase functions. These phase functions are obtained by determining the target focal length of each pixel in the SLM to create the image described by the smooth focal surface, and solving a linear least squares problem that approximates the phase function. With such a set of phase functions, the incoming light to the SLM will be modulated such that it appears to form the focal surface image.

The Oculus Research team determines that even one focal surface can compare to a multifocal display with four planes, and two focal surfaces is generally enough for an effective representation of a 3D scene. An optimised blending technique is used to determine the colour values in the target from the original image, and the resultant focal surface display is formed.

Although this method shows that natural focus cues in virtual images can be recreated using focal surface displays with high quality results, the technology is very new and is limited in many aspects. For instance focal surface displays as represented in the Oculus Research paper are not a wearable technology, limited by the hardware used to achieve the rendered image. The technology is also built around a fixed viewpoint with the assumption of no eye movement; it is stated that focal surfaces are not yet suitable for interactive scenes, and indicates a combination with eye tracking to display these images in a gaze-contingent manner may be the next step forward.

2.4 Light Field and Holographic Displays

Light field and holographic displays are inherently better at providing depth cues and minimizing VAC than conventional displays [7]. This is because unlike other display methods, light field and holographic displays attempt to reconstruct the original wavefront of the depicted image [12, 16]. This means that focal length is not fixed at the virtual projection distance, and in theory allows the vergence and accommodation of human eyes to work as they would when viewing a real scene.

Holographic displays are a specialised form of light field displays [7], the latter approximating an image's natural wavefront through samples of "ray bundles" whereas holographic displays allow for finer control of the wavefront [10, 16]. The tradeoff is usually higher computational complexity in holographic displays to achieve the same level of image quality [16]. However holographic displays do not suffer from the trade off between spatial and angular resolution due to diffraction that is inherent to light field displays [16]. Light field and holographic methods are more commonly employed in AR see-through HMD applications, with notable exemplars such as Microsoft's Hololens and the Magic Leap. However the concept is also applicable to VR, and is explored by Microsoft Research in their 2017 paper"Holographic Near-Eye Displays for Virtual and Augmented Reality" [16].

Microsoft Research presents various designs applying a phase-only SLM to implement a holographic projector that is used to display images in a near-eye system. They introduce multiple processing techniques that allow for more optimal computation of holographic images. Their designs are based on the principle of Fresnel holography, which they implement using a point light source and an SLM that is programmed through a GPU. The complex computations required to render a detailed 3D holographic image is performed via shader programs on the GPU, where input image data is converted into a phase-only representation of the 3D hologram, which is sent to the SLM. Another technique that is described is increasing display performance using eye tracking. In a similar fashion to foveated rendering through eye tracking [15], the Microsoft Research team show a method of reducing computational load in areas of the hologram that are on the periphery of the point of gaze fixation by reducing resolution, or adjusting other parameters within their processing steps based on the fovea region.

The paper presents four different prototypes incorporating the techniques they describe; two VR applications and two AR. The two VR prototypes showcase two different achievements: one is a holographic VR display with a FOV of 70 degrees, while the other is a holographic VR display with a true 3D hologram with accurate depth cues, and the ability to correct refractive vision errors. However these prototypes were limited in that they were both benchtop prototypes, and the rendering speed of the second was too great for real time processing. They address this second issue by applying a form of the eye-tracked processing shortcut, however it is based on mouse cursor position rather than through implementation of an eye tracker. This allowed them to display the 3D hologram in real time and found that there was insignificant computational penalty for adjusting the focus level to the cursor position for each frame. They leave the first issue for future work, aiming to implement all of these techniques into one head mounted unit with eye tracking.

2.5 Rendering Techniques

There have been many attempts to solve the VAC problem in a software-only graphics based approach, most often using a specialised rendering technique. The aim of such rendering techniques is usually to drive natural accommodation by rendering depth cues such as retinal blur, also called defocus blur [10, 12]. However many studies have shown that simulated retinal blur does not actually drive accommodation to vergence [10], and thus most recent studies focus on hardware reliant methods to recreate natural retinal blur by seemingly projecting light from the display to different depths [10, 12].

As previously mentioned, Cholewiak et al present a rendering technique that drives accommodation very effectively, which this thesis adapts and explores. In their 2017 and 2018 papers, they discuss the mechanisms by which the human eye actually accommodates correctly, and show that retinal blur or defocus blur, the most popular depth cue to recreate, does not actually carry signed information about the state of focus [2, 3]. In fact, a study by Marin-Franch et al shows that when retinal blur is the only depth cue, the eye can only accommodate via a trial-and-error mechanism of micro-fluctuations in the focal length of the lens [9]. When considering accommodation as a closed loop control system, retinal blur would be an odd-error signal. Cholewiak et al then proceed to identify depth cues that could potentially serve as even-error signals that drive accommodation. They look to higher order aberrations present in most human eyes and analyse the viability of astigmatism, spherical aberration, microfluctuations and longitudinal chromatic aberrations in serving as such an error signal [3].

Astigmatism occurs in the majority of adults and causes the approximated point spread function (PSF) of incident light on the retina to be elliptical, with the major axis being dependant on whether the point of fixation is in near or far focus. Spherical aberration causes the incident rays in the periphery of the retina to focus at different distances dependant on near or far focus. Microfluctuations provide a signal by the result in clarity from a fluctuation in a particular direction of focus, leading the trial-and-error mechanism to converge on the correct focus. Chromatic aberration describes the effect of the eye's lens having different refractive indices for different wavelengths of light, referred to as longitudinal chromatic aberration (LCA). It is present in most human eyes and is consistent in its effect across the population. LCA means that depending on the current focus of the eye, colours will be perceived with different sharpness, providing a signed depth cue to drive accommodation. This discrepancy in colour is not consciously perceived but has been shown in many studies to affect accommodation. This suggests that appropriate colour rendering may be able to drive accommodation.

Cholewiak et al produce images rendered with these aberrations present, and find that all except LCA are weak cues for driving accommodation. LCA based rendering however was shown in their user studies to very effectively and accurately drive accommodation, additionally LCA is the only effect among those examined that is constant for all eyes and can provide a general solution. They test their system by comparing the accommodation of a subject viewing an image through a focus adjustable lens presenting the image at varied focal lengths, and an image viewed through a fixed focus lens with LCA rendering to simulate changes in focus. They find that changes in accommodation are very similar in these two conditions. With promising results to show for their approach, they suggest that for use in VR applications their technique could be used in conjunction with gaze tracking to mitigate VR in a non-optical approach.

2.6 Eye/Gaze Tracking

Gaze tracking technology is only recently being explored for use in VR applications, with the advent of the foveated rendering technique that allows for greatly reduced computation time in rendering a display [10, 15]. Companies such as Tobii and Fove are at the forefront of eye tracking in VR HMDs, releasing development kits and prototypes with the eye tracking functionality [17, 18]. Eye tracking has many uses in VR, such as the aforementioned foveated rendering, or as an additional user input method, data collection, and in many cases it can be incorporated into various techniques of VAC mitigation like varifocal displays [10, 11]. With the installation of this single additional system, all of the above applications can be achieved in one HMD device. Future XR platforms will most likely all offer eye tracking functionality.

Chapter 3

Theory

Before relating the design process of the VR version of Chromablur developed for this thesis, a thorough understanding of the original is required.

3.1 Chromatic Aberration

The Chromablur algorithm works by taking advantage of the effect of LCA in the human eye. The refractive index of the eye's lens is dependent on wavelength, so when incoming light is focussed onto the retina, different wavelengths are focussed to different points, in the same way a prism refracts visible light and separates them into the colour spectrum. When light is not properly focussed on the retina, we experience a degree of defocus blur which is proportional to the difference in distance of this incorrect focal point to the retina. These conditions are called myopic and hyperopic focus, when light is focussed in front of and behind the retina respectively. The difference in focal distances for the different wavelengths of visible light are very small, so we do not perceive blurry vision due to this effect, but we do perceive different colours with different sharpness.

For example, when the eye is focussed on a white point then wavelengths corresponding to green would be the best focussed on the retina, with wavelengths corresponding to blue and red having myopic and hyperopic focus due to the degree of refraction each wavelength experiences. The mismatch in focal distance means that if our retina only detected light from either blue or red wavelengths, the image formed by our brains would be very slightly blurred. This slight blur manifests as coloured fringes around objects with step changes in depth or colour in our vision such as occluding objects or contrasting colours. Although this fringe is not perceived consciously, the colour of the fringe provides the eye with valuable information regarding the state of its focus [2]. For a well focussed eye receiving all wavelengths of visible light, green would be the sharpest while the red and blue ends of the spectrum are expected to be slightly out of focus, producing a purple fringe and sharp greens. For an eye that is myopically focussed, red would be brought into sharpness with a blueish fringe, with some defocus blur present due to the incorrect focussing. A hyperopically focussed eye would experience the reverse; sharp blues and a reddish fringe.

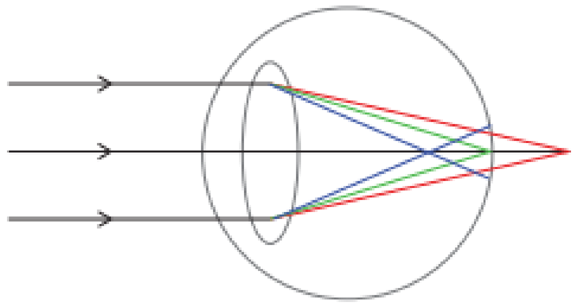

Figure 3.1: Here the effect of chromatic aberration is shown for a well focussed eye, green is focussed directly on the retina while blue and red are improperly focussed. Taken from Cholewiak et al [2]

The relationship between wavelength and the difference in focal distance is given by the following equation:

$$D(\lambda) = 2.071 - \frac{633.46}{\lambda - 214.10} \tag{3.1}$$

Where D is the difference in focal distance and lambda is wavelength in nanometers. 520nm is assumed to be in best focus, giving D(520) = 0. While the hyperbolic form of the equation is derived by Thibos et al [19] from Emsley's reduced eye model of refractive error and Cornu's hyperbolic formula for the refractive index in water, the exact numbers were obtained from regression co-efficients resulting from fitting the model to empirical data collated from many prior studies [3, 19].

3.2 Chromablur Algorithm

The importance of the role of LCA in the eye's accommodative response is a well studied area of vision science, and the Chromablur algorithm applies that understanding practically. The goal of Chromablur is to compute an image to display that when viewed by the aberrated but in focus human eye, will produce a retinal image that incorporates LCA and defocus blur, and will stimulate the eye to accommodate according to what kind of focus state the effect specifies. In contrast, conventional photorealistic rendering produces retinal images that lack any sort of aberration that could stimulate the accommodative response. The problem that the algorithm attempts to solve is expressed as follows:

$$I_{R,G,B}(x,y) = D_{R,G,B}(x,y) ** K_{R,G,B}(x,y) \tag{3.2}$$

Where I is the target retinal image that we wish to form in our eyes, D is the display image that will be shown on the screen and viewed by the eyes, and K is a wavelength dependent blur kernel that models the point spread function of light on the retina due to the effect of LCA. R, G and B represent the three colour channels of the image, and the ** operator denotes convolution. It can be seen that I is simply the convolution of D and K, although obtaining these parameters is a non trivial task.

There are two versions of the algorithm presented, one for 2D non depth varying images and one for 3D scenes. The 3D version will be the basis of this thesis as it is more applicable to the target environment of VR. The 3D Chromablur algorithm has two steps to solve the equation, an initial forward calculation and an inverse calculation to obtain the final result. In the forward calculation, the target retinal image I is calculated. Cholewiak et al do this through ray tracing, where the camera properties model the chromatically aberrated eye using equation 3.1, and each colour primary is approximated by one wavelength value. In the inverse step, D is calculated by deconvolving I with K. The blur kernel K is a point spread function derived from the complex aperture function of the eye, which Cholewiak et al approximate with cylinder kernels for the calculation. Noting that the resulting optimization cannot be solved analytically, the solution is obtained through ADMM optimization, which works iteratively until convergence to reach a solution.

Although the 2D algorithm is designed for static, non depth-varying images its process is useful to understand. The target retinal image I is obtained by convolving the original image with the proper form of K, then to find D the colour channels of the original input image are convolved with varying cylinder kernel approximations, and the RMS error between the target retinal image is minimised to arrive at the output image.

3.3 Assumptions and Limitations

Although Cholewiak et al conclude that their method is effective in driving accommodation after conducting multiple studies, there are some weaknesses to be acknowledged and assumptions that affect the accuracy of their result. The first most obvious weakness is that this method relies on the viewer having colour normal vision. Cholewiak et al conduct a study to investigate this, and confirm that people with colour deficient vision will not benefit from Chromablur, and additionally people suffering from presbyopia, which occurs in all people as they age, also do not benefit. People with refractive errors like myopia and hyperopia are able to take advantage of Chromablur once their vision has been corrected.

There are two key assumptions made by the Chromablur algorithm, that could affect it's performance, although this influence is not fully explored. The first is made when calculating the wavelength dependent refractive error; an assumption has to be made as to what monochromatic wavelength value will be used to represent each colour primary. Error is introduced in two fronts, firstly when the empirically derived equation 1 is adjusted such that a different wavelength is in focus, the equation must be vertically translated such that the desired wavelength is at 0 (Figure 3.2). Secondly, the wavelength of colour primaries emitted by displays are usually broadband and vary across manufacturers and models, versus the assumed monochromatic value. In supplementary material to their paper, Cholewiak et al investigate the RMS error between images produced with Chromablur and conventional rendering for the two conditions of monochromatic and broadband primaries using bandpass filters to control the light from the display. They find that across differing levels of defocus, the monochromatic case does not generally worsen the error, and so conclude that their assumption is valid.

The second assumption made by the algorithm is the value of pupil diameter. Pupil diameter influences the amount of blur we perceive, as well as refractive error. It is an input parameter to the blur kernel and so affects the final result to some degree. An assumption must be made rather than measurement as pupil diameter can change due to many different factors such as brightness, medication, hormonal changes, and the accommodative response itself. Cholewiak et al base their assumption on the accepted average pupil diameter found in literature, around 4mm, and provide insight into the validity of assuming a constant value. The point spread function of defocus blur is modelled by a disk kernel, whose diameter is given by:

$$\beta \approx A|\Delta D| \tag{3.3}$$

Where β is the disk diameter in angular units, A is pupil diameter, and D is refractive error causing the blur. This kernel is incorporated in the wavelength dependant kernel used in the Chromablur

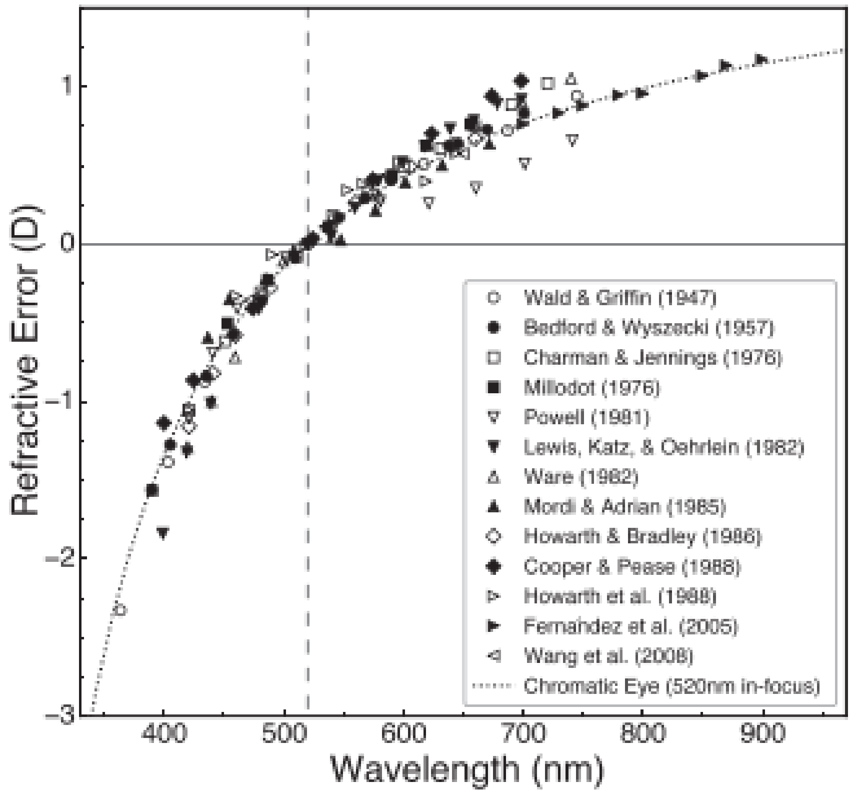

Figure 3.2: The hyperbolic curve of refractive error introduced by LCA. In the figure 520 is shown in focus, as described by equation 3.1. To bring other wavelengths into focus the curve must be shifted vertically. [3]

algorithm. The defocus of different wavelengths is described by this equation when the magnitude term is taken from the refractive error due to LCA. As the disk diameter is proportional to both pupil diameter and magnitude of the LCA effect, a scaling of one parameter is equivalent to scaling the other. According to literature, the effect of scaling the magnitude of the LCA effect has no correlation to the accommodative response stimulated, so Cholewiak et al conclude that the assumed value of pupil diameter will not affect the performance of Chromablur.

The final note to be made is the viability of applying Chromablur to a VR context. As mentioned above, the mechanisms by which Chromablur drives accommodation are rendering the LCA effect as well as defocus blur. The defocus blur is necessary as it is the stimulus that tells the eyes that an adjustment to accommodation must be made, while the LCA effect provides the signed information about which direction the accommodation should be in, myopic or hyperopic. However unlike in real life or the controlled setting of Cholewiak's experiments, in VR even though this effect could drive the eyes to accommodate to the new depth, the rendered defocus blur would not be minimised once the eye refocusses, resulting in a blurry image. This is noted and investigated by Cholewiak et al, as they hypothesise that after an initial accommodation in the direction specified by the LCA effect, the resulting blurrier image would cause the viewer to accommodate back to its original state if exposed to the stimulus for a longer time. They find that subjects tended to maintain the new accommodation specified by the LCA effect even after longer periods of exposure to the stimulus, when presumably the image had gotten blurrier. This implies that a tradeoff can be made in image quality and accuracy of accommodation, although both are important to maximise for the best VR experience.

Chapter 4

Design

4.1 Identifying Constraints and Capabilities of the Development Environment

To begin development, the tools and development environments available needed to be identified and set up. The available VR headset was the HTC Vive, and so Unity 3D was chosen as the development environment, due to its compatibility with the Vive. There are many ways to implement VR functionality within Unity, however the SteamVR API was chosen due to its ease of use.

Initially there were two methods considered to implement the Chromablur effect within Unity, one method was to implement it as a post-processing effect, and another was to pre-process the scene earlier on in the rendering pipeline. The post-processing approach had the advantage of being much easier to implement, with the disadvantage of the effect's calculations having to be performed twice due to the VR pipeline having to render the scene once per eye. Any impact on computation performance significantly affects the quality of the VR experience, so this was thought to be a large detriment. The idea behind the alternative method was to avoid having to double up on computations by performing them earlier on in the rendering pipeline, however it was uncertain whether this was possible to do, and if it would have the intended benefit. Attempting this method would be far more difficult and require modification of the SteamVR API. Unity provides a default rendering pipeline for all their 3D applications which SteamVR works with, and also two new scriptable render pipelines that allow customisation. However documentation of these scriptable pipelines was minimal, noting this feature was targeted at experienced graphics developers. Lacking the prerequisite graphics knowledge and skills, it was decided that the post-processing approach would be attempted first. The pre-processing method presented a risk to the project in that it would require a much larger time investment to learning, without guarantee of a better result. The problem of the post-processing method was recognised as a risk due to the theory of VR rendering, however there was potential that it may not significantly affect performance in practice. This risk could also be minimised by optimising the algorithm where possible.

In Unity post-processing effects are made up of a shader and script file, shaders being written with a combination of the Unity native ShaderLab language and standard HLSL, while scripts were written in C#. By taking the current image seen by the virtual camera as input, post-processing effects can manipulate the final output image with great freedom, by applying computations on each pixel. Before developing the post processing effect however, a 3D scene to apply it on was required. As the focus of the thesis was the visual effect, freely available online environment assets were downloaded from the Unity Asset Store and modified to meet needs. The following free to use Unity VR tutorial environment was downloaded for initial development.

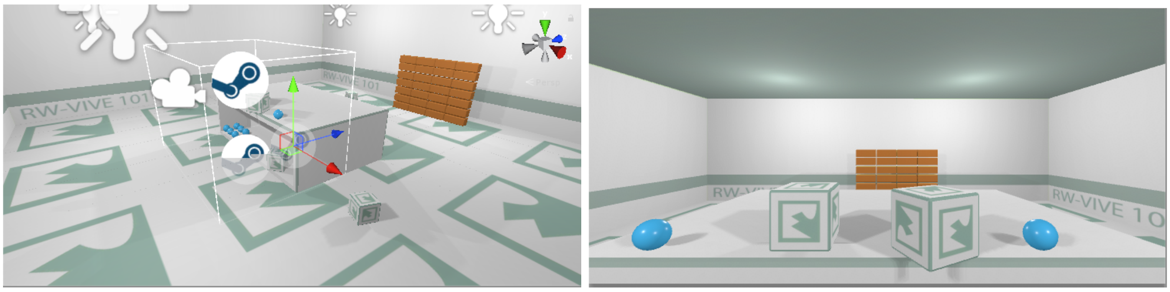

Figure 4.1: Left: Unity Editor Scene view allowing control over environment. Right: Unity Editor Game view showing a preview of the camera's rendered output. Environment asset provided by Eric Van de Kerckhove [4]

This asset was suitable for initial development of the post-processing effect due to having an array of objects with varying geometry in a simple environment, where it was easy to position the camera and edit objects to produce images with depth and colour variance that would be suitable to show the Chromablur effect. Although the goal was to develop the post-processing effect for VR, it was determined that for testing purposes using the preview window in the Unity editor was sufficient until significant progress was made.

4.2 Designing the Algorithm

To simplify Cholewiak et al's Chromablur algorithm, the first step was to assess the viability of their own suggestions presented in the discussion section. The first suggestion was to use OpenGL shader approximations of their algorithm utilizing depth varying disk blur in the three primary colour channels, this would be able to replace the ray-tracing operation performed in the reference quality version. This suggestion seemed highly viable, as an OpenGL shader performs very similarly to a HLSL shader, but for a different system. This suggestion was hypothesised by Cholewiak et al to be sufficient to drive accommodation, however it is noted that it would introduce errors where depth gradients are large in an image. The other suggestions consisted of improvements to the methods of the original algorithm such as using real time ray tracing rather than offline, or different methods of performing deconvolution in the inverse step.

4.2.1 Forward Step

The depth varying disk blur method was seen as a useful starting to point to the simplified algorithm, as it would replace the ray tracing operation in the forward step with simple convolutions. The goal of the disk blur method would be the same as the ray tracing – to produce a target retinal image that we desire to be formed in the eyes. The ray tracing method achieves this by very accurately modelling the behaviour of light and its interaction with the eye's lens, at the cost of very high computation complexity. The disk blur method would instead model the effect of chromatic aberration in the eye, as shown in figure 3.1. As described by Cholewiak et al, this translates to applying a depth varying amount of blur to the relevant colour channels, which will cause the final image to be rendered with some amount of colour fringing as the channels misalign. The blur in each channel is applied using a disk kernel, as we consider the light corresponding to that channel as being improperly focussed on the retina, so it must be rendered with the appropriate defocus blur. As mentioned previously, the point spread function due to defocus is a disk or cylinder function, so we can approximately consider the point spread function due to chromatic aberration to consist of multiple disk kernels corresponding to each colour channel.

Before implementing this approximate point spread function, the parameter of computational complexity was investigated. In graphics literature it seems to be the convention to simulate defocus blur using a 2D Gaussian kernel rather than a disk kernel [3]. This is due to the property of 2D Gaussians being the only circularly symmetric kernel that is also a separable kernel [20], meaning the 2D convolution can be split into two 1D convolutions which we will refer to as a horizontal and vertical pass. This reduces the computation complexity of a convolution operation from $O(H * W * N^2)$

to $O(H*W*N)$, where N represents the size of the symmetric kernel and H * W represents the image dimensions [21], which is a very significant difference. Considering the need for optimal computation complexity in the post-processing effect, a separable kernel was highly desirable.

However, in optics literature it is well known that a disk kernel models the effects of defocus blur most effectively, and although similar, a Gaussian kernel does not achieve the same effect [22]. Strasburger et al explore the difference between a 2D Gaussian kernel and a disk kernel in a note for clarity, and use the phenomenon of spurious resolution as a key difference [22]. Spurious resolution occurs when an image with high spatial frequency is viewed out of focus, and causes periodic renewal of contrast at frequencies higher than when contrast first dropped to zero, arising from negatives in the optical transfer function [23]. This is commonly demonstrated using radial grating stimuli, such as the Siemens Star object.

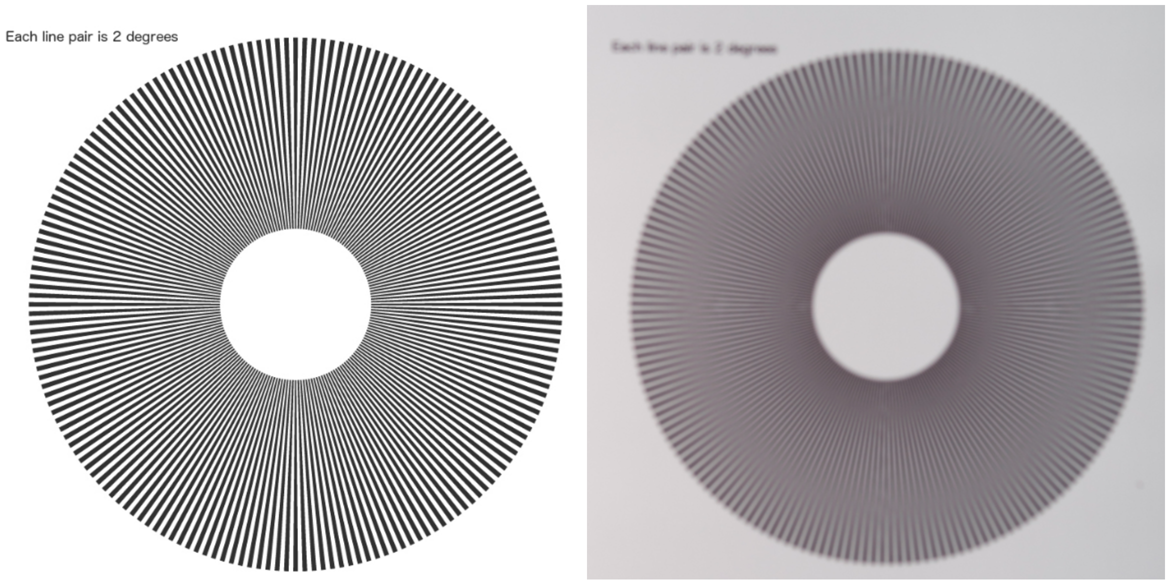

Figure 4.2: Left: The Siemens Star test object. Right: The Siemens Star with some defocus blur. Concentric circles of differing contrast are visible due to spurious resolution. [5]

Strasburger et al point out that while convolution with a disk kernel can produce spurious resolution, a Gaussian kernel cannot due to it's Fourier transform always remaining a Gaussian, and positive [22]. Although there may be cases where correcting spurious resolution with a Gaussian could be desirable, for the task of modelling accurate optics the disk kernel is more suitable.

This presented a situation where a decision had to be made between better performance with a separable Gaussian kernel, or a more accurate model with a disk kernel. Before fully committing to either side, a technique to gain the advantages of both options was investigated. First discussed by Olli Niemitalo [20] and later referenced by Kleber Garcia in SIGGRAPH 2017 [24], this technique aims to produce a circularly symmetric kernel that approximates a disk kernel using Gaussian kernels.

Niemitalo shows that like regular real valued Gaussian kernels, a complex Gaussian retains the separable and symmetric properties that make them desirable. He derives a method of producing kernels with these properties that approximate a disk by taking weighted sums of complex Gaussians [20], much like approximating a square wave as a sum of sinusoids, or optimal filter design. Although the 1D kernels produced are quite verbose, this technique would allow disk kernels to be used in the Chromablur algorithm while retaining the efficiency of a 2D Gaussian.

Figure 4.3: From left to right: a regular Gaussian kernel, a 1 component disk kernel approximation, and a 5 component approximation

The known disadvantage to this technique was it requires extensive pre computation of co-efficients depending on the parameters chosen that determine the accuracy of the approximation [20]. Increasing the number of complex Gaussian components will produce more accurate results, but require much more memory [25]. Additionally, programming of imaginary numbers is not natively supported in shaders, so helper functions to provide complex calculations had to be implemented, which require more operations than purely real numbers. The order of complexity of this method would still be $O(H*W*N)$, however it would take a substantially larger amount of calculations than a simple 2D Gaussian, as well as a significant amount of memory.

An initial implementation of a blur effect using separable disk kernels was attempted in Unity based on OpenGL Shadertoy implementation of Niemitalo's technique by Bart Wronski [25]. The co-efficients in this implementation were for a single component approximation, pre-calculated and made available by Niemitalo. Conversion from OpenGL to HLSL and from a the Shadertoy app to Unity was made challenging by the difference in architecture and standard macros and functions. Unlike the Shadertoy implementation, Unity shaders do not support multiple image inputs as easily, requiring storage as lookup textures. While the separated version of a 2D Gaussian would result in two "passes" of the shader and require one intermediary image buffer, the separation of the disk kernel as implemented by Wronski resulted in five passes and four intermediary buffers, and would require more to work in Unity. It began to seem unclear if this method would truly be more efficient especially for small kernels – Chromablur would only require a slight amount of blur so this increase in memory usage and shader passes may not give the desired advantage from using a separable kernel. It was decided that a simple 2D Gaussian or unseparated disk kernel would be better to use.

Due to the simplicity of implementing a 2D Gaussian, both methods were implemented for comparison. The 2D Gaussian convolution was achieved straightforwardly by convolving the columns and then the rows of the image by a 1D Gaussian taking the Gaussian parameters as input. It was found that Unity provided a set of pre-calculated disk kernels of various sampling density, consisting of co-ordinates taken from concentric rings within a unit circle that could be scaled as necessary [26]. As the disk kernel diameter would not be very large, the sparsest kernel was selected consisting of 16 samples. As well as these, a third method of blur called box blur was also implemented for comparison, which is also symmetric and separable but not circular. It was found that the Gaussian and box blurs are very similar, and from literature we know that multiple applications of simple box blurring can approximate a Gaussian blur by the central limit theorem [27]. Upon implementing these blur methods, the disk blur did not appear to be significantly slower than the two separable methods, likely due to using a small sample size, so it was chosen as the method to use for Chromablur.

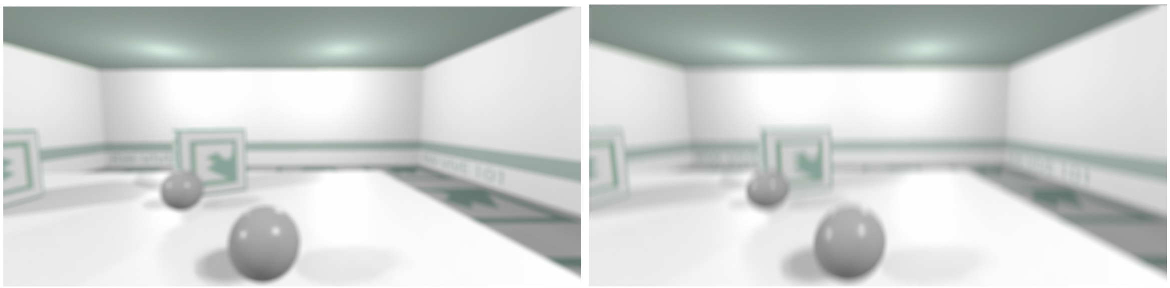

Figure 4.4: Left: Gaussian blurring with a mean of 0 and 0.5 standard deviation, 8 samples per pass and kernel size of 2% of the screen. Right: Disk blur using the 16 sample kernel, diameter of 2% of the screen

With the blur kernels ready, the next step was to tie the level of blur to defocus due to chromatic aberration. The two parameters that controlled the blur intensity was sampling density and kernel radius. To calculate the kernel radius, equations 3.1 and 3.3 were used to simulate the effect of LCA, while sampling density was kept constant. Equation 3.1 gives the wavelength dependent difference in focal distances in dioptres, which was input to equation 3.3 to find the radius of the disk kernel. As mentioned previously, equation 3.1 is given with the assumption of 520nm (green) being in focus, which is the "well focussed" condition of accommodation. To obtain the values for myopic and hyperopic accommodation, the equation needed to be adjusted so that blue and red wavelengths were assumed to be in focus. This was achieved by simply applying a vertical shift to the equation such that it gave zero when the wavelength input was 449nm for blue, and 617nm for red, rather than for 520nm.

These values were the same assumed values used by Cholewiak et al, along with an assumption of 4mm for pupil diameter.

By adjusting the LCA equation and inputting the assumed values, the appropriate blur kernel diameters were obtained, in degrees of visual angle. To translate this value to a pixel size that could be programmed within the shader, the following steps were taken. Visual angle refers to the angle that a visual stimulus subtends at the eye's lens, and the maximum possible visual angle that a stimulus can form is the eye's field of view (FOV). The visual angle of the blur disk diameter could then be taken as a percentage of the field of view, and this percentage could be applied to the image's pixel dimensions to obtain a corresponding pixel value. The official specifications of the HTC Vive state a FOV of 110^o, however FOV in VR headsets is known to be quite variable depending on variables like the interpupillary distance, eyebox distance, and the direction of viewing. As the lenses in VR headsets are not always symmetrical, the actual FOV provided is not a perfect circle, and thus a more appropriate measure of FOV should be given with a horizontal and vertical value. For the required calculations, the FOV value was taken from the maximum of the results obtained by Oliver Kreylos in an experiment comparing multiple headsets [28]. This value was 110^o by 113^o degrees, which were converted to radians for use in the shader program. Once the angular blur disk diameter was converted into a percentage of FOV, this was used to find the diameter in pixels by assuming that the displayed image took up the entire FOV of the headset. To make sure the kernel was indeed circular, the x dimension was scaled by the aspect ratio to account for the rectangular image, and then scaled to the new found disk diameter in pixels.

The final step was then to convolve the kernel with the image. There were three sets of calculations used to produce the final target retinal images corresponding to a hyperopic, myopic and well focussed accommodative state, as shown in figure 4.5.

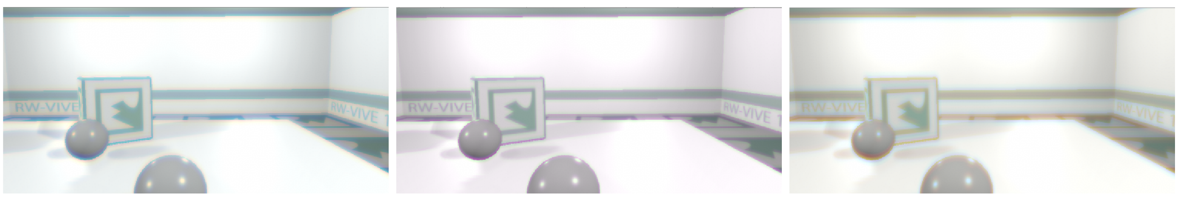

Figure 4.5: Left to right: Chromablur rendered scene showing hyperopic focus, good focus, myopic focus. The blur kernel diameters were increased for the purpose of demonstration

4.2.2 Inverse Step

The inverse step of the original Chromablur algorithm used an iterative technique to deconvolve the blur kernel from the target retinal image, producing the display image that would be the final output of the algorithm. The blur kernel used was nearly identical to the kernel described in the modified forward step: cylinder functions with diameters depending on wavelength and depth. This meant that following a similar process would simply result in the original input image being recovered after the deconvolution. In the original algorithm, the forward step represented the ideal result from which to recover the display image by deconvolving an approximate blur kernel, however the modified forward step described above is already an approximation with a certain amount of error. It was hypothesised that then an inverse step would not be necessary, and the modified algorithm could produce the desired result by directly outputting the target retinal image calculated in the forward step. As can be seen in figure 4.4, the target retinal image and Chromablur image are visually very similar, showing that in equation – the effect of the kernel K in being convolved with display image D does not observably change the image significantly.

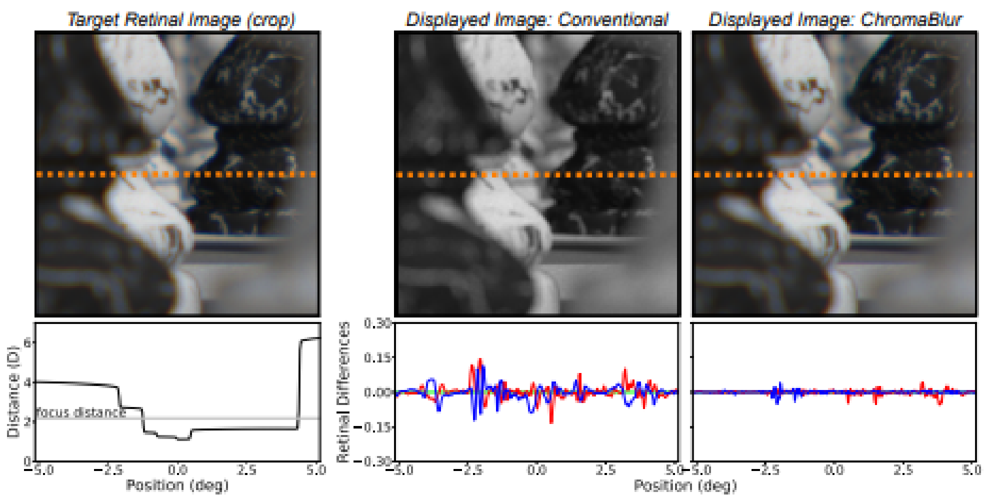

Figure 4.6: Shown in the figure are Cholewiak et al's results of applying their 3D Chromablur to a sample scene. The graphs from left to right show the scene's depth map across the dotted line, and the difference in intensity of of the display images to the target retinal image. [2]

This then finalised the modified version of Chromablur to be a process of convolving the input image with the approximate LCA kernel to output a target retinal image. Rather than solving equation 3.2 for D, the modified version outputs I, and takes the original input image as D. The blur kernel K is the approximate point spread function of LCA consisting of wavelength dependent disk kernels, which models the effect LCA has on incoming light in the eye. In theory this modified algorithm would mean that the retinal image formed in our eye is actually I * K not I, which lowers the accuracy of the algorithm compared to the original, however considering that K is already an approximation and not a completely accurate model of LCA, this lowering of accuracy was tolerable. The accurate model of LCA is given by the square of the Fourier transform of the eye's complex aperture function, which Cholewiak et al only use in the forward step of the 2D version of Chromablur and use the disk kernel approximation in all other calculations.

4.3 Application to VR

The final step of designing the new VR Chromablur algorithm was the method of its application to the scene. So far to test it, the entire scene or image was given the same effect uniformly – in focus, myopic or hyperopic. However the point of applying Chromablur to VR was to drive the viewer's accommodation to the same distance as vergence, thereby mitigating VAC. To do this we require a detailed understanding of the user's accommodative state while experiencing the VR scene.

It was found that the first environment that was prepared was not conducive to showcasing the Chromablur effect due to being quite limited in the depth range, and lacking fine details in textures and colours. A better test environment was found in the Unity Asset Store, called Viking Village which consisted of a very large, realistic looking 3D environment. The environment was edited to remove unnecessary components and the SteamVR API was added to provide VR functionality. The results of this section are shown in this new environment.

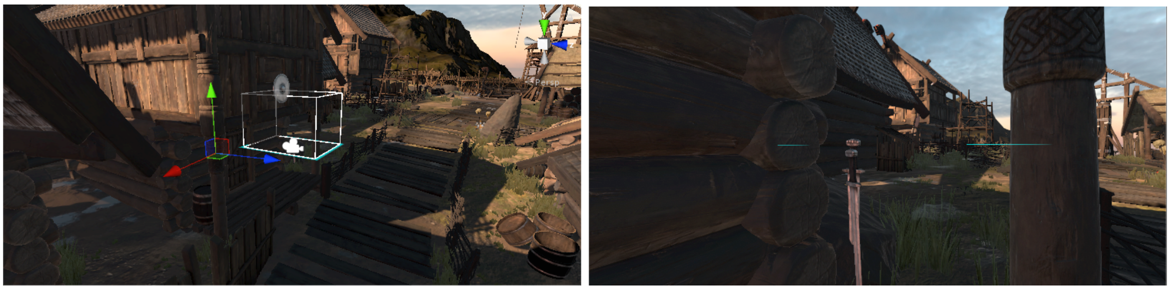

Figure 4.7: The Scene and Game views of the Viking Village environment taken from the Unity Asset Store. The horizontal light blue lines indicate the VR play area and are not visible when running the envionrment.

As defined previously, VAC arises due to the VR user's eyes always accommodating to the virtual projection distance of the screen, while their eyes can verge on any point in the virtual scene. It follows then that there may be times when the user's accommodative state matches vergence, and times when they are accommodated too far or too near compared to vergence. For the states where there is a mismatch we would like to stimulate the eyes to accommodate to the vergence distance. The factors we need to know to achieve this are the virtual screen projection distance, which will always be the accommodation distance, the current vergence distance, and the depth of each point in the scene. The depth of each point can be easily obtained by reading from the depth texture in the shader program which contains the necessary information, and virtual screen projection distance is a platform specific constant attribute of the VR headset. Unexpectedly, this specification is not commonly provided by manufacturers along with other specifications like resolution and FOV, but the information for the HTC Vive was found in early interviews and blogs of developers detailing the design of their product. The vergence distance however depends on what the user is currently looking at within the scene, and to determine this we would require some kind of eye tracking system. Although this would potentially provide the best method of applying Chromablur to mitigate VAC, one of the reasons for using this rendering only method was to keep the solution free of any hardware reliance so that the solution could be applied to all VR platforms and not require a completely re-designed headset. However without eye tracking there is no way to objectively measure the point of the user's gaze, so vergence distance would either have to be estimated based on other factors, or the solution would have to work independent of vergence distance.

In the initial stages of planning the design, the intention was to proceed with the gaze estimation method. This method was detailed in the progress seminar: the point of the user's gaze fixation could be estimated by a combination of monitored data such as head position and current activity being performed, and some kind of input data such as a probabilistic model of gaze position within the FOV. The estimation could be made even more accurate by prompting the user to look at certain parts of a scene through eye catching visual cues, or providing instructions and goals to complete a task. Once the gaze position was deduced from these factors, the depth texture could be read at that point and a surrounding error region to give an estimate of the average depth in that area of the scene, which we could then use as our vergence distance value. With the vergence distance identified, the difference between that and the accommodation distance could be found, and then Chromablur could be rendered to match the accommodation to the vergence distance. However there were some problems with this approach that would lead to inaccuracy or inefficient performance.

Firstly, the idea of finding the difference in vergence and accommodation distances does not work when accommodation distance is set to infinity, as it is in some platforms. Setting the virtual projection distance to infinity means that the light rays are parallel when they are incident on the eye's lens, which occurs in real life when looking at far distances. This would then mean that technically vergence and accommodation could not be matched exactly. Secondly, the eye is robust to a certain amount of accommodative error, so an exact matching of vergence and accommodation is not necessary to mitigate VAC. Rather than attempting to make Chromablur drive accommodation on a continuous distance scale, a discrete application would be more efficient, and this is what the simplified algorithm currently does already.

The Chromablur algorithm we have designed currently only renders three different states corresponding to our assumptions of the monochromatic representative wavelengths of the display's RGB channels. These can be described as blue in focus for the hyperopic "accommodating too far" state, green for "well accommodated" and red in focus for the myopic "accommodating too near". To make the model continuous we could have a range of wavelength values rather than one representative value, which would scale proportionally with the amount of defocus rendered. However it was hypothesised that the discrete model will be sufficient to mitigate VAC in most cases, and so spending valuable computing power on a continuous model would be inefficient.

After re-evaluating the initial plan of estimating vergence distance, it was decided that a better solution would be to apply Chromablur in a manner that was effective independent of the current vergence distance. The assumption for this method would be that the discrete accommodation model of too near, too far would be sufficient to mitigate VAC. The design of this method was simple, given that we know what distance the eye is always accommodating to, and we know the depth of each point in the scene, then we know which parts of the scene we would see with a myopic or hyperopic accommodative state. Once we have identified this, we can apply the appropriate mode of Chromablur to that part of the scene, for example if the user looks at a point very near to the eye, accommodation would be known to be in a hyperopic "too far" state, so Chromablur should be applied to stimulate the eyes to accommodate closer.

Although as part of the three modes, the "well focussed" state was developed such that a purple fringe would appear, it was decided that this mode would not be necessary in mitigating VAC. Cholewiak et al also do not render this state, instead rendering the parts of the image that are in focus conventionally with no Chromablur. Although for demonstration purposes the "well focussed" mode was occasionally used, in the final algorithm that was applied and evaluated the parts of the scene corresponding to the accommodation depth were rendered conventionally. This also meant less computation as the blur kernels did not have to be convolved in those areas.

The algorithm was now finalised, although depending on the device the application method must be adjusted. For example on the HTC Vive, which was the platform that was used to develop the algorithm, the headset lenses project the display to a virtual distance of infinity [29], meaning the eye would be in a hyperopic state compared to vergence for all parts of the scene except the horizon. To clarify understanding, this doesn't mean all parts of the scene except the horizon would be out of focus, but rather there is a vergence accommodation mismatch for those parts of the scene. To remedy this, the Chromablur algorithm would be applied in the "too far" mode, with its magnitude scaling with depth so that for the closest points in the scene near the eye it would be strongest, while at the horizon it would reduce to zero and become conventionally rendered. As evaluation of the algorithm was conducted using the HTC Vive, this was the method of application evaluated.

Figure 4.8: Chromablur applied with the depth scaling method. Image is shown with myopic focus, and increased kernel diameter for visibility of the effect.

For other devices such as the Oculus Rift, the projection distance is speculated to be around the 1.4m range. For such a case where accommodation distance is set at a middle distance in the scene, Chromablur would be applied so that points farther than the projection distance are rendered with the "too close" mode, points closer would be rendered with "too far" and points at the projection distance itself rendered conventionally, with a small buffer range either way, thus forming discrete "depth bins" in the scene. With the depth scaling method and the depth bins method, Chromablur could be applied appropriately to all VR headsets.

Figure 4.9: Chromablur applied with the depth bin method. Depth bin sizes were chosen arbitrarily for demonstration, with increased kernel diameter and the well-focused bin rendered for visibility of the effect.

Chapter 5

Results

5.1 Evaluation Methodology

There were two methods used to evaluate the effectiveness of the algorithm, a user study and a comparison of computing load. The user study consisted of two experiments, which were conducted in two different environments developed by Angie Yao and Jane Phoon, used with permission. These two experiments would simulate two common types of VR experiences, a close at hand task requiring manipulation of a controller and passive viewing of a surrounding environment. In the computational load comparison we compare the performance of Chromablur and conventional rendering within a single environment on the same machine, using measures such as frame rate, memory, CPU and GPU load.

5.1.1 User Studies

The goal of the user studies was to ascertain whether or not Chromablur was successful in its intended effect of mitigating VAC. This could be measured by observing the user's accommodative state during the VR experience and comparing the results for the test and control conditions, and by querying the participants for their subjective experiences to find a possible preference for one condition over another.

In both experiments the two conditions would be a control condition of conventional rendering with no post-processing effect applied, and the test condition with the Chromablur post-processing effect in the depth scaling mode. Each participant would participate in both experiments under the same condition, and within each condition the experiments would be delivered in alternating order. Both test groups would record the same data and use the same questionnaires.

The most reliable way to measure accommodation in the eye is to use an autorefractor, which is how Cholewiak et al evaluate their algorithm's capability of driving accommodation. However an autorefractor is a large piece of equipment, and cannot be used in conjunction with the VR headset. The experiment conducted by Cholewiak et al on a projector screen could have been recreated with the modified Chromablur, however it was observed that the effect when viewed in VR was slightly different to the static preview shown within the Unity Editor likely due to how the SteamVR API worked, so it would not be a reliable measure of the performance in VR which we are interested in.

Instead, another method was devised to provide a way of observing an accommodative response without actually measuring accommodation. Pupil diameter is known to change with accommodative response as the ciliary muscle contracts or relaxes to change the shape of the lens [30]. Although the pupil diameter is directly controlled by separate muscles in the iris, movement of the ciliary muscle for an accommodative response also causes the pupils to constrict or dilate slightly [30]. This new method would record pupil data using the Pupil Labs Eye Tracker add-on for the Vive while the user was viewing the environment, and if the modified Chromablur does indeed drive accommodation then the user's pupil diameter would change differently in the two conditions. Of course pupil diameter is dependent on many other factors also, as we explored in a prior section. By recording baseline data while the pupil size is maximum, for example with a dark screen, and keeping other factors such as brightness as constant as possible, it would be possible to normalise the pupil diameter's response and interpret the results as being resultant from accommodation.

Experiment 1

The first experiment consisted of passive viewing of the surrounding environment. The user would be instructed to simply enjoy their ride through a park-like environment populated by wild animals. This experiment aimed to compare the effect of Chromablur on the accommodative response and participants' preference for a condition for the specific act of medium to long distance viewing. It was hypothesised that participants' subjective responses in the two conditions would be very similar due to the Vive's virtual projection distance being infinity, meaning VAC would not be significant at larger distances. The defocus blur introduced by Chromablur would also be less noticeable at longer distances, meaning accommodation is less likely to be stimulated at those distances, so the change in pupil diameter was also hypothesised to be similar.

One caveat however was that this environment experienced some performance issues in some areas with frame rates dropping to around 20FPS, which in combination with involuntarily movement such as a riding a moving cart, is an easy way to provoke motion sickness within VR. This could potentially influence the results when surveying the user's feelings of sickness or nausea, so users were asked to try and identify the type of discomfort they experienced, and indicate if they had experienced similar feelings previously in VR.

Experiment 2

This experiment would consist of an interactive environment where the user was given one controller, which would appear as a large stick in VR. Users were first instructed to hold the stick out at arms length vertically and focus on it, then bring it in and out of depth as if to make themselves cross-eyed. They were then told to examine the object at close range from different orientations, before being told to interact with the environment as they wished. This experiment was designed to compare the effect of Chromablur to conventional rendering for close at hand tasks, where the symptoms of VAC are greatest. It was hypothesised that given the Chromablur algorithm was effective at driving accommodation, the change in pupil diameter of participants would be significantly different between the two conditions. For closer viewing, the defocus blur would be much more noticeable so it was hypothesised that even if participants had an accommodative response induced, they may rate the Chromablur condition more poorly than the conventional condition, although feelings of discomfort would likely be lesser with Chromablur.

5.1.2 Computation Performance

The second metric by which the modified Chromablur algorithm was evaluated was its effect on computation performance when the VR environment was running. The data was gathered through the Unity Editor's Profiler tool, which monitors and records various parameters such as frame rate, CPU and GPU load, the amount of vertices and triangles rendered in the current image, and more. The tests were performed with the same PC used to run the user study environments, which has hardware specifications:

- CPU: Intel Core i7-8750H (4.1GHz)
- GPU: NVIDIA GeForce RTX 2080 8GB GDDR6
- RAM: 2x 16GB DDR4 (2666MHz)
- Screen: 15.6" FHD (1920x1080) 144Hz
- OS: Windows 10 Home 64bit

The Viking Village environment was used to record data as it was the most consistent of all the environments - performance in the experiment environments was influenced by external factors such as the user's interactions. The Profiler tool keeps a record of the previous 300 rendered frames, so points were randomly sampled from these 300 frames and used to form average values. To record the data, the headset was placed on a spinning chair and rotated fully, roughly within the time taken to record the 300 frames, to obtain data from a 360^o viewing angle of the environment. The parameters measured were:

- Average CPU and GPU Frame Times
- The proportion of CPU and GPU frame time spent on the Unity Game player and the Editor itself
- Used Texture Memory
- Render Texture Memory
- Used VRAM
- Total Memory and GFX Driver Memory

The Profiler tool has some of its own processing and memory overhead, and shows this along with the data from the rendered output, and the resources used by the editor itself. Running the test from the editor rather than a standalone build meant that the performance was affected by editor processes, however we use the Profiler to examine these.

5.2 Data and Analysis

The raw data collected in the experiments consisted of over tens of thousands of data points for each participant, so have been ommitted from this document for brevity.

5.2.1 Questionnaire Responses

After analysis of the subjective questionnaire results, the hypotheses of the two experiments were evaluated. In experiment 1, it was hypothesised that the participants' responses would be similar across both conditions due to the nature of the activity. It was found that this was the case, however this result was influenced heavily by the performance of the environment. The majority of participants (8 of 12) reported feeling discomfort due to the drops in frame rate in some areas of the environment, which caused the display to appear lagged and blurry. In both conditions, participants responded similarly about the performance issues and motion sickness, indicating that this factor shaped the results more significantly than any difference due to Chromablur. Other parameters such as realism and depth perception in the two conditions were also rated very similarly negatively, with many linking this rating again to the performance issues. This result was thus considered inconclusive, despite aligning with the prediction of the hypothesis.

In experiment 2, the hypothesis for participant responses was that due to it involving closer viewing, the blur created by Chromablur would be more noticeable, and therefore participants would be more likely to rate the Chromablur condition worse overall, except feelings of discomfort and visual strain. It was found that particpants in the Chromablur condition reported much less visual strain than the conventional rendering (4 of 6 in the conventional group and 1 of 6 in the Chromablur though this participant commented the strain went away after some time). Both groups reported difficulty focussing on the stick at close range and commented about seeing blur, however the conventional rendering group reported feeling eye strain more often. Across both conditions, no participants reported any feelings of sickness or nausea during or after the environment. Due to the lack of player movement and good performance of the environment, there were less factors that could have induced such feelings, unlike experiment 1. For ratings of realism, Chromablur participants had a much larger range of ratings from 1-9 out of 10 with an average of 4.83, while the conventional group's response range was 3-7 out of 10 with an average of 5.33. This was also the case for ratings of depth perception, where the Chromablur group's responses ranged from 2-10, and the conventional group 6-9 out of 10, both with an average of 7. The conventional group's responses were more consistent and rated higher on average than the Chromablur group, as expected from the hypothesis.

5.2.2 Pupil Data

To ascertain if an accommodative response was causing the participants' pupils to respond differently in the two conditions, we recorded baseline pupil diameters with a fully dark screen so that pupil size would be maximised, then subtracted the pupil response within the experiments from the average baseline value to view the response as a change in diameter with respect to the baseline.

Each data point had an associated pupil confidence value, which is the eye-tracking software's certainty that the detected pupil object is indeed a pupil, a value of 0 meaning it is certainly not a pupil, and 1 being certain that a pupil has been detected. The provided information document from Pupil Labs indicates that a confidence value of 0.6 usually produces the best results, so data points with confidence values below this threshold were discarded. The baseline recording also served to indicate how well the software was calibrated to the participant's eyes, and when baseline confidence values were too low the settings were adjusted and data re-recorded.

It was found that due to discrepancies in participants' head shape and size the eye tracking cameras installed in the headset were occasionally in suboptimal angles for recording. This meant that even with good calibration the data capture was sometimes inconsistent, causing more data to be discarded for some participants than others. However by maximising the sampling rate the impact of this was minimized.

There was also inconsistency between the data of the left and right eyes across all participants, which is to be expected as they were derived from two different camera feeds. It is known that accommodation is yoked between the eyes [2], so for each participant the better quality of the two eyes' data was used.

Although maximising the sampling rate lowered the impact of data becoming unusable due to the errors in the detection algorithm, it also meant the total dataset size became larger than necessary, making it a detriment to the processing. As such, the data in both experiments was downsampled by appropriate factors.

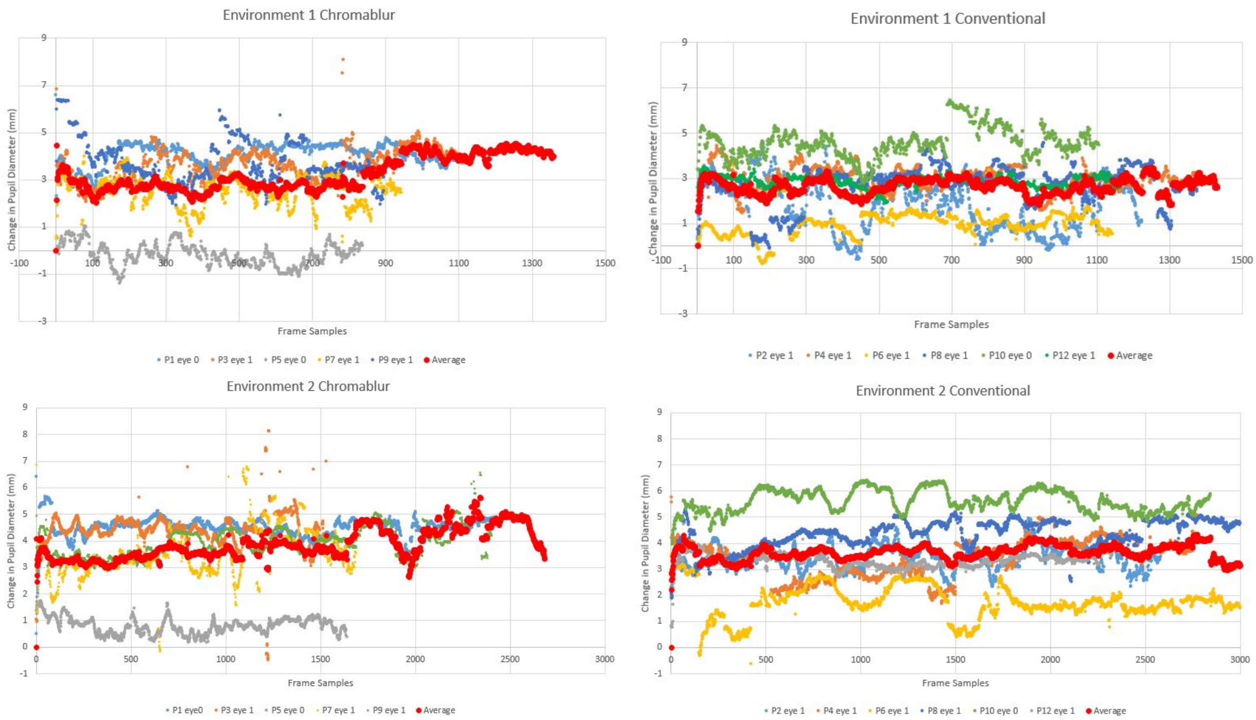

Figure 5.1: Top row: pupil diameter response for participants in Experiment 1, average shown in red. Bottom row: pupil diameter response for participants in Experiment 2, average shown in red. Participant 11's data was unusable due to low confidence values, and declined to re-record so the points have been ommitted from the Chromablur results.

It can be seen in figure 5.1 that the variation in pupil response was more consistent in the Chromablur conditions for both experiments. Participant 5 is an outlier in that it is the only one that diverges from the clustering seen in other participants for Chromablur. The fact that participant 5's data is close to 0 means that their pupil size in the experiments were not very different to the baseline values where it should have been much larger than in the bright environment. Although the confidence values were high, this data may have been subject to some calibration error. It is not impossible that the eye's pupil diameter did not change significantly from a dark to a bright environment, however it is much more common for the pupils to constrict visibly. Examining participant 5's data, we find that the average baseline values were around 4mm, which is much smaller than all other participants whose average baseline pupil diameter was between 6-8mm, and suggests some systematic error was introduced by the detection algorithm despite its high confidence rating in the baseline recording. If we discount participant 5's data, the Chromablur condition seems to have induced a more consistent response across participants.

The consistent response seen in the Chromablur condition and the large variance seen in the conventional condition could indeed be due to Chromablur stimulating the eye to accommodate as we intended, and having no such stimulation in the conventional condition. If Chromablur is driving the eyes of the participants to accommodate in a certain way, then it follows that the pupil responses will also all be similar and consistent. However for the Chromablur condition we have one less data set and are considering participant 5 an outlier. This means we are comparing the responses of four participants to six, which is not a fair test. Additionally, without measuring accommodation objectively we cannot conclusively say that the Chromablur algorithm has had the intended effect.

Although the Chromablur data shows a more consistent clustering of pupil responses, the variation in individual responses themselves are similar across both conditions, with some participants exhibiting a very constant response and some showing large fluctuations. In addition to the software recording actual changes in pupil diameter, the variation seen in each individual is also in part due to calculation error. The pupil detection software obtains pupil diameter by recording the apparent diameter in pixel size, which does not account for the angled camera perspective, then applying a proprietary transform algorithm to obtain pupil diameter in millimeters with perspective correction, based on average human eyeball size. The accuracy of this conversion is also influenced by the confidence value of the data point, so the high frequency fluctuations can be seen to be mostly due to this type of error, while lower frequency shifts reflect the participant's actual pupil diameter change.

The trend found in the questionnaires of Chromablur participants experiencing less eye strain might have been explained by a more constant pupil response, meaning the muscles in the eye were doing less work overall, but it seems this was not the case. The trend could have been caused by simple subjective experience, or another mechanism not linked to pupil diameter.

The thicker red lines show the average of the participants. They have been plotted separately for easier comparison below:

The plots of the average pupil responses show that despite the difference in consistency, the average response from the participants did not differ greatly in the experiments. A clear difference

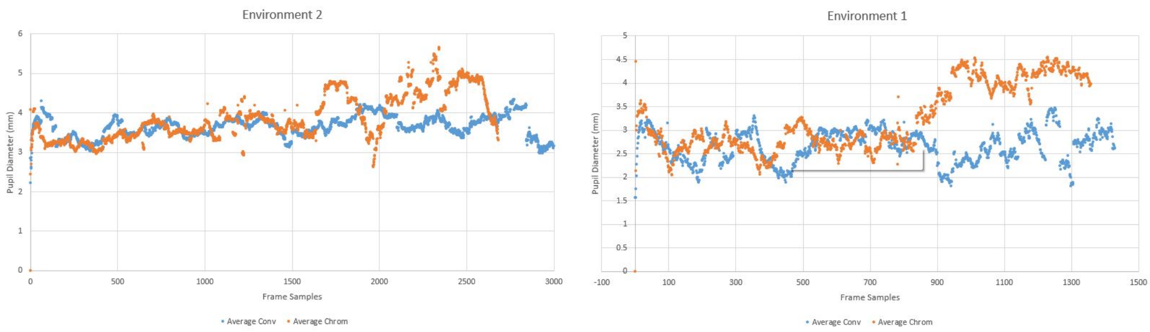

Figure 5.2: Average pupil responses

in the average responses would have supported the hypothesis of experiment 2, and provided some compelling evidence for the effectiveness of Chromablur. However apart from some divergence towards the end of the experiments, it appears that the average change in pupil diameter was similar across both conditions.

The data obtained shows some difference in the consistency of pupil responses of participants across the two conditions, which was one of the indicators we were looking for to evaluate the effectiveness of Chromablur. However the average responses were quite similar and does not support our hypothesis. Due to many sources of error and inaccuracy in the experiments, and a lack of participants to enlarge the total datasets, the results are considered inconclusive, however there are some promising aspects that merit further investigation with some refinements and changes to the methodology.

5.2.3 Computation Performance

Figures 5.3 shows the comparisons in performance for the Chromablur and conventional rendering tests.

The top row of graphs shows the average amount of time taken for the CPU and GPU to complete their total calculations for one frame of the output, and the proportion of this time spent on processing the Unity Game player, editor, and other processes. It must be noted that these frame time values are not reflective of the frame time of only the environment, but include the editor UI as well. We can obtain the more common measure of frame rate in frames per second (FPS) by taking the reciprocal of the frame time in seconds. The Profiler tool itself measures and records frame time of every process separately, while the statistics window in the Unity Game Player displays the frame rate of only the game window. For analysis purposes we will use frame time from the Profiler, as they are a more detailed measure of performance, however they are not reflective of the actual frame rate of the environment.

As expected, the Chromablur rendering takes longer to process on average for CPU processing

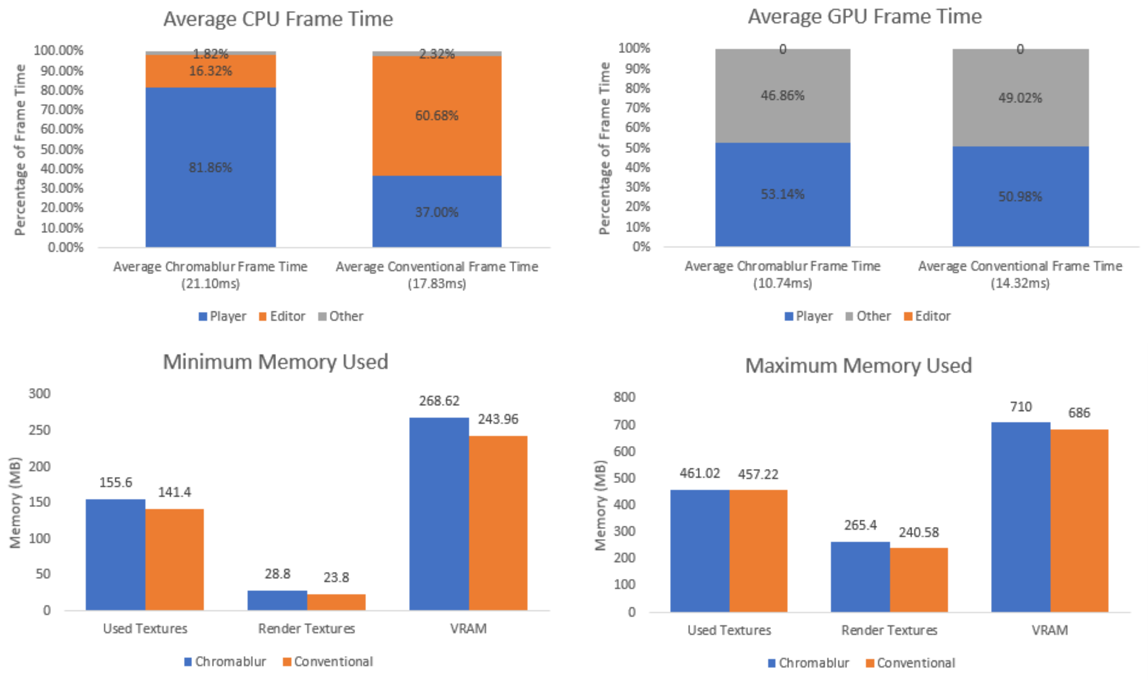

Figure 5.3: Top row: Proportional breakdowns of how the CPU and GPU spent their processing power. Bottom Row: Comparisons in minimum and maximum memory usage of the two conditions

with a 21.1ms average compared to 17.83ms, however for the GPU it's average frame time was actually lower with a 10.74ms average compared to 14.32ms. In the conventional rendering case only 37% of the CPU time was actually spent on rendering the image in the player, whereas with Chromablur on 81.86% of the frame time was spent on the player. This large difference is likely due to Chromablur requiring multiples C# scripts; the Chromablur script itself and all the overhead scripting required in the Unity Post Processing Stack overhead. The conventional rendering did not require any scripts at all, due to the environment having no dynamic or conditional behaviours implemented.

In the GPU frame time no processing was spent on the editor, as GPU's only perform graphics calculations. The frame time of Chromablur was unexpectedly lower, and the proportion of time spent rendering the player was only slightly higher for Chromablur. This shows that Chromablur had a negligible impact on GPU performance, which when considering the impact on CPU performance, suggests that the algorithm is not optimised for GPU calculations. As previously detailed, post-processing effects in Unity are always composed of a C# script and a shader that pass data back and forth, so both the CPU and GPU are utilised. The results of the performance experiment show that during the interaction of the script and shader of the Chromablur effect, more work is being done by the script, which increases the CPU load rather than GPU.

The bottom row of graphs show the memory usage of the two conditions. As expected, Chromablur consistently uses more memory than the conventional rendering, however it is not a significantly larger amount. The Profiler provided both minimum and maximum memory usage for each type category, and it was found Chromablur had both higher minimums and maximums.

Chapter 6

Discussion

6.1 Refinements to Evaluation Methodology

There were many factors that influenced the accuracy of results and sources of error in the experiment process. Here we will discuss the impact of these on the results, and how they could be minimised.

Firstly, the experiment design could be improved by using environments better suited to the tests we wish to conduct, and longer exposure to these environments. For the long distance passive viewing, the environment's performance issues greatly impacted the participants subjective experiences, and may have affected the eye's responses too. The performance standard for a comfortable VR is around 90FPS consistently, whereas this environment seemed to have a maximum of 70 and minimums between 10 and 20FPS. For the close viewing environment, the task that was performed was likely not detailed or complex enough to assess participants' performance well. An environment requiring participants to perform more common close-at-hand tasks in VR such as reading or manipulation of small objects would have been more suitable. Longer exposure to the environments is also something that would improve results, many people who experience some kind of sickness or discomfort in VR are usually fine with shorter experience like the experiments conducted. In real life usage of VR, exposure times are likely to be greater than 10 minutes, possibly up to a few hours. As with all population sample studies, we could also have benefited from more participants contributing data, reducing the impact of outliers on the results.

Another major improvement to the experiments would be a more accurate way of measuring accommodation. As previously mentioned, autorefractors are the best way of doing so however they are not suitable for use in conjunction with a VR headset. For these experiments we could not infer the actual accommodative state from the recorded pupil data, so investigation into such a method for use with the experiment should be conducted. Pupil capture itself could be improved by implementing a more systematic calibration method, and by recalibrating for each environment to obtain the best results.

Finally, the method used to evaluate the computational performance could have been made more accurate by performing the data capture on the built executable environments, rather than from within the Unity Editor, to remove the influence of the Editor's overhead processes. While it still provided a good indication of performance, the built environment would be strictly more accurate, and also make performance more optimal. The performance data also indicated that the GPU was being under utilised, so the post-processing effect could be optimised to solve this. Ideally, the effect should load the GPU more than the CPU but currently it is the other way around. This may be due to the way Unity compiles scripts and shaders; shader programs can be structured in many ways, some of which may influence how they are compiled. Further research into graphics development may be required to optimise the shader code and improve performance in this aspect.

6.2 Weaknesses and Limitations of the Design

The VR Chromablur algorithm designed in this thesis is a reduced and modified version of the original developed by Cholewiak et al. As such it has some weakness and limitations arising from this process, as well as from its application to VR.

Firstly, the algorithm is based off a simplified model of LCA and the human eye. The model as described by Cholewiak et al was also a reduced model, with the monochromatic wavelength assumption, pupil diameter assumption, and simplified kernels. However we diverge further by only implementing the forward step of the algorithm, and using the approximate blur kernel rather than a high fidelity method like ray tracing of the true kernel. Both of these points lowers the accuracy of the rendered LCA effect, but we proceed with the hypothesis that it may still be sufficient to stimulate accommodation.

Another difference is that our algorithm is applied in a VR context, not just a static image. This presented us with a large array of new problems and considerations. As detailed in the design section, the application of Chromablur to VR requires knowledge of the virtual projection distance to which the eyes are always accommodated, so for each platform it will be different. The effect we created had three states or modes that it could render the image with; the well focussed, hyperopic, and myopic accommodative states. This meant that we could only stimulate accommodation in two discrete ways, closer or farther than current by a constant amount, which although may be sufficient for many cases is strictly less complete than being able to stimulate accommodation on a continuous scale, for example 1D closer than current or 1.5D closer. The discrete model was motivated both by the hypothesis of its sufficiency for our goal, and also for performance reasons. Being able to render Chromablur to stimulate accommodation on a continuum would require the wavelength assumptions to become wavelength inputs, depending on the magnitude of accommodation we wished to stimulate. It was thought that any increase in computational load would be prohibitive to good performance, however after analysis of the computational performance it appears that the algorithm could realistically be more complex, as it did not add significant load compared to the conventional rendering.

As well as new theoretical considerations for the algorithm, the VR context also presented challenges arising from the hardware and software. Chromatic aberration is not a phenomenon unique to human eyes, and it was found that the HTC Vive's lenses actually had a visible amount of LCA, especially in the periphery. When viewing objects with large depth or colour gradient in the periphery, a clear red or blue outline was visible due to LCA, which would have affected the LCA that was being rendered by Chromablur in those areas. Being a result of the optical properties of the headset's lenses, this problem was unavoidable, however it was thought that it would not greatly affect Chromablur as along with the visible LCA, the image becomes distinctly blurrier at the lens periphery and so users are more likely to attempt to keep their gaze in the centre of the lens, where native LCA is not a problem.

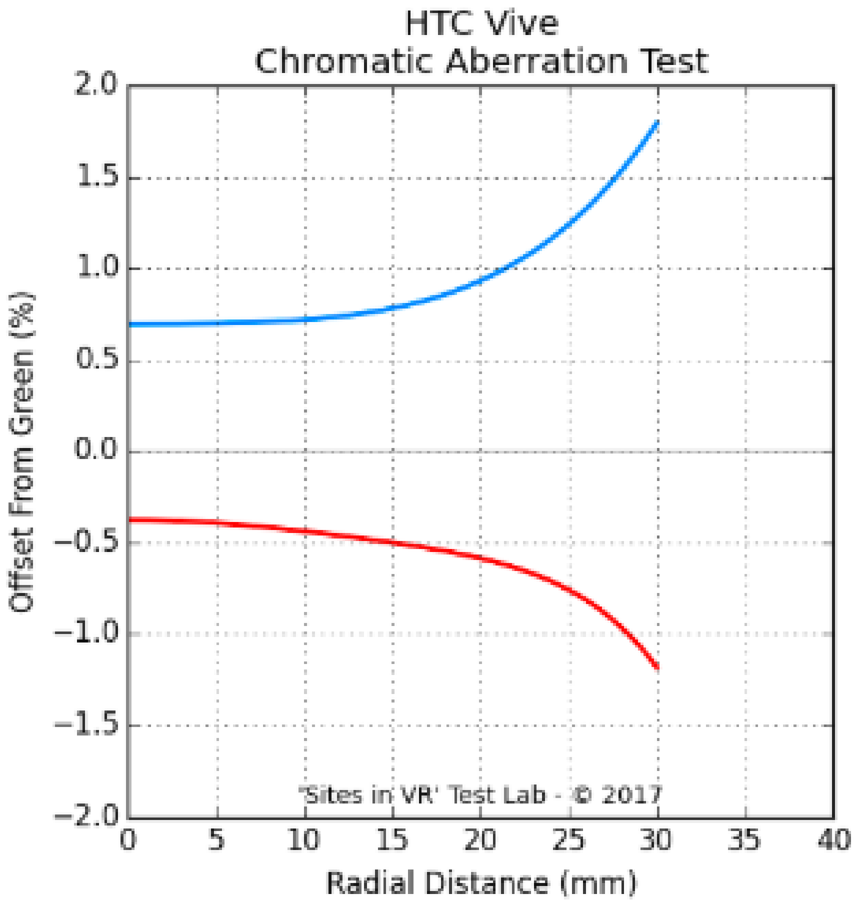

Figure 6.1: Colour shift of the lenses in the HTC Vive, taken from Sites in VR Test Lab [6]

Finally, there were two issues from the SteamVR API that had to be controlled. SteamVR provides VR functionality to any 3D scene by applying a stereoscopic transform to the output image, and rendering the left and right eyes appropriately. As well as calculating two slightly different perspectives based on the camera position, it also applies a spherical transform that accounts for the lens distortion in VR headsets. This process was the cause of the issues, the first being that the geometric transform exaggerated the effect of Chromablur from the preview screen in the editor. This meant that the calculated values were actually too big once the scene was actually rendered within the headset, causing significant colour artifacting instead of the expected subtle fringes. This required the calculated values to be scaled down by a constant factor until it matched the preview screen.

The second problem was that as its name suggests, post-processing effects are the final calculations applied in the render pipeline, and so by default the Chromablur effect was calculated after the VR transform. However the Chromablur effect was written based on input from a single image, whereas SteamVR splits this into two distorted images, making the texture read calls within the shader inaccurate, resulting in significant artifacting. The post-processing package only allows three different insertion points to the render pipeline for a custom effect: “Before Transparent”, “Before Stack” and “After Stack”. The before and after stack modes refer to the built-in post processing effects provided by Unity, and transparent refers to the built-in render pass that renders all transparent objects. To solve this issue we required the custom Chromablur effect to be calculated before the SteamVR pass, and it was found that only the “Before Transparent” mode allowed this. However this meant that the Chromablur algorithm is only applied to opaque objects, so all transparent objects would be rendered without the effect. To remedy this in future versions of the effect, more understanding and control of Unity’s VR rendering pipeline is required, and one of the scriptable render pipelines mentioned in the design section will likely have to be rather than the default.

6.3 VAC Mitigation and Image Quality Tradeoff

The fundamental idea behind Chromablur that is being used to mitigate VAC in VR is to drive accommodation to vergence distance by rendering a stimulus that will produce this response. We have explored the theoretical mechanisms that can achieve this, and implemented a post-processing effect based on these mechanisms. However as we mentioned at the end of the theory section, there is a trade off to be made between image quality and accuracy of accommodation.

The purpose of an accommodative response from the eye is to minimise the perceived defocus blur; when there is no blur in the image the eye is well focussed. This is the reason why we must render defocus blur as well as LCA in the image. The trade off is then that if we wish to mitigate

VAC by changing the accommodative state of the eye, we must apply some amount of blur to the presumably well focussed image, lowering the quality of the viewing experience as once the eye has accommodated it will be viewing a blurry image. This means that despite having made the appropriate accommodative response, the defocus blur in the eye has not gone away. We have already made note of Cholewiak et al's investigation into this problem, and found that it does not affect the maintenance of the new accommodative state.

However we now consider the implication of this blur for the overall effectiveness of Chromablur as a VAC mitigation solution for VR. Given that it is effective in matching the accommodation distance to vergence, the eyes now have the problem of blur, which is more immediately noticeable than VAC. It must be considered if it is worthwhile to transition from a state where despite viewing a sharp, clear image, there is risk of VAC related symptoms, to a state where VAC has been mitigated but the image is out of focus. Viewing an out of focus image causes its own symptoms, even if the defocus is mild. This is evident in people suffering from refractive errors: without their optical correction these people are prone to eye strain and headache. Of course, the Chromablur algorithm does not affect the entire image in VR, depending on the platform and what is currently being viewed there may be cases where there is little to no Chromablur rendering.

This problem of replacing the VAC problem with blur is not solved by implementing an effective Chromablur-like algorithm with no defocus blur. If we could drive accommodation in VR purely graphically without rendering any blur, we would actually still see blur. This is because the fundamental design of a VR headset that causes VAC to occur is also actually the mechanism that allows us to keep the flat display screen in focus at all times regardless of vergence. If we drive accommodation purely graphically, we may be mitigating VAC but we are also no longer focussing well on the display screen, so the scene will appear out of focus.

This suggests that Chromablur alone is not suitable as a complete solution to the VAC problem, but it could be used to enhance a hardware based approach. A purely graphical Chromablur could still improve accurate accommodation for distances that do not require a large defocus blur stimulus – if the blur created by Chromablur or by changing accommodation to a particular distance away from the headset's projection distance is tolerable by the viewer, then Chromablur is an effective solution until that point. This should be the case when viewing objects at far distances for headsets with infinite projection distance, as if we stimulate accommodation to change from infinity to a range of about 50m, the change in diopters is from 0 to 0.02, which when input to our algorithm would result in minimal blur. This gives a numerical understanding of why VAC is most severe at close ranges, as the change in focal length in diopters is greatest at short range. So for the most practically useful application of Chromablur, it should be used in conjunction with a hardware method.

Cholewiak et al also come to this conclusion and suggest that Chromablur can be used with focus tunable lenses to provide the best solution. Applying Chromablur to a system utilising focus tunable lenses and eye tracking would create a very robust and effective solution to VAC. The eye tracking would provide the gaze position utilised by both the algorithm and the tunable lens, and these two elements could work together to stimulate accommodation and minimise blur.

Chapter 7

Conclusion

7.1 Summary and Conclusion

To mitigate the problem of VAC we designed a graphics based software only solution that would take advantage of LCA within the eye to provide accurate depth cues rendered within the image. This was based on Cholewiak et al's earlier work, in which they present Chromablur - an image processing algorithm that produced images rendered with appropriate depth cues that could stimulate accommodation.

The algorithm worked by rendering LCA - a phenomenon occurring due to the refractive index of the eye's lens being wavelength dependent. This causes different colours to be focussed at different points on the retina, and depending on the state of accommodation different colour fringes would be seen, providing a signed depth cue that would allow the eye to accommodate correctly.

To take this algorithm from a static image context to a VR environment, we had to simplify and modify the original algorithm. This required an in depth understanding of the problem, assumptions and limitations of the original, and the problems arising from applying these in VR. We simplified the algorithm by using approximate blur kernels to model the LCA effect, and only recreated the forward step of the original, introducing some level of error but reducing computation complexity. We then had to design the appropriate method of application to a VR environment, such that the effect could mitigate VAC. The virtual projection distance of the HTC Vive was found to be set at infinity, so the modified Chromablur algorithm was applied in a depth scaling manner, so that the effect would be strongest at close range and taper away at long distance.

We conducted two user studies to evaluate the effectiveness of our algorithm, although there was an issue in that we could not objectively measure accommodation to show a response was stimulated. Instead we measured pupil diameters, as it is known that pupil diameter changes due to an accommodative response. By recording baseline data we normalised these values to account for other factors such as brightness, and found that in the Chromablur condition participants had a more consistent response. Participants also reported experiencing less eye strain in one experiment, however due to performance issues in the other affecting responses this could not be corroborated. The data overall showed some promise for the effectiveness of Chromablur, however was deemed inconclusive due to not objectively measuring accommodation.

We discussed the weaknesses, limitations and inaccuracies of the algorithm developed as well as the methods of evaluation used. There are many aspects of the design that we believe could be improved or changed after the evaluation, such as using a scriptable render pipeline to minimise the impact of the SteamVR transforms on the Chromablur effect, and implementing a continuous version of Chromablur rather than the discrete model we developed here. However a fundamental flaw in the idea of using a graphics only approach means that this algorithm by itself cannot effectively provide a solution to VAC, without introducing other problems.

7.2 Future Work

Our findings and shortcomings in this thesis provide some interesting avenues for future work, as well as straightforward improvements that can be made. The list of improvements and refinements is long, and includes the following.

- The best way to evaluate the effectiveness of the algorithm is to measure accommodation, and a method to do so without an autorefractor should be investigated.
- The script and shader code can be further optimised to increase performance, and the Unity project itself can be restructured to use a scriptable render pipeline that will allow better control over the final rendered output.
- The algorithm might have been over simplified and could potentially use a more accurate blur kernel to produce a target retinal image that would merit the deconvolution step.
- The methods of application could be refined to be more accurate, such as making the defocus blur scale as well as the LCA effect.
- The test environments used for the experiments could be improved to better facilitate observation of the simulated scenarios.
- Incorporation of eye tracking and some optical method of driving accommodation would upgrade and complete the solution provided by Chromablur.

There were some interesting concepts and under explored considerations while developing the algorithm. These could be potentially investigated further in future research:

- Niemitalo's method of producing separable disk kernels from complex Gaussian sums could be explored to implement a more optimal disk blur for larger sample sizes.
- In this thesis only the depth scaling method was tested, as the HTC Vive was the only platform that was accessible for development. Testing other methods of application and effectiveness of Chromablur in other platforms such as the Oculus Rift, PSVR, and even the Google Cardboard could be investigated. For lower quality platforms this method may prove more effective.
- The Chromablur algorithm as presented by Cholewiak is a graphics based software only method of driving accommodation. As they presented it, it is most effective at stimulating accommodation to a single, static distance, which we attempted to adapt and apply in VR to mitigate VAC. The ability to drive accommodation has applications outside of this, and it is most commonly used in the world to correct people's refractive errors. It is possible then that Chromablur could be used to correct myopia or hyperopia in a VR headset and eliminate the need for the user to wear the corrective lenses while using VR.
- The effect of changing the assumed parameters in the algorithm was not fully explored, further research into varying them could be conducted to refine the algorithm.

Bibliography

[1] D. Hoffman, A. Girshick, K. Akeley, M. Banks, Vergence-accommodation conflicts hinder visual performance and cause visual fatigue, Journal of vision 8 (2008) 33.1–30 (02 2008). doi:10.1167/8.3.33.

[2] S. A. Cholewiak, G. S. Love, P. P. Srinivasan, R. Ng, M. S. Banks, Chromablur: Rendering chromatic eye aberration improves accommodation and realism, ACM Transactions on Graphics (TOG) 36 (2017). doi:10.1145/3130800.3130815.
URL http://bankslab.berkeley.edu/publications/chromablur/

[3] S. A. Cholewiak, G. D. Love, M. S. Banks, Creating correct blur and its effect on accommodation, Journal of Vision 18 (9) (2018) 1–1 (09 2018). arXiv:https://jov.arvojournals.org/arvo/content_public/journal/jov/937491/i1534-7362-18-9-1.pdf, doi:10.1167/18.9.1.
URL https://doi.org/10.1167/18.9.1

[4] E. V. de Kerckhove, Htc vive tutorial for unity, Online Tutorial Website (Jan. 2019).
URL https://www.raywenderlich.com/9189-htc-vive-tutorial-for-unity

[5] T. Axford, A curious effect - ”spurious resolution” in an out-of-focus image, Online Forum (Feb. 2016).
URL https://www.dpreview.com/forums/thread/3970862

[6] S. in VR Test Lab, Htc vive vr headset lens review, Website (2016).
URL http://www.sitesinvr.com/viewer/htcvive/index.html

[7] O. Kreylos, Accomodation and vergence in head-mounted displays, Online Blog (Jul. 2017).
URL http://doc-ok.org/?p=1602

[8] M. Lambooij, M. Fortuin, I. Heynderickx, W. IJsselsteijn, Visual discomfort and visual fatigue of stereoscopic displays: A review, Journal of Imaging Science and Technology (2009).

[9] I. Marín-Franch, A. J. Del Águila Carrasco, P. Bernal-Molina, J. J. Esteve-Taboada, N. López-Gil, R. Montés-Micó, P. B. Kruger, There is more to accommodation of the eye than simply minimizing retinal blur, Biomedical optics express 8 (29082097) (2017) 4717–4728 (Sep. 2017).
URL `https://www.ncbi.nlm.nih.gov/pmc/PMC5654812/`

[10] G. Kramida, Resolving the vergence-accommodation conflict in head-mounted displays, IEEE transactions on visualization and computer graphics 22 (2015) 1912 – 1931 (08 2015). `doi:10.1109/TVCG.2015.2473855`.

[11] N. Padmanaban, R. Konrad, T. Stramer, E. A. Cooper, G. Wetzstein, Optimizing virtual reality for all users through gaze-contingent and adaptive focus displays, Proceedings of the National Academy of Sciences 114 (9) (2017) 2183–2188 (2017). `arXiv:https://www.pnas.org/content/114/9/2183.full.pdf`, `doi:10.1073/pnas.1617251114`.
URL `https://www.pnas.org/content/114/9/2183`

[12] N. Matsuda, A. Fix, D. Lanman, Focal surface displays, ACM Trans. Graph. 36 (4) (2017) 86:1–86:14 (Jul. 2017). `doi:10.1145/3072959.3073590`.
URL `http://doi.acm.org/10.1145/3072959.3073590`

[13] T. Zhan, Y.-H. Lee, S.-T. Wu, High-resolution additive light field near-eye display by switchable pancharatnam–berry phase lenses, Opt. Express 26 (4) (2018) 4863–4872 (Feb 2018). `doi:10.1364/OE.26.004863`.
URL `http://www.opticsexpress.org/abstract.cfm?URI=oe-26-4-4863`

[14] S. Lee, J. Cho, B. Lee, Y. Jo, C. Jang, D. Kim, B. Lee, Foveated retinal optimization for see-through near-eye multi-layer displays, IEEE Access 6 (2018) 2170–2180 (2018). `doi:10.1109/ACCESS.2017.2782219`.

[15] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, A. Lefohn, Towards foveated rendering for gaze-tracked virtual reality, ACM Transactions on Graphics (TOG) 35 (6) (2016) 1–12 (2016).

[16] A. Maimone, A. Georgiou, J. S. Kollin, Holographic near-eye displays for virtual and augmented reality, ACM Trans. Graph. 36 (4) (2017) 85:1–85:16 (Jul. 2017). `doi:10.1145/3072959.3073624`.
URL `http://doi.acm.org/10.1145/3072959.3073624`

[17] Tobii, Online Blog (May 2017). [link].
URL `https://blog.tobii.com/eye-tracking-vr-devkit-for-htc-vive-311cbca952df`

[18] FOVE, October update, Online Blog (Oct. 2015).
URL https://blog.getfove.com/2015/10/30/october-update/

[19] L. N. Thibos, M. Ye, X. Zhang, A. Bradley, The chromatic eye: a new reduced-eye model of ocular chromatic aberration inhumans. (opthalmic and visual optics), Applied Optics 31 (19) (1992) 3594 (1992).

[20] O. Niemitalo, Circularly symmetric convolution and lens blur, Online Blog (Aug. 2010).
URL http://yehar.com/blog/?p=1495

[21] S. Maji, Advanced edge detection lecture, Online Lecture Slides (Feb. 2016).
URL http://www-edlab.cs.umass.edu/~smaji/cmpsci370/slides/hh/lec02_hh_advanced_edges.pdf

[22] H. Strasburger, M. Bach, S. P. Heinrich, Blur unblurred-a mini tutorial., i-Perception 9 (2018) 2041669518765850 (Mar-Apr 2018).

[23] J. I. Yellott, Correcting spurious resolution, JOV 5 (12) (2005) 97–97 (2005).
URL https://doi.org/10.1167/5.12.97

[24] K. Garcia, Circular separable convolution depth of field, in: ACM SIGGRAPH 2017 Talks, SIGGRAPH '17, ACM, New York, NY, USA, 2017, pp. 16:1–16:2 (2017). doi:10.1145/3084363.3085022.
URL http://doi.acm.org/10.1145/3084363.3085022

[25] B. Wronski, Separable disk-like depth of field, Online Blog (Aug. 2017).
URL https://bartwronski.com/2017/08/06/separable-bokeh/

[26] J. Flick, Depth of field, Online Blog (2017).
URL https://catlikecoding.com/unity/tutorials/advanced-rendering/depth-of-field/

[27] P. Kovesi, Arbitrary gaussian filtering with 25 additions and 5 multiplications per pixel, The University of Western Australia (Sep. 2009).
URL http://web.csse.uwa.edu.au/research/?a=826172

[28] O. Kreylos, Optical properties of current vr hmds, Online Blog (Apr. 2016).
URL http://doc-ok.org/?p=1414

[29] C. Faliszek, Egx 2015 developer session htc vive steam vr, Online video recording, taken from timestamp 37:41 (2015).
URL https://www.youtube.com/watch?v=o8Ea6NqxnY4&feature=youtu.be&t=2261

[30] M. Motlagh, Physiology, accommodation, Statpearls (2019).